Privacy in Multiple On-line Social Networks - Re-identification and Predictability

David F. Nettleton[*], **Vladimir Estivill-Castro**[*], **Julián Salas**[**]

[*]Web Science and Social Computing Research Group, Department of Information and Communications Technology (DTIC), Universitat Pompeu Fabra, UPF Tanger Building, 08018 Barcelona, Catalonia, Spain.

[**]Internet Interdisciplinary Institute (IN3), Universitat Oberta de Catalunya (UOC), Parc Mediterrani de la Tecnologia (Edifici B3), Av. Carl Friedrich Gauss, 5, 08860 Castelldefels (Barcelona), Spain.

E-mail: david.nettleton@upf.edu, vladimir.estivill@upf.edu, jsalaspi@uoc.edu

Received 11 February 2018; received in revised form 1 November 2018; accepted 19 November 2018

Abstract. We consider the re-identification of users of on-line social networks when they participate in several different on-line social networks, potentially using several different accounts. The re-identification of users serves several purposes: (i) commercial use so as to avoid redundant mailing to the same user; (ii) enhancement of the information available about these users by unifying information from different sources; (iii) consolidation of accounts by on-line social network providers; (iv) identification of potentially malicious users and/or bots. We highlight that all this should occur within the bounds of the data protection and privacy laws as well as the users' expectations on such matters to avoid backlash. In this paper, we explore this situation first by a formalization using the *SAN* model to conceptually structure information as a graph, which includes user and attribute type nodes. This formalization enables us to reason on two issues. First, how to identify that two or more user-accounts belong to the same user. Second, what gains in predictability are obtained after re-identification. For the first issue, we show that a set-difference approach is remarkably effective. For the second issue we explore the impact of re-identification on the predictability by two different machine learning algorithms: C4.5 (decision tree induction) and SVM-SMO (Support Vector Machine with SMO kernel). Our results show that as predictability improves, in some cases different *SAN* metrics emerge as predictors.

Keywords. Data Privacy, On-line Social Networks, Graph Representation, Re-Identification, User Link and Attribute Prediction.

1 Introduction

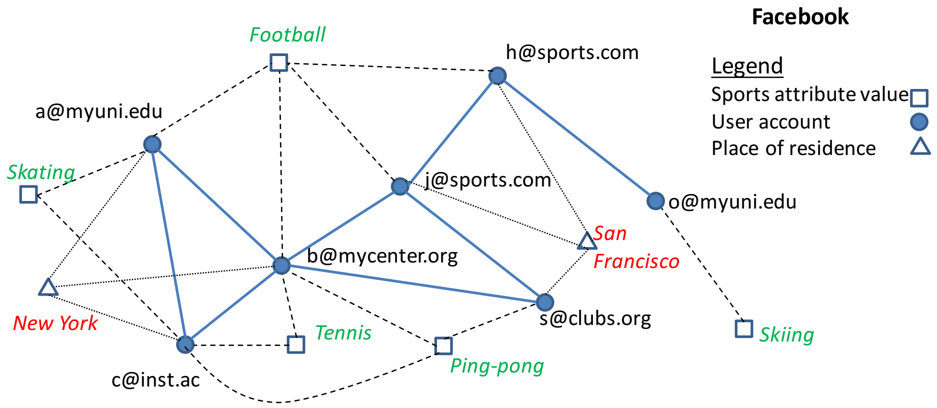
Users subscribe to different on-line applications, such as On-line Social Networks (OSNs), content providers and e-mail services because they offer complementary functionalities which give a rich and diverse user experience on the Internet. Also, the information a user defines in each application and their activity in those applications offers the potential to construct an aggregated footprint [20]. A data analyst can then use such personal footprints to improve targeting of commercial services or consolidate user-accounts within an online application. Recent trends in the IT Industry which have led to the amalgamation of major Web application providers into the same group of companies set up the context for such analyses. For example, Google now owns YouTube (formerly, one a search engine and the latter, a video sharing service), as well as the Google+ social network. Facebook has acquired WhatsApp. Should such corporations be able to analyze users' usage-data as different services become combined? Should users be provided with alerts of the inferences obtained by such amalgamations? The organizations managing such personal data could also benefit the user experience by re-identification. In particular, redundant marketing can be avoided, and consolidation of accounts and an improved service could be offered. In this paper, we evaluate and quantify how users of multiple on-line social networks who may also have multiple user-accounts in the same OSN can be re-identified based on user-user and user-attribute links which form a distinctive 'footprint' pattern.

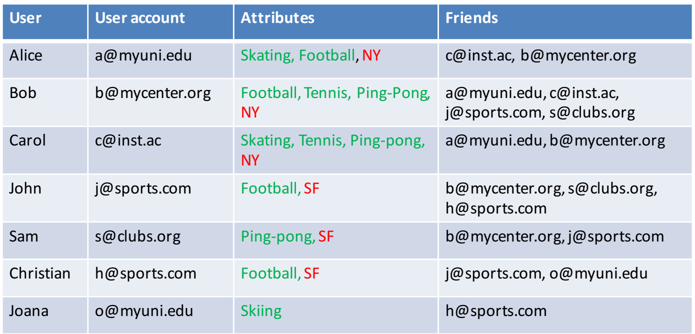
The original contributions of the paper are: (1) to verify that the prediction obtained from the fusion of two (or more) OSNs is superior to the prediction obtained from using them separately; (2) to show how the privacy of an individual is closely related to the privacy of others, so that the revelation of sensitive attributes from quasi-identifiers of a given individual can lead to the revelation of sensitive attributes of other individuals; (3) to show how some attributes and social relations of OSN users' may be predicted in one social network, using data from a different social network; (4) the use in this context of a graph data model called *SAN* (Social-Attribute Network) which allows us to represent and reason on user-user relations and user-attribute relations in a unified scheme.

In order to address these challenges, two relevant issues have been tackled: firstly, how to identify that two or more user-accounts belong to the same user, and secondly, what type of gains in predictability are obtained after re-identification. For the first issue we show that a set-difference approach is remarkably effective. For the second issue we explore the impact of re-identification on the predictability by two different machine learning algorithms: C4.5 (decision tree induction) and SVM-SMO (Support Vector Machine with `SMO` kernel). Our results show that as predictability increases, different combinations of *SAN* metrics emerge as the predictors.

By generating such models we also study the possibility of attribute disclosure, considering how a combination of attribute values from several users may predict the values for another users' attributes, even when the latter have not been provided by the users in the network. In this sense, such attributes play the role of quasi-identifiers (or predictors) for other sensitive attributes. This has been previously studied in [8, 9, 10] with the aim of empowering users and raising their awareness on the possibility of involuntarily revealing their confidential attributes by publishing non-confidential ones.

The paper is organized as follows: Section 2 presents related work; Section 3 presents a formalization of the problem, and briefly describes the machine learning algorithms which we use for building predictive models. In this section, we also present the *SAN* model and metrics used as features for learning and inference. For illustration, in Section 4 we provide an example of two OSNs and their amalgamation, represented using the *SAN* model. In Section 5 we apply the ideas proposed in the examples of Section 4 to a bigger dataset derived from a real on-line social network. We demonstrate empirically that a set-difference results in a remarkably effective re-identification prediction. Such re-identification then enables building a larger *SAN* and expand on the prediction of user's links and attribute-values. However, by exploring using two state-of-the-art supervised learning algorithms (`C4.5`, `SVM-SMO`), we observe that several predictors exist. This phenomena could be seen as detrimental to the privacy of individuals but beneficial to the aim of completing the footprint image of users. Section 6 summarizes the present work.





User	User account	Attributes	Friends
Alice	a@myuni.edu	Skating, Football, NY	c@inst.ac, b@mycenter.org
Bob	b@mycenter.org	Football, Tennis, Ping-Pong, NY	a@myuni.edu, c@inst.ac, j@sports.com, s@clubs.org
Carol	c@inst.ac	Skating, Tennis, Ping-pong, NY	a@myuni.edu, b@mycenter.org
John	j@sports.com	Football, SF	b@mycenter.org, s@clubs.org, h@sports.com
Sam	s@clubs.org	Ping-pong, SF	b@mycenter.org, j@sports.com
Christian	h@sports.com	Football, SF	j@sports.com, o@myuni.edu
Joana	o@myuni.edu	Skiing	h@sports.com

Figure 1: A sample social network of hobbies and ludic interests

2 Related Work

In a recent study, Singh et al. [20] analyzed the privacy risk of a user's publicly available information from multiple on-line social applications, such as Google+ and LinkedIn. Naturally, if more data is available, machine learning algorithms can usually build more accurate classifiers, and if more predictor attributes are known about an individual, more alternative classifiers can be constructed to infer an attribute-value pair that a user considers confidential. Fig. 1 displays an example of a simple social network of hobbies and ludic interests. The service provider may not know the identity of the users, as it only knows the user-accounts identified by e-mail addresses. While none of this data may be confidential, more data about the users of this OSN may enable inferring attribute values these users may consider confidential (for example Joana has not disclosed her city of residence). The work by Singh et al. [20] considered as predictors only the attributes of the user itself, without taking into account neighbor connections and neighbor attributes. Singh et al. [20] define a *footprint* for OSN users which represents the union of the attribute-value pairs derived from different on-line, public domain sources. Much more information can be derived when the neighbor connections and neighbor attributes are also taken into account [18, 19], but previous work considered separate OSNs. The *SAN* [11] model formalizes an on-line social network as the mathematical model of a graph. This model uses two types of vertices: *social vertices* which represent the users and *attribute vertices* which represent the attributes. Edges can exist between social vertices and between social vertices and attribute vertices. Yin et al. [24, 25] presented a first version of the *SAN* model under the name of *augmented graph*. We remark that the *SAN* model for social vertices represents user-accounts and not user's themselves. For example, in Fig. 1, user Christian is known to the service operator as h@sports.com (but it is possible that the same user created more than one user-account). The original *SAN* represented all the information available as vertices and edges. For example, in Fig. 1, solid edges represent social neighbors (the friends). On the other hand, black dashed edges signify known information about a user-account, such as o@myuni.edu has *Skiing* as *sport* (we represent different attributes with different dashing styles). The attributes *sport* having the value *Skiing* is known as an attribute-value pair for user-account o@myuni.edu. However, in this paper we are going to consider the link between user Joana and her user-account o@myuni.edu as unknown. But we consider such knowledge as an inferable link (however, if Joana participates in the same OSN with two user-accounts, their identification, while profitable for the service provider may be an invasion of privacy from Joana's point of view). Similarly, if user-accounts in different OSNs are identified as belonging to the same person.

The *SAN* enables the definition of a family of metrics for two classes of edges. The CN metric, defined by Liben-Nowell and Kleinberg [13], is a metric on two social vertices (from our perspective, two user-accounts) that returns a count of all their mutual connections. The AA metric, defined originally by Adamic and Adar [1], takes as input a user and an attribute and returns a count of all users and attributes which they have in common. Backstrom and Leskovec [2] defined a link-prediction algorithm which used the number of common neighbors as an edge attribute. Gong et al. [11] built on such previous work, using and extending the *SAN* model for link prediction and attribute inference. In their work, Gong et al. [11] introduced the metric, CN-*SAN*, which given two users, returns all their mutual connections plus their mutual attributes. We re-introduce the formal definitions later in Section 3.5.

Estivill et al. [8] first demonstrated how to calculate personalized risk for users, by using an information gain calculation on sets of attributes found heuristically using forests of decisions trees derived from the OSN's data. That is, the information gain is used to quantify the risk for quasi-identifiers concerning a sensitive attribute designated by the individual user. It is important to emphasize that the risk is a personalized risk value calculated for each user u and a specified attribute-value pair. In our example, Joana may wish to keep her *place of residence* confidential although others disclose it. This focus on the individual's choice of what is sensitive differentiates the approach from the typical method of building a general model for all users which is then used to predict a risk value for each user. The approach is tested only on the attribute-data (no evaluation is made of the links that appear between users in social networks) [8]. Estivill and Nettleton [9, 10] extended this method [8] to OSN graphs using Gongs *SAN* model. That is, the *SAN* metrics (see Section 3.5) can be included as quasi-identifiers (they become derived features) to predict an attribute-value pair a particular user considers confidential. The *SAN* metrics can be interpreted as the strength of the relation between two social nodes or between an attribute-value pair and a social node. In essence, such inference is predicting an edge of those which should be illustrated as dotted lines in Fig 1, but which is not present. In our example, the connection between o@myuni.edu (i.e. Joana) and h@sports.com (i.e. Christian), who has disclosed *San Francisco*, may be sufficient to infer this confidential information about o@myuni.edu as his/her residence. As we pointed out, in some cases, the relation between two user-accounts is what is considered a sensitive attribute, and the *SAN* metrics can act as quasi identifiers (of those edges illustrated as solid in Fig 1). Also, a specific attribute-value pair, such as *political orientation=left*, *religion=Muslim* or *illness=diabetes* can be considered a sensitive attribute. In the empirical section of this paper we extend beyond the information gain heuristic calculation [9, 10] and we use supervised learners (`C4.5` [15], `SVM-SMO` [14, 22]) to identify predictors of sensitive attributes.

This can be said to have an analogy to the imputation of missing data in a dataset. Both of these machine learning approaches enable the discovery of which attributes are more influential, or provide more information in the prediction of the class. That is, they identify predictor attributes; and thus, they have been used to protect confidential attributes in on-line social networks [9]. Therefore, our approach could be considered analogous to decision-tree-based missing-value imputation (DMI) [17], k-decision tree based missing value imputation (kDMI) [16] and k-nearest neighbour based imputation (kNNI) [3]. Although missing value imputation approaches may also result in user-account re-identification, our approach reveals what attributes or connections are more influential for re-identification. The comparison of techniques for the assignment of missing values is outside the scope of the present work.

3 Definitions and Formalization of the Problem

3.1 Encoding Social Networks as a *SAN*

As motivated already, we shall represent the i-th social network S^i as a graph $G^i = (V^i, E^i)$ with two types of vertices. The first type of vertices in V^i are user-account vertices; each account with credentials in the social network S^i is such a user-account vertex. The second type of vertices in V^i are attribute-value vertices A^i. The set of all the edges in S^i is denoted by E^i. Note that each vertex in A^i is an attribute-value pair. For example, if the attribute is *income* and the attribute-value may be *low*, *medium* or *high*, then the domain of *income* is $\{low, medium, high\}$ and then all attribute-value pairs are vertices $a_i \in A^i$. When the attribute is self-evident, we may identify an attribute-value pair just with the value. To emphasize, an attribute-value pair refers to an attribute having a specific value, like *income=medium*. That is, for a particular user-account u, we know the attribute-value pair, then we link the user-account vertex u in the social network S^i with an edge in E^i between the vertex u and the corresponding attribute-value vertex representing the attribute-value pair. For example, if we know user-account franco@myunit.edu's income, and that such income is *medium*, we place an edge between franco@myuni.edu and the vertex *income=medium*. Note that there may be another attribute (for example *promiscuity*) that also has the same domain $\{low, medium, high\}$ as *income*, but franco@myunit.edu's *promiscuity* is *low*; so there will also be an edge in E^i between the vertex representing franco@myunit.edu and the attribute-value pair *promiscuity=low*.

Also, a connection between two user-accounts in the social network S^i will be represented by an edge in E^i. A user may hold several accounts within the same social network, and the credentials to participate through the account may be different for each account within the same social network, and also within other social networks. Therefore, we will not assume that the vertex corresponding to the same user-account identifier in different networks correspond to the same vertex u in all the graphs G^i. In fact, we assume that each $G^i = (V^i, E^i)$ is a sub-graph of a much larger *SAN* which also has edges of the type "*user holds user-account*" but currently such information may not be known. For example, suppose the user Samuel Franco uses franco@myuni.edu as a professional social network and s@clubs.org for his sports interests (and therefore, he uses each e-mail address as his ID in two different social networks). Thus, there is a user vertex Samuel Franco which we may know exists, but we do not know the information that creates the edges in the super *SAN*

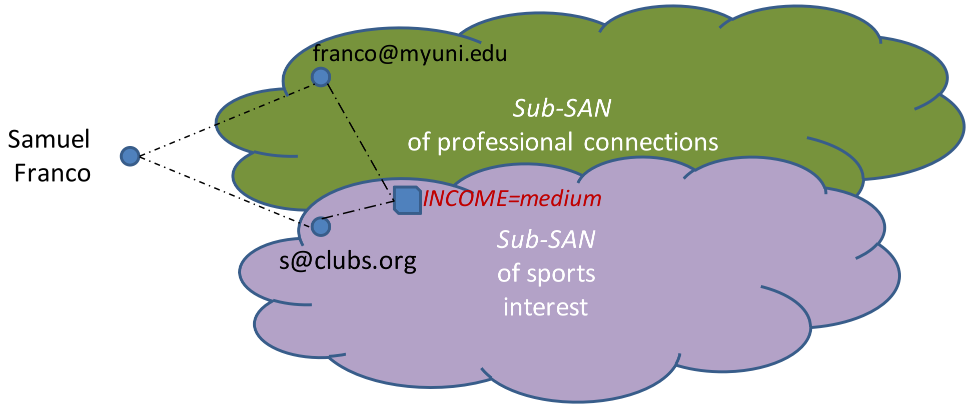


Figure 2: We suggest a super-*SAN* for what previously have been separate *SAN*s

$income = low \in S^i \rightarrow income = low \in S^j$
$income = medium \in S^i \rightarrow income = medium \in S^j$
$income = high \in S^i \rightarrow \{income = high, income = very\ high\} in S^j$
$income = low \in S^j \rightarrow income = low \in S^i$
$income = medium \in S^j \rightarrow income = medium \in S^i$
$income = high \in S^j \rightarrow income = high \in S^i$
$income = very\ high \in S^j \rightarrow income = high \in S^i$

Table 1: An illustration of a mapping defined between attributes that exists in two social networks S^i and S^j.

that incorporates the sub-*SAN*s(refer to Figure 2).

However, we assume there is a corresponding well-known mapping for the vertices representing attribute-value pairs for compatible/same attributes that belong to two different social networks. For example, suppose franco@myuni.edu uses another social network S^j and that in this network *income* is an attribute with domain $\{low, medium, high, very\ high\}$, then we assume that analysts have a correct mapping so that each attribute-value pair in S^j can be mapped to the correct value pair in S^i. For example, one such mapping is to map values as in Table 1. The important point is that we assume that attribute-value pairs for the same attribute (property) are easily identified across different social-networks. With these definitions and notation we represent the relations between user-accounts and the known properties (attribute-values pairs) for a set of several social networks.

3.2 Encoding Social Networks as an Adjacency List

The representation of the *SAN* is a conceptual model to organize the information on a social network. It may actually be represented in slightly different forms in implementations for performing analytics with its information. Typically, because the *SAN* is sparse it is represented as an adjacency list. We make the following observations, which are derived from the adjacency-list representation of a *SAN*. All social vertices have a unique integer identifier. All attribute-value pairs also have a unique integer identifier. The set of identifiers for user-accounts and the set of integers for attribute-value pairs are disjoint.

user	*income*			*promiscuity*		
	low	*medium*	*high*	*low*	*medium*	*high*
Franco	false	true	false	true	false	false

Table 2: Fragment of attributes for just one row of the tabular model for the social network information.

The information for a user-account has two lists: the list of its social neighbors and the list of its attributes. Thus, a user who holds a user-account in each *SAN* would have a record in each. When a user participates with a user-account u^1 in one social network S^1, we denote by

$$\langle u^1, \{c_{1,1}, c_{1,2}, \ldots, c_{1,t_1}, a_{1,1}, a_{1,2}, \ldots, a_{1,s_1}, \}\rangle \tag{1}$$

the record in SAN S^1 , where each $c_{1,i}$ is a connection to another user-account in SAN S^1 and each $a_{1,j}$ is an attribute-value pair also in SAN S^1. Analogously,

$$\langle u^2, \{c_{2,1}, c_{2,2}, \ldots, c_{2,t_2}, a_{2,1}, a_{2,2}, \ldots, a_{2,s_2}, \}\rangle \tag{2}$$

is the record for user-account u_2 in SAN S^2.

3.3 Encoding Social Networks as a (Boolean) Table

We present another conceptual model that enables us to organize the information for the context of machine learning. In this tabular model, we have a matrix M^i with Boolean values, for the social network S^i where each row corresponds to a user-account, and each column corresponds to an attribute-value pair. Thus, in our running example, there will be a row for franco@myuni.edu and there will be 3 columns to indicate the known income=medium. For further illustration, Table 2 shows another set of columns for a second attribute. In a real social network, the number of columns is large and the number of rows corresponds to the number of user-accounts in the social-network. However, this tabular model corresponds to the canonical representation of instances in supervised machine learning. In this case, a classifier is learned from the instances which are considered a learning set. When a user does not disclose an attribute-value pair, and considers this information confidential, the data available for this user-account and the data about many other user-accounts may enable inference of this user's attribute-value pair. This could be useful for account re-identification and to avoid duplicate marketing, or simply for issuing an alert.

3.4 The Supervised Learning Methods

Once we have the social network conceptually defined as a tabular training set, we can use machine learning methods to learn a classifier that predicts a particular attribute-value pair that a user has not disclosed. The analyst has the freedom to permute which is the dependent variable and which are the independent variables.

We have selected two methods which enable us to contrast different learning paradigms, and which are considered to be among the top ten algorithms in data mining [23]: (1) Quinlan's C4.5 algorithm [15] is widely used in data mining and induces trees in a top down fashion by splitting attributes based on information gain, which in turn has a solid theoretical basis in Shannon's information theory. This is particularly relevant for the present work; (2) The SMO [14] Support Vector Machine [22] is also widely used in data mining and offers a contrasting theoretical basis for splitting the information based on finding ″support vectors″ in the solution space.

The particular implementation we will use for Quinlan's C4.5 algorithm is J48 as denoted in the Weka software [12]. Canonical decision trees learners build the tree from the root down. At each node expansion, the algorithm selects the attributes to be placed at the current node of the tree using heuristics that aim at minimizing the height of the tree (and thus increase generalization power). A common approach is to pick an attribute to expand a node based on information gain. There is a rationale to using the information gain, namely information gain is correlated to how informative the attribute is for determining the class (and such rationale has also been used for feature selection).

For the second learning method, SMO Support Vector Machines (SVM−SMO), the particular implementation we will use is LIBSVM [5]. Using sequential minimal optimization (SMO) as the learning process is an alternative to the original formulation that required to solve a large quadratic programming (QP) optimization problem. Support Vector Machines find a hyper-plane separating the class in a high-dimensional space where instances are taken virtually, depending on the choice of a kernel base. Such hyper-plane is chosen as the one that maximizes the margin. As a result of this learning (optimization), some of the instances are identified as support vectors that are within the margin, or close to the margin and thus define the classification.

3.5 The Metrics in the *SAN*

The tabular form of the *SAN* presented earlier suggests only attribute-value pairs as predictors. We show in this section that the data for learning from the *SAN* includes metrics that summarize influence within the local neighborhood of each social vertex. These features (or additional columns to the tabular data model) are *SAN* metrics.

To describe these metrics we need some notation. First we need notation for the social neighbors of a user-account u. These are all other accounts in the social network connected to u, and this 1-hop neighborhood will be denoted $\Gamma_{s+}(u)$ where $\Gamma_{s+}(u) = \{v | v \text{ is a social vertex and } (u,v) \in E\}$. This notation will also be used to describe the set of all user-accounts that hold a particular attribute-value pair. In that case $\Gamma_{s+}(a = \text{value})$ is defined for an attribute-value pair, and is also the 1-hop neighborhood $\Gamma_{s+}(a = \text{value}) = \{v | v \text{ is a social vertex and } (v, a = \text{value}) \in E\}$. Again, when the attribute is evident from the value, we write $\Gamma_{s+}(a = \text{value})$ as $\Gamma_{s+}(\text{value})$. Note the use of the subindex with s to indicate these neighborhoods consist only of social vertices (user-accounts in the social network). The subindex $+$ stands for distance one neighborhood, so Γ_{s++} will be all social vertices at distance two.

The next definition of neighborhood does not restrict the vertices to social vertices and therefore we no longer use the subindex s. Thus,

$$\Gamma_{+}(u) = \Gamma_{s+}(u) \cup \{a = \mathtt{A} | \text{ user } u \text{ has value } \mathtt{A} \text{ for attribute } a\}.$$

The summarization of data into informative features leads to the definition of some metrics. As we suggested earlier, the number of neighbors in common between two vertices u_i and u_j is indicative of high interconnectivity between these two vertices independently of whether u_i and u_j are connected. Thus CN [13] (*common neighbor*) is defined as follows.

$$m_{\text{CN}}(u_i, u_j) = \sum_{u \in \Gamma_{s+}(u_i) \cap \Gamma_{s+}(u_j)} w(u), \tag{3}$$

where $w(u)$ is a weight associated to user-account u (which could take into consideration other relevance factors).

While CN considers only the interaction of u_i and u_j through social networks, the metric CN_*SAN* [11] also includes the attributes in common.

$$m_{\mathrm{CN}-SAN}(u_i, u_j) = \sum_{u \in \Gamma_+(u_i) \cap \Gamma_+(u_j)} w(u). \qquad (4)$$

A metric widely used to construct a feature for attribute prediction is the Adamic-Adar statistic [1] (AA) which evaluates the information between a social vertex u and an attribute-value pair $a = \mathtt{A}$.

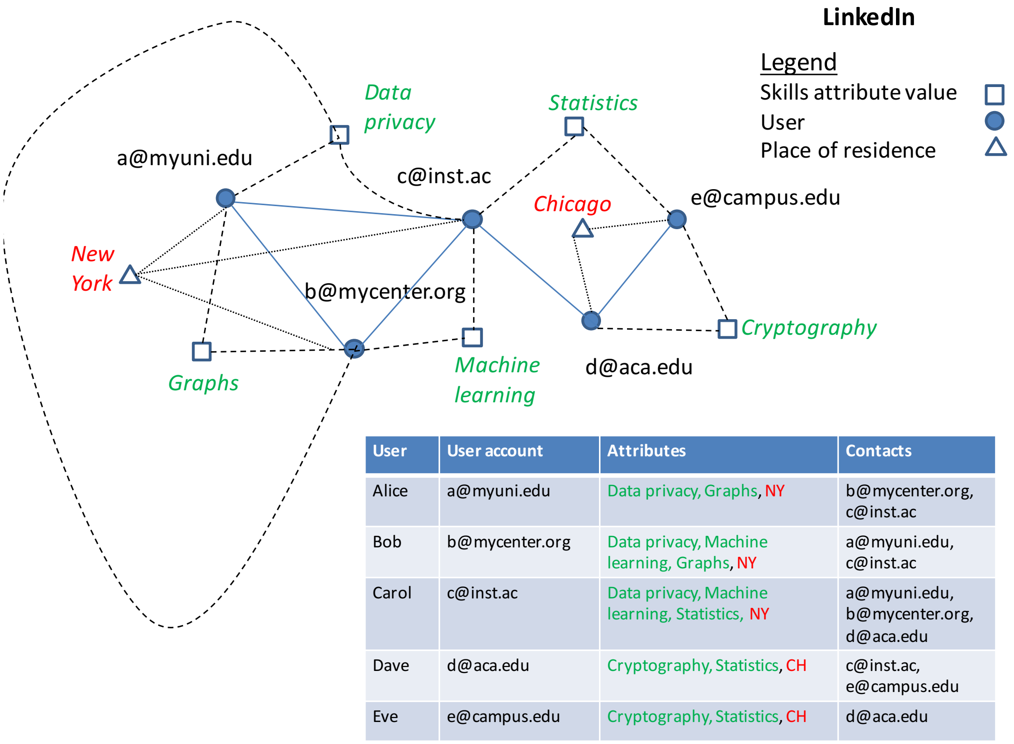
$$m_{\mathrm{AA}}(u, a = \mathtt{A}) = \sum_{u_i \in \Gamma_{s+}(u) \cap \Gamma_{s+}(a = \mathtt{A})} \frac{w(u_i)}{\log |\Gamma_{s+}(u_i)|}. \qquad (5)$$

Note again that $m_{\mathrm{AA}}(,)$ can be calculated even if the graph does not have the edge between user-account u and the attribute-value pair $a = \mathtt{A}$, nevertheless it counts all social neighbors of u that have attribute-pair $a = \mathtt{A}$ and weighs them by their neighborhood sizes. That is, it scales the importance of a common neighbor by the inverse of the social degree of that neighbor.

4 Multiple Network Membership

To aid the discussion, in this section we will provide some simple examples of *SAN*s. In Figure 3, we see five LinkedIn users and five attribute-value vertices which represent specific skills. We can see that *data privacy* is the most frequent *skill* (3 users have it) and all other skills have two edges. We can also see that b@mycenter.org (the account used by Bob) and c@inst.ac (the account used by Carol) have the most skills (3 apiece). Also, we see a general tendency that users who are connected also tend to have edges to common attribute-value vertices. For example, c@inst.ac and b@mycenter.org are connected and also have two skills in common (*data privacy* and *machine learning*). For illustration, the attribute *skill* will be considered a quasi-identifier (that is, we assume this is information users make public and is available to infer other attributes). Also in Figure 3, we see two attribute-value vertices which represent where people live (the attribute *place of residence*). We see that a@myuni.edu (used by Alice), b@mycenter.org and c@inst.ac all are accounts that indicate their users live in *New York*, and e@campus.edu (used by Eve) and the account used by Dave both live in *Chicago*. To further the illustration, the *place of residence* attribute values will be considered as confidential attribute-value pairs. That is, not all users divulge this attribute.

In Figure 1 we saw seven Facebook user-accounts and five attribute-value vertices which represent specific sporting interests. We can see that *Football* is the most frequent *sport* (4 user-accounts like it) and *Skiing* is the least frequent (only Joana likes it). As before, we see a general tendency that user-accounts who are connected tend to have edges to common attribute-value vertices. For example, b@mycenter.org is connected to a@myuni.edu and j@sports.com and all three of them like *Football*. However, if we compare Figure 3 with Figure 1 we see that the user-account sets and attribute sets vary. For our discussion, in the Facebook example, the *sporting interest* attribute values will be considered as quasi-identifiers. Here we assume people disclose this attribute to find others who they



User	User account	Attributes	Contacts
Alice	a@myuni.edu	Data privacy, Graphs, NY	b@mycenter.org, c@inst.ac
Bob	b@mycenter.org	Data privacy, Machine learning, Graphs, NY	a@myuni.edu, c@inst.ac
Carol	c@inst.ac	Data privacy, Machine learning, Statistics, NY	a@myuni.edu, b@mycenter.org, d@aca.edu
Dave	d@aca.edu	Cryptography, Statistics, CH	c@inst.ac, e@campus.edu
Eve	e@campus.edu	Cryptography, Statistics, CH	d@aca.edu

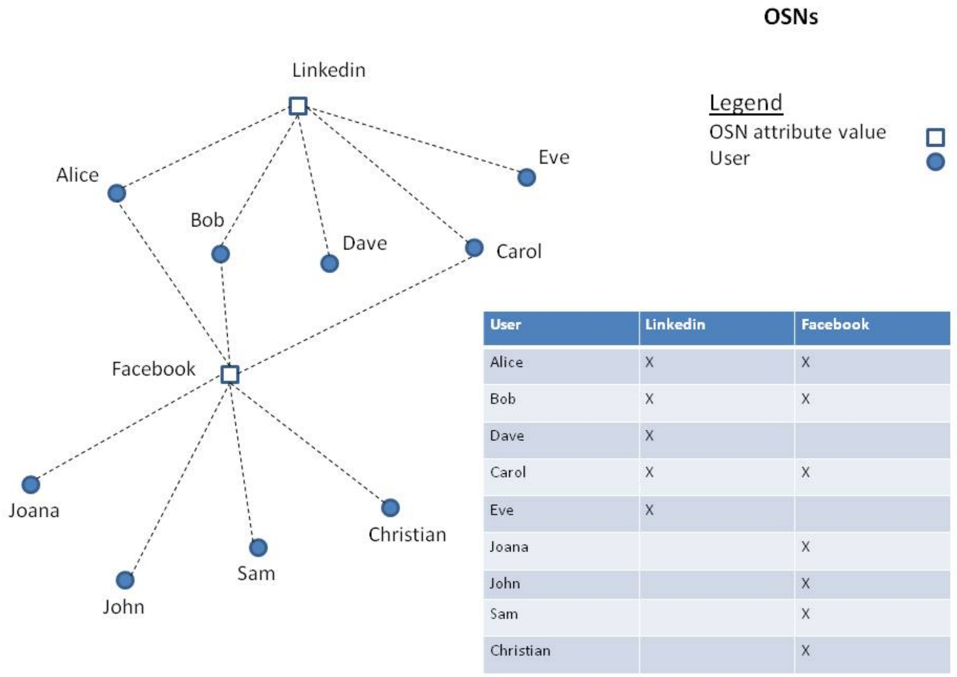
Figure 3: A sample social network of professional connections where users disclose attributes regarding their profession

can engage in the common interest. Also in Figure 1 we saw two attribute-value vertices which represent where people live (*place of residence*). We see that the users of accounts a@myuni.edu, b@mycenter.org and c@inst.ac live in *New York*, whereas Christian, John and Sam all live in *San Francisco*. The residence attribute values will again be considered as confidential attributes.

Figure 4 shows an amalgamation of the LinkedIn and Facebook graphs we saw in Figure 3 and Figure 1. We see that three user-accounts (Alice, Bob and Carol) who have an account in LinkedIn also have an account in Facebook. Furthermore, all three of these users who are connected in LinkedIn are also connected in Facebook.

4.1 Revisiting Re-identification with Several OSNs

Consider Figure 1 (our sample Facebook graph), and user-account Bob. Let us suppose that *residence* is confidential for Bob and that the network predictive precision is under a given threshold for Bob. That is, an analysis [8, 9] of the information available on the Facebook network finds that there are no attributes that can be used to predict Bob's *residence* with a confidence above θ. However, Bob has two friends (Alice and Carol) who also live in *New York*. It is known that people who are connected in social networks tend to live geographically close [6], so it could be inferred that Bob has a higher than average likelihood of living in *New York* even though he has not stated it. Also, other attributes that Bob still has public (such as liking *Tennis*) may be a high predictor of *residence=New York*. Then, if



User	Linkedin	Facebook
Alice	X	X
Bob	X	X
Dave	X	
Carol	X	X
Eve	X	
Joana		X
John		X
Sam		X
Christian		X

Figure 4: Two meta attribute-value nodes (OSN name), nine user-account vertices and associated edges from the amalgamation of Figure 3 and Figure 1

we consider the LinkedIn graph as well (Figure 3), the relation between Bob and Carol will be potentiated, and as Carol likes *Tennis*, this would probably increment the likelihood that there is a predictive model that infers that Bob lives in *New York* with certainty above the threshold θ.

We illustrate the power of the features introduced in the previous section furthering this example. First, we can calculate the neighborhoods. We will use the superscript F to denote that the calculations are with respect to the social network in Figure 1. The one-hop *social* neighborhood of Alice is $\Gamma^F_{s+}(\text{Alice}) = \{\text{Bob}, \text{Carol}\}$. Similarly, the one-hop *social* neighborhood of Bob is $\Gamma^F_{s+}(\text{Bob}) = \{\text{Alice}, \text{Carol}, \text{John}, \text{Sam}\}$. The one-hop neighborhood (including attribute-value pairs) of these two user-accounts is as follows.

$$\begin{aligned}
&\Gamma^F_{+}(\text{Alice}) \\
&\quad = \{\text{Bob}, \text{Carol}, \textit{sporting interest} = \textit{Football}, \textit{sporting interest} = \textit{Skating}, \\
&\qquad \textit{residence} = \textit{New York}\} \\
&\Gamma^F_{+}(\text{Bob}) \\
&\quad = \{\text{Alice}, \text{Carol}, \text{John}, \text{Sam}, \textit{sporting interest} = \textit{Football}, \\
&\qquad \textit{sporting interest} = \textit{Ping Pong}, \textit{sporting interest} = \textit{Tennis}, \\
&\qquad \textit{residence} = \textit{New York}\}
\end{aligned}$$

While the user-accounts with the *residence* are as follows.

$$\Gamma^F_{+}(\textit{New York}) = \{\text{Alice}, \text{Bob}, \text{Carol}\}.$$

Assuming all weights are 1, we can illustrate the three metrics with respect to the social network also as seen in Figure 1.

$$\begin{aligned}
&m_{\text{CN}}(\text{Alice}, \text{Bob})^F \\
&\quad = |\{\text{Carol}\}| = 1. \\
&m_{\text{CN}-SAN}(\text{Alice}, \text{Bob})^F \\
&\quad = |\{\text{Carol}, \textit{sporting interest} = \textit{Football}, \textit{residence} = \textit{New York}\}| = 3. \\
&m_{\text{AA}}(\text{Alice}, \textit{residence} = \textit{New York})^F \\
&\quad = |\{\text{Carol}, \text{Bob}\}| = 2.
\end{aligned}$$

Under these conditions, the analysis to predict Bob's *residence*, finds that the rule

$$\textit{connected to Carol} \rightarrow \textit{residence} = \textit{New York}$$

is a predictor.

4.2 Multiple OSNs

We now propose the generation of features from the metrics of several multiple overlapping OSNs. For this, assume that the two social networks have a set C_u of common user-accounts. That is, the e-mail, or some other identifier, is shared and the set C_u are users who have explicitly revealed they are the same person. In our example, we see that Alice, Bob and Carol have all used the same e-mail as their identifier for both social networks in Figure 1 and Figure 3. Moreover, in our example, there are no users that are using a different credential across the network. Similarly, although the on-line social networks may have different ambitions, they share a set C_a of common attributes. For example the user's *place of residence* is a property that appears in both networks in Figure 1 and Figure 3 although there are other properties specific to each of the social networks.

More interesting is the fact that we can consider the sub-*SAN* consisting of the intersection of all the on-line social networks we are fusing. Each user-account in each sub-*SAN* has a projected record; in particular, the original record in S^1 (given by Equation (1)) and the original record in S^2 (given by Equation (2)) are projected. If the corresponding user-accounts have not been identified, the record is shorter. The record is projected on the attributes of the intersection. For SAN S^1 we have the following record.

$$\begin{aligned}&\text{record}(u^1, S^1 \cap S^2)\\ &= \quad \langle u^1, (\{c_{1,1}, c_{1,2}, \ldots, c_{1,t_1}\} \cap C_u) \cup (\{a_{1,1}, a_{1,2}, \ldots, a_{1,s_1} \cap C_a)\}\rangle.\end{aligned}$$

And similarly for SAN S^2.

$$\begin{aligned}&\text{record}(u^2, S^1 \cap S^2)\\ &= \quad \langle u^2, (\{c_{2,1}, c_{2,2}, \ldots, c_{2,t_2}\} \cap C_u) \cup (\{a_{2,1}, a_{2,2}, \ldots, a_{2,s_2} \cap C_a)\}\rangle.\end{aligned}$$

Essentially we project the records to the intersection of S^1 and S^2.

If the user accounts are identified and $u^1 = u^2$, then the record is projected to the intersection as

$$\begin{aligned}&\text{record}(u^1, S^1 \cap S^2)\\ &= \quad \langle (u^1 = u^2), (\{(c_{1,1}, c_{1,2}, \ldots, c_{1,t_1}, c_{2,1}, c_{2,2}, \ldots, c_{2,t_2})\} \cap C_u)\\ &\qquad \cup(\{(a_{1,1}, a_{2,1}), (a_{1,2}, a_{2,2}), \ldots, (a_{1,\|C_a\|}, a_{1,\|C_a\|})\} \cap C_a)\}\rangle.\end{aligned}$$

We note that some attributes are potentially multi-valued because the user may have supplied value $a_{1,1}$ for the first attribute of C_a in *SAN* S^1 and a different value $a_{2,1}$ in *SAN* S^2 (but in most instances this would be the same). In our running example the three users in the intersection, Alice, Bob and Carol, all have matching values for the attribute in the intersection.

Similarly, the connections may belong to identified or non-identified user-accounts. In our simple illustration the three users Alice, Bob and Carol in the intersection have used the same user-id across the two on-line social networks. So, their record in the intersection *SAN* is simply of a list of connections to the others (in the adjacency list representation).

The following observation constrains the values of the metrics when the metrics are considered relative to a particular *SAN*. The metrics count edges (to social vertices, attribute vertices or both). Also, the intersection *SAN* is a subset of each of the *SAN*s being merged, who themselves are sub-graphs of the union (or super-*SAN*). Therefore, the metric value on the intersection is less than the same metric value on the *SAN* being merged, which is less that the metric value relative to the super-*SAN*. For example,

$$m_{\text{CN}}(u_i, u_j)^{S^1 \cap S^2} \leq m_{\text{CN}}(u_i, u_j)^{S^1} \leq m_{\text{CN}}(u_i, u_j)^{S^1 \cup S_2}.$$

For illustration, if we have our Facebook and LinkedIn networks, we can construct the metric CN^F for Facebook and CN^L for LinkedIn provided the user-accounts of u_i and u_j are identified across the OSNs (we will discuss re-identification later).

Even the metric AA can be computed for all those attributes C_a that the networks have in common. However, the intersection graph would have fewer attributes than each *SAN* and the super *SAN* that results from the union will list all attributes in all the sub-*SAN*s. The numerator of the AA metric will also have the monotonic relationship of values: the count relative to the intersection is a smaller value than the count relative to each of the *SAN*s, and this value, in turn, is smaller than the count for the union of the *SAN*s. If we make the denominator constant to the users in C_u (users for whom we have identified their user-accounts), the metric values are also monotonic for the metric AA.

To illustrate these concepts, we now return once more to our two examples of Figure 1 and Figure 3, which are combined in Figure 4. We see that three user-accounts (Alice, Bob, and Carol) are members of both OSNs, and that all three are connected in both of the social networks.

The neighborhoods for Alice and Bob with respect to Figure 3 are as follows.

$$\begin{aligned}
\Gamma^{L}_{s+}(\text{Alice}) &= \{\text{Bob}, \text{Carol}\} \\
\Gamma^{L}_{s+}(\text{Bob}) &= \{\text{Alice}, \text{Carol}\} \\
\Gamma^{L}_{+}(\text{Alice}) &= \{\text{Bob}, \text{Carol}, \textit{skill} = \textit{data privacy}, \textit{residence} = \textit{New York}\} \\
\Gamma^{L}_{+}(\text{Bob}) &= \{\text{Alice}, \text{Carol}, \textit{skill} = \textit{data privacy}, \\
&\quad \textit{skill} = \textit{machine learning}, \textit{residence} = \textit{New York}\}
\end{aligned}$$

Now we can evaluate the metrics for the OSN in Figure 3 as we did earlier for Figure 1.

$$\begin{aligned}
m_{\text{CN}}(\text{Alice}, \text{Bob})^{L} &= |\{\text{Carol}\}| \\
&= 1 \\
m_{\text{CN}-SAN}(\text{Alice}, \text{Bob})^{L} &= |\{\text{Carol}, \textit{skill} = \textit{graphs}, \textit{skill} = \textit{data privacy}, \\
&\quad \textit{residence} = \textit{New York}\}| \\
&= 4 \\
m_{\text{AA}}(\text{Alice}, \textit{residence} = \textit{New York})^{L} &= |\{\text{Bob}, \text{Carol}\}| \\
&= 2.
\end{aligned}$$

If we consider the intersection *SAN*, because the only attribute in common is *place of residence*, some of the counts are smaller. For example,

$$m_{\text{CN}-SAN}(\text{Alice}, \text{Bob})^{L\cap F} = |\{\text{Carol}, \textit{residence} = \textit{New York}\}| = 2$$

However, the information of the combined OSNs show these neighborhoods.

$$\begin{aligned}
&\Gamma^{F\cup L}_{s+}(\text{Bob}) \\
&\quad = \{\text{Alice}, \text{Carol}, \text{John}, \text{Sam}\}, \\
&\Gamma^{F\cup L}_{+}(\text{Bob}) \\
&\quad = \{\text{Alice}, \text{Carol}, \text{John}, \text{Sam}, \\
&\qquad \textit{skill} = \textit{graphs}, \textit{skill} = \textit{data privacy}, \textit{skill} = \textit{machine learning}, \\
&\qquad \textit{interest} = \textit{Football}, \textit{interest} = \textit{Tennis}, \textit{interest} = \textit{Ping Pong}, \\
&\qquad \textit{residence} = \textit{New York}\}
\end{aligned}$$

Clearly, this is a richer picture (digital footprint) of the user-account Bob in terms of his professional contacts and skills, as well as other (more informal) contacts and interests. All of this information could be used to identify user-accounts that are not currently identified and to infer attribute-value pairs for those user-accounts with identified connections, as well as existing but unidentified connections.

5 Illustrative Example

In this section we will first attack the problem of re-identification. Our results suggest that it is possible to re-identify users across different social networks even if the intersection sets C_u and C_a are relative small with respect to the size of the typical on-line social networks we have in mind. For this, we will describe the results of applying a set-theoretical re-identification technique. We follow this by the results of applying predictive models built using machine learning techniques with *SAN* metrics as inputs. We investigate the sensitivity of the CN, CN_*SAN* and AA metrics of Subsection 3.5 to their consideration with respect to the sub-*SAN*s involved. In particular, which of these metrics remain predictors of a binary condition (whether a user-account has a certain attribute-value pair although not disclosed or whether two user-accounts are connected although not yet disclosed).

5.1 Experimental Setup, Dataset and Metrics

For both approaches, we use a real dataset [11] to demonstrate the concepts proposed in Section 3 and Section 4. The dataset ***infer-attrib/SEP4.txt*** is about applicants and their background in areas of information technology. This data set has more than 5,000 user-accounts and over 10,000 attributes. On average, each user-account has about 4 attributes. Because attributes can have many values, the corresponding *SAN* will still have many attribute-vertices. In the file for this *SAN*, each line represents the information of a user-account in the adjacency-list format introduced before. We identified 7 attribute-nodes in the corresponding SAN whose total count is 92 or more in the complete dataset.

For each user and each attribute, we took the position that the user wants to keep confidential which indicates whether they have or don't have the attribute. Data cleansing of this data set consists of filtering by two key attributes: education and employment [10]. Among the most frequent attributes, were "bangalore institute of technology", "aspiag service srl", "fh aachen", "general studies" and "post graduate in marketing". From these nodes, the edges are constructed from coincidences where people studied and worked, and what they studied or do.

In order to facilitate the reproducibility of the experiments by other researchers, we have made the Java programs and files publicly accesible at: https:// github.com/ dnettlet/ OverlappingOSNsUserPrivacy , under the GNU GENERAL PUBLIC LICENSE v3.0.

5.2 Re-identification using a Set-theoretical Approach

In this section we provide an illustration of the power of combining two social networks to identify users of different user-accounts. In particular, we consider a user participation with a user-account u^1 in one social network S^1 and another user-account u^2 in a second social network S^2. Can the combination of data from both social networks establish that the two user-accounts correspond to the same person? Such re-identification [21, Page 123] can have many variants, and our approach here can be seen as *data matching* [21, Page 133], record linkage [7] or entity resolution [4].

5.2.1 Empirical Evaluation of Re-identification using Set-theoretical Approach

Re-identification of users in multiple OSNs can be related to distance based record linkage methods on databases [21, Section 5.4.8]. In distance based record linkage, the record assigned in one file is the nearest record on the other file. In our case, we will match the nearest records (user accounts) under Jaccard distance, after projecting them on the intersection. That is, we will use the intersection of two SANs and the set C_u of common user-accounts and the set C_a of common attributes, and focus our attention on users whose user-accounts are not identified. Recall that in this case, we project the records onto the intersection of S^1 and S^2.

Can we identify when two user-accounts u^1 and u^2 correspond to the same user? We argue that this is remarkably likely without having to resort to the calculation of metrics of the *SAN* model either. We argue that we just need to find for each user-account u^1 the user-account u^2 most similar in the intersection of S^1 and S^2, where most similar is simply the largest intersection between record$(u_1, S^1 \cap S^2)$ and record$(u_2, S^1 \cap S^2)$[1].

We substantiate this claim by the following analysis. A record of a user-account in the intersection of S^1 and S^2 is nothing more that a subset of $C_u \cup C_a$, when expressed as in Equation (1). If we assume that the record$(u^1, S^1 \cap S^2)$ consists of r values and all records are taken as being equally likely, the probability of another record matching it is

$$\frac{1}{\binom{\|C_u \cup C_a\|}{r}}.$$

We note that if we let $I = \|C_u \cup C_a\|$ be the size of common information between the two SANs, then because

$$\frac{I^r}{r^r} \leq \binom{I}{r} \leq \frac{I^r}{r!},$$

we see that very rapidly most user-accounts will have a record which is different from any other user as the number of connections or attributes that appear in both networks increases. When I is large and r is much smaller than I,

$$\binom{I}{r} \approx \frac{(I/r - 0.5)^r e^r}{\sqrt{2\pi r}}. \tag{6}$$

This shows the rapid decrease in probability that, by randomly picking r connections and attributes in the common space of the two *SAN*s, we choose a given user with such a record. Therefore, the probability of two user-accounts being the same given that r edges match, rapidly increases with r.

Although not all connections nor attributes are equally likely, this still suggests that user-accounts belonging to different users will display different records in the intersection of the two *SAN*s. Conversely, two user-accounts of the same user, even if not identified, will hold very similar records. In the terminology of probabilistic record linkage [21, Section 5.5]. we are saying that the probability of re-identification grows exponentially with the number r of matches in the coincidence vector.

We created a procedure to split a *SAN* into two *SAN*s randomly. The intention is to emulate the existence of two sub*SAN*s with some common identified users, but also with some common unidentified users. Furthermore, the two sub*SAN*s will have a mixture of common shared attributes together with some non-shared attributes. Our splitting method requires a value $p_p \in [0, 100]$ for the percentage of the total number of users that have unidentified user-accounts. That is, if the original *SAN* has $\|U\|$ user-accounts, the two new *SAN*s

[1] Because $\|S^1 \cap S^2\|$ is constant across user-accounts, we are basically maximizing the Jaccard coefficient.

will also have $\|U\|$ user-accounts, but as many as $p_p\|U\|/100$ user-accounts will be masked completely in the second network. That is, a set P of private users with $||P|| = p_p\|U\|/100$ user-ids is selected randomly and a random permutation $\pi : P \to P$ is chosen by which each network identifier c_i in the first network is replaced by $\pi(c_i)$ in the second network. Records within P are randomly shuffled among themselves in the production of the second network, so the order of file records for the second *SAN* provides no information about the identification mapping π.

The attributes are split separately, in this case a value $p_a \in [0, 100]$ indicates the size of the overlap of the two new *SAN*s relative to the original set A of attributes. Namely,

$$p_a = 100 \times \frac{\|A_1 \cap A_2\|}{\|A\|}. \tag{7}$$

Thus, this is implemented by randomly selecting a subset of the attributes of size $\|A_1 \cap A_2\|$ from A and then tossing a binary coin for each remaining attribute to place it in one or (exclusively) the other network. For example $p_a = 25$ for 10,000 attributes means that the networks will have 2,500 attributes in common and approximately 3,750 attributes will be exclusively to one and not the other.

This setting allows us to have two *SAN*s where we can attempt to discover π. That is, which is the corresponding user-account in the first network that corresponds to the other user-account in the second network. Using the data of the entire ***infer-attrib/SEP4.txt*** *SAN* with $p_p = 10\%$ and $p_a = 20\%$ we have two *SAN*s where the challenge is to re-identify 520 user-accounts.

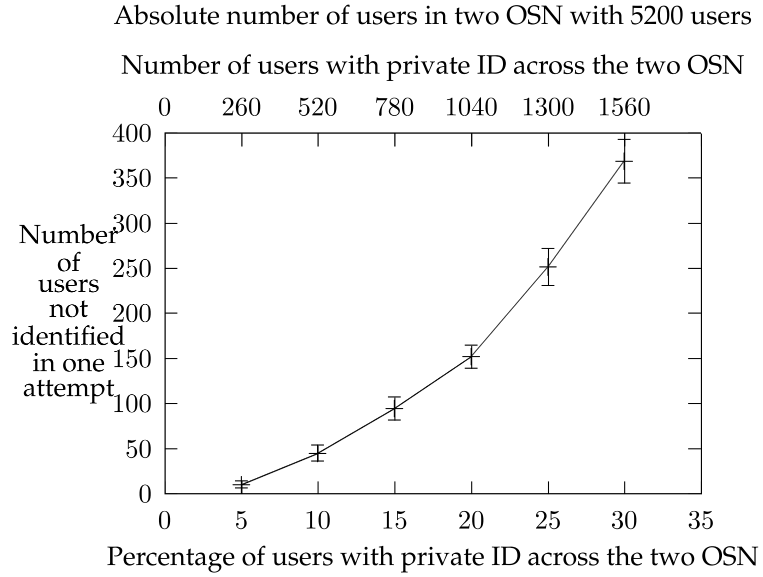
Our proposed re-identification method of a user-account u^1 simply searches in the other SAN for a user-account u^2 that maximizes the intersection of the records. With the ***infer-attrib/SEP4.txt*** *SAN*, one pass of this technique typically identifies 92% of the user-accounts (less that 9% of user-accounts remain un-identified after a first pass). All user-accounts that in the original SAN had more than 7 items are identified in the first pass.

When are records not identified? Usually when records have very little information. For example, in one instance of our experiment, record 380 had only one connection in network one to user-account 2535 (no attributes). In the other network, this user has a different id ($\pi(380) = 926$) and has only one user-account as a connection (user-account 1237) and two attributes. Both 2535 and 1237 happen to be un-identified user-accounts, but in the first pass of the method $\pi(2535) = 1237$ is identified.

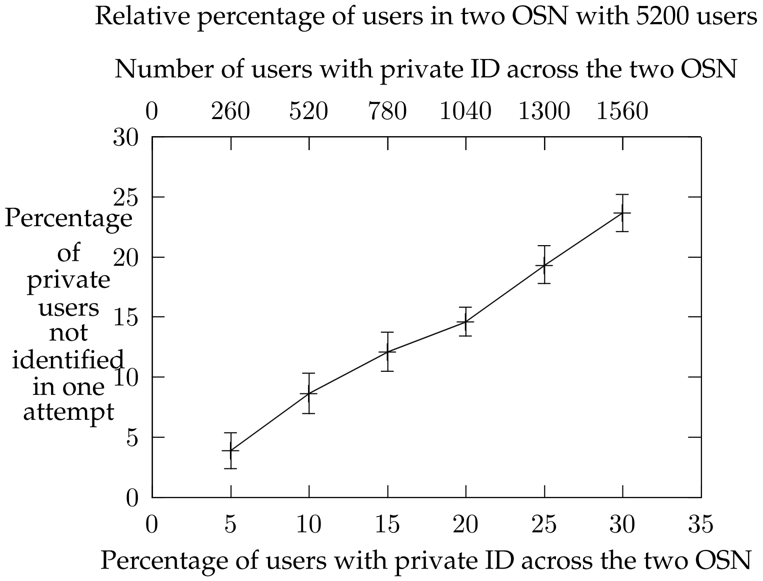
In fact, every time the method identifies that a user-account in one network matches a user account on the other, the set C_u grows. The identification of 92% of the user-accounts was performed where each identification exercise is using the initial C_u. In fact, if one updates the identified accounts and performs a second pass, all user-accounts are identified.

We also note that we recorded as unidentified in the first pass those user-accounts for which there is not a unique best match. However, for all user-accounts, the first pass narrows the identification to only two possibilities (there is a tie in the size of the intersection of the records).

Figure 5 shows the results of repeating 30 times the random generation of two *SAN*s from the ***infer-attrib/SEP4.txt*** *SAN*, with different values for $p_p \in \{5, 10, 15, 20, 25, 30, 35\}$ (the bars in the figure are 95% confidence intervals). The lower tics of the x-axis plot show the value of p_p while the upper tics of the x-axis show the size of the set P of private users. In particular, we see in Figure 5a that when the number of the private users (those who have not identified themselves in the two OSNs) is larger, then more of those users cannot be re-identified using only the common part of the two SANs. The plot in Figure 5a



(a) Absolute number of private users.



(b) Relative percentage to the number of private users

Figure 5: Illustration that the smaller the set of private users, the more those users are identifiable

would suggest that a larger number of unidentified user-accounts increases exponentially the number of those that are safe from re-identification. But, surprisingly, in relative terms, this is not so. Figure 5b shows that in percentage terms, the unidentified accounts are only linearly related with a shift of about 5%. That is, when the two online social networks have 30% of unidentified users, about 25% of those users can be identified with this information on the very first attempt.

We note again that in fact an exercise of identification of two user-accounts would probably be performed incrementally. That is, as soon as two user-accounts are identified as the same, then the set C_u would grow by one. The data of this experiment shows that in any case, the vast majority of the unidentified users can be identified immediately giving us many candidates by which to enlarge C_u.

5.3 Predictive Modeling Approach using *SAN* Metrics

In this section we contrast predictive models of the overall *SAN* with those of separate sub-*SAN*s. We build predictive models using machine learning techniques and incorporating the CN, CN_*SAN* and AA metrics as independent variables. In the following we refer to the two machine learning algorithms as J48 (the Weka implementation of Quinlan's C4.5 algorithm) and SVM-SMO (Support Vector Machine, Sequential Minimal Optimization), which have been described previously in Section 3.4.

5.3.1 Data Pre-processing

We identified, in the whole graph, the top five most highly connected users and the two most frequent attributes. The most highly connected users will be labeled as *topid-0*, *topid-1*, *topid-2*, *topid-3*, and *topid-4*. We refer to the two most frequent attributes as *topatt-0* and *topatt-1*. The objective is to predict, for a given user, to which top users it is connected and which top attributes it possesses. We contrast this relatively easy task when the data for the whole graph is available, against a split into two sub-graphs. The two sub-graphs will be different because top users and frequent attributes are partitioned across the sub-graphs. Using the data of the entire ***infer-attrib/SEP4.txt*** *SAN* we construct two subgraphs of it (S^1 and S^2) with the following properties.

1. Graph S^1 is the subgraph induced by *topid-2*, *topid-3*, and *topid-4*. That is, it includes these users and all their neighbors, but excludes *topid-0* and *topid-1*. It includes all information about *topatt-0*, but excludes *topatt-1*. Similarly, S^2 is the subgraph induced by *topid-0*, and *topid-1*, and which excludes *topid-2*, *topid-3* and *topid-4*. It also excludes all information about *topatt-0*, but includes *topatt-1*.

2. We calculate the *SAN* metrics (CN, CN_*SAN* and AA) for all 5 top users and the 2 top attributes.

3. The data set is organized so that the property to predict is whether a user is connected or not to a *topid* and also whether a user has *topatt* as an attribute-value pair.

Predictive models: for S^1 we have three potential links to predict and one attribute (4 classifiers in this case). For S^2 we have two potential links to predict and one attribute (3 classifiers in this case). For the entire *SAN* we have 5 links and 2 attributes. (7 classifiers in this case). However, we also tried predicting the users and attributes which were not present in S^1 and S^2, using the class labels from the entire *SAN* as outputs. Thus we had a total of 21 predictive models (7 for each *SAN*).

Classifier	Prediction	*topid-0*		*topid-1*		*topid-2*		*topid-3*		*topid-4*		*topatt-0*		*topatt-1*	
Technique	type	*Prec*	*Recall*	*Prec*	*Recall*	*Prec*	*Recall*	*Prec*	*Recall*	*Prec*	*Recall*	*Prec*	*Recall*	*Prec*	*Recall*
J48	No edge	0.976	0.732	0.884	0.633	1.000	0.829	0.852	0.767	0.982	0.966	1.000	1.000	0.955	0.889
J48	Edge exists	0.583	0.955	0.371	0.722	0.400	1.000	0.417	0.556	0.905	0.950	1.000	1.000	0.273	0.500
SVM-SMO	No edge	0.846	0.786	0.936	0.733	0.984	0.871	0.866	0.967	0.982	0.948	0.966	1.000	0.958	0.958
SVM-SMO	Edge exists	0.538	0.636	0.484	0.833	0.438	0.875	0.818	0.500	0.864	0.950	1.000	0.964	0.500	0.500

Table 3: Learning on 50% of all users of of the entire ***infer-attrib/SEP4.txt*** *SAN* (randomly selected). Precision and recall reported on remaining unseen 50% users. The classifier aims to predict whether the user has (or does not have) a social edge to one of the top users as well as the two most frequent attributes.

Classifier	Prediction	*topid-0*		*topid-1*		*topid-2*		*topid-3*		*topid-4*		*topatt-0*		*topatt-1*	
technique	type	*Prec*	*Recall*	*Prec*	*Recall*	*Prec*	*Recall*	*Prec*	*Recall*	*Prec*	*Recall*	*Prec*	*Recall*	*Prec*	*Recall*
J48	No edge	0.857	0.783	0.966	0.933	1.000	0.667	0.667	0.571	0.955	0.778	1.000	0.462	0.846	0.379
J48	Edge exists	0.545	0.667	0.000	0.000	0.500	1.000	0.700	0.778	0.333	0.750	0.650	1.000	0.053	0.333
SVM-SMO	No edge	0.895	0.739	0.968	1.000	0.857	0.750	1.000	0.500	0.931	1.000	0.806	0.692	0.882	0.517
SVM-SMO	Edge exists	0.538	0.778	0.000	0.000	0.455	0.625	0.720	1.000	1.000	0.500	0.730	0.833	0.067	0.333

Table 4: Learning happens on S^1 (the subgraph induced by *topid-2*, *topid-3*, and *topid-4*, excluding any information on *topatt-1*. The training set is 50% of randomly chosen users. Precision and recall for predictions reported on other 50% of users as testing set.

For all social networks (the entire *SAN*, S^1 and S^2), the users (social-nodes) were always randomly split into two sets each with 50% of the users. One set was used as the training set and the other as an unseen test set. To use the vanilla learning algorithms from WEKA [12], all attributes were converted to nominal categories (in particular, the metrics values were discretized to ranges of 10% percentiles). For the training set, the WEKA boost function was used to balance the output class. In all cases, we report the precision and recall for instances in the unseen dataset. Precision and recall have their standard definitions: Precision = TP/(TP+FP) and Recall = TP/(TP+FN), Where TP is True Positive, FP is False Positive, and FN is False Negative.

5.3.2 Results

The results support the hypothesis that an overall vision with social edges and attribute knowledge from multiple graphs with common users improves the predictability of the users' social edges and their attributes.

For example, the prediction of whether a user has a social edge to *topid-4* is not accurate when using the data just from S^1 or just from S^2, but can be predicted well from the fusion of these sub-*SAN*s. The pattern is similar for whether a user has a social edge to *topid-3*: this issue is poorly predicted with data in S^2, but much better with the data of S^1 or from the fusion of the *SAN*s. Similarly, whether a user holds *topatt-0* is poorly predicted from data in S^2, whereas predictions are more accurate for S^1, but the accuracy is even higher for the fusion of the *SAN*s.

Table 3 displays the precision and recall for the entire *SAN*. Table 4 shows the results for S^1 while Table 5 shows the results for S^2. The tables evaluate how well the existence or non-existence of a edge is predicted. With respect to the five top users, these *SAN*-edge represent a connection between user accounts. With respect to the top two frequent attributes, the *SAN*-edge represents whether the user has that value for such attribute or not.

Results for Table 3 are based on the fusion of subgraphs S^1 and S^2. We see that predicting whether a user is connected or not to one of the five users with most connections has good

Classifier	Prediction	*topid-0*		*topid-1*		*topid-2*		*topid-3*		*topid-4*		*topatt-0*		*topatt-1*	
technique	type	*Prec*	*Recall*	*Prec*	*Recall*	*Prec*	*Recall*	*Prec*	*Recall*	*Prec*	*Recall*	*Prec*	*Recall*	*Prec*	*Recall*
J48	No edge	0.625	0.227	0.563	0.474	0.900	0.346	0.933	0.500	0.960	0.857	1.000	0.462	0.913	0.750
J48	Edge exists	0.227	0.625	0.286	0.364	0.105	0.667	0.000	0.000	0.000	0.000	0.222	1.000	0.000	0.000
SVM-SMO	No edge	0.724	0.955	0.538	0.368	0.897	1.000	0.952	0.714	0.966	1.000	0.818	0.692	0.933	1.000
SVM-SMO	Edge exists	0.000	0.000	0.294	0.455	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000

Table 5: Learning happens on S^2 (the subgraph induced by *topid-0*, and *topid-1*, excluding any information on *topatt-0*. The training set is 50% of randomly chosen users. Precision and recall for predictions reported on other 50% of users as testing set.

predictability for all such popular users except for *topid-2*. Also remarkable is that *topatt-0* has improved its predictability with respect to the predictability with only S^1. However, the predictability of *topatt-1* seems surprisingly low.

In Table 4 we see the predictive results for S^1. This subgraph is induced by *topid-2*, *topid-3*, and *topid-4*. Thus, it is not surprising to see that *topid-3*, and *topid-4* display (relatively) good predictive precisions. It is surprising that this subgraph has also a strong predictive precision for *topid-0* even when the social edge from a user to *topid-0* has been removed. This implies that the social edges to the other top users and to other attributes are very informative and act as quasi-predictors for *topid-0*.

In Table 5 we see the predictive results for subgraph S^2. This is the induced graph on *topid-0* and *topid-1* and including *topatt-1*. We see that none of the users or attributes have a significant predictability.

Contrasting the results presented from Tables 3, 4 and 5, we can see that amalgamating the subgraphs improves the predictability of *topid-1*, *topid-3*, and *topid-4* (but not so much for *topid-2*). The edges for *topid-0* can be predicted as well with the data of S^1 as with the entire *SAN*, but they cannot be predicted as well with only the data of S^2. Also, the predictability of *topatt-0* improves but not so for *topatt-1* (however this may be due to the low predictability of *topatt-1*, noting that this attribute exhibits a high class unbalance).

Figure 6 and Figure 7 show a histogram representation of the SMO results. These figures plot the F-score derived from the precision and recall data from Tables 3, 4 and 5. The F-score is derived from the precision P and the recall R by the following formula:

$$F_1 = \frac{P \times R}{P + R} \times 2. \tag{8}$$

Thus the F-score provides a weighted measure taking into account both precision and recall.

Figure 6 shows the predictive results for identifying when there is no social edge (No edge) to a top user, or when the user does not have a given (top) attribute.

Examination of Figure 6 (prediction that there is no edge), reveals that, for the five top users and two top attributes, using the entire *SAN* enables the best results in 3 cases, equal first in another three cases and comes second in only one. On the other hand, using S^1 comes first in 1 case, equal first in 2 cases and equal second in one case. Finally, using S^2 comes first in 1 case, equal first in 3 cases and second in one case. Thus it is clear that overall, the entire *SAN* provides the most information for learning a predictor to identify when there is no edge. The sub-*SAN*s S^1 and S^2 are less informative.

Fig. 7 shows the predictive results for when there is an edge that should be identified. This signifies that a user is connected to a given (top) user or the user has a given (top) attribute. It can be seen that the predictive accuracy in general is much lower than for predicting if an edge is not there (Fig. 6). This is derived from the sparse nature of the *SAN* and its sub-

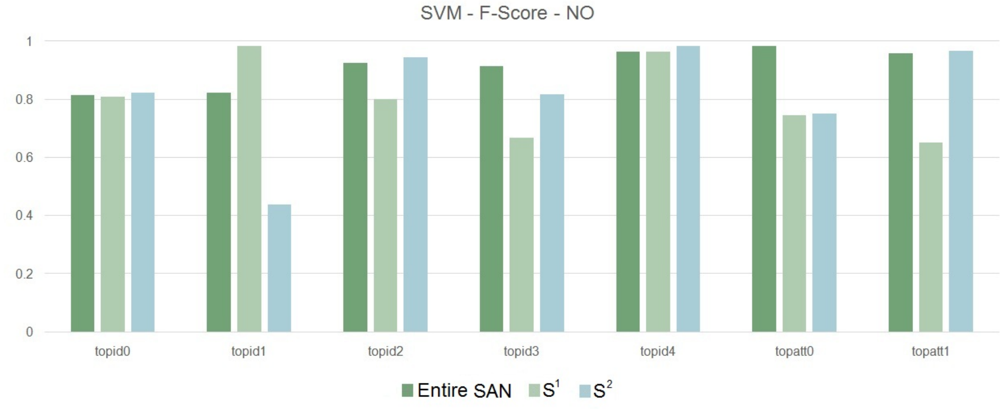


Figure 6: F-Score for No edge in the SANs considered (entire SAN, S^1 and S^2) for all top users and attributes

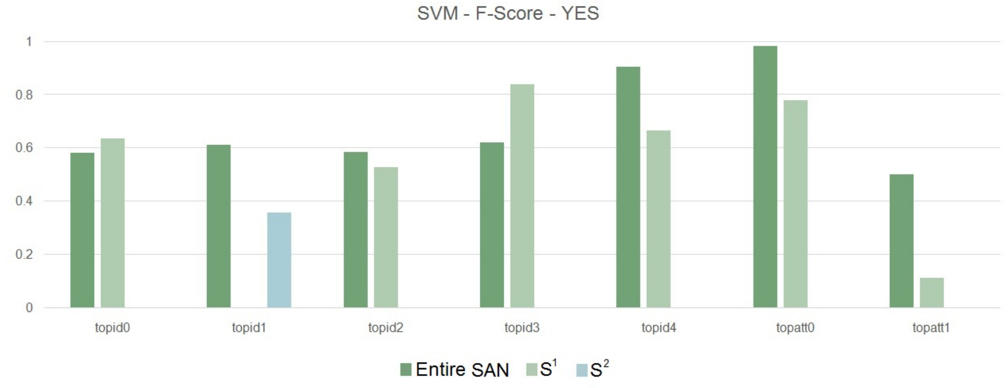


Figure 7: F-Score for predicting existing edges for all SANs, top users and attributes

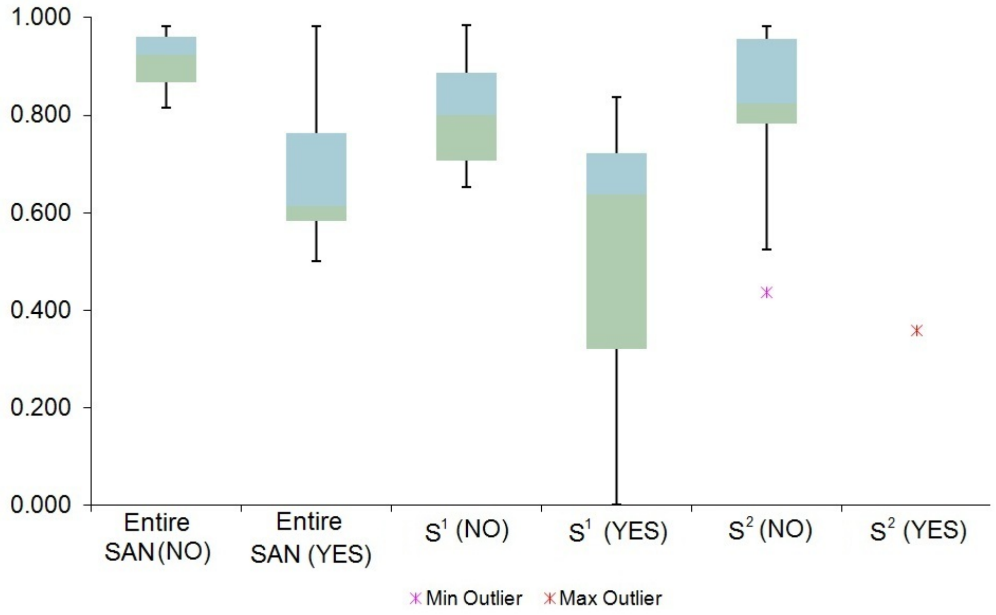


Figure 8: Box-plot showing distribution of predictive precisions for different *SAN*s and classification types

*SAN*s. The graph is generally sparse and this sparsity results in class unbalance (although we used the "boost" techniques as a pre-process to improve the training).

For the five top users and two top attributes, the entire *SAN* has the best results in five cases, whereas S^1 comes first in two cases. The sub-*SAN* S^2 only obtains a non-zero score in one case. However, the sub-*SAN* S^1 does show usefully high (above 60%) predictive results for *topid-3 topid-4* and *topatt-1*.

Fig. 8 shows a box with whiskers representation for the distribution of the predictive precision for different *SAN* datasets and classification attribute. It can be seen that the entire *SAN* has the lowest respective spread of predictive precision (which is desirable) over the quartiles and more of the spread is in the higher precision range. On the other hand, S^1 performs much better than S^2 for the "Edge exists" class (the latter having no displayable results) and somewhat worse than the entire *SAN*.

5.3.3 Top Input Factors used by Predictive Models

In this section we explore whether some subset of the available features are identified as high predictors of a social edge (indicating a connection between users of the on-line social network), or an attribute edge (indicating the user holds a certain attribute). This is relevant for privacy as users may not be aware that such features are available in the data for data mining from the structure of the *SAN*, and as the inferred edge may be regarded as confidential by the user.

The definition of the —SAN metrics (refer to Section 3.5) are fundamentally edge counts. While CN only accounts for social edges, the other two (CN_*SAN* and AA) involve the attributes of social neighbors for given attribute-values. Hence, the information value of CN_*SAN* and AA would be expected to be greater than plain CN for social-edge prediction as well as attribute prediction. We will now see that the results are concordant with this expectation. The CN_*SAN* and AA metrics are identified as those which produce the most informative features by the machine learning algorithms.

Table 6 contrasts which features are identified by the two learning algorithms as highly informative and therefore, predictors of the corresponding *SAN* edge. For the J48 classifier, the features are ordered by their frequency of use in the induced rules (note that J48 uses an information gain calculation to decide which features to include), and for the SVM-SMO classifier the features are ordered by the attribute weightings generated by the algorithm. The information in Table 6 contrasts how the source *SAN* influenced which are regarded as the most influential (high predictive value) features. For example, the same classifier learning algorithm J48 generates classifiers from the entire *SAN* where the value of the $m_{\mathrm{CN}-SAN}(\textit{topid-3}, u)$ metric and $m_{\mathrm{CN}-SAN}(\textit{topid-4}, u)$ metrics are considered high predictors of whether a user has the attribute *topatt-0*. But, when the source of information is the subgraph S^1, the highest predictor is the value of the metric $m_{\mathrm{CN}-SAN}(\textit{topid-2}, u)$. Whether a user has a social edge with *topid-0* is also interesting. With the entire *SAN* as input, the J48 ranks several metrics (CN, AA and CN_*SAN*) as predictors, while when restricted to the sub-*SAN* S^1, the same algorithm finds only the metric $m_{\mathrm{CN}-SAN}(\textit{topid-1}, u)$ as a predictor. The SMO algorithm on the other hand also used a diversity of AA, CN and CN_*SAN* metrics as predictors for *topid-0*, whereas in S^1, the same algorithm used $m_{\mathrm{CN}-SAN}(\textit{topid-1}, u)$, $m_{\mathrm{CN}-SAN}(\textit{topid-4}, u)$, $m_{\mathrm{AA}}(\textit{topid-1}, a)$, $m_{\mathrm{AA}}(\textit{topid-2}, a)$, and $m_{\mathrm{AA}}(\textit{topid-4}, a)$. Thus, although there are similarities in the choice of features derived from metrics that are the main predictors, the models for the sub-*SAN* S^1 identified fewer predictors.

We note that the largest number of times a feature derived from a *SAN* metric was selected as a predictor in the entire graph was four times for $m_{\mathrm{CN}-SAN}(\textit{topid-2}, u)$, $m_{\mathrm{AA}}(\textit{topid-0}, a)$ and $m_{\mathrm{AA}}(\textit{topid-4}, a)$. Also, the largest number of appearances is 4 when the data is S^1, but giving a different set of features identified as predictors: $m_{\mathrm{CN}-SAN}(\textit{topid-1}, u)$, $m_{\mathrm{AA}}(\textit{topid-1}, a)$ and $m_{\mathrm{AA}}(\textit{topid-2}, a)$. Thus, it appears that the algorithms have used different sub-metrics but the same metric families (CN_*SAN* and AA).

6 Summary and Conclusions

To summarize, we can say that re-identification in on-line social networks seems to present a significant risk for those who would like to remain un-identified from one on-line social network to another. Particularly, if such users are connected to a few users who do not wish to reveal they are the same owner of two corresponding user-accounts. This seems to be in radical contrast with the notion of k-anonymity. The connections in common (to user-accounts) and attributes between one network and another, even if several connections are missing or some attributes are not revealed from one network to another, are sufficient to re-identify the user. This is mainly because to whom a user connects is a finger-print that even if slightly distorted is sufficiently different from any other user fingerprint, so that matching most similar finger-prints re-identifies the user-accounts.

Figure 5 of Section 5.2 showed that the re-identification accuracy was 75%-95% over corresponding overlap sets with the overlap going down from 30% to 5% across the two graphs. That is, a smaller overlap set size gives a greater re-identification precision. Interestingly, across the different overlap values, about 5% of users with non-identifiable ID are always re-identified in the first attempt.

On the other hand, in terms of predictive precision of the user links and attributes, Fig-

Edge	Classifier	Entire *SAN*	S^1
topatt-0	J48	$m_{\mathrm{CN}-SAN}(\textit{topid-3}, u)$, $m_{\mathrm{CN}-SAN}(\textit{topid-4}, u)$	$m_{\mathrm{CN}-SAN}(\textit{topid-2}, u)$
topatt-0	SVM-SMO	$m_{\mathrm{CN}-SAN}(\textit{topid-2}, u)$ $m_{\mathrm{CN}-SAN}(\textit{topid-3}, u)$, $m_{\mathrm{CN}-SAN}(\textit{topid-4}, u)$, $m_{\mathrm{AA}}(\textit{topid-1}, a)$, $m_{\mathrm{AA}}(\textit{topid-4}, a)$	$m_{\mathrm{CN}-SAN}(\textit{topid-0}, u)$, $m_{\mathrm{CN}-SAN}(\textit{topid-1}, u)$, $m_{\mathrm{CN}-SAN}(\textit{topid-2}, u)$, $m_{\mathrm{CN}-SAN}(\textit{topid-3}, u)$, $m_{\mathrm{CN}-SAN}(\textit{topid-4}, u)$, $m_{\mathrm{AA}}(\textit{topid-0}, a)$, $m_{\mathrm{AA}}(\textit{topid-2}, a)$
topid-0	J48	$m_{\mathrm{AA}}(\textit{topid-0}, a)$, $m_{\mathrm{CN}}(\textit{topid-0}, u)$, $m_{\mathrm{CN}-SAN}(\textit{topid-3}, u)$, $m_{\mathrm{CN}-SAN}(\textit{topid-2}, u)$, $m_{\mathrm{CN}-SAN}(\textit{topid-1}, u)$	$m_{\mathrm{CN}-SAN}(\textit{topid-1}, u)$
topid-0	SVM-SMO	$m_{\mathrm{AA}}(\textit{topid-0}, a)$, $m_{\mathrm{AA}}(\textit{topid-1}, a)$, $m_{\mathrm{AA}}(\textit{topid-3}, a)$, $m_{\mathrm{AA}}(\textit{topid-4}, a)$, $m_{\mathrm{CN}}(\textit{topid-0}, u)$, $m_{\mathrm{CN}}(\textit{topid-3}, u)$, $m_{\mathrm{CN}-SAN}(\textit{topid-1}, u)$, $m_{\mathrm{CN}-SAN}(\textit{topid-2}, u)$	$m_{\mathrm{CN}-SAN}(\textit{topid-0}, u)$, $m_{\mathrm{CN}-SAN}(\textit{topid-1}, u)$, $m_{\mathrm{CN}-SAN}(\textit{topid-2}, u)$, $m_{\mathrm{AA}}(\textit{topid-1}, a)$
topid-4	J48	$m_{\mathrm{CN}}(\textit{topid-2}, u)$, $m_{\mathrm{AA}}(\textit{topid-4}, a)$, $m_{\mathrm{CN}}(\textit{topid-3}, u)$, $m_{\mathrm{CN}}(\textit{topid-0}, u)$	$m_{\mathrm{CN}}(\textit{topid-2}, u)$ $m_{\mathrm{AA}}(\textit{topid-1}, a)$, $m_{\mathrm{AA}}(\textit{topid-2}, a)$
topid-4	SVM-SMO	$m_{\mathrm{CN}}(\textit{topid-2}, u)$, $m_{\mathrm{CN}-SAN}(\textit{topid-2}, u)$, $m_{\mathrm{AA}}(\textit{topid-0}, a)$, $m_{\mathrm{AA}}(\textit{topid-2}, a)$, $m_{\mathrm{AA}}(\textit{topid-3}, a)$, $m_{\mathrm{AA}}(\textit{topid-4}, a)$	$m_{\mathrm{CN}-SAN}(\textit{topid-1}, u)$, $m_{\mathrm{CN}-SAN}(\textit{topid-4}, u)$, $m_{\mathrm{AA}}(\textit{topid-1}, a)$, $m_{\mathrm{AA}}(\textit{topid-2}, a)$, $m_{\mathrm{AA}}(\textit{topid-4}, a)$
topid-3	J48		$m_{\mathrm{CN}}(\textit{topid-2}, u)$
topid-3	SVM-SMO		$m_{\mathrm{CN}}(\textit{topid-2}, u)$, $m_{\mathrm{CN}-SAN}(\textit{topid-2}, u)$, $m_{\mathrm{AA}}(\textit{topid-0}, a)$, $m_{\mathrm{AA}}(\textit{topid-1}, a)$, $m_{\mathrm{AA}}(\textit{topid-2}, a)$, $m_{\mathrm{AA}}(\textit{topid-4}, a)$

Table 6: Top input factors used by predictive models for selected graphs, topids and topatts.

ure 8 of Section 5 showed that the average precision for the entire *SAN* (the fusion of both sub-graphs) for all top users and attributes was between 60% and 75% for the second and third quartiles (that is, the bulk of the distribution). However, for S^1 the average precision for the same quartiles was down to between 35% and 70%, and for S^2 the precision was residual and not displayed. Thus the amalgamation of S^1 and S^2 produced a very significant increase in the average predictability of the user and attribute links. Finally, Table 6 showed that the top predictors were found to be the *SAN* metrics CN_*SAN* and AA, which confirms the effectiveness of the *SAN* derived metrics.

Acknowledgments

Julián Salas acknowledges the support of a UOC postdoctoral fellowship. This work is partly funded by the Spanish Government through grant TIN2014-57364-C2-2-R "SMART-GLACIS".

References

[1] L. Adamic and E. Adar. Friends and neighbors on the web. *Social Networks*, 25:211–230, 2001.

[2] L. Backstrom and J. Leskovec. Supervised random walks: Predicting and recommending links in social networks. In *Proceedings of the Fourth ACM International Conference on Web Search and Data Mining*, WSDM '11, pages 635–644, New York, NY, USA, 2011. ACM.

[3] G. E. A. P. A. Batista and Monard. M. C. An analysis of four missing data treatment methods for supervised learning. *Applied Artificial Intelligence*, 17(5-6):519–533, 2003.

[4] Indrajit Bhattacharya and Lise Getoor. Collective entity resolution in relational data. *ACM Trans. Knowl. Discov. Data*, 1(1), March 2007.

[5] C.-C. Chang and C.-J. Lin. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2:27:1–27:27, 2011. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

[6] R. Compton, D. Jurgens, and D. Allen. Geotagging one hundred million twitter accounts with total variation minimization. In J. Lin, J. Pei, X. Hu, X. Chan, R. Nambiar, C. Aggarwal, N. Cercone, V. Honavar, J. Huan, B. Mobasher, and Saumyadipta Pyne, editors, *2014 IEEE International Conference on Big Data, Big Data 2014*, pages 393–401, Washington, DC, USA, October 27-30 2014. IEEE.

[7] Halbert L. Dunn. Record linkage. *American Journal of Public Health and the Nations Health*, 36(12):1412–1416, 1946.

[8] V. Estivill-Castro, P. Hough, and M.Z. Islam. Empowering users of social networks to assess their privacy risks. In *Big Data (Big Data), 2014 IEEE International Conference on*, pages 644–649, Oct 2014.

[9] V. Estivill-Castro and D. F. Nettleton. Can on-line social network users trust that what they designated as confidential data remains so? In *2015 IEEE Trustcom/BigDataSE/ISPA*, volume 1, pages 966–973, Aug 2015.

[10] V. Estivill-Castro and D. F. Nettleton. Privacy tips: Would it be ever possible to empower online social-network users to control the confidentiality of their data? In *Proceedings of the 2015 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining 2015*, ASONAM '15, pages 1449–1456, New York, NY, USA, 2015. ACM.

[11] N. Z. Gong, A. Talwalkar, L. W. Mackey, L. Huang, E. C. R. Shin, E. Stefanov, E. Shi, and D. Song. Joint link prediction and attribute inference using a social-attribute network. *ACM Trans. Intell. Syst. Technol.*, 5(2):27, 2014.

[12] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten. The WEKA data mining software: an update. *SIGKDD Explorations*, 11(1):10–18, 2009.

[13] D. Liben-Nowell and J. Kleinberg. The link prediction problem for social networks. In *Proceedings of the Twelfth International Conference on Information and Knowledge Management*, CIKM '03, pages 556–559. ACM, 2003.

[14] J. Platt. Fast training of support vector machines using sequential minimal optimization. In *Advances in kernel methodssupport vector learning*, page 169185. MIT Press, 1998. Chapt 12. Schlkopf B, Burges CJC, Smola AJ (eds).

[15] J.R. Quinlan. *C4.5 programs for machine learning*. Morgan Kaufmann, San Mateo, 1993.

[16] M. G. Rahman and M. Z Islam. kDMI: A novel method for missing values imputation using two levels of horizontal partitioning in a data set. In H. Motoda, Z. Wu, L. Cao, O. Zaiane, M. Yao, and W. Wang, editors, *Advanced Data Mining and Applications*, pages 250–263, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg.

[17] M. G. Rahman and M. Z Islam. Missing value imputation using decision trees and decision forests by splitting and merging records: Two novel techniques. *Knowledge-Based Systems*, 53:51 – 65, 2013.

[18] K. J. Reza, M. D. Islam, and V. Estivill-Castro. 3lp: Three layers of protection for individual privacy in facebook. In *Proceedings of the ICT Systems Security and Privacy Protection - IFIP SEC 2017*, May 29th - 31st 2017. to appear.

[19] E. Ryu, Y. Rong, J. Li, and A. Machanavajjhala. Curso: Protect yourself from curse of attribute inference: A social network privacy-analyzer. In *Proceedings of the ACM SIGMOD Workshop on Databases and Social Networks*, DBSocial '13, pages 13–18, New York, NY, USA, 2013. ACM.

[20] L. Singh, H. Yang, M. Sherr, Y. Wei, A. Hian-Cheong, K. Tian, J. Zhu, S. Zhang, T. Vaidya, and E. Asgarli. Helping users understand their Web footprints. In *Proceedings of the 24th International Conference on World Wide Web*, WWW '15 Companion, pages 117–118, New York, NY, USA, 2015. ACM.

[21] V. Torra. *Data Privacy: Foundations, New Developments and the Big Data Challenge*. Springer International Publishing, Cham, Switzerland, 2017.

[22] V.N. Vapnik. *The nature of statistical learning theory*. Springer Verlag, New York, 1995.

[23] X. Wu, V. Kumar, J. Ross Quinlan, J. Ghosh, Q. Yang, H. Motoda, G. J. McLachlan, A. F. M. Ng, B. Liu, P. S. Yu, Z.-H. Zhou, M. Steinbach, D. J. Hand, and D. Steinberg. Top 10 algorithms in data mining. *Knowl Inform Syst*, 14(1):1–37, 2008.

[24] S. Yin, M. Gupta, T. Weninger, and J. Han. A unified framework for link recommendation using random walks. In *2010 International Conference on Advances in Social Networks Analysis and Mining*, pages 152–159, Aug 2010.

[25] Z. Yin, M. Gupta, T. Weninger, and J. Han. LINKREC: A unified framework for link recommendation with user attributes and graph structure. In *Proceedings of the 19th International Conference on World Wide Web*, WWW '10, pages 1211–1212, New York, NY, USA, 2010. ACM.

Notation

This section serves as a reference for notation consistency.

S We use S for a social network and S^i for the i-th social network.

V We use V for a a set of vertices and V^i for the vertices of the i-th social network.

E We use E for a a set of links and E^i for the edges of the i-th social network.

connections We try to say two users are *connected* in a social network and avoid two users are *friends* (specific to Facebook), two users are *linked* (specific to LinkedIn), two users are *related* (somewhat confusing).

u An account in the social network.

a An Attribute value pair.

$\mathtt{A}$ A value for an attribute, we have a macro $\mathtt{A}$

user I am still uncertain on how to use the word user, I would prefer to use *user-account*

vertex we use this for the basic elements of a graph and avoid *node*

node we use this for internal tests on attributes in a decision tree, while the class determination happens at a *leaf* of the decision tree

edge we use this for the basic elements of a graph and avoid *link*, *arc*.

skill We have a macro for names of attributes.

data privacy We have a macro for values of attributes.

Franco We have a macro for names of people.

$|\ |$ The cardinality of a set.

$\Gamma()$ neighborhood in a graph