Divide-and-Learn: A Random Indexing Approach to Attribute Inference Attacks in Online Social Networks

Sanaz Eidizadehakhcheloo, Bizhan Alipour Pijani, Abdessamad Imine, Michaël Rusinowitch

Divide-and-Learn: A Random Indexing Approach to Attribute Inference Attacks in Online Social Networks*

Sanaz Eidizadehakhcheloo[1], Bizhan Alipour Pijani[2], Abdessamad Imine[2], and Michaël Rusinowitch[2]

[1] Sapienza Università di Roma, 00185, Roma, Italy
eidizadehakhcheloo.1772528@studenti.uniroma1.it
[2] Lorraine University, Cnrs, Inria, 54506 Vandœuvre-lès-Nancy, France
bizhan.alipourpijani@loria.fr, abdessamad.imine@loria.fr, rusi@loria.fr

Abstract. We present a Divide-and-Learn machine learning methodology to investigate a new class of attribute inference attacks against Online Social Networks (OSN) users. Our methodology analyzes commenters' preferences related to some user publications (e.g., posts or pictures) to infer sensitive attributes of that user. For classification performance, we tune Random Indexing (RI) to compute several embeddings for textual units (e.g., word, emoji), each one depending on a specific attribute value. RI guarantees the comparability of the generated vectors for the different values. To validate the approach, we consider three Facebook attributes: gender, age category and relationship status, which are highly relevant for targeted advertising or privacy threatening applications. By using an XGBoost classifier, we show that we can infer Facebook users' attributes from commenters' reactions to their publications with AUC from 94% to 98%, depending on the traits.

Keywords: Social Networks · Privacy · Attribute Inference Attack · Random Indexing.

1 Introduction

Although OSN users are getting more cautious about their privacy, they remain vulnerable to attribute inference attacks where the principle is to illegitimately gain private attributes (such as age, gender or political orientation) from publicly available information. In general, attribute inference attacks are either based on data directly generated by the target user (such as pseudos) or data obtained by exploring the user vicinity network. The need to protect the sensitivity of some attributes compels users to conceal information that may disclose them. For instance, the authors of [6] investigate Facebook users' privacy awareness and show that age has the lowest exposure rate. Less than 3% of the users (about

* This work is funded by DigiTrust (http://lue.univ-lorraine.fr/fr/article/digitrust/).

495k users) reveal their age, which shows the sensitivity of this attribute. They also show that half of their members hide their gender and 37.3% conceal their friend list. Therefore collecting user-generated data is a difficult task in a real scenario. We note that most attribute inference attacks get inoperative in the case where no data are provided by target users in their profiles. For instance many attribute inference systems are based on network homophily [25] and do not apply without the availability of friend lists.

Even with more user awareness, the privacy risks can come from other sources. Indeed, we will show that an attribute can be inferred from non-user generated data such as the reactions of other social media users on the target user's shared contents. Users in OSN share contents, so-called *publications*, to give people a better sense of who they are and what they care about. For example, Twitter users share a tweet to express their opinion. As for Facebook, users publish pictures, posts or status updates to display their beliefs, favorite brands, grab attention and nourish relationships. While many users hide their sensitive attributes (e.g., gender, age or political view), these publications are still available to the public. However, many users do not realize that even though they are cautious about their writing style, other users' reactions to their publications can reveal their sensitive information. As a typical reaction, people spontaneously *comment* on publications. This option allows them to engage and impress their personal opinion, which might create a sense of connectedness between the publication author and commenters (especially when the commenters are friends). We consider these comments as part of the publication metadata. We also consider *alt text* as metadata which is freely available textual information describing the picture contents (faces, objects, and other themes) and which is generated by some OSN platforms (like Facebook) for blind people who use screen readers.

In this paper, we describe how to infer attributes from publications metadata. We suspect that OSN users' publications metadata convey sensitive information, even though the users did not contribute to generating them directly. We will show that a target user's attribute can indeed be leaked from other users' reactions (e.g., comments) to the target user's publications, even without analyzing the publication content. Note that this new type of attack is difficult to defeat as the user has no control over other users' reactions. Demonstrating the risks of attribute inference attacks raised by publication metadata gives us concrete grounds for alerting about their sensitivity.

A central problem and our solution. An important problem when learning attributes from metadata is that the same term often appears with different contexts in different attribute values. Such a term will be called an *overlapped* term and the value-specific contexts will be called *non-overlapped* contexts. Example 1 shows female and male-owned pictures with generated alt-text to describe the picture content and comments posted by commenters. The tag *1person* and the word *baby* are overlapped terms, while *you*, *miss*, and *cray* are non-overlapped contexts. The commenters employ different neighboring words for the term *baby* when commenting on female and male-owned pictures, which demonstrates the

commenters' usage preferences. As a result, there is a variation in style and usage of the same term in the context of different attribute values.

Example 1. Metadata of two pictures.

Metadata of an image published by a female user
Generated alt-text: 1person
Comment: miss you baby

Metadata of an image published by a male user
Generated alt-text: 1person
Comment: cray cray baby 😆

To discover these variations, we apply a semantic space model, known as Distributional Semantic Model (DSM). Classical word embeddings (such as word2vec [20]) uncover the semantic relations among terms by scanning through the whole corpus and detecting co-occurrences in a fixed context window. They build a global view of terms co-occurrence in the entire dataset. In Example 1, *baby* is used as a romantic term of endearment in the female-owned picture, whereas in the male-owned picture it is about making fun of and teasing the picture owner. Hence generating a vector for each word using the entire dataset can mix and combine many possible word contexts.

We need to adapt the distributional semantic model so that an *overlapped* term with *non-overlapped* context will get different corresponding vector representations. These various representations should be comparable since attribute prediction often relies on computing similarities between vector representations of terms and users. However, due to different random initialization processes on the sub-datasets (corresponding to distinct attribute values), the generated vectors are not comparable by the standard similarity measures, such as cosine similarity [13]. To avoid this problem, we apply Random Indexing (RI) [26], which is an incremental and scalable method for constructing a vector space model. RI requires few computational resources for similarity computations and allows comparison of word spaces created over different attribute values [2]. Accordingly, from the same term, we can generate vectors in different attribute values and compare them.

As a case study, we conduct an intensive analysis of three Facebook attributes: age, gender, and relationship status, as they are recognized as key privacy concerns in the Internet era. Some Facebook users choose to hide gender to camouflage themselves against stalking, sexual harassment, or reducing discrimination [28]. Our attack receives promising results as we preserve the vector representation of each word for each attribute value. The result confirms that splitting the dataset can boost the attacker's performance.

Paper Organization. In Section 2 we review the related works. Section 3 presents our Divide-and-Learn methodology to incorporate attribute values in word vector generation. We outline our attribute inference attack steps in Section 4. We present a case study in Section 5 to evaluate our attacks. Finally, we conclude in Section 6 by discussing the capability of our attribute inference attacks on other OSN platforms and giving possible future work.

2 Related works

User profiling based on their available data on social media has obtained remarkable attention in the past decade. It is a key ingredient of recommendation systems. Researchers have investigated popular social media platforms and have leveraged all possible available data such as content sharing [3], friendship [9], behavior [17] to perform their attribute inference attacks. [29] considers users' purchase data for predicting multiple demographic attributes simultaneously. The authors of [1] show how an attacker can leverage seemingly harmless interests to reveal sensitive information about users. In particular, they infer user private attributes based on music interest similarities. [30] shows that a movie rating recommender system can infer the user's gender without additional metadata. [10] combines network structure and node attribute information to perform link prediction and attribute inference attacks. These are motivated by the observed interaction and homophily between network structure and node attributes. An active attack on privacy-preserving publication of social graphs is presented in [19]. Demographic variable attacks on Twitter users based on whom they follow are presented in [4]. Research efforts have been also focused on user writing style (i.e., users' messages, posts, and status updates) [21] and word usage [27] to infer undisclosed attributes. They apply language analysis to the text generated by users and implement machine learning approaches to achieve the attack.

To sum up, the mentioned works mainly require exploration of user vicinity networks (e.g., friend lists) and digital records (e.g., profile attributes, joined groups and liked pages), which might be unavailable in a real scenario or computationally costly to collect. This personal data as well as writing styles can be modified by the target user to escape inference attacks. In contrast, we infer target user attributes from commenters and Facebook generated data (both called here *publication metadata*). This metadata is easily available. Since our approach does not need to explore the target user's vicinity network, groups and pages, inference attacks are efficient and can even be launched *online*.

Our work is related to [23,5] where gender and age are inferred from Facebook picture metadata. However, here we do not limit ourselves to specific attributes and our attacks apply to many social media by leveraging commenters' preferences related to pictures, posts, and status updates. Our approach is inspired by recent techniques for analyzing word semantic changes over time [8].

3 Divide-and-Learn methodology

In this section, we first explain how the training dataset is divided into sub-datasets according to the different values of the attribute to be inferred. Next, we introduce the distributional semantic space and our proposed value-based random indexing approach. The notations used in this paper are summarized in Table 1.

Notations	Descriptions
D	collected training dataset
U	set of users
$u \in$ U	user in U
W	vocabulary of the training dataset
$w \in$ W	word in W
$\boldsymbol{w}$	distributional vector of w
c	context
$C(w)$	set of contexts of w in D
$P(u)$	set of publications of $u \in U$
l	an attribute
l_m	mth value of l
U^l_m, U_m	set of users s.t. attribute l has mth value (U_m when l is implicit)
W_m	set of words in comments of publications from users in U_m
$C_m(w)$	set of contexts of w in D_m
$\boldsymbol{u}$	vector for u

Table 1: Notations.

3.1 Dividing training datasets

Here, we introduce our splitting conditions and their computation. In the following we use *term* as a shorthand for *word/emoji/tag*.

Criteria for splitting. We can argue that splitting is not beneficial if the commenters use (i) a majority of different terms while commenting on users' publications in different attribute values, or (ii) more frequently the same terms co-occurring more often with similar contexts than dissimilar ones (see Subsection 5.3). Examples 2 and 3 illustrate cases where splitting the dataset would not be beneficial.

Example 2. Metadata of two pictures.

Metadata of an image published by female user
Generated alt-text: 2people
Comment: Wooooooow, its NICE

Metadata of an image published by male user
Generated alt-text: outdoor, sunglasses
Comment: look at the long beard

Example 3. Metadata of two pictures.

Metadata of an image published by female user
Generated alt-text: selfie, closeup
Comment: great picture

Metadata of an image published by male user
Generated alt-text: selfie, closeup
Comment: great picture

However, in the complementary cases, our experiments have shown the neat benefit of splitting for accuracy (see Subsection 5.3). For instance, if males and

females are commented with the same terms and the contexts of those terms are mostly specific to an attribute value (see Example 1), we can take advantage of this variation and split the dataset to generate vectors that are biased towards that attribute value. We, therefore, propose two conditions to be jointly satisfied in order to split the training dataset. To express these conditions, we define the importance of a set L that contains terms or contexts w as

$$Q(L) = \sum_{w \in L} freq(w)$$

where *freq(w)* is the number of occurrences of w in the dataset.

Condition 1 is satisfied when the set of overlapped terms is more important than the set of non-overlapped terms.

Condition 2 is satisfied when the set of non-overlapped contexts is more important than the set of overlapped contexts.

In Example 1, the first condition is satisfied by the overlapped terms (*1person* and *baby*), and the second condition is met as the overlapped terms have different contexts (*miss you* and *cray cray*). None of the conditions are satisfied in Example 2 and only the first condition is satisfied in Example 3. Therefore, a unique term representation is sufficient in the two latter cases without missing the variations in the contextual meaning of the terms.

Dataset dividing algorithm. We label the original training dataset D in such a way that the ith sub-datasets D_i contains users labeled with the ith attribute value and words appearing in their publications comments. Algorithm 1 has for inputs D and D_is, and it returns a boolean that is true if D has to be split into sub-datasets D_i. We introduce the following notations:

1. $l_1, l_2, ..., l_k$ represents the attribute values. If l is gender attribute, then l_1 ="male" and l_2 = "female".
2. D_i is the ith sub-dataset containing the set of users with attribute value l_i (with $i \in \{1, .., k\}$). For gender attribute, we obtain D_1 and D_2 as sub-datasets annotated by male and female.
3. W_i is the set of words in comments of pictures published by users in U_i.
4. $UTop$, and $BTop$ are integer parameters.
5. $Unigram(D_i, UTop)$ computes Uni_i, the set of $UTop$ most frequent terms in D_i.
6. $Bigram(Uni_i, BTop)$ computes Big_i, the set of $BTop$ most frequent terms in D_i co-occurring with terms in Uni_i.
7. $Tcount(t) = |\{i \in \{1, \dots k\} \mid t \in Uni_i\}|$, (resp. $Ccount(c) = |\{i \in \{1, \dots k\} \mid c \in Big_i\}|$) is the number of sets Uni_i (resp. Big_i) where a term t (resp. a context c) appears.

Suitable values of $UTop$ and $BTop$ will be determined from experiments (see Subsection 5.2).

Algorithm 1: Dataset dividing algorithm

input : $D, D_1, \ldots, D_k$
output: true iff the dataset D has to be split in $D_1, \ldots, D_k$

Step1:
for $i = 1, \ldots k$ **do**
  $Uni_i \leftarrow Unigram(D_i, UTop)$
  $Big_i \leftarrow Bigram(Uni_i, BTop)$
Step 2:
$OT \leftarrow \bigcap_{i=1}^{k} Uni_i$; $OC \leftarrow \bigcap_{i=1}^{k} Big_i$
$T \leftarrow \bigcup_{i=1}^{k} Uni_i$; $C \leftarrow \bigcup_{i=1}^{k} Big_i$
$NT \leftarrow \{t \in T \mid Tcount(t) = 1\}$; $NC \leftarrow \{c \in C \mid Ccount(c) = 1\}$
if $Q(OT) > Q(NT)$ ***and*** $Q(OC) < Q(NC)$ **then** true;
else false;

3.2 Random Indexing

Random Indexing (RI) is a fast dimensionality reduction method that transforms high-dimensional data into a lower-dimensional one by using a random matrix. It generates distributional representations that approximate similarities in sets of co-occurrence weights. RI assigns a randomly generated vector to each unique term in the text, the so-called index vector. These index vectors are sparse, n-dimensional, and ternary. They consist of a small number of randomly distributed non-zero elements, $\{-1, +1\}$. Each unique term is also represented by an n-dimensional initially empty vector, called distribution vector. RI incrementally updates the n-dimensional distribution vector of each word by summing the n-dimensional index vector(s) of all co-occurring words within a small window of text. As a result, terms appearing in a similar context tend to have a similar distributional vector. Let $c = [c_{-n}, \ldots c_{-1}, w, c_1, \ldots c_n]$ be the context of w with window from $-n$ to n (n chosen between 1 and 5) and let $\boldsymbol{c}$ be the vector obtained by accumulating word's index vector co-occurring with w in context c. We update the distributional vector $\boldsymbol{w}$ by using RI as follows:

$$\boldsymbol{w} = \sum_{c \in C(w)} \sum_{\substack{-n \leqslant j \leqslant n \\ j \neq 0}} \boldsymbol{c_j} \tag{1}$$

The problem with this approach is that the entire dataset potentially contributes to each term vector representation. Therefore, the vectors are affected by the different attribute values and lose their discriminating power for attribute inference attacks. To remedy this problem, we propose to generate several value-based vectors for each term [12].

Values-based random indexing. Despite its simplicity, RI struggles to capture the relation between commenters' words/emojis usage preferences and the owners' publication profile. However, it can be adjusted for our task. Given a set of users U and an attribute l with k values $l_1, l_2, \ldots, l_k$, we introduce the subsets

$U_1, U_2, .., U_k$ of U, where U_m is the set of users whose attribute value is the mth value of l. Similarly, if W is the vocabulary of all comments for publications of users in U, we consider k sub-vocabularies $W_1, W_2, ...W_k$, such that each W_m records the commenters' preferences for a user in U_m.

In this way, we distinguish the different contexts of a term appearing within profiles with different attribute values. It is a key aspect of our inference attacks since the vector of a term occurring in W_m will be computed from its co-occurrences with other terms from W_m. Formally, instead of computing with standard RI, a single vector $\boldsymbol{w}$ from the entire W, we compute k vectors $w_1, w_2, ...w_k$, where w_m is derived from W_m, as follows:

$$\boldsymbol{w}_m = \sum_{c \in C_m(w)} \sum_{\substack{-n \leqslant j \leqslant n \\ j \neq 0}} \boldsymbol{c}_j \tag{2}$$

From Equation 2, we generate distinct vectors for the same term for different values. Previous word embedding approaches generate a single vector for each term appearing in Example 1 by combining different context terms. These approaches miss word semantical variations corresponding to different attribute values. In our approach, we can rely on Equation 2 to compute several vector representations for the same term, each one corresponding to an attribute value.

Generating index vectors. RI relies on two important hypotheses. First, in high dimensional space, there exists a much larger number of almost orthogonal than orthogonal directions, according to Hecht-Nielsen [11]. Second, if we project points of a vector space into a randomly selected high dimensionality subspace, the distances between these points are approximately preserved (Johnson-Lindenstrauss-Schechtman lemma [16]). The choice of random matrix is an essential aspect of RI to satisfy these two hypotheses. In this work, we train a machine learning algorithm to find the best parameters of RI, namely the dimension and non-zero elements (see Subsection 5.2). By following the steps mentioned above, our approach provides suitable vectors to perform attribute inference with higher accuracy than with alternative embeddings.

4 Attribute inference attacks

We consider an attacker who intends to infer OSN users' attributes from a set of publications P where each publication contains metadata. Thus, the OSN is exposed to potential privacy violations by an attacker (external or OSN user) collecting, storing and analyzing publications metadata from user profiles.

Once we learn the vector representations of terms for each attribute value (see Equation 2), we compute a vector representation of u_m by aggregating all the terms that appear in his/her publications as follows:

$$\boldsymbol{u}_m = \sum_{w \in P(u_m)} \boldsymbol{w}_m \tag{3}$$

We introduce a set S of vectors computed by Equation 3. For the target user t, we generate a set of user vectors $T = \{\boldsymbol{t}_1, \boldsymbol{t}_2, ...\boldsymbol{t}_k\}$, where vector $\boldsymbol{t}_m$ is obtained as follows:

$$\boldsymbol{t}_m = \sum_{w \in P(t)} \boldsymbol{w}_m \tag{4}$$

We compute cosine similarity [20,22] between T and S to check which user in S has the most similar vector to the vector of a user in T as follows:

$$(t_\mu, u_\mu) = \arg\max_{\substack{\boldsymbol{t} \in T \\ \boldsymbol{u} \in S}} (cosine(\boldsymbol{t}, \boldsymbol{u}))$$

The *arg max* function outputs a tuple (t_μ, u_μ) and we infer that the target attribute value is l_μ. Cosine similarity is a common similarity measure for vectors when their magnitude is not relevant (e.g., they represent linguistic items in distributional semantics [14]).

As a concrete example, we consider a gender inference attack against a user named *Target*. Given two gender values l_1 ="male" and l_2 = "female", we generate two vectors $\boldsymbol{w}_1$ and $\boldsymbol{w}_2$ for *Target* where $\boldsymbol{w}_1$ and $\boldsymbol{w}_2$ correspond to vector of w in D_1 and D_2, respectively. Vectors *Target_F* and *Target_M* (red points in Figure 1) represent female and male hypotheses vectors, respectively. Using cosine similarity, we compute in Figure 1 the closest users of S to *Target_F* (blue dots) and the closest users of S to *Target_M* (blue dots). As *Raymond* is closer to *Target_M* than *Berkeley* to *Target_F*, we label therefore *Target* by *male*.

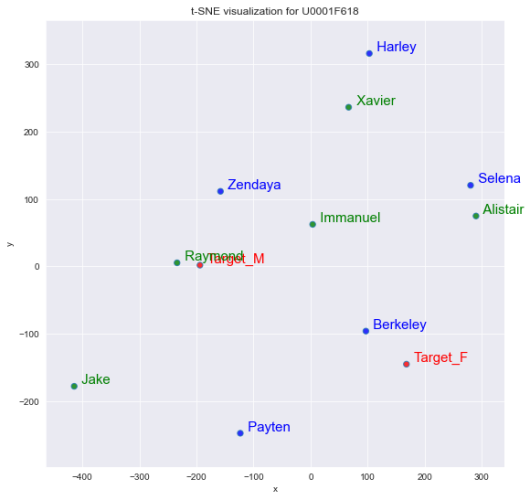

Fig. 1: Nearest users to female and male hypothesis vectors.

To sum up, the attacker can extract ground truth data and select the attribute to attack. Then, he checks if splitting the dataset is necessary by using

Algorithm 1. Next, value-based random indexing from Subsection 3.2 is applied to generate word vectors that are meaningful for each attribute value. Finally, the attribute inference is achieved by following the above steps.

5 Case study: Facebook

In this section, we apply our methodology to implement attribute inference attacks on Facebook. We first describe the experimental setup containing our dataset, evaluation metrics and parameter settings related to the classifier and Algorithm 1. Next, we assess and compare our approach with word2vec, a state-of-the-art method for representing language semantics [20].

5.1 Dataset

As a case study, we concentrate on Facebook. More precisely, we consider photos published by their owners. Compared to other publications (such as post and status update), pictures on Facebook receive an extra comment from Facebook, namely alt-text. The generated alt-text has two advantages. First, it alleviates the image processing tasks. Second, it provides additional information for the attacker. We show the success of our attack on three Facebook sensitive attributes: age, gender and relationship status. The data collection was conducted from October 2019 to April 2020 by utilizing a python crawler for academic research purpose. For the ground truth, we focus on the profiles where these attributes are public, and we collect the required information from the HTML files of the corresponding images. For every picture, we extract data such as comments, alt-texts, publication time and attribute's value of the owner. We have collected *9280* users' profiles: *7611* users published their gender, *4604* users shared their relationship status, and *3813* users announced their age. We randomly selected Facebook users to avoid usage bias by region or country. Overall we have collected *399,076* pictures and their *686,859* messages. We have leveraged the available picture metadata (either alt-text, comments or both) in our attack process. In order to get a representative and useful dataset, we perform the following pre-processing steps:

- Purifying the *conjunction*, *redundant tag*, and *text* from the generated alt-text.
- Cleaning the extracted comments by removing stop words and re-formulating flooded characters, misspelled words, and abbreviations.

Generating a vector representation from all user's pictures might be inaccurate when, for example, they are published at different time periods. Therefore, in addition to the above pre-processing steps, we use pictures' publication time range to prevent these impacts. For instance, if Alice shared *20* images in her profile from *January* to *June*, we create two users (*Alice_jan-mar* and *Alice_apr-jun*) by assigning pictures published from *January* to *March* to *Alice_jan-mar* and pictures of *April* to *June* to *Alice_apr-jun*. Moreover, Facebook users can

share photos on each other profiles. Considering those pictures owned by someone else might hinder the inference attacks as the publisher owner and the user might have different attribute values. In this study, we only examine images published by the user and filter out photos owned by others.

5.2 Experimental setup

Our attack relies on XGBoost classifier. We utilize different metrics to evaluate our approach. First, we use AUC (area under the ROC curve) as it is not sensitive to the label distribution [15]. Second, we use *macro* and *micro* average to evaluate our inference attacks. A macro-average computes the metric independently for each class and then takes the average (hence treating all classes equally), whereas micro-average aggregates the contributions of all classes to compute the average metric.

Parameter settings. We tune three different sets of parameters related to: classifier, RI and Algorithm 1.

For the classifier, we tune (i) the *learning_rate* to adjust weights on each step and make the model robust, (ii) *max_depth* and (ii) *min_child_weight* to control over-fitting. The *objective* specifies the learning task (e.g., binary classification) and the corresponding learning objective, *n_estimators* represents the number of trees to fit, and *subsample* indicates the fraction of observations to be randomly sampled for each tree. We set their default values and evaluate how different values affect our performance. Except for the objective depending on the number of classes, we create an array of different values for each parameter and use a python notebook tool called *GridSearchCV* to automatically find the best value of that array. For example, we assigned to *learning_rate* the array [0.01, 0.03, 0.05, 0.07], and use *GridSearchCV* to find the value giving the best result.

As for RI, [7] proposes a grid of sample values. We set *dimension= 500* and select two non-zero elements of the index vector to {-1, +1}, which maximize the result of our inference attacks.

Our dividing algorithm (see Algorithm 1) comprises parameters $UTop$ and $BTop$ to select the most frequent overlapped and non-overlapped *Unigram* and *Bigram*, respectively. The best parameter values are $UTop$=90 and $BTop$=110. They have been learned from our dataset by a grid search with $UTop, BTop \in \{10, 20, \ldots, 200\}$.

5.3 Inference results

For the age inference attack, we consider the following classes: 20 to 25, 25 to 30, 30 to 35, 35 to 40, 40 to 45, and 45 to 50. We chose these age groups to have a compromise between the accuracy of age prediction and the balancing of datasets. The age categories in our dataset reflect, in general, the most active ones on Facebook. We do not consider ages under 20 or over 50 as it is time consuming to collect enough data and keep all age categories balanced. As for relationship status, we collect datasets for *single*, *married* and *engaged* users. Consequently,

we consider three classes. Finally, we reduce gender inference attacks to a binary classification problem with *female* and *male* classes[3]. For age and relationship status, we set the XGBoost classifier *objective* to $multi : softprob$ that gives the probability of each class, while for gender we set it to *logistic*. We have used train-test splitting, as it performs faster, and split the entire dataset into the train, validation and test datasets. We have leveraged these datasets for training, parameter pruning and testing our XGBoost classifier, respectively. We have trained word2vec on the same dataset and compared its performance with our approach. Figure 2 shows the AUC result of word2vec and our approach over three different attributes.

	C1				***C2***			
	AUC	*Precision*	*Recall*	*Fscore*	*AUC*	*Precision*	*Recall*	*Fscore*
word2vec / class Female	**82**	**74.3**	**69.9**	**72.2**	**79**	**67.4**	**67.9**	**67.7**
RI-split /class Female	61	53.8	49.1	51.3	60	53.4	49.5	51.4
word2vec /class Male	**82**	**74.9**	**78.8**	**76.8**	**79**	**75.2**	**74.5**	**74.8**
RI-split /class Male	61	62.6	67.1	64.6	60	61.6	65.3	63.4

Table 2: Comparison with word2vec when splitting conditions are not satisfied.

Figure 2 (a) shows the result of the age inference attack by using word2vec where the age classes are inferred with AUC from 70% to 90%. In contrast, Figure 2 (b) represents the result of the same attribute inference attack by using our approach, which gets a tremendous boost with a substantial gain in performance. For example, our approach infers the class *35 - 40* with 99% AUC, where it was 77%. In addition to AUC, the *micro* and *macro* average increased to 98% and 95%, respectively. Figure 2 (c and d) display word2vec and our approach performance to relationship status inference attack. Our approach can accurately infer the relationship status attribute of the target user in comparison to word2vec. The class *Engaged* obtains 96% AUC in our approach, where word2vec infers this class inadequately (slightly better than random). Lastly, Figure 2 (e and f) depict the gender inference attack. Similarly, our approach outstands word2vec model. The attacker can infer the user's gender by using our approach with 98% AUC, where it drops to 70% in word2vec.

As mentioned in Subsection 3.1, we split the dataset only if two conditions are satisfied. To justify this, Table 2 represents some results of RI with splitting when the conditions are not satisfied. From a crawled dataset labeled by gender, we have synthetized two datasets $C1$ and $C2$. In $C1$, the first condition is satisfied and the second one is not satisfied. In $C2$, the first condition does not hold (and we ignore the status of second condition). We note that in both cases it is better to use word2vec than RI with splitting. Moreover, word2vec generates only one vector for each word which is space economical compared to RI with splitting. To sum up, this result confirms that by relying on Algorithm 1, splitting is

[3] We do not ponder other genders and relationship status by lack of training samples.

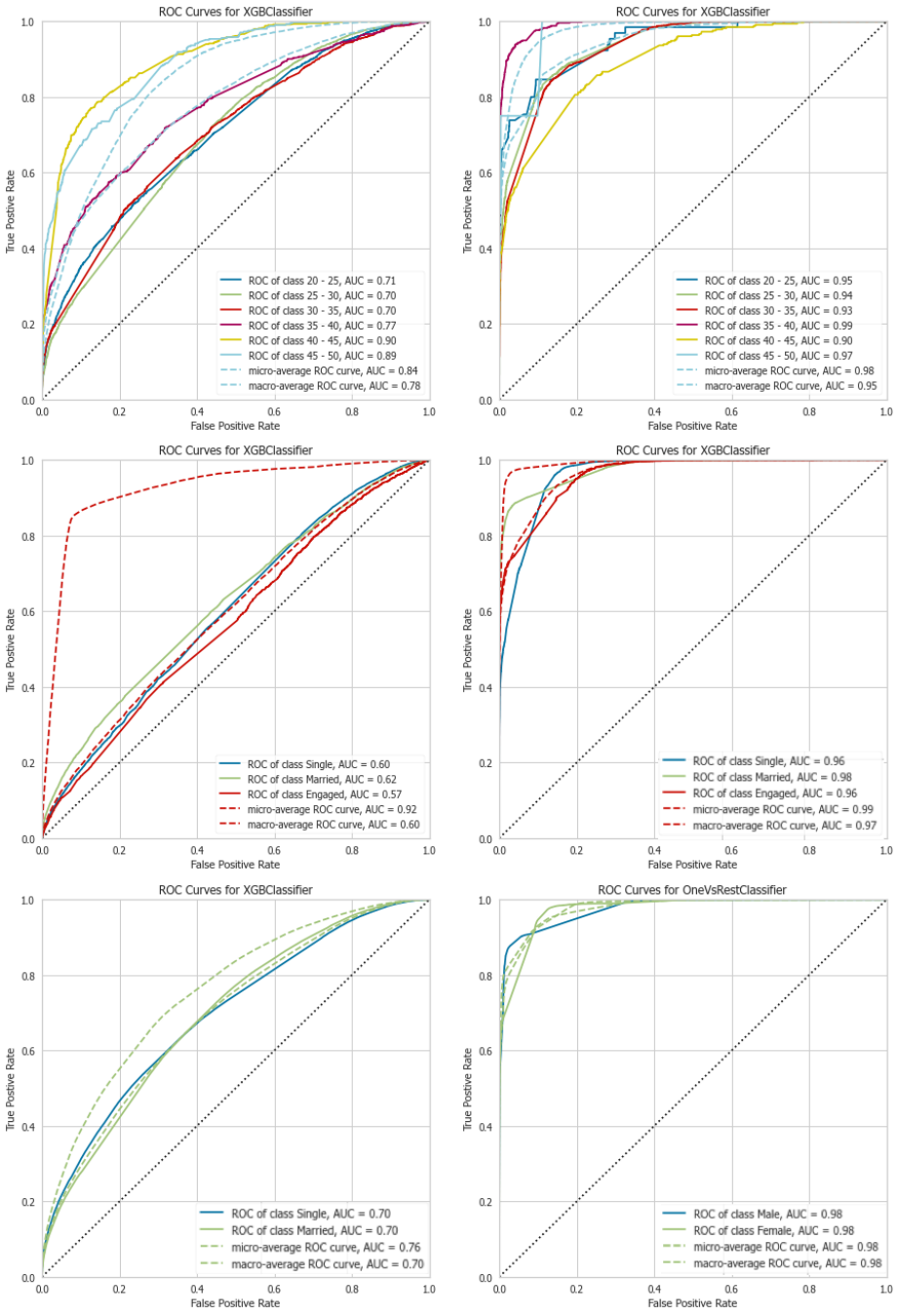

Fig. 2: Inference attacks performance measured by AUC for word2vec (left column) and our approach (right column): (line 1) age, (line 2) relationship status (line 3) gender

applied in our approach only when it is beneficial. When Conditions 1 and 2 (see Subsection 3.1) are satisfied, RI with splitting captures the commenters' words/emojis usage preferences adequately and this boosts the accuracy result. Otherwise, word2vec is more performant.

6 Conclusion

In this paper, we have presented a new perspective on attribute inference attacks based on reactions to target user publications. We have shown that if a term appears in diverse contexts, it can be represented by different vectors. To capture these diverse contexts, we have divided the dataset based on attribute values. We have also defined some conditions to prevent useless splittings. We have relied on the Random Indexing method to compute the term vectors in each attribute value, as the generated vectors need to be comparable. Based on the intensive analysis of *399,076* pictures and their *686,859* comments on Facebook, we have demonstrated that picture metadata conveys sensitive information and some private attributes such as gender, age category and relationship status are leaked by the variations in commenter's words/emojis usage preferences and picture owner sharing style. Our attacks are suitable for online execution as they do not require exploring user behavioral data and vicinity networks. They generalize easily to other social media platforms such as Twitter.

Even though some Facebook users limit themselves to sharing pictures with friends only, it is often not hard for an attacker to be added to a large list of friends and have access to the picture metadata.

As future work, we plan to extend our tool by explainable machine learning techniques [24,18] to offer users several means to reduce attribute inference risks (e.g., deleting or reducing the influence of leaking terms in comments). We also intend to enlarge the user age boundary and consider users over 50. Furthermore, using recent state-of-the-art tools such as BERT or ELMo for word embeddings would a promising direction to improve our work.

Ethical Statement. Our experiments have been performed on publicly available OSN data collected from Facebook. Although this data is public, it may lead to infer private information and we are therefore committed to keeping it in secure storage and only for the time necessary to carry out this work.

References

1. Abdelberi, C., Ács, G., Kâafar, M.A.: You are what you like! information leakage through users' interests. In: 19th Annual Network and Distributed System Security Symposium, NDSS. The Internet Society, San Diego, California, USA (2012)
2. Basile, P., Caputo, A., Semeraro, G.: Temporal random indexing: A system for analysing word meaning over time. Italian Journal of Computational Linguistics **1**(1), 55–68 (2015)

3. Choudhury, M.D., Sharma, S.S., Logar, T., Eekhout, W., Nielsen, R.C.: Gender and cross-cultural differences in social media disclosures of mental illness. In: Proceedings of the Conference on Computer Supported Cooperative Work and Social Computing, CSCW. pp. 353–369. ACM, Portland, OR, USA (2017)
4. Culotta, A., Kumar, N.R., Cutler, J.: Predicting the demographics of twitter users from website traffic data. In: Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, January 25-30, 2015, Austin, Texas, USA. pp. 72–78 (2015)
5. Eidizadehakhcheloo, S., Pijani, B.A., Imine, A., Rusinowitch, M.: Your age revealed by facebook picture metadata. In: ADBIS, TPDL and EDA 2020 Common Workshops and Doctoral Consortium Lyon, France, August 25-27, 2020, Proceedings. Communications in Computer and Information Science, vol. 1260, pp. 259–270. Springer (2020)
6. Farahbakhsh, R., Han, X., Cuevas, Á., Crespi, N.: Analysis of Publicly Disclosed Information in Facebook Profiles. CoRR **abs/1705.00515** (2017)
7. Fernández, A.M., Esuli, A., Sebastiani, F.: Lightweight random indexing for polylingual text classification. Journal of Artificial Intelligence Research **57**, 151–185 (2016)
8. Giulianelli, M., Tredici, M.D., Fernández, R.: Analysing lexical semantic change with contextualised word representations. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020. pp. 3960–3973. Association for Computational Linguistics (2020)
9. Gong, N.Z., Liu, B.: You are who you know and how you behave: Attribute inference attacks via users' social friends and behaviors. In: 25th Security Symposium. pp. 979–995. USENIX, Austin, TX, USA (2016)
10. Gong, N.Z., Talwalkar, A., Mackey, L.W., Huang, L., Shin, E.C.R., Stefanov, E., Shi, E., Song, D.: Joint link prediction and attribute inference using a social-attribute network. ACM Trans. Intell. Syst. Technol. **5**(2), 27:1–27:20 (2014)
11. Hecht-Nielsen, R., et al.: Context vectors: general purpose approximate meaning representations self-organized from raw data. Computational intelligence: Imitating life **3**(11), 43–56 (1994)
12. Jurgens, D., Stevens, K.: Event detection in blogs using temporal random indexing. In: Proceedings of the Workshop on Events in Emerging Text Types. pp. 9–16 (2009)
13. Kutuzov, A., Øvrelid, L., Szymanski, T., Velldal, E.: Diachronic word embeddings and semantic shifts: a survey. In: Proceedings of the 27th International Conference on Computational Linguistics, COLING 2018, Santa Fe, New Mexico, USA, August 20-26, 2018. pp. 1384–1397 (2018)
14. Levy, O., Goldberg, Y.: Linguistic regularities in sparse and explicit word representations. In: Proceedings of the Eighteenth Conference on Computational Natural Language Learning, CoNLL 2014, Baltimore, Maryland, USA, June 26-27, 2014. pp. 171–180. ACL (2014)
15. Lichtenwalter, R., Lussier, J.T., Chawla, N.V.: New perspectives and methods in link prediction. In: Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Washington, DC, USA, July 25-28, 2010. pp. 243–252 (2010)
16. Lindenstrauss, W.J.J.: Extensions of lipschitz maps into a hilbert space. Contemp. Math **26**, 189–206 (1984)
17. Ludu, P.S.: Inferring gender of a twitter user using celebrities it follows. CoRR **abs/1405.6667** (2014)

18. Lundberg, S.M., Lee, S.: A unified approach to interpreting model predictions. In: Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA. pp. 4765–4774 (2017)
19. Mauw, S., Ramírez-Cruz, Y., Trujillo-Rasua, R.: Robust active attacks on social graphs. Data Min. Knowl. Discov. **33**(5), 1357–1392 (2019)
20. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781 (2013)
21. Nguyen, D., Gravel, R., Trieschnigg, D., Meder, T.: How old do you think I am? A study of language and age in twitter. In: Proceedings of the Seventh International Conference on Weblogs and Social Media, ICWSM 2013, Cambridge, Massachusetts, USA, July 8-11, 2013. The AAAI Press (2013)
22. Pennington, J., Socher, R., Manning, C.D.: Glove: Global Vectors for Word Representation. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL. pp. 1532–1543 (2014)
23. Pijani, B.A., Imine, A., Rusinowitch, M.: You are what emojis say about your pictures: language-independent gender inference attack on facebook. In: SAC '20: The 35th ACM/SIGAPP Symposium on Applied Computing, online event, Brno, Czech Republic, March 30 - April 3, 2020. pp. 1826–1834. ACM (2020)
24. Ribeiro, M.T., Singh, S., Guestrin, C.: "why should I trust you?": Explaining the predictions of any classifier. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, August 13-17, 2016. pp. 1135–1144 (2016)
25. Ryu, E., Rong, Y., Li, J., Machanavajjhala, A.: curso: protect yourself from curse of attribute inference: a social network privacy-analyzer. In: Proceedings of the 3rd ACM SIGMOD Workshop on Databases and Social Networks, DBSocial 2013, New York, NY, USA, June, 23, 2013. pp. 13–18. ACM (2013)
26. Sahlgren, M.: An Introduction to Random Indexing. In: Methods and applications of semantic indexing workshop at the 7th international conference on terminology and knowledge engineering (2005)
27. Sap, M., Park, G.J., Eichstaedt, J.C., Kern, M.L., Stillwell, D., Kosinski, M., Ungar, L.H., Schwartz, H.A.: Developing age and gender predictive lexica over social media. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP. pp. 1146–1151. ACL, Doha, Qatar (2014)
28. Sherwin, G., Bhandari, E.: Facebook settles civil rights cases by making sweeping changes to its online ad platform. `https://www.aclu.org/blog/womens-rights/womens-rights-workplace/facebook-settles-civil-rights-cases-making-sweeping` (2019)
29. Wang, P., Guo, J., Lan, Y., Xu, J., Cheng, X.: Your cart tells you: Inferring demographic attributes from purchase data. In: Proceedings of the Ninth ACM International Conference on Web Search and Data Mining, San Francisco, CA, USA, February 22-25, 2016. pp. 173–182 (2016)
30. Weinsberg, U., Bhagat, S., Ioannidis, S., Taft, N.: Blurme: inferring and obfuscating user gender based on ratings. In: Sixth ACM Conference on Recommender Systems, RecSys '12, Dublin, Ireland, September 9-13, 2012. pp. 195–202 (2012)