Efficient Quantification of Profile Matching Risk in Social Networks Using Belief Propagation

Anisa Halimi[1] and Erman Ayday[1,2(✉)]

[1] Case Western Reserve University, Cleveland, OH, USA
{anisa.halimi,erman.ayday}@case.edu
[2] Bilkent University, Ankara, Turkey

Abstract. Many individuals share their opinions (e.g., on political issues) or sensitive information about them (e.g., health status) on the internet in an anonymous way to protect their privacy. However, anonymous data sharing has been becoming more challenging in today's interconnected digital world, especially for individuals that have both anonymous and identified online activities. The most prominent example of such data sharing platforms today are online social networks (OSNs). Many individuals have multiple profiles in different OSNs, including anonymous and identified ones (depending on the nature of the OSN). Here, the privacy threat is profile matching: if an attacker links anonymous profiles of individuals to their real identities, it can obtain privacy-sensitive information which may have serious consequences, such as discrimination or blackmailing. Therefore, it is very important to quantify and show to the OSN users the extent of this privacy risk. Existing attempts to model profile matching in OSNs are inadequate and computationally inefficient for real-time risk quantification. Thus, in this work, we develop algorithms to efficiently model and quantify profile matching attacks in OSNs as a step towards real-time privacy risk quantification. For this, we model the profile matching problem using a graph and develop a belief propagation (BP)-based algorithm to solve this problem in a significantly more efficient and accurate way compared to the state-of-the-art. We evaluate the proposed framework on three real-life datasets (including data from four different social networks) and show how users' profiles in different OSNs can be matched efficiently and with high probability. We show that the proposed model generation has linear complexity in terms of number of user pairs, which is significantly more efficient than the state-of-the-art (which has cubic complexity). Furthermore, it provides comparable accuracy, precision, and recall compared to state-of-the-art. Thanks to the algorithms that are developed in this work, individuals will be more conscious when sharing data on online platforms. We anticipate that this work will also drive the technology so that new privacy-centered products can be offered by the OSNs.

Keywords: Social networks · Profile matching · Deanonymization · Privacy risk quantification

1   Introduction

Many individuals, to preserve their privacy and to protect themselves against potential damaging consequences, choose to share content anonymously in the digital space. For instance, people share their opinions about different topics or sensitive information about themselves (e.g., their health status) without sharing their real identities, hoping that they will remain anonymous. Unfortunately, this is non-trivial in today's interconnected world, in which different activities of individuals can be linked to each other. An attacker, by linking anonymous activities of individuals to their real identities (via other publicly available and identified information about them), can obtain privacy-sensitive information about them. Thus, individuals need tools that show them the scale of their vulnerability against such privacy risks when they share content. In this work, we tackle this problem by focusing on data sharing on online social networks (OSNs).

An OSN is a platform, in which, individuals share vast amount of information about themselves such as their social and professional life, hobbies, diseases, friends, and opinions. Via OSNs, people also get in touch with other people that share similar interests or that they already know in real-life [8]. With the widespread availability of the Internet, OSNs have been a part of our lives more than ever. Most individuals have multiple OSN profiles for different purposes. Furthermore, each OSN offers different services via different frameworks, leading individuals share different types of information [9]. Also, in some OSNs, users reveal their real identities (e.g., to find old friends), while in some OSNs, users prefer to remain anonymous (especially in OSNs in which users share anonymous opinions or sensitive information about themselves, such as their health status). Here, the privacy risk is the deanonymization of the anonymous OSN profile of a user using their other OSN profiles, in which the user is identified.

Such profile matching across OSNs (i.e., identifying profiles belonging to the same individuals) is a serious privacy threat, especially for individuals that have anonymous profiles in some OSNs and reveal their real identities in others. If an attacker can link anonymous profiles of individuals to their real identities, it can obtain privacy-sensitive information about individuals that is not intended to be linked to their real identities. Such sensitive information can then be used for discrimination or blackmailing. Thus, it is very important to quantify and show the risk of such profile matching attacks in an efficient and accurate way.

An OSN can be characterized by (i) its graphical structure (i.e., connections between its users) and (ii) the attributes of its users (i.e., types of information that is shared by its users). The graphical structures of most popular OSNs show strong resemblance to social connections of individuals in real-life (e.g., Facebook). Existing work shows that this fact can be utilized to link accounts of individuals from different OSNs [29]. However, without sufficient background information, just using graphical structure for profile matching becomes computationally infeasible. Furthermore, some OSNs or online platforms either do not have a graphical structure at all (e.g., forums) or their graphical structures do not resemble the real-life connections of the individuals (e.g., health-related OSNs such as PatientsLikeMe [3]). In these types of OSNs, an attacker can uti-

lize the attributes of the users for profile matching. Thus, to show the scale of the profile matching threat, it is crucial to process both the graphical structure and the other attributes of the users in an efficient and accurate way.

In this work, we efficiently model the profile matching problem in OSNs by considering both the graphical structure and other attributes of the users, a step towards delivering real-time information to OSN users about their privacy risks for profile matching due to their sharings on online platforms. Designing efficient privacy risk quantification tools is non-trivial, especially considering the scale of the problem. To overcome this challenge, we develop a novel, graph-based model generation algorithm to solve the profile matching problem in a significantly more efficient and accurate way than the state-of-the-art.

We formulate the profile matching problem as finding the marginal probability distributions of random variables representing the possible matches between user profile pairs from the joint probability distribution of many variables. We factorize the joint probability distribution into simpler local functions to compute the marginal probability distributions efficiently. To do so, we formulate the model generation for profile matching by using a graph-based algorithm. That is, we formulate the problem on a factor graph and develop a novel belief propagation (BP)-based algorithm to generate the model efficiently and accurately (compared to the state-of-the-art). The outcome of the model generation will pave the way towards developing real-time risk quantification tools (i.e., inform users about their privacy loss and its consequences as they share new content).

Our results show that the proposed model generation algorithm can match user profiles with an accuracy of up to 90% (depending on the amount of information and attributes that users share). As more information is collected about the users profiles in social networks, the accuracy of the BP-based algorithm increases. Also, by analyzing the effect of social networks' size to obtained precision and recall values, we show the scalability of the proposed model generation algorithm. We also show that by controlling the structure of the proposed graphical model, we can simultaneously improve the efficiency of the proposed model generation algorithm and increase its accuracy.

The rest of the paper is organized as follows. In Sect. 2, we discuss the related work. In Sect. 3, we provide the threat model. In Sect. 4, we describe the proposed framework in detail. In Sect. 5, we implement and evaluate the proposed framework using real-life datasets belonging to various OSNs. In Sect. 6, we discuss how the proposed scheme can be used for real-time privacy risk quantification, potential mitigation techniques, and generalization of the proposed scheme for different OSNs. Finally, in Sect. 7, we conclude the paper.

2   Related Work

Several works in the literature have proposed profile matching schemes that leverage network structure, publicly available attributes of the users, or both of them. Profile matching based only on network (graph) structure is widely known as the deanonymization problem. Graph deanonymization (DA) attacks can be classified as (i) seed-based attacks [16,18,19,29,30], in which a set of seeds (users'

accounts in two different networks which belong to the same individual) are known; and (ii) seed-free attacks [33,39], in which no seeds are used. Narayanan and Shmatikov were among the first that proposed a graph deanonymization algorithm [29]. Nilizadeh et al. [30] improved the attack proposed by Narayanan et al. by proposing a community-level deanonymization attack. Korula and Lattanzi [19] proposed a DA attack that by starting from a set of seeds, iteratively matches user pairs with the most number of neighboring mapped pairs. Ji et al. [16] quantified the deanonymizability of social networks from a theoretical perspective (i.e., focusing on social networks that follow a distribution model). Pedarsani et al. [33] proposed a Bayesian-based model to match users across social networks without using seeds. Their model uses node degrees and distances to other nodes. Sharad et al. [36] showed that users' re-identification (deanonymization) in anonymized social networks can be automated. Ji et al. [17] evaluated several anonymization techniques and deanonymization attacks and showed that all state-of-art anonymization techniques are vulnerable to modern deanonymization attacks. Recently, Zhou et al. [43] proposed DeepLink, a deep neural network based algorithm that leverages network structure for user linkage.

Another line of works [11,14,15,23,24,26,27,31,34,38,41,42] have leveraged public information in the users' profiles (such as user name, profile photo, description, location, and number of friends) for profile matching. Shu et al. [37] provided a broad review of the works that use public information for profile matching. Malhotra et al. [26] built classifiers on various attributes to determine whether two user profiles are matching or not. On the other hand, Zafarani et al. [42] explored user name by analyzing the behaviour patterns of the users, the language used, and the writing style to link users across social media sites. Goga et al. [11] showed that attributes that are hard to be controlled by users, such as location, activity, and writing style, may be sufficient for profile matching. Liu et al. [25] proposed a framework that mainly consists of three steps: behavior similarity modeling, structure consistency modeling, and multi-objective optimization. Goga et al. [12] conducted a detailed analysis of user profiles and their attributes identifying four properties: availability, consistency, non-impersonability, and discriminability. Andreou et al. [5] combined attribute and identity disclosure across social networks. Recently, Halimi et al. [13] proposed a more accurate profile matching framework based on machine learning techniques and optimization algorithms. One common thing about most of these aforementioned approaches is that they rely on training classifiers to determine whether a user pair is a match or not. We implemented some of these approaches and compared with the proposed framework in Sect. 5.

Contribution of this Paper: Previous works show that there exists a non-negligible risk of matching user profiles on offline datasets. Showing the risk on offline datasets is not effective since users need tools that guide them at the time of data sharing in digital world. However, building algorithms that will pave the way towards real-time privacy risk quantification is non-trivial considering the scale of the problem. In this paper, we develop a novel belief propagation

(BP)-based algorithm to generate the model efficiently and accurately (compared to the state-of-the-art). The proposed algorithm has linear complexity with respect to the number of user pairs (i.e., possible matches), while Hungarian algorithm [21], state-of-the-art that provides the highest accuracy (as shown in Sect. 5.4), has cubic complexity with respect to the number of users. We also show that the proposed algorithm achieves comparable accuracy with the Hungarian algorithm while providing this efficiency advantage.

3   Threat Model

We assume the attacker has access to user profiles in different OSNs. For simplicity we consider two OSNs: user profiles in OSN A (the auxiliary OSN) are linked to their identities, while in OSN T (the target OSN), the profiles of the individuals are anonymized. The attacker's goal is to match one or multiple user profiles from OSN T to the profiles in OSN A in order to determine the real identities of the users in OSN T. To do such profile matching, we assume that the attacker can only use the publicly available attributes of the users from OSNs A and T.

We study the extent of profile matching risk by means of two attacks: targeted attack and global attack. Targeted attack represents a scenario in which the attacker identifies the anonymous profile of a victim (or a set of victims) in OSN T and aims to find the corresponding unanonymized profile of the same victim in OSN A. Global attack represents the case in which the attacker aims to link all profiles in OSN T to their corresponding matches in OSN A.

4   Proposed Model Generation

Let A and T represent the auxiliary and the target OSNs, respectively, in which people publicly share attributes such as date of birth, gender, and location. We represent the profile of a user i in either A or T as U_i^k, where $k \in \{A, T\}$. We focus on the most common attributes that are shared in many OSNs and we categorize the profile of a user i as $U_i^k = \{n_i^k, \ell_i^k, g_i^k, p_i^k, f_i^k, a_i^k, t_i^k, s_i^k, r_i^k\}$, where n denotes the user name, ℓ denotes the location, g denotes the gender, p denotes the profile photo, f denotes the freetext provided by the user in the profile description, a denotes the activity patterns of the user (i.e., time instances at which the user posts), t denotes the interests of the user (based on the sharings of the user), s denotes the sentiment profile of the user, and r denotes the (graph) connectivity pattern of the user. As discussed, the main goal of the attacker is to link the profiles between two OSNs. The overview of the proposed framework is shown in Fig. 1. In the following, we describe the details of the proposed model generation algorithm.

4.1   Categorizing Attributes and Defining Similarity Metrics

Once the attributes of the users are extracted from their profiles, we first categorize them so that we can use them to compute the similarity values of attributes
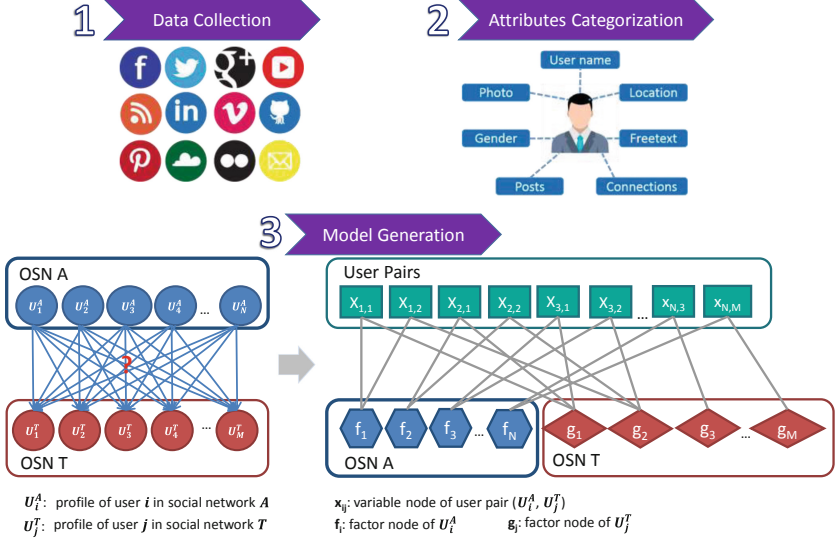
Fig. 1. Overview of the proposed framework. The proposed framework consists of 3 main steps: (1) data collection, (2) categorization of attributes and computation of attribute similarities, and (3) generation of the model.

between different users. In this section, we define the similarity metrics for each attribute between a user i in OSN A and user j in OSN T.

User Name Similarity - $S(n_i^A, n_j^T)$**:** We use Levenshtein distance [22] to compute the similarity between user names of profiles.

Location Similarity - $S(\ell_i^A, \ell_j^T)$**:** Location information collected from the users' profiles is usually text-based. We convert the textual information into coordinates via GoogleMaps API [1] and calculate the geographic distance between the corresponding coordinates.

Gender Similarity - $S(g_i^A, g_j^T)$**:** Availability of gender information is mostly problematic in OSNs. Some OSNs do not publicly share the gender information of their users. Furthermore, some OSNs do not even collect this information. In our model, if an OSN does not provide the gender information publicly (or does not have such information), we probabilistically infer the gender information by using a public name database. That is, we use the US social security name database[1] and look for a profile's name (or user name) to probabilistically infer the possible gender of the profile from the distribution of the corresponding name (among males and females) in the name database. We then use this probability as the $S(g_i^A, g_j^T)$ value between two profiles.

---

[1] US social security name database includes year of birth, gender, and the corresponding name for babies born in the United States.

Profile Photo Similarity - $S(p_i^A, p_j^T)$**:** We calculate the profile photo similarity through a framework named OpenFace [4]. OpenFace is an open source tool performing face recognition. OpenFace first detects the face (in the photo), and then preprocesses it to create a normalized and fixed-size input for the neural network. The features that characterize a person's face are extracted by the neural network and then used in classifiers or clustering techniques. OpenFace notably offers higher accuracy than previous open source frameworks. Given two profile photos p_i^A and p_j^T, OpenFace returns the photo similarity, $S(p_i^A, p_j^T)$, as a real value between 0 (meaning exactly the same photo) and 4.

Freetext Similarity - $S(f_i^A, f_j^T)$**:** Freetext data in an OSN profile could be a short biographical text or an "about me" page. We use NER (named-entity recognition) [10] to extract features from the freetext information. The extracted features are location, person, organization, money, percent, date, and time. To calculate the freetext similarity between the profiles of two users, we use the cosine similarity between the extracted features from each user.

Activity Pattern Similarity - $S(a_i^A, a_j^T)$**:** To compute the activity pattern similarity, we find the similarity between observed activity patterns of two profiles (e.g., likes or post). Let a_i^A represent a vector including the times of last $|a_i^A|$ activities of user i in OSN A. Similarly, a_j^T is a vector including the times of last $|a_j^T|$ activities of user j in OSN T. First, we compute the time difference between every entry in a_i^A and a_j^T and we determine $\min(|a_i^A|, |a_j^T|)$ pairs whose time difference is the smallest. Then, we compute the normalized distance between these $\min(|a_i^A|, |a_j^T|)$ pairs to compute the activity pattern similarity between two profiles.

Interest Similarity - $S(t_i^A, t_j^T)$**:** OSNs provide a platform in which users share their opinions via posts (e.g., tweets or tips), and this shared content is composed of different topics. In highlevel, first, we create a topic model using Latent Dirichlet Allocation (LDA) [7]. Then, by using the created model, we compute the topic distribution of each post generated by the users of the auxiliary and the target OSNs. Finally, we compute the interest similarity from the distance of the computed topic distributions.

Sentiment Similarity - $S(s_i^A, s_j^T)$**:** Users typically express their emotions when sharing their opinions about certain issues on OSNs. To determine whether the shared text (e.g., post or tweet) expresses positive or negative sentiment we use sentiment analysis through Python NLTK (natural language toolkit) Text Classification [2]. Given the text to analyze, the sentiment analysis tool returns the probability for positive and negative sentiment in the text. Users' moods are affected from different factors, so it is realistic to assume that they might change by time (e.g., daily). Thus, we compute the daily sentiment profile of each user, and daily sentiment similarity between the users. For this, first, we compute the normalized distribution of the positive and negative sentiments per day for each user, and then we find the normalized distance between these distributions for each user pair.

Graph Connectivity Similarity - $S(r_i^A, r_j^T)$**:** As in [36], for each user i, we define a feature vector $F_i = (c_0, c_1, ..., c_{n-1})$ of length n made up of components of size b. Each component contains the number of neighbors that have a *degree* in a particular range, e.g., c_k is the count of neighbors with a degree in range $[k \cdot b, (k+1) \cdot b]$. We use the feature vector length as 70 and bin size as 15 (as in [36]).

4.2   Generating the Model

We denote the set of profiles that are extracted for training from OSNs A and T as A_t and T_t, respectively. Profiles are selected such that some profiles in A_t and T_t belong to the same individual. We let set G include pairs of profiles (U_i^A, U_j^T) from A_t and T_t that belong to the same individual (i.e., coupled profiles). Similarly, we let set I include pairs of profiles that belong to different individuals (i.e., uncoupled profiles). For each pair of users in sets G and I, we compute the attribute similarities based on the categorizations of the attributes (as discussed in Sect. 4.1). We label the pairs in sets G and I (as coupled and uncoupled) and add them to the training dataset. Then, to identify the weight (contribution) of each attribute, we use logistic regression.

Next, we select the profiles to be matched and construct the sets A_e (with size N) and T_e (with size M). Then, we compute the general similarity $S(U_i^A, U_j^T)$ between every user in A_e and T_e using the identified weights of the attributes to obtain the $N \times M$ similarity matrix R. Our goal is to obtain a one-to-one matching between the users in A_e and T_e that would also maximize the total similarity. One way of solving this problem is to formulate it as an optimization problem and use the Hungarian algorithm, a combinatorial optimization algorithm that solves the assignment problem in polynomial time [21]. It is also possible to formulate profile matching as a classification problem and solve it using machine learning algorithms. Thus, we evaluate and compare the solution of this problem by using both the Hungarian algorithm and other off-the-shelf machine learning algorithms including k-nearest neighbor (KNN), decision tree, random forest, and SVM.

Evaluations on different datasets (we will provide the details of the datasets later in Sect. 5.2) show us that Hungarian algorithm provides significantly better precision, recall, and accuracy compared to other machine learning techniques (we will provide the details of our evaluation in Sect. 5.4). However, assuming N users in set A_e and M users in set T_e, the running time of the Hungarian algorithm for the above scenario is $O(\max\{N, M\}^3)$, and hence it is not scalable for large datasets. This raises the need for efficient, accurate, and scalable algorithms for model generation that will pave the way towards real-time privacy risk quantification.

4.3 Belief Propagation-Based Efficient Formulation of Model Generation

Inspired from the effective use of the message passing algorithms in information theory [35] and reputation management [6], in this research, for the first time, we formulate profile matching as an inference problem that infers the coupled profile pairs and develop an algorithm that relies on belief propagation (BP) on a graphical model. BP algorithm is based on a message-passing strategy for performing efficient inference using graphical models [32]. The problem we consider is different from [6,35] and so is the formulation. In this section we formalize our approach and present the different components that are needed to quantify the profile matching risk. Our goal is to obtain comparable precision, recall, and accuracy values as in the Hungarian algorithm with significantly better efficiency.

We represent the marginal probability distribution for a profile pair (i, j) to be a coupled pair as $p(x_{i,j})$, where $x_{i,j} = 1$ if profiles are matched as a result of the algorithm and $x_{i,j} = 0$, otherwise. Then, we formulate the profile matching (i.e., determining if a profile pair is coupled or uncoupled) as computing the marginal probability distributions of the variables in set $X = \{x_{i,j} : i \in A, j \in T\}$, given the similarity values between the user pairs in the similarity matrix R. Since the number of users in OSNs is high, it is computationally infeasible to compute the marginal probability distributions from the joint probability distribution $p(X|R)$. Thus, we propose to factorize $p(X|R)$ into local functions using a factor graph and run the BP algorithm to compute the marginal probability distributions in linear time (with respect to the number of profile pairs).

A factor graph is a bipartite graph containing two sets of nodes (variable and factor nodes) and edges between these two sets. We form a factor graph by setting a variable node for each variable $x_{i,j}$ (i.e., each profile pair). Thus, each variable node represents the marginal probability distribution of that profile pair being coupled or uncoupled. We use two types of factor nodes: (i) "auxiliary" factor node (f_i), representing each user i in OSN A and (ii) "target" factor node (g_j), representing each user j in OSN T. Each factor node is connected to the variable nodes representing its potential matches. Factor nodes represent the statistical relationships between the user attributes and profile matching. Using the factor nodes, the joint probability distribution function can be factorized into products of several local functions, as follows:

$$p(X|R) = \frac{1}{Z} \left[\prod_{i=1}^{N} f_i(x_{\sigma f_i}, R) \prod_{j=1}^{M} g_j(x_{\sigma g_j}, R) \right], \qquad (1)$$

where Z is a normalization constant, and σf_i (or σg_j) represents the indices of the variable nodes that are connected to factor node f_i (or g_j).

Figure 2 shows the factor graph representation of a toy example with 3 users from OSN A and 2 users from OSN T. Here, each user corresponds to a factor node in the graph (shown as a hexagon or rhombus, respectively). Each profile pair is represented by a variable node and shown as a rectangle. Each factor node is connected to the variable nodes it acts on. For example f_i is connected to all variable nodes (profile pairs) that contain U_i^A. The BP algorithm iteratively exchanges messages between the variable and the factor nodes, updating the beliefs on the values of the profile pairs (i.e., being a coupled or an uncoupled profile) at each iteration, until convergence.

Next, we introduce the messages between the variable and the factor nodes to compute the marginal distributions using BP. We denote the messages from the variable nodes to the factor nodes as μ. We also denote the messages from the auxiliary factor nodes to the variable nodes as λ and from the target factor nodes to the variable nodes as β. The message $\mu_{k \to i}^{(v+1)}\left(x_{i,j}^{(v)}\right)$ denotes the probability of $x_{i,j}^v = r$ ($r \in \{0,1\}$), at the vth iteration. Also, $\lambda_{i \to k}^{(v)}\left(x_{i,j}^{(v)}\right)$ denotes the

Fig. 2. Factor graph representation of 3 users from OSN A and 2 users from OSN T. (a) The users in both OSNs A and T. (b) Factor graph representation of all the possible profile pairs combinations between users in OSNs A and T.

probability that $x_{i,j}^v = r$ ($r \in \{0,1\}$) at the vth iteration given R (the messages β can be also expressed similarly). In the following, we describe the message exchange between the variable node $x_{1,4}$, the auxiliary factor node f_1, and the target factor node g_4 in Fig. 2. For clarity of presentation, we denote the variable and factor nodes $x_{1,4}$, f_1, and g_4 as i, k, z, respectively.

Following the general rules of BP [20], the variable node i generates its message to auxiliary factor node k by multiplying all the messages it receives from its neighbors, excluding k. Note that each variable node has only two neighbors (one auxiliary factor node and one target factor node). Thus, the message from the variable node i to the auxiliary factor node k at the vth iteration is as follows:

$$\mu_{i \to k}^{(v)}\left(x_{1,4}^{(v)}\right) = \beta_{z \to i}^{(v-1)}\left(x_{1,4}^{(v-1)}\right). \tag{2}$$
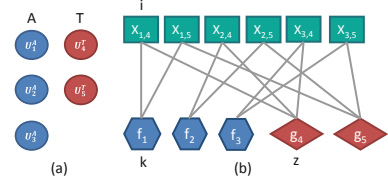
This computation is done at every variable node. The message from the variable node i to the target factor node z is also constructed similarly.

Next, factor nodes generate their messages. The message from the auxiliary factor node k to the variable node i is given by:

$$\lambda_{k \to i}^{(v)}\left(x_{1,4}^{(v)}\right) = \frac{1}{Z} \times S(U_1^A, U_4^T) \times \prod_{d \in (\sim i)} f_d\left(\mu_{d \to k}^{(v)}\left(x_{1,4}^{(v)}\right)\right), \tag{3}$$

where $(\sim i)$ means all variable node neighbors of k, except i. We compute function f_d as:

$$f_d\left(\mu_{d \to k}^{(v)}\left(x_{1,4}^{(v)}\right)\right) = \left(1 - \mu_{d \to k}^{(v)}\left(x_{1,4}^{(v)}\right)\right). \tag{4}$$

The above computation must be performed for every neighbor of each auxiliary factor node. The message from the target factor node z to the variable node i is also computed similarly.

The next iteration is performed in the same way as the vth iteration. The algorithm starts at the variable nodes. In the first iteration (i.e., $v = 1$), all the variable nodes send to their neighboring factor nodes the same value $(\lambda_{i \to k}^{(1)} \left(x_{i,j}^{(1)} \right) = 1/N)$, where N is the total number of "auxiliary" factor nodes. The iterations stops when the probability distributions of all variables in X converge. The marginal probability distribution of each variable in X is computed by multiplying all the incoming messages at each variable node.

4.4 ϵ-Accurate Model Generation

We also study the limitations and properties of the proposed BP-based model generation algorithm. We particularly analyze if the proposed algorithm maintains any optimality in any sense. For this, we use the following definition:

Definition 1 ϵ-***Accurate Model Generation.*** *We declare a model generation algorithm as ϵ-accurate if it can match at least ϵ% of the users accurately.*

Here, accuracy is the number of correctly matched coupled pairs by the proposed algorithm over the total number of coupled pairs. The above definition can also be made in terms of precision or recall (or both) of the proposed algorithm. Thus, for a fixed ϵ, we study the conditions for an ϵ-accurate algorithm. This also helps us understand the limits of profile matching in OSNs. To have an ϵ-accurate algorithm with a high ϵ value, it can be shown that, we require the BP-based algorithm to iteratively increase the accuracy until it converges. This brings about the following sufficient condition about ϵ-accuracy.

Definition 2 *Sufficient Condition.* *Accuracy of the model generation algorithm increases with each successive iteration (until convergence) if for all coupled profiles i and j, $Pr(x_{i,j}^{(2)} = 1) > Pr(x_{i,j}^{(1)} = 1)$ is satisfied.*

Depending on the fraction of the coupled profile pairs that meet the sufficient condition, ϵ-accuracy of the proposed algorithm can be obtained. In Sect. 5, we experimentally explore the cases in which this sufficient condition is satisfied with high probability.

5 Evaluation of the Proposed Mechanism

In this section, we evaluate the proposed BP-based algorithm by using real data from four OSNs. We also study the impact of various parameters to the ϵ-accuracy of the proposed algorithm.

5.1   Evaluation Metrics

To evaluate the proposed model, we mainly consider the global attack, in which the goal of the attacker is to match all profiles in A_e to all profiles in T_e. In other words, the goal is to deanonymize all anonymous users in the target OSN (who have accounts in the auxiliary OSN). For the evaluation metrics, we use precision, recall, and accuracy. Hungarian algorithm and the proposed BP-based algorithm provide a one-to-one match between all the users. However, we cannot expect that all anonymous users in the target OSN have profiles in the auxiliary OSN. Therefore, some of the provided matches are useless for us. Thus, we select a "similarity threshold" ("probability threshold" for machine learning techniques) for evaluation. Each matching scheme returns 1 (i.e., true match) if the similarity/probability of user pair is higher than the threshold, and 0 otherwise. So, we consider as true positives the pairs that are correctly matched by the algorithm and whose similarity/probability is greater than the threshold. We also compute accuracy as the number of correctly matched coupled pairs identified by the algorithm over the total number of coupled pairs.

5.2   Data Collection

To evaluate our proposed framework, we use three datasets: (i) Dataset 1 (D1): Google+ - Twitter [13], (ii) Dataset 2 (D2): Instagram - Twitter, and (iii) Dataset 3 (D3): Flickr social graph [40]. To collect the coupled profiles in D1 and D2, social links in Google+ profiles and about.me (a social network where users provide links to their OSN profiles) were used, respectively. In terms of dataset sizes, (i) D1 consists of 8000 users in each OSNs where 4000 of them are coupled profiles; (ii) D2 consists of more than 10000 coupled profiles (and more content about the OSN users compared to D1); and (iii) D3 consists of 50000 users. In D1, we use Twitter as our auxiliary OSN (A) and Google+ as our target OSN (T); in D2, we use Twitter as our auxiliary OSN (A) and Instagram as our target OSN (T); and in D3, we generate the auxiliary and the target OSN graphs as in [28,36] by using a vertex overlap of 1 and an edge overlap of 0.9.

5.3   Evaluation Settings

Since the model generation process is the same for all three datasets, in the rest of the paper, we hold the discussion over a target and auxiliary network. From each dataset, we select 3000 profile pairs (1500 coupled and 1500 uncoupled) for training. We also select 500 users from the auxiliary OSN and 500 users from the target OSN to construct sets A_e and T_e, respectively. Note that none of these users are involved in the training set. Among these profiles, we have 500 coupled pairs and 249500 uncoupled pairs, and hence the goal is to make sure that these 500 users are matched with high confidence in a global attack scenario. Note that we do not use cross-validation because we consider all the possible user combinations to test our model and it is time-consuming to compute all similarity metrics for all combinations. Considering all the combinations instead

of randomly selecting some user pairs is a more realistic evaluation setting since one can never know which users pairs the attacker will have access to. In cases that there are missing attributes (that are not published by the users) in the dataset, we assign a value for the attribute similarity based on the distributions of the attribute similarity values between the coupled and uncoupled pairs.
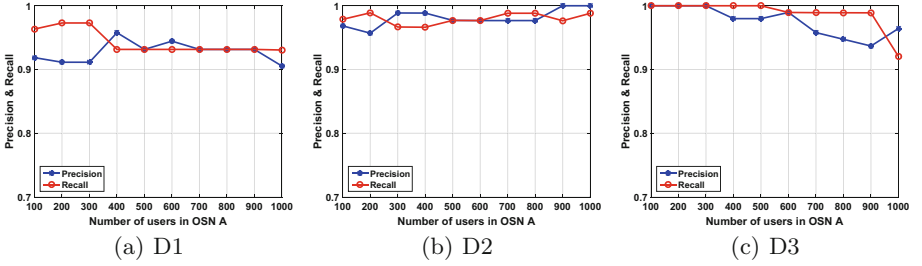
5.4   Evaluation of BP-Based Model Generation

In Fig. 3, we show the comparison of the proposed BP-based model generation to [12,13,26,31,36] for each dataset (D1, D2, and D3). [12,26,31,36] use machine learning-based techniques (k-nearest neighbor (KNN), decision tree, random forest, and/or SVM), while [13] uses the Hungarian Algorithm. Our results show that the proposed scheme provides comparable precision and recall compared to the state-of-the-art Hungarian algorithm and it significantly outpowers machine leaning-based algorithms. For instance, the proposed algorithm provides a precision value of around 0.97 (for a similarity threshold of 0.5) in D1. This means, if our proposed algorithm returns a matched profile pair that has a similarity value above 0.5, the corresponding profiles belong to same individual with a high confidence. At the same time, the complexity of the proposed algorithm scales linearly with the number of user pairs, while the Hungarian algorithm suffers from cubic complexity. Note that precision and recall values obtained from D2 are higher compared to the ones from D1 as we collected more information about users in D2. We also compare the BP-based model generation to the deep neural network based algorithm (DeepLink) [43] in D3. For DeepLink, we use the same settings as in [43]. DeepLink achieves an accuracy of 84% in D3 which is slightly less than the one obtained by the proposed algorithm (90%). DeepLink achieves a precision of 0.84, and a recall of 1 while the BP-based algorithm achieves a precision of 0.93 and a recall of 0.9. DeepLink provides a match for each user even if that user does not have a match.

(a) D1   (b) D2   (c) D3
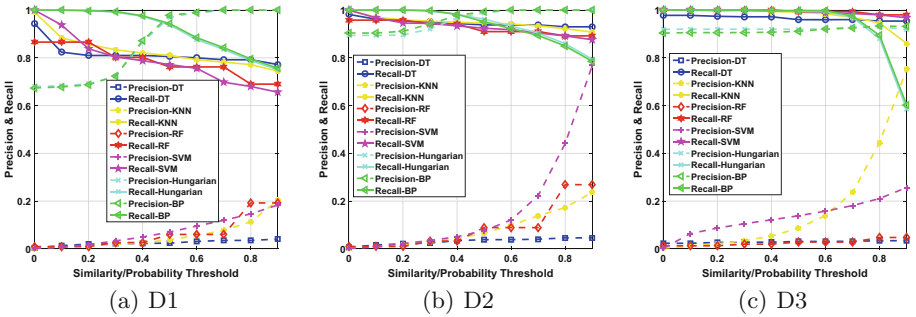
Fig. 3. Comparison of the proposed BP-based scheme with the Hungarian algorithm and machine learning techniques (decision tree - DT, KNN, random forest - RF, and SVM) in terms of precision and recall for D1, D2, and D3.

We also study the effect of the OSNs' size to precision and recall of the proposed algorithm. In Fig. 4, for each dataset, we show the precision/recall values of the BP-based algorithm when the number of users in the auxiliary OSN (OSN A) increases while the number of target users (i.e., users in OSN T) is fixed. We set the number of users in OSN T as 100 and increase the number of users in OSN A from 100 to 1000 in steps of 100. We observe that the precision/recall values of the proposed algorithm only slightly decrease with the increase of auxiliary OSN's size, which shows the scalability of our proposed algorithm. We achieve similar results for the other two scenarios: (i) when we fix the number of users in OSN A and vary the number of users in OSN T; and (ii) when we increase the number of users in both OSNs A and T. Due to the space constraints, we present the details of the results of scenarios (i) and (ii) in Fig. 7 and 8, respectively, in Appendix A.
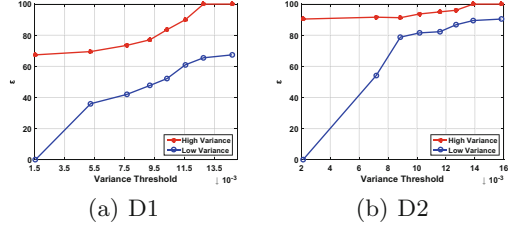

Fig. 4. The effect of auxiliary OSN's (OSN A) size to precision/recall when the size of target OSN (OSN T) is 100 in D1, D2, and D3.
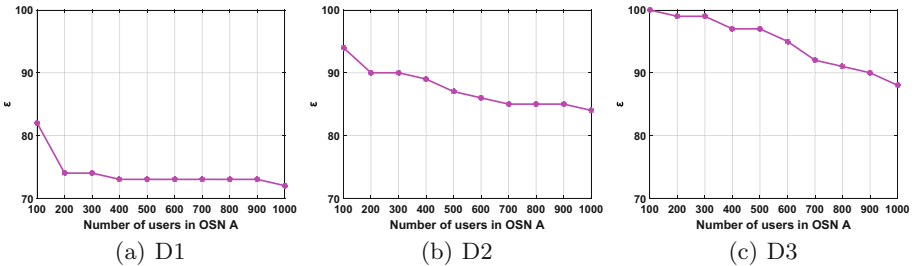
Next, we evaluate the ϵ-accuracy of the proposed model generation algorithm (introduced in Sect. 4.4). There are many parameters to consider to analyze the ϵ-accuracy of the proposed algorithm, such as the average degree of factor nodes, total similarity of each user in the target OSN with the ones in the auxiliary OSN, and number of users in the target and auxiliary OSNs. Here, we experimentally analyze and show the ϵ-accuracy of the proposed algorithm considering such parameters. For evaluation, we use all datasets and pick 500 users from OSN T. For all the studied parameters, we observe that at least 97% of coupled profiles that can be correctly matched by the BP-based algorithm satisfy the sufficient condition (introduced in Sect. 4.4). We observe that ϵ value is inversely proportional to the average degree of the factor nodes. In D1, the ϵ-accuracy of the proposed algorithm is $\epsilon = 67$ and $\epsilon = 84$ when the average degrees of the factor nodes are 500 and 22, respectively (we discuss more about the results of this experiment in Sect. 5.5).

To study the impact of user pairs' similarity, for each user in OSN T, we compute the variance of the similarity values between that user and all users in OSN A. Then, we compute the accuracy of the proposed BP-based algorithm on users with varying variance values. For evaluation, we use D1 and D2. Our results show that ϵ-accuracy of the proposed algorithm is higher for users with higher variances (as shown in Fig. 5). For instance, in D1, we observe the ϵ-accuracy as $\epsilon = 42$ and $\epsilon = 90$, when we run the proposed algorithm only for the users

(a) D1   (b) D2

Fig. 5. The effect of variance threshold to ϵ-accuracy in D1 and D2. For "high variance", our goal is to match users in OSN T that have a variance (for the similarity values between a user in OSN T and all users in OSN A) greater than the variance threshold, while for "low variance", we consider users that have a variance value smaller than the variance threshold.

with low variance (lower than 0.008) and high variance (higher than 0.012), respectively. These results show that the vulnerability of the users in an OSN (for the profile matching attack) can be identified by analyzing particular characteristics of the OSNs.

Furthermore, we observe that ϵ value is inversely proportional to the number of users in OSN A (as shown in Fig. 6). The proposed algorithm achieves an ϵ-accuracy of $\epsilon = 82$ and $\epsilon = 72$ in D1 when the number of users in A is 100 and 1000, respectively while the number of users in T is 100. This decrease in accuracy can be considered as low considering that the number of possible matches increases 10 times (from 10000 to 100000 user pairs). In D2, we observe a similar trend to D1. In D3, accuracy decreases faster with the increase in number of users in OSN A (compared to D1 and D2). This is because, in D3, we only use the graph connectivity attribute for profile matching. Thus, as the number of users in OSN A increases, the number of users with similar graph connectivity patterns also increases causing the decrease in accuracy.

(a) D1   (b) D2   (c) D3

Fig. 6. The effect of auxiliary OSN's (OSN A) size to ϵ-accuracy in D1, D2, and D3. The size of target OSN (OSN T) is fixed to 100.

5.5   Complexity Analysis of the BP-Based Algorithm

The complexity of the BP-based algorithm is linear in the number of variable (or factor nodes). In the proposed BP-based algorithm, we generate a variable node for all potential matches between the target and the auxiliary OSNs. Assuming N users in both target and auxiliary OSNs, this results in N^2 variable nodes in the graph. To analyze the effect of number of variable nodes on the performance, we experimentally try to change the graph structure and limit the number of variable nodes for each user in the target

Table 1. Evaluation of the proposed BP-based algorithm with varying the number of variable nodes. N denotes the number of users in OSN T, and each value in the complexity column shows the number of variable nodes (i.e., the number of users pairs used for profile matching).

Dataset	Complexity	Precision	Recall	Accuracy
D1	N^2	0.978	0.944	67.4%
	$N\sqrt{N}$	0.955	0.939	84.6%
	$N \log N$	0.975	0.966	96%
D2	N^2	0.978	0.982	90.4%
	$N\sqrt{N}$	0.977	0.954	94.8%
	$N \log N$	0.934	0.946	90.6%
D3	N^2	0.925	0.976	90.4%
	$N\sqrt{N}$	0.92	0.973	91%
	$N \log N$	0.932	0.976	95%

OSN. We heuristically decrease the average degree of the factor nodes from N to $\sqrt{N}$ and $\log N$ by only leaving the variable nodes (user pairs) with the highest similarity values. For evaluation, we use all datasets (D1, D2, and D3). We pick 500 users from OSNs A and T to construct the test dataset (where there are 500 coupled and 249500 uncoupled profile pairs initially). In Table 1, we show the results with varying number of variable nodes. For instance, in D1, we obtain an accuracy of 67.4% when all the potential matches are considered (i.e., with N variable nodes for each user in T) and an accuracy of 96% when we use only $\log N$ variable nodes for each user. These results are important since they show that while reducing the complexity of the proposed BP-based algorithm, we can further improve its accuracy.

6   Discussion

In this section we discuss how the proposed framework can be utilized for sensitive OSNs, and potential mitigation techniques against the identified profile matching risk.

6.1   Profile Matching on Sensitive OSNs

Note that in D1 and D2, users provide the links to their social networks publicly. It is quite hard to obtain coupled profiles from social networks where users share sensitive information such as PatientsLikeMe. We expect to obtain similar results as long as users share similar attributes across OSNs. Considering that these users are more privacy-cautious, mostly non-obvious attributes such as interest, activity, sentiment similarity, or writing style can be used.

6.2   Mitigation Techniques

We foresee that the OSN can provide recommendations to the users (about the content they share) to reduce their risk for profile matching attacks. Such recommendation may include (i) generalizing or distorting some shared content of the user (e.g., generalizing the shared location or posting a content at a later time); or (ii) choosing not to share some content (especially for attributes that are hard to generalize or distort, such as interest or sentiment). When generating such recommendations, there are two main objectives: (i) content shared by the user should not increase user's risk for profile matching and (ii) utility of the content shared by the user (or utility of user's profile) should not decrease due to the applied countermeasures. Using a utility metric for the user's profile, the proposed framework (in Sect. 4.3) can be used to formulate an optimization between the utility of the user's profile and privacy of the user. The solution of this optimization problem can provide recommendations to the user about how to (or whether to) share a new content on their profile.

7   Conclusion

In this work, we have proposed a novel message passing-based framework to model the profile matching risk in online social networks (OSNs). We have shown via simulations that the proposed framework provides comparable accuracy, precision, and recall compared to the state-of-the-art, while it is significantly more efficient in terms of its computational complexity. We have also shown that by controlling the structure of the proposed BP-algorithm we can further decrease the complexity of the algorithm while increasing its accuracy. We believe that the proposed framework will be instrumental for OSNs to educate their users about the consequences of their online sharings. It will also pave the way towards real-time privacy risk quantification in OSNs against profile matching attacks.

Appendix

A   Scalability of the BP-Based Algorithm

We study the effect of the OSNs' size to precision and recall of the proposed algorithm. In Sect. 5.4, we provided the results when the number of users in OSN T is fixed. Here, we provide the results of the other two scenarios. In Fig. 7, for each dataset, we show the precision/recall values of the BP-based algorithm when the number of users in the target OSN (OSN T) increases while the number of auxiliary users (i.e., users in OSN A) is fixed. We set the number of users in
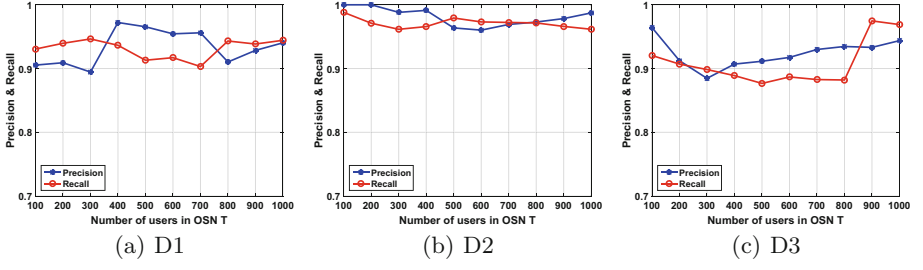
Fig. 7. The effect of target OSN's (OSN T) size to precision/recall when the size of auxiliary OSN (OSN A) is 1000 in D1, D2, and D3.
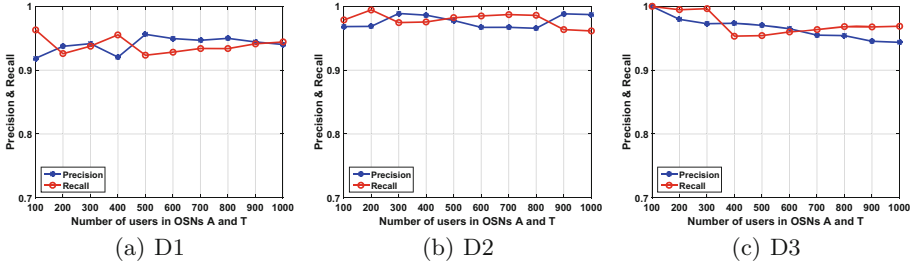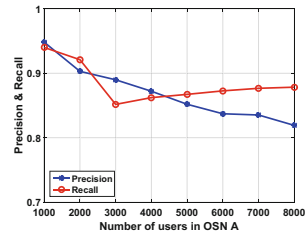

Fig. 8. The effect of auxiliary and target OSNs' (OSN A and T) size to precision/recall in D1, D2, and D3.

OSN A as 1000 and increase the number of users in OSN T from 100 to 1000 in steps of 100.

In Fig. 8, for each dataset, we show the precision/recall values of the BP-based algorithm when the number of users in both OSNs (i.e., OSN A and T) increases from 100 to 1000 in steps of 100. In both scenarios, we observe that the precision/recall values of the proposed algorithm only slightly decrease with the increase of the number of users in the target OSN, or the increase of the number of users in both OSNs, which shows the scalability of our proposed algorithm.

To further check the effect of the auxiliary OSN's size to precision and recall of the BP-based algorithm, we quantify the precision/recall values obtained by the proposed algorithm for larger scales in D3. We fix the number of users in the target OSN (i.e., OSN T) to 1000 while the number of users in the auxiliary OSN (i.e., OSN A) increases from 1000 to 8000 in steps of 1000 (in Fig. 4 the number of users in OSN T was fixed to 100 while the number of users in OSN A

Fig. 9. The effect of auxiliary OSN's (OSN A) size to precision/recall when the size of target OSN (OSN T) is 1000 in D3.

was increasing from 100 to 1000). We show the results for D3 in Fig. 9. The precision/recall values slightly decrease with the increase of the number of users in OSN A, confirming the scalability of the proposed algorithm. Note that, in D3, we only use the graph connectivity attribute for profile matching. We expect that the decrease in precision/recall values will be smaller when both the graphical structure and other attributes of the users are used to generate the model.

References

1. Google maps API (2020). https://developers.google.com/maps/
2. Natural language toolkit (2020). http://www.nltk.org/
3. Patienslikeme (2020). https://www.patientslikeme.com/
4. Amos, B., Ludwiczuk, B., Satyanarayanan, M.: OpenFace: a general-purpose face recognition library with mobile applications. Technical report, CMU-CS-16-118, CMU School of Computer Science (2016)
5. Andreou, A., Goga, O., Loiseau, P.: Identity vs. attribute disclosure risks for users with multiple social profiles. In: Proceedings of the IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (2017)
6. Ayday, E., Fekri, F.: Iterative trust and reputation management using belief propagation. IEEE Trans. Dependable Secur. Comput. **9**(3), 375–386 (2012)
7. Blei, D.M., Ng, A.Y., Jordan, M.I.: Latent Dirichlet allocation. J. Mach. Learn. Res. **3**, 993–1022 (2003)
8. Boyd, D.M., Ellison, N.B.: Social network sites: definition, history, and scholarship. J. Comput.-Mediat. Commun. **13**(1), 210–230 (2007)
9. Debnath, S., Ganguly, N., Mitra, P.: Feature weighting in content based recommendation system using social network analysis. In: Proceedings of the International Conference on World Wide Web (2008)
10. Finkel, J.R., Grenager, T., Manning, C.: Incorporating non-local information into information extraction systems by gibbs sampling. In: Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (2005)
11. Goga, O., Lei, H., Parthasarathi, S.H.K., Friedland, G., Sommer, R., Teixeira, R.: Exploiting innocuous activity for correlating users across sites. In: Proceedings of the 22nd International Conference on World Wide Web (2013)
12. Goga, O., Loiseau, P., Sommer, R., Teixeira, R., Gummadi, K.P.: On the reliability of profile matching across large online social networks. In: Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (2015)
13. Halimi, A., Ayday, E.: Profile matching across unstructured online social networks: threats and countermeasures. arXiv preprint arXiv:1711.01815 (2017)
14. Iofciu, T., Fankhauser, P., Abel, F., Bischoff, K.: Identifying users across social tagging systems. In: Proceedings of the International AAAI Conference on Web and Social Media (2011)
15. Jain, P., Kumaraguru, P., Joshi, A.: @i seek 'fb.me': identifying users across multiple online social network. In: Proceedings of the 22nd International Conference on World Wide Web (2013)
16. Ji, S., Li, W., Gong, N.Z., Mittal, P., Beyah, R.: On your social network deanonymizablity: quantification and large scale evaluation with seed knowledge. In: Proceedings of the Network and Distributed System Security Symposium (2015)

17. Ji, S., Li, W., Mittal, P., Hu, X., Beyah, R.: SecGraph: a uniform and open-source evaluation system for graph data anonymization and de-anonymization. In: Proceedings of the 24th USENIX Security Symposium (2015)

18. Ji, S., Li, W., Srivatsa, M., Beyah, R.: Structural data de-anonymization: quantification, practice, and implications. In: Proceedings of ACM SIGSAC Conference on Computer and Communications Security, pp. 1040–1053. ACM (2014)

19. Korula, N., Lattanzi, S.: An efficient reconciliation algorithm for social networks. Proc. VLDB Endowment **7**(5), 377–388 (2014)

20. Kschischang, F.R., Frey, B.J., Loeliger, H.A.: Factor graphs and the sum-product algorithm. IEEE Trans. Inf. Theory **47**(2), 498–519 (2001)

21. Kuhn, H.W.: The Hungarian method for the assignment problem. Naval Res. Logist. Q. **2**(1–2), 83–97 (1955)

22. Levenshtein, V.I.: Binary codes capable of correcting deletions, insertions, and reversals. Sov. Phys. Dokl. **10**(8), 707–710 (1966)

23. Liu, J., Zhang, F., Song, X., Song, Y.I., Lin, C.Y., Hon, H.W.: What's in the name?: an unsupervised approach to link users across communities. In: Proceedings of ACM International Conference on Web Search and Data Mining (2013)

24. Liu, S., Wang, S., Zhu, F.: Structured learning from heterogeneous behavior for social identity linkage. IEEE Trans. Knowl. Data Eng. **27**(7), 2005–2019 (2015)

25. Liu, S., Wang, S., Zhu, F., Zhang, J., Krishnan, R.: Hydra: large-scale social identity linkage via heterogeneous behavior modeling. In: Proceedings of the ACM SIGMOD International Conference on Management of Data (2014)

26. Malhotra, A., Totti, L., Meira Jr., W., Kumaraguru, P., Almeida, V.: Studying user footprints in different online social networks. In: Proceedings of the International Conference on Advances in Social Network Analysis and Mining (2012)

27. Motoyama, M., Varghese, G.: I seek you: searching and matching individuals in social networks. In: Proceedings of the 11th International Workshop on Web Information and Data Management (2009)

28. Narayanan, A., Shi, E., Rubinstein, B.I.P.: Link prediction by de-anonymization: how we won the kaggle social network challenge. In: Proceedings of the International Joint Conference on Neural Networks (2011)

29. Narayanan, A., Shmatikov, V.: De-anonymizing social networks. In: Proceedings of IEEE Symposium on Security and Privacy (2009)

30. Nilizadeh, S., Kapadia, A., Ahn, Y.Y.: Community-enhanced de-anonymization of online social networks. In: Proceedings of ACM Conference on Computer and Communications Security (2014)

31. Nunes, A., Calado, P., Martins, B.: Resolving user identities over social networks through supervised learning and rich similarity features. In: Proceedings of ACM Symposium on Applied Computing (2012)

32. Pearl, J.: Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference (1988)

33. Pedarsani, P., Figueiredo, D.R., Grossglauser, M.: A Bayesian method for matching two similar graphs without seeds. In: Proceedings of the 51st Annual Allerton Conference on Communication, Control, and Computing (2013)

34. Perito, D., Castelluccia, C., Kaafar, M.A., Manils, P.: How unique and traceable are usernames? In: Fischer-Hübner, S., Hopper, N. (eds.) PETS 2011. LNCS, vol. 6794, pp. 1–17. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-22263-4_1

35. Pishro-Nik, H., Fekri, F.: Performance of low-density parity-check codes with linear minimum distance. IEEE Trans. Inf. Theory **52**(1), 292–300 (2005)

36. Sharad, K., Danezis, G.: An automated social graph de-anonymization technique. In: Proceedings of the 13th ACM Workshop on Privacy in the Electronic Society (2014)
37. Shu, K., Wang, S., Tang, J., Zafarani, R., Liu, H.: User identity linkage across online social networks: a review. ACM SIGKDD Explorations Newsletter **18**(2), 5–17 (2017)
38. Vosecky, J., Hong, D., Shen, V.Y.: User identification across multiple social networks. In: Proceedings of the International Conference on Networked Digital Technologies (2009)
39. Wondracek, G., Holz, T., Kirda, E., Kruegel, C.: A practical attack to de-anonymize social network users. In: Proceedings of IEEE Symposium on Security and Privacy (2010)
40. Zafarani, R., Liu, H.: Social computing data repository at ASU (2009). http://socialcomputing.asu.edu
41. Zafarani, R., Liu, H.: Connecting corresponding identities across communities. In: Proceedings of the 3rd International AAAI Conference on Web and Social Media (2009)
42. Zafarani, R., Liu, H.: Connecting users across social media sites: a behavioral-modeling approach. In: Proceedings of ACM SIDKDD Conference on Knowledge Discovery and Data Mining (2013)
43. Zhou, F., Liu, L., Zhang, K., Trajcevski, G., Wu, J., Zhong, T.: DeepLink: a deep learning approach for user identity linkage. In: Proceedings of IEEE International Conference on Computer Communications, pp. 1313–1321. IEEE (2018)