Profile Matching Across Unstructured Online Social Networks

Volkan Kucuk[†] and Erman Ayday[‡⋆]

Bilkent University, Computer Engineering Department
Ankara, Turkey
[†]volkan.kucuk@bilkent.edu.tr
[‡]erman@cs.bilkent.edu.tr

Abstract. In this work, we propose a profile matching (or deanonymization) attack for unstructured online social networks (OSNs) in which similarity in graphical structure cannot be used for profile matching. We consider different attributes that are publicly shared by users. Our proposed framework mainly relies on machine learning techniques and graphical models. Our preliminary results indicate that profiles of the users in different OSNs can be matched with high probability by only using publicly shared attributes and without using the underlying graphical structure of the OSNs.

1 Introduction

An online social network (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 [2]. With the widespread availability of the Internet, especially via mobile devices, 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 [1].

The most common (and basic) types of information that is shared by the individuals in OSNs include name, last name, age, gender, location, date of birth, profile photo, and e-mail address [2]. However, such information is usually incomplete and inconsistent across different OSNs. Also, in some OSNs (e.g., Facebook), users mostly reveal their real identities (e.g., to find old friends), while in some OSNs users mainly prefer to remain anonymous (especially in OSNs in which users share sensitive information about themselves, such as health status).

⋆ Erman Ayday is supported by MSCA Individual Fellowship from the European Commission and by the Scientific and Technological Research Council of Turkey, TUBITAK, under Grant No. 115C130.

It is trivial to link profiles of individuals across different OSNs in which they share their real identities. However, such profile matching is both nontrivial and sometimes undesired if individuals do not reveal their real identities in some OSNs. While profile matching is useful for online service providers to build whole profiles of individuals (e.g., to provide better personalized advertisement), it also has serious privacy concerns. If an attacker can link anonymous profiles of individuals to their real identities (via their other OSN accounts in which they share their real identity), he 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 against the individuals for discrimination or blackmailing. Thus, it is very important to quantify and show the risk of such attacks and provide countermeasures against such profile matching attacks.

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). Therefore, it is natural to expect that the graphical structures of such OSNs will be similar to each other as well. Existing work shows that this similarity in graphical structure (along with some background information) can be utilized to link accounts of individuals from different OSNs [13]. 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 structure does not resemble the real-life connections of the individuals. A good example for the latter is patientslikeme.com. In this OSN, people share sensitive information about themselves such as their health conditions, diseases, diagnosis, and the drugs they use, and hence most users do not share their real identity. Users also follow each other, especially if they have similar diseases (e.g., to get more information about the alternative treatments of the disease). Thus, the OSN has a graphical structure, but this structure has no similarity to the real-life connections of the users, and hence an attacker cannot use the graphical structure to link the accounts of patientslikeme users to their accounts in other OSN (in which they share their identifiers). This does not mean that users of such OSNs are protected against profile matching (or deanonymization) attacks. In this type of OSNs, an attacker can utilize the attributes of the users across different OSNs to do the profile matching.

In this work, we propose a profile matching (or deanonymization) scheme that quantifies and shows the risk of the profile matching attack in unstructured OSNs. We show the threat between an auxiliary OSN (in which users share their real identities) and an anonymous OSN (in which users prefer to make anonymous sharings). The proposed scheme matches user profiles across multiple OSNs by using machine learning and graphical techniques. We mainly focus on two types of attacks (i) targeted attack, in which the attacker selects a set of victims from the auxiliary OSN and wants to determine the profiles of the victims in the anonymous OSN, and (ii) global attack, in which the attacker

wants to deanonymize the profiles of all the users that are in the anonymous OSN (assuming they have accounts in the auxiliary OSN). Our preliminary results show that by using a linear regression model, individuals' profiles can be matched with %80 accuracy. Note that the proposed framework can also be used in structured OSNs (along with the structural information).

The rest of the paper is organized as follows. In the next section, we discuss the threat model. In Section 3, we detail the proposed framework for profile matching in unstructured OSNs. In Section 4, we show the results of the proposed model by using real data. In Section 5, we summarize the related work and the main differences of this work from the existing work in the area. Finally, in Section 6, we discuss the future work and conclude the paper.

2 Threat Model

For simplicity, we consider two OSNs to describe the threat: (i) A, the auxiliary OSN that includes the profiles of individuals with their identifiers, and (ii) T, the target OSN that includes anonymous profiles of individuals. In general, the attacker knows the identity of the individuals from OSN A and depending on the type of the attack, he wants to determine the real identities of the user(s) in OSN T by only using the attributes of the users (i.e., information that is publicly shared by the users). The attacker can be a part (user) of both OSNs and it can collect publicly available data from both OSNs (e.g., via crawling). We assume that the attacker is not an insider in T. That is, the attacker cannot use the IP address, access patterns, or sign up information of the victim for profile matching (or deanonymization).

We consider two different attacks (i) targeted attack, and (ii) global attack. In the targeted attack, the attacker wants to deanonymize the anonymous profile of a victim (or a set of victims) in OSN T, using the unanonymized profile of the same victim in OSN A. In the global attack, the attacker's goal is to deanonymize the anonymous profiles of all individuals in T by using the information in A.

3 Proposed Model

Let A and T represent two OSN (A is the auxiliary OSN and T is the target OSN) in which people publicly share attributes such as date of birth, gender, and location. Profiles of a user in either A or T is also represented as U_i^k, where $k \in \{A, T\}$. In this work, we consider the profile of a user i as $U_i^k = \{n_i^k, \ell_i^k, g_i^k, p_i^k\}$, where n denotes the user name, ℓ denoted the location, g denotes the gender, and p denotes the profile photo. As discussed, the main goal of the attacker is to link the profiles between these two OSNs. General notations that are used in proposed model are presented in Table 1. Furthermore, the overview of the proposed framework is shown in Figure 1.

In general, the proposed scheme is composed of two main parts: (i) Steps 1-4 (In Figure 1) constitute the training part and they are the offline steps of the algorithm, and (ii) Steps 5-7 are the attack part. In Step 1, profiles and attributes

U_i^A	Profile of user i in OSN A
U_j^T	Profile of user j in OSN T
$S(U_i^A, U_j^T)$	General similarity of profiles two profiles U_i^A and U_j^T
$S(n_i^A, n_j^T)$	Username similarity of U_i^A and U_j^T
$S(\ell_i^A, \ell_j^T)$	Location similarity of U_i^A and U_j^T
$S(g_i^A, g_j^T)$	Gender similarity of U_i^A and U_j^T
$S(p_i^A, p_j^T)$	Photo similarity of U_i^A and U_j^T

Table 1: Symbols and notations used in this work.

of a set of users are obtained from both OSNs to construct the training dataset. We denote the set of profiles that are extracted for this purpose from OSNs A and T as $\mathrm{A_t}$ and $\mathrm{T_t}$, respectively. We assume that profiles are selected such that some profiles in $\mathrm{A_t}$ and $\mathrm{T_t}$ belong to the same individuals and some do not (more details on collecting such profiles can be found in Section 4.1).[1] We let set G include pairs of profiles (U_i^A, U_j^T) from $\mathrm{A_t}$ and $\mathrm{T_t}$ that belong to the same individual. That is, both profiles in the pair (U_i^A, U_j^T) belongs to the same individual. Similarly, we let set I include pairs of profiles (U_i^A, U_j^T) from $\mathrm{A_t}$ and $\mathrm{T_t}$ that belong to different individuals.

In Step 2, for each user in set $\mathrm{A_t}$, we compute the attribute similarity with each user in set $\mathrm{T_t}$ by using the metrics that are discussed in Section 3.1. In Step 3, we label the pairs in sets G and I and add them to the training dataset. If the pair is in set G, we label the pair as "1", otherwise we label it as "0". In Step 4, we fit our dataset into a linear regression (LR) model to learn the contribution (or weight) of each attribute to the profile matching attack (details of this step are discussed in Section 3.2). In Step 5, the attack type is determined and profiles to be matched are selected, and hence sets A and T are constructed. For simplicity, we assume set A includes N users from A and set T includes N users from T.[2] In Step 6, every profile in set A is paired with every profile in set T and the similarity between each pair is computed by using the created LR model. At the last step (Step 7), profiles in sets A and T are paired by maximizing similarities using a graph-based algorithm as discussed in Section 3.3.

3.1 Similarity Metrics

To create the model, it is required to define similarity metrics for the attributes in U_i^k. In this section, we provide the details of the similarity metrics we propose for each attribute.

User name similarity ($S(n_i^A, n_j^T)$) We use Levenshtein distance [10] to calculate the similarity between user names of profiles. Given the user names n_i^A and n_j^T of two profiles i and j with profiles U_i^A, and U_j^T, and assuming m and n represent the indices of the characters in these user names, Levenshtein distance

[1] Such profiles are required to construct the ground-truth for training.
[2] Sets A and T do not include any users from sets $\mathrm{A_t}$ and $\mathrm{T_t}$.

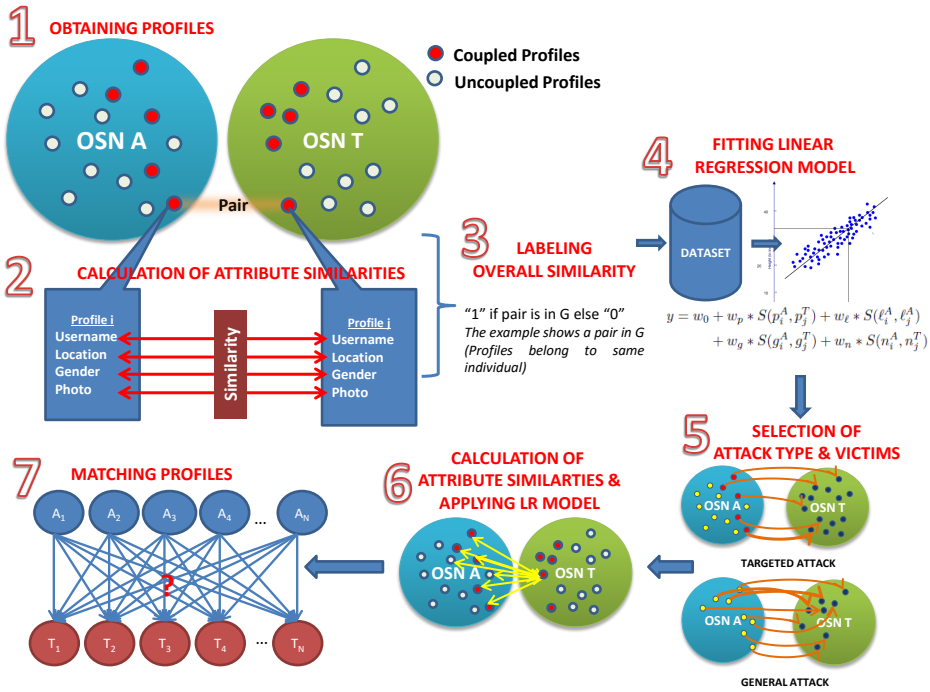


Fig. 1: Overview of the proposed profile matching framework.

between the user names can be computed as below.

$$lev_{(n_i^A, n_j^T)}(m,n) = \begin{cases} max(m,n), & \text{if } min(m,n) = 0 \\ min \begin{cases} lev_{n_i^A, n_j^T}(m-1,n) + 1 \\ lev_{n_i^A, n_j^T}(m,n-1) + 1 \\ lev_{n_i^A, n_j^T}(m-1,n-1) + 1_{n^A{}_i \neq n^T{}_j}, \end{cases} & \text{otherwise} \end{cases} \tag{1}$$

Thus,

$$S(n_i^A, n_j^T) = 1 - \frac{lev_{(n_i^A, n_j^T)}(m,n)}{K}, \tag{2}$$

where, K is a normalization constant.

Location similarity ($Sim(\ell_i^A, \ell_j^T)$) Almost all OSNs has location or hometown information available in user profiles. Even if such location information is not directly available, other types of information can be used to predict location

of a profile. For instance, timezone of a profile, location of a tweet, or location-related information in the freetext can be used to predict of the location (or hometown) of a profile. Location information collected from the users' profiles is usually text based, and hence one needs to convert the text-based location information into latitude/longitute data to calculate the geodesic difference between two profiles. We do this conversion via GoogleMaps API [3] and compute the location similarity between two user profiles as below.

$$\begin{aligned} Lat(\ell_i^A), Lon(\ell_i^A) &= GetCoordinate(\ell_i^A) \\ Lat(\ell_j^T), Lon(\ell_j^T) &= GetCoordinate(\ell_j^T) \\ GD = GeographicDistance(Lat(\ell_i^A), Lon(\ell_i^A), Lat(\ell_j^T), Lon(\ell_j^T)) \\ S(\ell_i^A, \ell_j^T) &= 1 - Norm(GD) \end{aligned} \tag{3}$$

Here, $Norm$ is a normalization function based on all computed values in the dataset.

Gender similarity ($\boldsymbol{S(g_i^A, g_j^T)}$) 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 such cases, a profile's gender can be predicted from other information. In our model, if an OSN does not provide 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[3] and look for a profile's name (or user name) to probabilistically infer the possible gender of the profile. We then use this probability as the $S(g_i^A, g_j^T)$ value between two profiles.

Profile photo similarity ($\boldsymbol{S(p_i^A, p_j^T)}$) Profile photo similarity is calculated through Microsoft FaceAPI [4] which is Microsoft's state-of-the-art cloud-based application to detect and recognize human faces in images. FaceAPI can perform face recognition, as well as age and gender estimation. Given two profile photos p_i^A and p_j^T, FaceAPI returns the photo similarity, $Sim(p_i^A, p_j^T)$, as a real value from $[0, 1]$.

3.2 Learning Weights of Attributes

Prior work shows that linear regression is an efficient way to learn attribute weights for social network related data [1]. Therefore, we apply linear regression to our training dataset in order to learn the contribution (or weight) of each attribute for the profile matching attack. Notations we use for this part are listed in Table 2.

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

w_n	Weight of user name
w_ℓ	Weight of location
w_g	Weight of gender
w_p	Weight of profile photo

Table 2: Notations used in the linear regression.

As discussed, we first construct sets $\mathrm{A_t}$ and $\mathrm{T_t}$ for training. Also, set G includes pairs of profiles (U_i^A, U_j^T) that belong to the same individual and set I includes pairs of profiles (U_i^A, U_j^T) from $\mathrm{A_t}$ and $\mathrm{T_t}$ that belong to different individuals. We refer to the pairs in G as "coupled profiles" and the ones in I as "uncoupled profiles".

We first compute the individual attribute similarities between each profile in $\mathrm{A_t}$ and in $\mathrm{T_t}$ using the similarity metrics described in Section 3.1. Then, we learn the contribution (or weight) of each attribute for the profile matching attack via linear regression using (4). Note that to avoid over sampling on coupled profiles, uncoupled profile pairs are also added to the training dataset.

$$S(U_i^A, U_j^T) = \begin{cases} 1 = y, & \text{if } (U_i^A, U_j^T) \in \mathrm{G} \\ 0 = y, & \text{if } (U_i^A, U_j^T) \in \mathrm{I} \end{cases} \tag{4}$$

where,

$$\begin{aligned} y = w_0 + w_p * S(p_i^A, p_j^T) + w_\ell * S(\ell_i^A, \ell_j^A) \\ + w_g * S(g_i^A, g_j^T) + w_n * S(n_i^A, n_j^T) \end{aligned} \tag{5}$$

As a result of this process, we learn the weights of the attributes which completes the training part of the proposed scheme (Steps 1-4 in Figure 1). Then, we use these learnt weights to compute the general similarity between each user pair in A and T. That is, we compute $S(U_i^A, U_j^T)$ between every user in A and T. Next, we use these general similarity values for the profile matching attack as discussed in the next section.

3.3 Matching Profiles

As discussed, for profile matching attack, we consider the users in sets A and T from the auxiliary and the target OSNs. For simplicity, we also assume that both sets include N users.[4] Before the actual profile matching, individual attribute similarities between each profile in A and in T are computed using the similarity metrics described in Section 3.1. Then, the general similarity $S(U_i^A, U_j^T)$ is computed between every user in A and T using the weights determined in Section 3.2. Let Z be a $N \times N$ similarity matrix that is constructed from the pairwise similarities between the users in A and T as below:

$$Z = \begin{bmatrix} S(U_0^A, U_0^T) & \dots & S(U_0^A, U_N^T) \\ \vdots & \ddots & \vdots \\ S(U_N^A, U_0^T) & \dots & S(U_N^A, U_N^T) \end{bmatrix}$$

[4] The case when the sizes of the OSNs are different can be also handled similarly (by padding one OSN with dummy users to equalize the sizes).

Our goal is to obtain a one-to-one matching between the users in A and T that would also maximize the total similarity. To achieve this matching, we use the Hungarian algorithm, a combinatorial optimization algorithm that solves the assignment problem in polynomial time [9]. The objective function of the Hungarian algorithm can be expressed as below:

$$max \sum_{i=1}^{N} \sum_{j=1}^{N} -Z_{ij} x_{ij},$$

where, $-Z_{ij}$ represents the similarity between U_i^A and U_j^T (i.e., $S(U_i^A, U_j^T)$). Also, x_{ij} is a binary value, that is, $x_{ij} = 1$ if profiles U_i^A and U_j^T are matched as a result of the algorithm, and $x_{ij} = 0$ otherwise. After performing the Hungarian algorithm to the Z matrix, we obtain a matching between the users in A and T that maximizes the total similarity. Note that we multiply Z_{ij} values with -1, in order to obtain maximum similarity (profit).

4 Evaluation

In this section, we evaluate the proposed algorithm by using real data from two OSNs.

4.1 Data Collection

In the literature there are limited datasets that can be used for profile matching between unstructured OSNs. Thus, to evaluate our proposed framework, we created a dataset that consist of users from two OSNs with several attributes. The flowchart of our data collection process in shown in Figure 2. The most challenging part of data collection was to obtain the "coupled" profiles between OSNs that belongs to same person in real-life (i.e., to generate set G in Section 3.2). To automate the coupled profile collection process, we used another social network, About.me [5]. About.me is a social directory, in which people share their social account links from most popular OSNs such as, Facebook, Twitter, Foursquare, and LinkedIn.

For the evaluation, we focused on two major OSNs: Twitter and Foursquare. In this setup, we used Foursquare as our auxiliary OSN (A) and Twitter as our target OSN (T). As shown in the Figure 2, if a given About.me profile includes links to both Twitter and Foursquare accounts of a user, we pair those accounts as a coupled profile. Furthermore, we also randomly pair uncoupled profiles which are used for training and testing the algorithm. In general, we used About.me API, Twitter API (through twitter4j [6] Java library), and Foursquare API for the data collection. From each coupled or uncoupled profile, we extracted the following information: (i) from Foursquare; firstname/lastname, last tip of user (a freetext comment about a location), gender, location, and profile photo, and (ii) from Twitter; name/screenname, last tweet time of the user, location, profile photo, and timezone. As a result, the constructed dataset consists of 1500 profile pairs, of which 900 are coupled and 600 are uncoupled profiles.

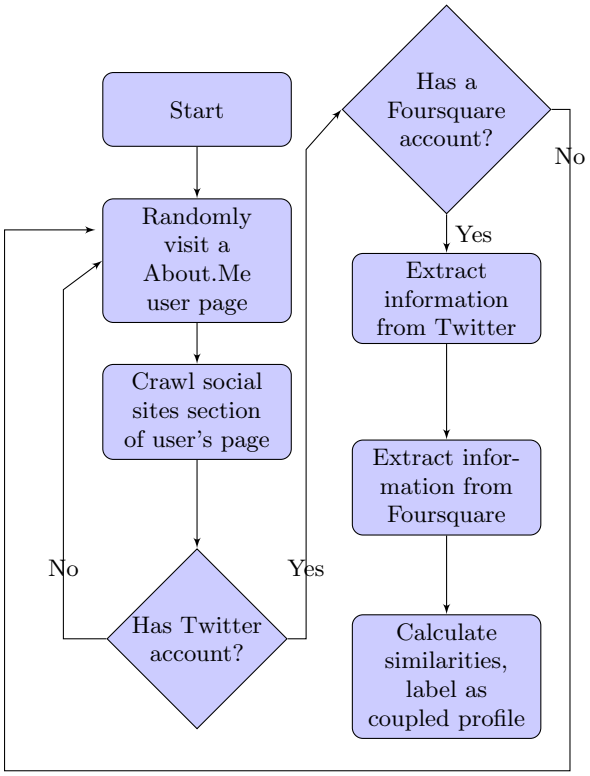


Fig. 2: Data collection flowchart to obtain the coupled profiles.

4.2 Learning Weights of Attributes

After creating the dataset, we selected 800 profile pairs for training. These pairs consists of 500 coupled and 300 uncoupled profile pairs. To learn weights of attributes, we performed linear regression to our dataset (as discussed in Section 3.2). As a result, we obtained the weights in Table 3. We observe that location, user name, and profile photo attributes are the most important attributes to determine whether two profiles belong to same individual or not.

Attribute	Weight
user name	0.8909
location	0.7468
gender	0.2786
profile photo	0.4278

Table 3: Weights of attributes computed as a result of linear regression.

Following our linear regression model, we compute the general similarity between profiles U_i^A and U_j^T as follows:

$$\begin{aligned} S(U_i^A, U_j^T) = 0.4278 * S(p_i^A, p_j^T) + 0.7468 * S(\ell_i^A, \ell_j^T) \\ +0.2786 * S(g_i^A, g_j^T) + 0.8909 * S(n_i^A, n_j^T) - 1.0034 \end{aligned} \tag{6}$$

After learning the attribute weights, we selected 300 users from Twitter and 300 users from Foursquare. Note that none of these users were involved in the training set. Among these profiles we had 50 coupled pairs. Thus, we evaluated the success of our proposed framework based on the matching between these coupled profiles. All similarity calculations were conducted by using (6) to create the similarity matrix Z, as discussed in Section 3.3.

4.3 Profile Matching

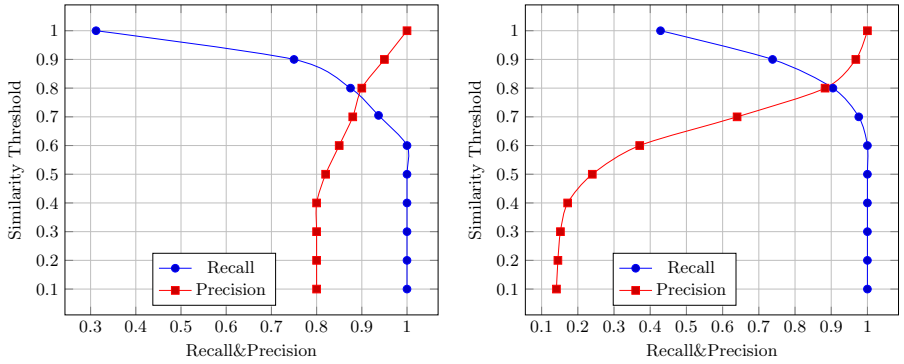
To evaluate our model, we consider two types of profile matching attacks: (i) targeted attack, and (ii) global attack. In targeted attack, the goal of the attacker is to match the anonymous profiles of 20 target individuals from T (Twitter) to their corresponding profiles in A (Foursquare). In the global attack, the goal of the attacker is to match all $N = 300$ profiles in A to all $N = 300$ profiles in T. In other words, the goal is to deanonymize all anonymous users in the target OSN (who has accounts in the auxiliary OSN). Note that in our dataset, only 50 users are common (i.e., coupled) between the two OSNs, and hence the goal is to make sure that these 50 users are matched with high confidence. In both targeted and global attacks, we use the Hungarian algorithm for profile matching between the auxiliary and the target OSN (as discussed in Section 3.3). The evaluation criteria of the proposed framework is based on the type of attack. Thus, next, we present the metrics we use for the evaluation.

4.4 Evaluation Metrics and Results

Hungarian algorithm provides a one-to-one match between all the users in the auxiliary and the target OSN. However, as mentioned, we cannot expect that all anonymous users in the target OSN to have profiles in the auxiliary OSN (we are only interested in the ones that have profiles in both OSNs). Therefore, some matches provided by the Hungarian algorithm are useless for us. Thus, we define a confidence value and we only consider the matches that are above the confidence value to evaluate our framework. For this purpose, we set a "similarity threshold". We examine the general similarity values between the matches as a result of the Hungarian algorithm and we only consider the matches that are above the similarity threshold.

For the evaluation, we consider the precision and recall values obtained as a result of both attack scenarios. As shown in Figure 3, we obtained a precision value of around %84 for a similarity threshold of 0.8, which means that we could correctly match 42 coupled profiles out of 50 (that has an overall similarity over 0.8) in global attack. Thus, if our linear model (Equation 6) returns a similarity

value that is above 0.8 for a given profile pair, we can say that the corresponding profiles belong to same individual with a high confidence. Furthermore, in targeted attack we able to match 16 profiles out of 20. These preliminary results indicate the importance of publicly shared attributes of users for profile matching.



(a) Targeted Attack (b) Global Attack

Fig. 3: Precision-recall curve for different values of similarity threshold.

5 Related Work

In the literature, most works focus on profile matching (or deanonymization) by using structural information which mainly relies on network structure of OSN. Narayanan and Shmatikov show that statistical approaches on high-dimensional micro-data can deanonymize potentially sensitive information [12]. In another work, Narayanan and Shmatikov propose a framework for analyzing privacy and anonymity in social networks and a deanonymization algorithm that is based purely on the network topology [13]. Ji et.al propose a secure graph data sharing/publishing system [8] in which they implement and evaluate graph data anonymization algorithms, data utility metrics, and modern Structure-based De-Anonymization (SDA) attacks. Furthermore, Ji et.al quantify deanonymizability and partial deanonymizability of real world social networks with seed information where a social network follows an arbitrary distribution model [7]. As opposed to these works, we use only publicly available unstructural data for profile matching.

In one of the recent works, Liu et. al propose a framework called HYDRA that uses both structural and unstructural information to match profiles across OSNs [11]. In a nutshell, the proposed framework has three steps named behavior similarity modeling, structure consistency modeling, and multi-objective optimization. In this work, we use a different technique for profile matching, consider different attributes, and we do not utilize the structural information at all.

6 Conclusion and Future Work

In this work, we proposed a framework for profile matching in unstructured online social networks (OSNs). Our results show that using only unstructural information, users' profiles in different OSNs can be matched with high precision. As future work, we will (i) increase the dataset size to obtain more generalized results, (ii) target OSNs with sensitive data to show how an attacker can abuse sensitive information, (iii) apply different machine learning algorithms on collected data, (iv) directly compare our framework with other proposed solutions in the literature, (v) consider more features (especially the ones that are not obvious identifiers) such as freetext generated by the users, activity patterns of the users, and interests of the users, and (vi) propose countermeasures against the proposed attack.

References

1. Souvik Debnath, Niloy Ganguly, and Pabitra Mitra. Feature weighting in content based recommendation system using social network analysis. In *Proceedings of the 17th international conference on World Wide Web*, pages 1041–1042. ACM, 2008.
2. Nicole B Ellison et al. Social network sites: Definition, history, and scholarship. *Journal of Computer-Mediated Communication*, 13(1):210–230, 2007.
3. https://developers.google.com/maps/.
4. https://www.microsoft.com/cognitive-services/en-us/face api.
5. http://www.about.me.
6. http://www.twitter4j.org.
7. Shouling Ji, Weiqing Li, Neil Zhenqiang Gong, Prateek Mittal, and Raheem A Beyah. On your social network de-anonymizablity: Quantification and large scale evaluation with seed knowledge. In *NDSS*, 2015.
8. Shouling Ji, Weiqing Li, Prateek Mittal, Xin Hu, and Raheem Beyah. Secgraph: A uniform and open-source evaluation system for graph data anonymization and de-anonymization. In *24th USENIX Security Symposium (USENIX Security 15)*, pages 303–318, 2015.
9. Harold W Kuhn. The hungarian method for the assignment problem. *Naval research logistics quarterly*, 2(1-2):83–97, 1955.
10. Vladimir I Levenshtein. Binary codes capable of correcting deletions, insertions, and reversals. In *Soviet physics doklady*, volume 10, pages 707–710, 1966.
11. Siyuan Liu, Shuhui Wang, Feida Zhu, Jinbo Zhang, and Ramayya Krishnan. Hydra: Large-scale social identity linkage via heterogeneous behavior modeling. In *Proceedings of the 2014 ACM SIGMOD international conference on Management of data*, pages 51–62. ACM, 2014.
12. Arvind Narayanan and Vitaly Shmatikov. Robust de-anonymization of large sparse datasets. In *Security and Privacy, 2008. SP 2008. IEEE Symposium on*, pages 111–125. IEEE, 2008.
13. Arvind Narayanan and Vitaly Shmatikov. De-anonymizing social networks. In *Security and Privacy, 2009 30th IEEE Symposium on*, pages 173–187. IEEE, 2009.