On the Effectiveness of Obfuscation Techniques in Online Social Networks

Terence Chen[1,2], Roksana Boreli[1,2],
Mohamed-Ali Kaafar[1,2,3], and Arik Friedman[1,2]

[1]NICTA, Australia [2]UNSW, Australia [3] INRIA, France
{firstname.lastname}@nicta.com.au

Abstract. Data obfuscation is a well-known technique for protecting user privacy against inference attacks, and it was studied in diverse settings, including search queries, recommender systems, location-based services and Online Social Networks (OSNs). However, these studies typically take the point of view of a single user who applies obfuscation, and focus on protection of a single target attribute. Unfortunately, while narrowing the scope simplifies the problem, it overlooks some significant challenges that effective obfuscation would need to address in a more realistic setting. First, correlations between attributes imply that obfuscation conducted to protect a certain attribute, may influence inference attacks targeted at other attributes. In addition, when multiple users conduct obfuscation simultaneously, the combined effect of their obfuscations may be significant enough to affect the inference mechanism to their detriment. In this work we focus on the OSN setting and use a dataset of 1.9 million Facebook profiles to demonstrate the severity of these problems and explore possible solutions. For example, we show that an obfuscation policy that would limit the accuracy of inference to 45% when applied by a single user, would result in an inference accuracy of 75% when applied by 10% of the users. We show that a dynamic policy, which is continuously adjusted to the most recent data in the OSN, may mitigate this problem. Finally, we report the results of a user study, which indicates that users are more willing to obfuscate their profiles using popular and high quality items. Accordingly, we propose and evaluate an obfuscation strategy that satisfies both user needs and privacy protection.

1 Introduction

With the growing popularity of Online Social Networks (OSNs) in the past decade, users are sharing an increasing amount of personal information, ranging from their personal details and interests, to their habits and opinions. Access to some of this personal information can be restricted by configuring the OSNs' built-in privacy settings, but despite the OSN users' growing privacy awareness, a lot of this data is still considered harmless and made publicly accessible. This and other user-generated data is collected and mined by companies that provide personalized services, including recommendations and targeted advertising.

Users' privacy can be compromised when the public information in their profiles is used to derive information they are not willing to reveal. Previous studies have shown that private attributes[1] can indeed be easily inferred based on information from others who revealed those attributes, either by utilizing social graph characteristics like social connections of the target user (the homophily principle) [8, 11, 13], or based on publicly shared items in common with other users, utilizing statistical inference/maximum likelihood approaches [3].

As users have full control of the publicly available OSN information, they can combat inference by obfuscating their public profiles, i.e., by adding or removing selected information, while still keeping their true purpose of information sharing. Obfuscation has been proposed to protect users' privacy against inference attacks in the context of search queries [14, 18], movie ratings [17] and location-based services [1]. In the OSN domain, He and Chu [7] assumed that social relationships are publicly available, and proposed a protection method based on obfuscating (removing specific existing, or adding fake) social links. In this paper, we study generic statistical inference in OSNs, using a machine learning approach. We consider both the inference and the protection mechanisms, based on easily accessible interest items (activities, movies, music, etc.) from the OSN profiles. We assume that obfuscation is based on, e.g., an obfuscation application, that recommends a choice of items to users who wish to protect selected (one or more) attributes. These recommendations are based on a sample dataset collected from other OSN users. We show how obfuscation can be applied to protect user privacy and derive a practical (acceptable to users) and effective obfuscation strategy. We evaluate the practical aspects of obfuscation when users are protecting multiple attributes and in a system where there is a mix of privacy conscious users and users who do not share privacy concerns. Our contributions are as follows.

Using a dataset of close to 1.9 million Facebook profiles, we evaluate the effectiveness of different obfuscation strategies (to select items to be used for obfuscation) and obfuscation policies, including adding, removing or replacing selected items, with respect to a number of commonly used classifiers. We propose a novel obfuscation strategy, based on the χ^2 feature selection metric, which does not require knowledge of the classifier that the attacker is using (in general such knowledge is required for optimum obfuscation strategies). We show that this strategy can significantly reduce the inference accuracy, e.g., by 45% for the case when 40% of interest items are added to the user profile, compared to 60% reduction offered by the optimum strategy. The advantage of the optimum strategy decreases for the case when the same proportion of items are removed or replaced, or when a lower proportion of items are obfuscated.

This is the first work that evaluates the obfuscation of multiple attributes. We show that a strategy targeted towards protecting a single attribute can also offer protection to other sensitive attributes. For example, our results show that while obfuscation targeting the *gender* attribute can reduce the inference accuracy by

[1] We use the term "private attribute" to describe the information a user does not share publicly or, more generally, information that is not available online.

a factor of 3 (from 87% to 30%) when all users obfuscate 50% of their items, having a strategy targeting a different attribute, *relationship*, still results in an improvement factor of 1.3 for *gender*. A strategy targeting both attributes simultaneously can, for the same percentage of obfuscated items, increase this to 1.45. When attributes are correlated, however, a comparable protection level can be achieved while targeting both, e.g., for the case of *gender* and *interested in*, with a reduced inference accuracy of close to 30%.

We evaluate the effect of the obfuscation strategy that a group of users adopt on the privacy of all users, both in a static setting, and in a dynamic setting in which the strategy is adjusted based on the most recent OSN system data. We show that a static obfuscation strategy will not protect users in a system where other users share the same strategy. For example, with only 20% of users obfuscating their profiles by adding (or replacing) items, the accuracy of inference is increased from 45% when a single user obfuscates their profile, to 85%. Removing items results in a less significant, but still notable increase. Using a dynamic strategy can improve the resulting obfuscation gain for privacy conscious users, although a significant gain is only achieved for a small proportion of all users, indicating the need for further study of dynamic obfuscation strategies.

We evaluate the user preferences for various obfuscation strategies via a user study and show how the study results can be applied to derive a user-friendly obfuscation mechanism, which is both effective and practical. Our results indicate that quality and popularity are important factors in the choice of items for obfuscation and that having a mechanism that incorporates these factors is imperative for an effective solution.

The remainder of the paper is organized as follows. In Section 2, we discuss the background on inference techniques, we present the attack model and discuss the performance of different classifiers. Obfuscation is addressed in Section 3, including the obfuscation approaches and performance evaluation. Section 4 evaluates the performance of static and dynamic obfuscation techniques. The user preference study is presented in Section 5. We discuss the related work on inference and obfuscation techniques in Section 6 and conclude in Section 7.

2 Inferring Personal Information

In this section we evaluate the efficiency of private attribute inference attacks using machine learning in the context of OSNs. We first describe our attacker model, and then we introduce the feature selection process and the classifiers used in this study. Using a dataset consisting of 1.9 million Facebook profiles, we evaluate the performance of different classifiers on different types of background information.

2.1 Attack Model

The goal of the inference attack is to obtain the value of a user's private (not publicly accessible) attribute of their OSN profile, by analyzing publicly available background information using machine learning techniques. We assume the

adversary is able to learn from a large set of static public profiles. We note that in this section we do not consider the impact of obfuscation on the performance of inference, this will be addressed in Section 4. We also note that, while finding the optimal inference attack model or inference algorithm is not the main focus of this study, our evaluations are based on state-of-the-art inference techniques that are used in the literature, e.g., in [17] and [9].

Assume an OSN profile comprises a total of k attributes (both public and private): $P = \{A_1, A_2, ..., A_k\}$. The attributes can have either a single value, e.g. for *age*, *gender*, and *relationship status*, or a list of values, e.g., for interest related items in *favourite movies*, *music*, *books*, etc. We define the target attribute of the attacker as $A_x \in P$, and the background attribute used for learning as $A_b \in P$. Throughout the paper, we use the terms "items" and "features" to refer to specific attribute values.

To infer the value of A_x based on A_b, we model the inference task as a document classification problem. We split the dataset into training and test groups of user profiles. First, from the training set and based on the background attribute A_b belonging to a set of users $N = \{1, 2, ..., n\}$ and a set of items (features) $M = \{1, 2, ..., m\}$, we construct a $n \times m$ binary matrix X_{train}. Matrix elements $x_{ij} \in \{0, 1\}$, where $i \in N$ and $j \in M$, represent the user-item relationship: $x_{ij} = 1$ indicates the item j is in user i's profile, and $x_{ij} = 0$ otherwise. Similarly, vector $Y_{train} = \{y_1, y_2, ..., y_n\}$, $y_i \in C$, represents the user-class relationship, with values of the target attribute A_x taken as classes $C = \{1, 2, .., c\}$. The classifier is trained with X_{train} and Y_{train}, resulting in a prediction function $F(\cdot)$. The test profiles are used to construct X_{test} and Y_{test} and the value A_x is then predicted by the trained classifier function $F(\cdot)$ as $Y'_{test} = F(X_{test})$, where the output Y'_{test} is the predicted value of A_x. The predicted results Y'_{test} are compared to the actual values in Y_{test} to evaluate the classifier performance, $E(Y'_{test}, Y_{test})$, where E denotes the performance metrics. For E, we use accuracy (the sum of all correct classifications divided by the total number of classifications) and Area Under Curve (AUC).

2.2 Feature selection: χ^2

Feature selection is the process of selecting a subset of the terms occurring in the training set and using this subset as features in the classification task. This not only reduces the size of the training and test sets, but also increases the classification accuracy, by eliminating noise introduced by feature overfitting. Considering that wrapper and embedded feature selection methods are computationally expensive and specific to a prediction algorithm [6], we adopt the filter method [12], as it provides a fast pre-processing step while still retaining the utility of the feature set. Most importantly, the filter method is independent from the used prediction algorithm.

The filter method first computes a utility measure $S(t, c)$ for each term t and class c, and then selects k features that have the highest value of $S(t, c)$. In this study, we use the chi-square (χ^2) correlation coefficient, one of the most effective

feature selection metrics for text classification [19], as the utility measure. The χ^2 score for term t to class c is computed by:

$$\chi^2(t,c) = \sum_{e_t \in \{0,1\}} \sum_{e_c \in \{0,1\}} \frac{(N_{e_t e_c} - E_{e_t e_c})^2}{E_{e_t e_c}} \tag{1}$$

Where e_t is an indication of the document containing the term t and e_c indicates whether the document is in class c. $N_{e_t e_c}$ is the observed frequency of t in the document with class c and $E_{e_t e_c}$ is the expected frequency, with $e_t \in \{0,1\}$ and $e_c \in \{0,1\}$. E.g., E_{11} is the expected frequency of the term $t = 1$ and class $c = 1$ occurring jointly in a document.

2.3 Selected Classifiers

A number of techniques can be used for document classification. After evaluating the performance of a number of state-of-the-art classifiers (including Naïve Bayes, Decision Tree, Random Forest, Support Vector Machines and Logistic Regression), we selected the top three performing classifiers for our experiments (we assume an attacker could easily perform a similar evaluation).

Bernoulli Naïve Bayes Classifier Naïve Bayes classifier assumes the *independence* of features, in our case the presence of background attribute values in A_b, and that each of the features contributes independently to the probability that the prediction instance belongs to a class. We select the Bernoulli Naïve Bayes classifier because the features are binary values (corresponding to the presence of these features, or of interest related items, in a user profile). Maximum a posteriori (MAP) decision rule is used for class prediction, i.e., to select the most probable hypothesis. Given the background attribute contains m features, the predicted class label c is calculated below:

$$y' = \underset{c}{argmax}\ p(y=c) \prod_{i=1}^{m} p(A_i = a_i | y = c) \tag{2}$$

Logistic Regression Classifier The logistic regression classification model is used for predicting the outcome of *dependent* features. Logistic Regression assumes a parametric form for the distribution $P(Y|X)$, then directly estimates its parameters $W = \{w_0, w_1, ..., w_m\}$ from the training data. The prediction is based on the following probability:

$$p(y = c|A) = \frac{1}{1 + exp(w_0 + \sum_{i=1}^{m} w_i A_i)} \tag{3}$$

Logistic regression is a binary classifier. For multi-class target attributes, we have used One-Versus-All strategy.[2]

Random Forest Classifier Random Forest is an ensemble classification approach that combines a set of binary decision trees. At the learning stage,

[2] http://scikit-learn.org/stable/modules/multiclass.html

each tree is constructed using a portion of the training data and a subset of data features. Given a fixed set of features Φ that model the training data, $\log(\Phi)$ features and around 2/3 of the training data are randomly selected to construct each tree. Within the forest trees, each node uses for the decision making a single feature $f \in \Phi$, which is the best performing feature out of the selected subset of features. The class of an instance is determined by the majority voting of the terminal nodes reached when traversing the trees [2].

Age: 13,308 users (5.3%)			**Gender: 169,509 users (67.8%)**		
classes	code	percentage	**classes**	code	percentage
13-17	0	35.59%	female	0	57.4%
18-24	1	42.85%	male	1	42.6%
25-34	2	13.49%			
35+	3	8.05%			
			Interested in : 70,476 users (28.2%)		
Relationship: 93,855 users (37.5%)			**classes**	code	percentage
classes	code	percentage	men	0	23.89%
in a relationship	0	62.2%	women	1	39.32%
single	1	37.8%	men and women	2	36.78%

Table 1. Target attribute availability and their classes; total users: 249,847

2.4 The Performance of the Inference Attack

Dataset Used for Evaluation We use a dataset of randomly sampled Facebook public profiles, comprising approximately 1.9 million profiles. We extract users' interest related items, i.e., activities, books, films, interests, movies, music, television, etc., and user personal information attributes, i.e., *age*, *gender*, *relationship* and *interested in*. The availability of target attributes in the used dataset is summarized in Table 1, and the distribution of the number of items per attribute and per user are summarized in Tables 2 and 3, respectively.

attribute	**# records**	**unique**	**avg # users**	**25%**	**50%**	**75%**	**95%**
activities	918,525	11,405	80.54	15	27	66	277
books	371,142	5,125	72.42	13	21	45	240
films	313,679	3,577	87.69	15	28	66	303
interests	233,478	2,862	81.58	13	21	44	250
movies	975,105	6,693	145.69	15	30	89	555
music	2,055,576	16,313	126.01	13	22	53	375
television	1,362,780	6,462	210.89	14	26	72	658
all attributes	6,230,285	52,437	118.81	14	24	61	373

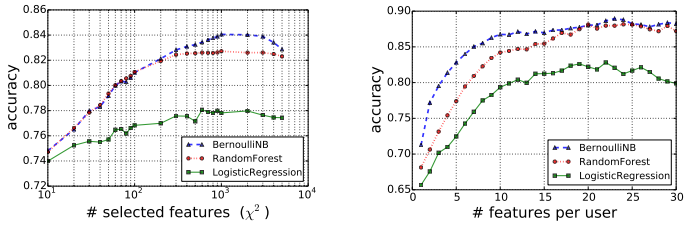
Table 2. Distribution of the number of items per attribute

Inference Results The inference attack on selected attributes is evaluated using the Facebook dataset, for *gender*, *age*, *relationship* and *interested in* (the actual values of classes are shown in Table 1), using the following shared interest attributes: *activities*, *books*, *films*, *interests*, *movies*, *music*, *television* and also *all attributes*, where we consider all possible public interest items from users' profiles as features. We use 10-fold cross validation and compute the average accuracy

attribute	# users	avg # items	25%	50%	75%	95%
activities	95,779	9.59	2	3	7	28
books	134,219	2.77	1	2	4	6
films	50,547	6.21	2	4	7	18
interests	65,730	3.55	1	3	4	9
movies	174,119	5.6	3	5	5	15
music	236,653	8.69	3	5	8	29
television	233,893	5.83	3	5	6	15
all attributes	249,847	24.94	12	15	24	69

Table 3. Distribution of the number of items per user

and AUC across all folds. For each combination of target attribute and interest type, we first extract all users who have revealed this target attribute and at least one item in the selected interest type, we then construct the user-item matrix based on this subset of users.



(a) Accuracy vs. the number of selected features (χ^2)

(b) Accuracy vs. the number of features that a user has

Fig. 1. Accuracy for inferring gender based on movie features, using different classifiers.

To understand how the number of selected features affects the inference, we perform two preliminary measurements on (a) accuracy vs. the number k of features selected for inference, (b) accuracy vs. the number of features per user. As an example, Figure 1(a) shows the results for the accuracy of inferring *gender* using movie items. We observe that the accuracy improves as the number of selected features increases, for all classifiers, and becomes stable when k reaches 1000. Interestingly, we observe that by selecting only 10 features, all classifiers achieved an accuracy higher than 74%. Based on this observation, we use $k = 1000$ features for the remainder of this study.

Figure 1(b) shows the performance of gender inference when using movie features, and for a varying number of features that a user has (regardless of whether they were selected). We can again observe improved accuracy, in line with an increasing amount of available information. We note that the accuracy become stable when a user has more than 10 features.

Attribute	**age** inference						**gender** inference					
	NB		**LR**		**RF**		**NB**		**LR**		**RF**	
	ACC	AUC	ACC	AUC	ACC	AUC	ACC	AUC	ACC	AUC	ACC	AUC
activities	0.501	0.536	0.502	0.529	0.551	0.556	0.80	0.718	0.777	0.604	0.813	0.643
books	0.502	0.536	0.533	0.509	0.494	0.522	0.843	0.709	0.795	0.535	0.809	0.564
films	0.591	0.564	**0.616**	0.505	0.598	0.52	0.814	0.800	0.763	0.70	0.80	0.77
interests	0.444	0.500	0.456	0.496	0.452	0.509	0.841	0.649	**0.806**	0.548	0.823	0.604
movies	0.545	0.581	0.520	0.528	0.551	0.571	0.837	**0.824**	0.774	0.712	0.827	**0.785**
music	0.594	0.595	0.568	0.547	0.585	0.575	0.752	0.724	0.703	0.654	0.737	0.698
television	0.601	0.605	0.552	0.558	0.572	0.586	**0.85**	0.823	0.790	0.686	**0.834**	0.774
all attributes	**0.624**	**0.613**	0.568	**0.573**	**0.605**	**0.586**	0.822	0.808	0.736	**0.723**	0.796	0.778

Attribute	**relationship** inference						**interested in** inference					
	NB		**LR**		**RF**		**NB**		**LR**		**RF**	
	ACC	AUC	ACC	AUC	ACC	AUC	ACC	AUC	ACC	AUC	ACC	AUC
activities	0.629	0.620	**0.592**	**0.570**	0.589	0.563	0.526	0.474	0.476	0.510	0.518	0.508
books	0.593	0.591	0.561	0.539	0.577	0.567	0.47	0.549	0.456	0.488	0.456	0.523
films	0.591	0.591	0.556	0.553	0.567	0.567	0.485	0.473	0.518	0.497	0.541	0.604
interests	0.560	0.499	0.581	0.498	0.568	0.503	0.464	0.561	0.440	0.532	0.467	0.565
movies	0.610	0.609	0.561	0.560	0.578	0.577	**0.745**	**0.783**	**0.688**	**0.729**	**0.726**	**0.771**
music	0.604	0.607	0.549	0.546	0.565	0.562	0.529	0.508	0.463	0.510	0.480	0.511
television	0.614	0.608	0.572	0.557	0.578	0.570	0.542	0.573	0.472	0.584	0.495	0.559
all attributes	**0.640**	**0.632**	0.573	0.563	**0.602**	**0.591**	0.740	0.487	0.617	0.489	0.6876	0.465

Table 4. Performance of inferring **age**, **gender**, **relationship**, **interested in**, using different information and classifiers; ACC: accuracy, AUC: area under curve, NB: Naïve Bayes, LR: Logistic Regression, RF: Random Forest.

Predictive Power of Interests Table 4 shows the inference accuracy and AUC of predicting *age*, *gender*, *relationship* and *interested in* for all interest types and for the three selected classifiers. The values corresponding to the best performance are highlighted in each column. We observe that predicting *gender* is the least challenging target, with prediction accuracy of up to 85% using television (program) items. This is followed by *interested in*, where the classifier can predict as high as 74% of the cases correctly and with AUC of 78.36%. Inferring *age* is more difficult, as there are more classes and little difference between classes. Comparing the performance of different classifiers, we observe that Bernoulli Naïve Bayes classifier outperforms the other classifiers for this inference task. We can observe that the predictive power for different types of interests changes when inferring a different attribute, e.g., movie items are a good indicator of *interested in* and using television items results in the best performance when inferring *gender*. It is interesting to note that considering all available items from different interest categories does not always result in the best performance.

3 Obfuscating User Profiles

We assume that obfuscation of a user profile would be realized via an application, which would recommend changes to specific attribute values based on the knowledge of profiles of other users (contributed by users in, e.g., a crowdsourced

scenario, and/or collected by the application back-end). The goal of obfuscation is to mitigate the inference attack on a selected (target) attribute, or a set of such attributes. We stress that in this work, we do not aim to provide provable privacy properties, but rather propose an empirical analysis of the state-of-the-art obfuscation techniques to protect users from inference attacks that leverage machine-learning approaches.

3.1 Obfuscating a Single Attribute: Strategies and Performance Evaluation

We first define the terms used to describe and evaluate the obfuscation approaches. For the sake of simplicity, we consider the case when a single attribute is targeted (and protected by obfuscation), although the following is also applicable to multiple target attributes. Obfuscation is accomplished by altering one or more background attributes A_b, so that the accuracy of inferring the class (value) of the target attribute A_x is minimized.

The attribute value (item) is chosen in the following way. In the initial step, all available items (from all users) are organized into classes, corresponding to the target attribute. Obfuscation is done using a selected item, from a class different from the one the target attribute belongs to. The user profile is modified according to a selected *obfuscation policy*, that can be: adding, removing or replacing an item. The item is chosen in line with a specific *obfuscation strategy*. The resulting improvement is measured as the reduction of the inference accuracy for the target attribute class, which can be perceived as *obfuscation gain*.

We consider a number of obfuscation strategies to choose the obfuscation items, based on rankings corresponding to selected metrics. For the purpose of obfuscation strategy evaluation, we first consider the optimal case of obfuscation, where the classifier used by the attacker is known to the (obfuscating) application. Therefore, we can choose the feature that achieves the highest *obfuscation gain*, based on the available data. To compute the *obfuscation gain* δ_k of a feature $k \in \{1, 2, ..., m\}$, we first set the k^{th} element in each user's profile to 1, resulting in the test set $\hat{X}_{test}$. Then, we compute $\delta_k = E(F(X_{test}), Y_{test}) - E(F(\hat{X}_{test}), Y_{test})$, where $E(X, Y)$ represents accuracy.

Figure 3.1 shows the distribution of *obfuscation gain* for the *gender* attribute, and for three representative classifiers using all features. For all classifiers, the effect of adding a single feature to the profile has a limited effect on the prediction accuracy, as more than 90% of the features only result in a 10% accuracy reduction. The CDF curves indicate that Logistic Regression is most resilient to obfuscation using a single feature, while Naïve Bayes has the lowest level of resilience.

However, there are many possible classifiers an attacker may use and, in practice, the assumption of prior knowledge of the classifier is not realistic. We therefore consider the following classifier-independent *obfuscation strategies*:

- χ^2: Ranking the candidate obfuscation items based on the χ^2 feature selection metrics, defined in Section 2.3.

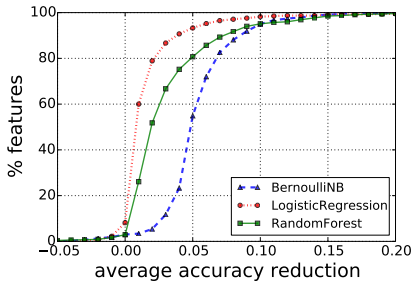


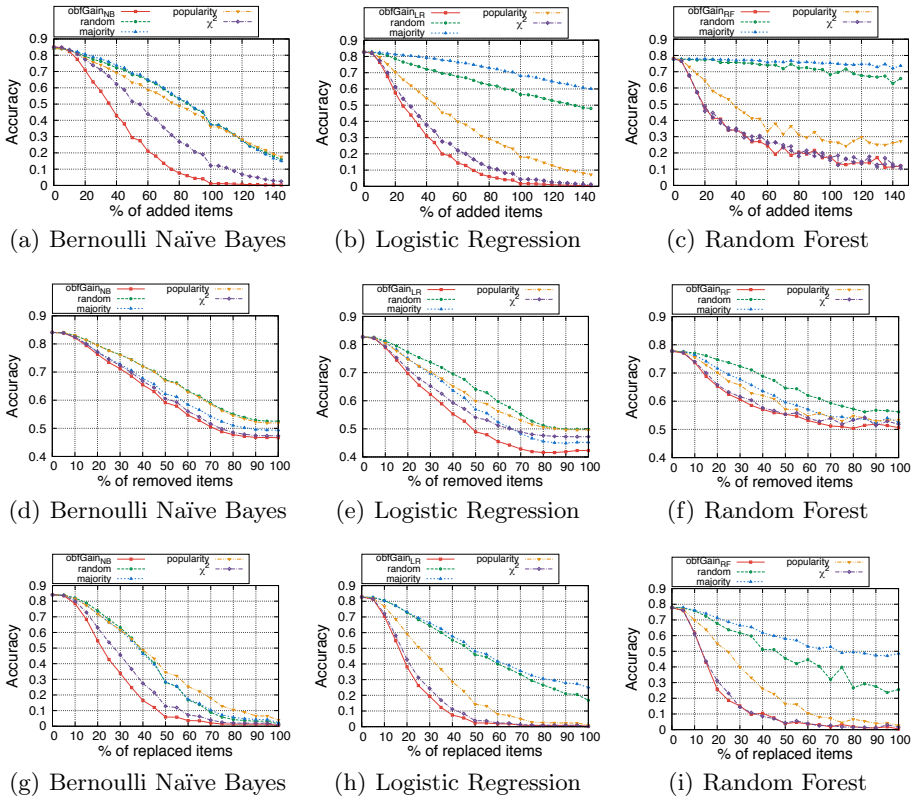
Fig. 2. Accuracy reduction distribution (CDF) for all features, using selected classifiers: Naïve Bayes, Logistic Regression and Random Forest.

- **Popularity**: Ranking the candidate items based on the number of users who have these items.
- **Majority**: Ranking the items based on the proportion of users in a specific class who possess them.
- **Random**: Randomly selecting the obfuscation items from a specific class.

For the evaluation, the set of items from a user's profile A_b is transformed into a binary vector v, with elements corresponding to the presence of items in the profile. *Adding* an item to the profile is equivalent to changing the corresponding element from 0 to 1; *removing* an item is equivalent to reversing a 1 to 0; and *replacing* an item is the combination of *removing* one item and *adding* another. The performance of obfuscation is evaluated by comparing the accuracy of inference for the original profile v and the obfuscated profile $\hat{v}$. To simplify the evaluation, we use *movies* as the background attribute to infer the target attribute *gender*. The obfuscation framework can be easily applied to other attribute combinations.

Figure 3 shows the performance of different *obfuscation strategies* for selected classifiers, with inference accuracy computed for a varying level of obfuscated items in the user profiles. As expected, the classifier-optimized *obfuscation strategy* results in the best performance. However, as noted, this requires prior knowledge of the classifier and the metric is computationally expensive to generate. The *obfuscation policy* of replacing items results in the highest level of obfuscation. Comparing the classifiers, Random forest and Logistic regression are more resilient to random noise than the Bernoulli Naïve Bayes. Finally, the χ^2 *obfuscation strategy* is closest in performance to the optimal strategy for all classifiers, indicating that this would be a good choice for a practical and cost effective solution.

To better understand the number of items that are needed to effectively obfuscate a user profile, Figure 4 shows the distribution of the number of items for all the users in our dataset, including the number of features before and after the χ^2 feature selection. Recall that for a meaningful inference, we filter out the users who have less than 10 items before the experiments, therefore the

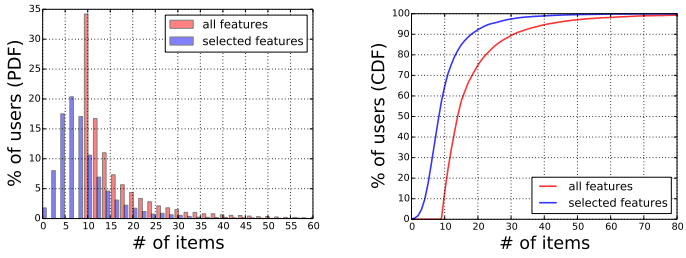


(a) Bernoulli Naïve Bayes (b) Logistic Regression (c) Random Forest

(d) Bernoulli Naïve Bayes (e) Logistic Regression (f) Random Forest

(g) Bernoulli Naïve Bayes (h) Logistic Regression (i) Random Forest

Fig. 3. Obfuscating *gender* inference by adding, removing and replacing movie items to user profiles using three selected classifiers: (a) - (c): Adding items; (d) - (f): Removing items; (g) - (i): Replacing items.



(a) PDF (b) CDF

Fig. 4. Distribution of the number of (selected) movie items per user.

distribution starts with 10 items. The values shown in Figure 3 indicate that only 20-30% of added items are sufficient to obfuscate a profile (the resulting inference accuracy is below 50%). The CDF curve in Figure 4(b) indicates that

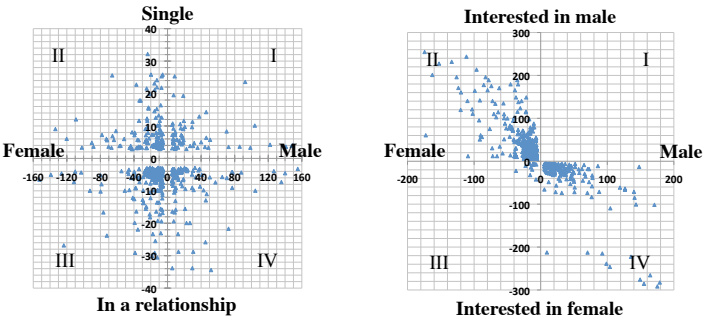
over 60% of the users have less than 10 selected items, which suggests that they only need to add 2 to 3 items to obfuscate their profile.

3.2 Obfuscating multiple attributes

We now consider how obfuscating a selected attribute may impact the inference of a different attribute and how multiple attributes may be jointly obfuscated.

We study the problem using two example pairs of target attributes: *gender* and *relationship* status, and *gender* and *interested in*. We focus on the χ^2 feature selection metric, shown to be a successful *obfuscation strategy* for single attributes in Section 3.1. To understand how each feature may contribute to the obfuscation of both target attributes, we show the χ^2 score distribution of each movie item in a 2-dimensional Cartesian plane in Figure 5. As the χ^2 score only indicates the inference strength of an item, but does not indicate the class it belongs to, we represent one of the classes related to an attribute by negative χ^2 values. In the example on Figure 5, we define *gender* "female", *relationship* status "in a relationship" and *interested in* "female" as negative values.

We observe that χ^2 score values are distributed in all four quadrants for the *gender-relationship* pair shown in Figure 5(a), while the values for *gender-interested in* combination in Figure 5(b) are mainly distributed in quadrants II and IV, indicating that the latter two attributes are strongly correlated. We note that our dataset contained no features indicating a combination of "male" and "interested in male", and included only a few weak features indicating "female" and "interested in female."



(a) (b)

Fig. 5. χ^2 score of items for inferring pairs of attributes: (a) *gender* vs. *relationship*; (b) *gender* vs. *interested in*.

We now evaluate the impact of obfuscating both single and multiple attributes on the inference of the selected attributes. Figures 6(a) and 6(b) show the inference accuracy for varying levels of obfuscation, for *gender* and for *relationship*, respectively, using the *obfuscation policy* of adding items, when either

a single or multiple attribute classes are obfuscated. The *obfuscation strategy* is χ^2, modified for the case when both attribute classes are obfuscated. For this, we choose items from a quadrant (see Figure 5) determined by both attribute classes, e.g., if a user is protecting the "male" and "single" classes, obfuscating items are chosen from quadrant III ("female" and "in a relationship" classes). For the case of an attribute having multiple classes, or for multiple target attributes, a general rule would be to select items that are not in the target class, for all attributes that the user wishes to protect.

We observe from Figure 6(a) that, in line with Figure 5(b), *gender* and *interested in* are strongly correlated, and therefore obfuscating *interested in* also obfuscates *gender*, achieving a similar *obfuscation gain* as obfuscation based on *gender*. We can also observe that obfuscating based on *relationship* decreases the inference accuracy of *gender*, but by a much lower value, indicating that items introduced by obfuscating *relationship* are less relevant to *gender*. When considering multiple attributes, both *relationship-gender* and, in Figure 6(b), *relationship-gender* strategy, are very close, respectively, to the performance of obfuscation based on the *gender* and *relationship* attributes alone. However, when jointly targeting *interested-gender*, it is difficult to achieve a high obfuscation performance level.

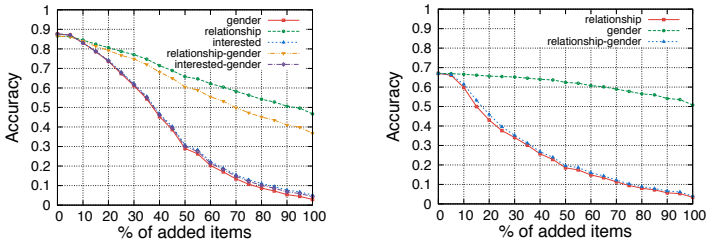


Fig. 6. (a) *Gender* inference accuracy while obfuscating *relationship* and *interested in*, and when considering multiple attributes. (b) *relationship* inference accuracy when obfuscating *gender*, and when considering both *gender* and *relationship*.

4 The Impact of Obfuscation

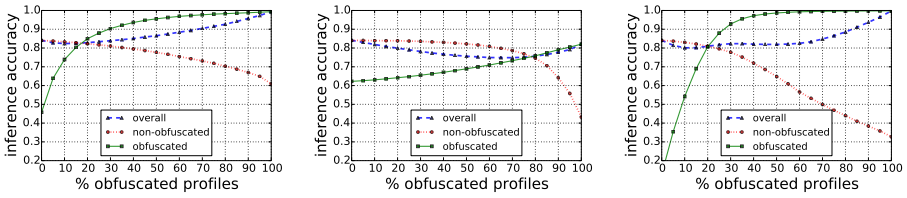
In the previous section, we assumed that the attacker learns from an initial version of user profiles, where no obfuscation was applied. As the number of obfuscating users increases, it is reasonable to assume that the attacker becomes aware of the obfuscation activity and adjusts the attack model. In this section, we examine how the obfuscation may affect the efficiency of the inference attacks, when the attacker re-learns constantly from the most up-to-date dataset (which

includes obfuscated profiles). Then we discuss how the attacker may use the knowledge of the users' obfuscation strategy to further improve the attack.

For ease of presentation, we use the example of *gender* inference, however we note that very similar results and trends were observed for other attributes we tested. In the following, we observe the impact of attribute obfuscation on the inference accuracy of (i) regular users who do not take part in the obfuscation process (non-obfuscated), and as such keep their original public profiles unchanged; (ii) users who have adopted a specific obfuscation strategy; and (iii) all users in the system.

4.1 Static Obfuscation Strategy

We start by studying a static obfuscation strategy that relies on an initial (non-obfuscated) dataset, and always delivers a fixed set of items to users. We split the dataset into a training set, representing the background knowledge of the attacker, and a test set, representing the set of target users. Then, we "pollute" a portion of the training set using χ^2 obfuscation strategy we discussed in the previous section. We assume that the obfuscating users modify (add, remove or replace) 50% of their profile items (this achieves a reasonably low inference accuracy, as per Section 3).



(a) Adding items (b) Removing items (c) Replacing items

Fig. 7. The impact of obfuscation on users' privacy: regular users, obfuscating users and overall users, with different portions of obfuscated profiles.

The inference accuracy for different percentages of obfuscated profiles in the training set is shown in Figure 7, for the cases of adding, removing and replacing items from user's profiles. Our results show that the inference accuracy for obfuscated profiles increases as the proportion of other obfuscated profiles becomes higher. This suggests a paradox, where privacy loss increases as more users become privacy-conscious.

To better understand why the obfuscation fails as the obfuscating population increases, we analyze the new information that is injected into the system (as part of the training data) by the obfuscating users. A static obfuscation strategy always suggests the same set of items to achieve the best obfuscation. As a result, the suggested items establish strong correlation with the value that the user is

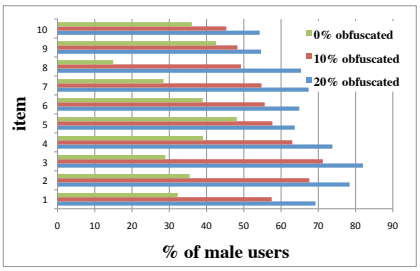


Fig. 8. The change in class bias for top 10 obfuscating features, as the percentage of obfuscated users increases.

trying to hide. For example, Figure 8 shows the percentage of *male* users who have one of the top-10 *female*-associated items (based on the χ^2 score) when there are 0%, 10% and 20% of obfuscated profiles. Initially all items are female dominant and hence have strong association with *female* during inference. As the number of *male* users who obfuscate their profiles by injecting these items increases, the *male*/*female* ratio changes and the *male* users become dominant at 10% and 20% obfuscated profiles in the system. Adding these (reversed) items to users' profiles hence becomes a strong indication for the class the users are attempting to hide.

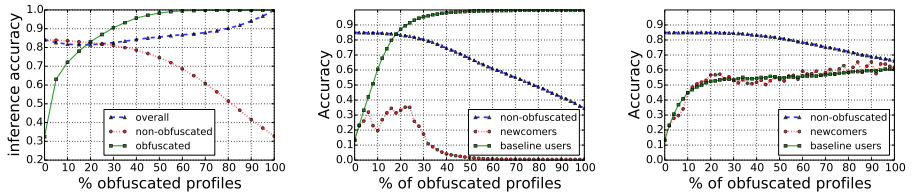
Notably, the inference accuracy for regular users decreases as the number of users applying the obfuscation increases. This is again due to the attacker learning from a set of items consisting of more obfuscated items, which equates to adding noise to the "clean" profiles.

4.2 Dynamic Obfuscation Strategy

The main issue for the static strategy is that the injection of a similar set of false items into the system, eventually results in those items becoming indicative of the attributes users are attempting to keep private. We now investigate two obfuscation strategies that recommend items dynamically: first we revisit the experiment using a random obfuscation strategy; second, we assume the obfuscation engine refreshes the strategy based on the most up-to-date data.

Recall that the random strategy chooses items from a different class randomly, regardless of the obfuscation strength of the items. Although this approach has inferior performance compared to more optimal strategies, it introduces diversity, which increases the difficulty of identifying the obfuscated items. We perform the same experiment as in Section 4.1 using the random obfuscation strategy, and show the inference accuracy in Figure 9(a). We observe a similar result to what was achieved by the static χ^2 strategy (Figure 7(c)), which suggests that spreading the recommended items randomly does not resolve the issue.

We then consider two dynamic obfuscation scenarios: first, we assume there is a baseline of x% obfuscated profiles using the initial static strategy (static baseline). We evaluate the obfuscation gain for newcomers who obfuscate their



(a) random replace (b) static baseline replace (c) dynamic baseline replace

Fig. 9. Impact of obfuscation on users' privacy using dynamic strategy, (a) random strategy is used by all users. (b) training set polluted using the baseline static strategy (c) training set polluted using the baseline dynamic strategy

profile based on up-to-date data; the results are shown in Figure 9(b). We observe that the early adopters (baseline users) of obfuscation face a high risk of successful inference. In contrast, newcomers who adopt the dynamic strategy, have well obfuscated profiles.

In the second scenario, we consider a realistic situation where obfuscation is adopted by users gradually over time; at each time period, we introduce 1% obfuscated profiles to the training dataset (dynamic baseline). We then show how the inference accuracy evolves as the portion of obfuscated profiles increases for both early obfuscation adopters (baseline users) and newcomers. The results are shown in Figure 9(c). We observe an increase in the inference accuracy for between 0% to 20% of the obfuscated profiles in the system, both for baseline users and newcomers. Unlike the previous scenarios, the inference accuracy becomes stable for both strategies. This observation suggests that using dynamic and the most up-to-date strategy is beneficial both for early adopters of obfuscation and for newcomers.

4.3 Limitations of the Obfuscation Strategies

In response to the obfuscation activities, the attacker can adopt stronger attack models that actively identify polluted items and detect obfuscated profiles and attributes. With such a capability, an attacker may use the knowledge of the obfuscating items as recommended by the obfuscation engine to filter out noisy items before launching the inference attacks. Likewise, an attacker may take advantage of detected obfuscation behavior to infer the private attribute that the user is trying to hide.

There are several ways to detect an obfuscated profile. Firstly, the attacker may access the obfuscation application in the same way as a legitimate user. This method is effective for static obfuscation strategies as the attacker can easily obtain a list of the most likely obfuscation items.

For the dynamic strategies, a powerful attacker may detect obfuscation by monitoring abrupt changes of profile items that are inconsistent with previous

inference results. Arguably, the use of dynamic obfuscation strategy and progressive introduction of obfuscation items would benefit the users towards resisting more sophisticated inference attacks.

5 Considering User Preferences

We examined users' preferences for the choice of obfuscation items via a survey, performed on the online crowdsourcing survey platform CrowdFlower.[3]

Our survey first asked the respondents to rate the level of sensitivity of the personal information in their profiles on a scale of $1-5$. Then, the survey asked them to rate the preference for adding specific movies to their profile, in order to protect their *gender* and *relationship* information from being inferred by a third-party. The evaluation focused on three factors that may affect a user's decision to include an item in his/her profile: (1) *privacy protection level*: we included three levels, high, medium and low, corresponding to values of χ^2; (2) *popularity*: high or low, related to the number of users with this item in our dataset; and (3) *quality*: high or low, based on IMDb[4] movie ratings. The list of movies in the survey was carefully selected so that each item could be related to a combination of these factors. For different user groups, i.e., for different combinations of gender and relationship status, we provided a bespoke version of the survey. The full survey can be found in the technical report [4].

We received 254 responses, with 158 responses that we considered valid (we removed completed surveys that took less than 2 minutes to complete or that had identical responses to almost all questions). We restricted the survey to users from English speaking countries, with the respondents being from: US 36.7%, Canada 35.4%, UK 17.7%, Australia 6.3%, New Zealand 3.8%. The *gender-relationship status* distribution of the respondents was: male-single 17%, male-in a relationship 19.6%, female-single 12.6% and female-in a relationship 50.6%.

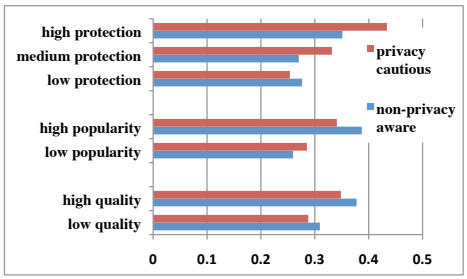


Fig. 10. User provided ratings for different criteria; privacy conscious user: average attribute sensitivity rating higher than 3; non-privacy aware user: average attribute sensitivity rating lower than 3

[3] http://crowdflower.com/
[4] http://imdb.com/

Based on the average rating for the level of sensitivity of their personal information, we classified users into two groups: privacy conscious users (rating above 3) and non-privacy aware users (rating below 3). The proportion of positive ratings (movies that were rated as acceptable for obfuscation) for the two groups is shown in Figure 10. We observe that the privacy conscious users have a higher likelihood of accepting movies that provide high protection, with a 17% difference between the high and low protection levels; the trend is less noticeable for non-privacy aware users, with only 7% difference between the two. Similarly, the respondents also prefer movies that are popular and of high quality.

In addition, we also sought participants' opinion about obfuscating OSN profiles in general. As this was an open question, we did not quantify the opinions, but based on the received comments the participants can be categorized to users who: (1) do not understand the inference attack; (2) do not wish to add any non-genuine content to their profiles; (3) would add any items to protect their privacy; (4) accept only items that are consistent with the image that they wish to present to others, i.e., items the users genuinely like. The last comment is representative of a non-negligible portion of users (12 out of 65 who answered this question) and motivates the design of a user-friendly obfuscation strategy.

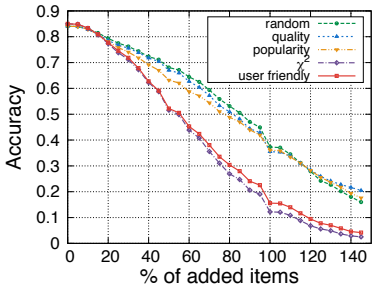


Fig. 11. Inference accuracy for gender using movie items.

5.1 User-friendly obfuscation strategy

We now consider the results of the user study, which indicate that *popularity* and *quality* of items need to be considered to have an acceptable obfuscation strategy. As shown in Figure 11, if we apply the obfuscation strategy solely based on these factors, the performance of the obfuscation is quite poor, i.e., similar to the random strategy.

We therefore propose a user-friendly obfuscation strategy that while taking into account both factors of *popularity* and *quality*, yields a significant improvement in obfuscation performance compared to the strategy that exclusively uses these factors. We first select items that are in the top n most popular and top n highest quality lists, where n is a variable that controls the size of the intersection between the popularity and quality sets. We heuristically choose $n = 100$,

which results in a subset of 69 common items (that is sufficient for the majority of users to obfuscate their profiles). We then apply the χ^2 score metric to the set of items as the obfuscation strategy; χ^2 provides a solid obfuscation performance, as shown in Section 3.1. The performance of the new strategy is close to χ^2, as shown in Figure 11, while satisfying users' preferences for the choice of obfuscation items.

6 Related Work

Kosinski et al [9] have shown that potentially sensitive attributes like sexual orientation and political views can be inferred, with high precision, using Facebook likes. A number of other research works [8, 13] address the inference of users' sensitive attributes from social links, and the protection mechanisms based on selective adding or removal of such links [7]. Ryu et al. [15] evaluated the performance of inferring sensitive attributes using a deterministic algorithm, logistic regression and matrix factorization. They also studied the impact of friends' privacy policies (for selected sensitive attributes) on the potential to infer these attributes for users who do not publicly reveal them. Their approach bases the inference on social links and contact information, while in this paper we only relied on on the information publicly available in the user's own profile.

Weinsberg et al. [17] studied the performance of obfuscation methods using different classifiers, and the impact of obfuscation on the utility of recommendations, in a movie recommender system scenario. They evaluated a number of classifiers and selected obfuscation methods, including greedy, sampled and random choice of obfuscation items. Their work is closest to the study presented in this paper, however their obfuscation approaches assume prior knowledge of the classifier used for the inference attack. Salman et al. [16] proposed a practical methodology to prevent statistical inference (relying on the theoretical framework of [5]); the proposed mechanism distorts the data before making it publicly available, while providing a guarantee of the data utility. Li at al. [10] also present a mechanism for preventing inference attacks (association rules) in a data publishing scenario. All these works considered, in a system setting, only the resulting loss of data utility for other system users and a static privacy mechanism. Our work evaluates the impact of obfuscation on the privacy of other users, and also considers a dynamic mechanism.

7 Conclusion

This paper investigated a set of practical problems related to the design of a usable obfuscation system, to mitigate inference attacks in OSNs. The user study indicates that a number of factors, not directly related to the performance of obfuscation mechanisms, need to be considered to progress towards a system that may be acceptable to privacy-conscious users. We believe that our user-friendly obfuscation strategy, which is designed to integrate the user preferences into an effective solution is a first step in the direction of user acceptability. However,

there are a number of additional challenges. The proposed strategy needs to be verified in a real world setting, with development of such an application planned for future work. Then, although the dynamic obfuscation mechanism can provide improved performance compared to the static case, a level of coordination between users may be required in order to achieve an acceptable obfuscation performance in the long term.

References

1. C. A. Ardagna, M. Cremonini, E. Damiani, S. D. C. di Vimercati, and P. Samarati. Location privacy protection through obfuscation-based techniques. In *Proceedings of the 21st IFIP Working Conference on Data and Applications Security*, 2007.
2. L. Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001.
3. A. Chaabane, G. Acs, and M. A. Kaafar. You are what you like! Information leakage through users' Interests. In *Proceedings of the 19th NDSS*, 2012.
4. T. Chen, R. Boreli, M.-A. Kaafar, and A. Friedman. On the effectiveness of obfuscation techniques in online social networks. Technical Report 1833-9646-8065, National ICT Australia, April 2014.
5. F. du Pin Calmon and N. Fawaz. Privacy against statistical inference. In *Allerton Conference*, pages 1401–1408, 2012.
6. I. Guyon and A. Elisseeff. An introduction to variable and feature selection. *J. Mach. Learn. Res.*, 3:1157–1182, Mar. 2003.
7. J. He and W. W. Chu. Protecting private information in online social networks. In *Intelligence and Security Informatics*, pages 249–273. Springer, 2008.
8. J. He, W. W. Chu, and Z. V. Liu. Inferring Privacy Information from Social Networks. In *ISI*, Berlin, Heidelberg, 2006.
9. M. Kosinski, D. Stillwell, and T. Graepel. Private Traits and Attributes are Predictable from Digital Records of Human Behavior. *PNAS*, 110(15), 2013.
10. C. Li, H. Shirani-Mehr, and X. Yang. Protecting individual information against inference attacks in data publishing. In *Proceedings of DASFAA*, 2007.
11. J. Lindamood, R. Heatherly, M. Kantarcioglu, and B. Thuraisingham. Inferring Private Information Using Social Network Data. In *WWW*, 2009.
12. C. D. Manning, P. Raghavan, and H. Schütze. *Introduction to Information Retrieval*. Cambridge University Press, New York, NY, USA, 2008.
13. A. Mislove, B. Viswanath, K. P. Gummadi, and P. Druschel. You Are Who You Know: Inferring User Profiles in Online Social Networks. In *WSDM*, 2010.
14. S. T. Peddinti and N. Saxena. On the privacy of web search based on query obfuscation: A case study of trackmenot. In *PETS*, 2010.
15. E. Ryu, Y. Rong, J. Li, and A. Machanavajjhala. Curso: Protect yourself from Curse of Attribute Inference: a Social Network Privacy-analyzer. In *WDSN*, 2013.
16. S. Salman, Z. Amy, d. P. C. Flavio, B. Sandilya, F. Nadia, K. Branislav, O. Pedro, and T. Nina. How to Hide the Elephant or the Donkey in the Room: Practical Privacy Against Statistical Inference for Large Data. In *GlobalSIP*, 2013.
17. U. Weinsberg, S. Bhagat, S. Ioannidis, and N. Taft. Blurme: inferring and obfuscating user gender based on ratings. In *RecSys*. ACM, 2012.
18. S. Ye, F. Wu, R. Pandey, and H. Chen. Noise Injection for Search Privacy Protection. In *Conference on Computational Science and Engineering*, 2009.
19. Z. Zheng, X. Wu, and R. Srihari. Feature selection for text categorization on imbalanced data. *SIGKDD Explor. Newsl.*, 6(1):80–89, June 2004.