Attribute inference attacks on social media publications

Bizhan Alipour Pijani

UNIVERSITÉ DE LORRAINE

École doctorale IAEM Lorraine

Attaques par inférence d'attributs sur les publications des réseaux sociaux

Attribute Inference Attacks on Social Medias Publications

THÈSE

présentée et soutenue publiquement le March 10, 2022

pour l'obtention du

Doctorat de l'Université de Lorraine

(mention informatique)

par

Bizhan Alipour Pijani

Composition du jury

Rapporteurs :	Esma Aimeur	Professeure - Université de Montréal
	Maritta Heisel	Professeure - Université de Duisburg-Essen
Examinateurs :	Fadila Bentayeb	Professeure - Université Lyon 2
	Marine Minier	Présidente du juy - Professeure - Université de Lorraine
	Abdessamad Imine	Maitre de Conférence HDR - Université de Lorraine - Co-advisor
	Michaël Rusinowitch	Directeur de recherche - INRIA - Co-advisor

Laboratoire Lorrain de Recherche en Informatique et ses Applications — UMR 7503

Mis en page avec la classe thesul.

Remerciements

My thanks and gratitude go to Allah, Most Gracious, Most Merciful, from whom I seek forgiveness, guidance, and acceptance.

A special thanks to my beloved family and my lovely wife Sanaz, who supported me in this journey. Without their precious support, it would not be possible to succeed.

I would also like to thank my supervisors, Abdessamad Imine and Michael Rusinowitch, for their support, guidance, motivation, and suggestions that helped me in all the research and writing of this thesis.

My warm thanks go to my Friends, close or far away, and work colleagues who also contributed to the success of this work.

Besides my advisors, I would like to thank the rest of my thesis jury: Prof. Esma Aimeur, Prof. Maritta Heisel, Prof. Fadila Bentayeb, and Prof. Marine Minier for their insightful comments and encouragement.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Contents

1.1 Research Context

Online Social Networks (OSN) are a melting-pot of personal information (such as age, gender, and political beliefs), often unconsciously published by users to be visible and attract new connections. From a wide variety of users' data and behaviors, OSN allow for (i) suggesting contacts based on traits similarity, (ii) delivering content based on user's interests, and (iii) launching targeted advertising by third parties. However, the popularity and growth of OSN have rendered their platforms vulnerable to malicious activities and increased user privacy concerns. For instance, [Boshmaf et al., 2013] showed that Facebook users accept friend requests from unknown people but have several mutual friends. Even with user control through the privacy settings available in OSN, users remain vulnerable to attribute inference attacks where an attacker seeks to illegitimately obtain their personal attributes (such as the gender attribute) from publicly available information. An attribute inference attack is a powerful mean to breach user privacy for malicious purposes or targeted advertisements [Belinic, 2009].

Disclosure of personal information can have serious outcomes ranging from spam [Gao et al., 2012] to online bullying and sexual harassment [Chowdhury et al., 2019]. A cloning attack, one of the insidious attacks on social media, is another outcome of this disclosure. In this attack, the target user's personal information is inferred to build a fake profile [Kontaxis et al., 2011]. Besides, numerous fake or hijacked identities have been leveraged for mounting various attacks, such as advertising and collecting personal private information [Wu and Chen, 2017]. Moreover, companies can exploit the harvested personal information

and jeopardize some users' privacy to tailor online ads according to their profile [Tucker, 2014].

Some demographic attributes, such as age, gender, and zip code, may dramatically affect ads delivery. Gender targeting is only one way for OSN to help advertisers focus on particular users and exclude others. For instance, females get fewer ads related to high-paying jobs than men [Datta et al., 2015, Tobin and Merrill, 2018]. The American Civil Liberties Union (ACLU)[1] accused Facebook of allowing discriminatory job ads on its social network that prevents women from receiving job offers for certain well-paid positions. On another side, a study has shown how employers interpret different types of self-presentation data on OSN and how the data pulls employers into hiring decisions [Brouer et al., 2017].

To increase awareness of social media users about threats to their privacy, we are motivated to show the feasibility of inference attacks from minimal data, even when the user hides all their profile information and comments. This approach allows us to inform users of privacy threats arising from seemingly innocent data and propose effective countermeasures to mitigate those threats.

Approaching the privacy risks surrounding OSN users requires examining the types of data available to an attacker and the methods of deploying attribute inference attacks. To that end, in the rest of this chapter, we categorize data that an attacker can explore to discover attribute values, present privacy attacks, and overview some protection schemes. Finally, we conclude the chapter by summarizing our thesis contributions.

1.2 Social Media Data

1.2.1 Data generated by target user

In this thesis, given a user that an attacker targets, we call *target-user generated data* the data owned and published by this user.

Attributes. Attributes are elements of information that the user chooses to publish in the profile. These attributes help social media friend's suggestion algorithms (friends recommendation system) look for the user's best match. The trait's value is specified by picking from a predefined list (select female, male, etc ...) or typing by the user (adding a bio in the profile).

Sharing. Users can share many types of information to express their feelings and thoughts, such as pictures, videos, links, and posts containing all mentioned information. A social media analysis [Shutterstock, 2015] explains the psychology behind sharing as follows: "*perhaps one of the strongest forces driving our motivation to share is based on our sense of identity, more specifically, the desired version of ourselves that we want to project onto the world.*"

Behaviours. Behaviors are characterized by various user activities, for instance, following celebrities' or athletes' profiles, joining sports clubs or university groups, and pages to receive updates from them.

[1] https://www.aclu.org/blog/womens-rights/womens-rights-workplace/facebook-settles-civil-rights-cases-making-swe

1.2.2 Other data

In this thesis, given a user that an attacker targets, we call *non target-user generated data* the data generated and published by other users or by the OSN itself. These data can be triggered or not by the target user publications.

Data that are triggered by the target user publications. The target user can post various publications, including posts, videos, links, or photos. Other users can interact with these publications by animated and/or textual reactions. There are seven animated reactions: *Like*, *Love*, *Haha*, *Angry*, *Sad*, *Wow*, and *Thankful*. As a typical textual reaction, people spontaneously comment on these publications. This option allows them to engage and impress their personal opinion, creating a sense of connectedness between the target user and the commenters, especially when they are friends. Target user publications trigger this data sharing. Unlike posts, videos, and links posted by the target user, pictures receive an automatic description from social media such as Facebook and Instagram. Below we review the information added this way to images and that we call *metadata*.

(i) Automatic alt-text. Automatic alt-text (AAT) is a system that applies computer vision technology to identify faces, objects, and themes from pictures displayed by OSN users to generate descriptive alt-text that a screen reader can process. They have proposed this system to help blind people to feel more connected and involved in OSN. The alt-text always starts by *May be* and is followed by a list of recognized objects. For example, Facebook designs a list of 97 objects and themes that provides different sets of information about the image, including people *(e.g., people count, smiling, child, baby)*, objects *(e.g., car, building, tree, cloud, food)*, settings *(e.g., inside restaurant, outdoor, nature)*, and themes *(e.g., close-up, selfie, drawing)* [Wu et al., 2017]. AAT provides free, additional information about photos and makes blind people feel more included and engaged in photos.

(ii) Comments. Other users (target friends, friends of friends, or ordinary users) show their feelings by posting comments underneath the picture. These comments contain potentially sensitive information and are often available to the attacker. Commenters can express their feeling by simply using words (English and non-English comments), and/or they can use different types of communication (emoji/emoticons) that we discuss below.

(i) Emoji. Users in social media use emojis to express their feelings directly. Since *2010*, emojis emerged into communication where Oxford Dictionaries[2] announced 😂, commonly known as *face with tears of joy*, as the word of the year.

(ii) Emoticons. An emoticon[3] represents human facial expressions using only keyboard characters such as letters, numbers, and punctuation marks. They express emotions differently through facial gestures inside text-based communication.

Data that are not triggered by the target user publications. Target user friends or friends of friends can publish their attributes, interests, groups they joined in their profiles. They can also react to other users' publications, excluding the target user. Target user publications do not trigger this data sharing.

[2] `https://languages.oup.com/press/news/2019/7/5/WOTY`
[3] `https://en.wikipedia.org/wiki/Emoticon`

1.3 Inference Attacks and Positioning of Thesis

An attacker can perform attribute inference attacks using target user generated data and non target-user generated data. The inference attacks using only *target-user generated data* are known as behavior-based [Chaabane et al., 2012] inference attacks. The attacks follow the intuition that *you are how you behave*. In these attribute inference attacks, the attacker monitors user behavior such as pages liked and groups joined by the target user to infer private attributes. On the other hand, inference attacks exploit both data generated by the target user and data produced by other means. For instance, homophily attacks rely on friend-list published by the target and attributes announced by the target friends (see also [Gong and Liu, 2016]). These attacks follow the intuition that *you are who you know*. They work in two steps: the attacker first collects the friend list of the target user inside the target user profile, and then from the target user and friend's available data infers target hidden attributes.

Existing inference techniques are behavior-based or friend-based attacks. However, in real cases, the amount of available information to an attacker is small. Typically, essential attributes (e.g., age, gender, relationship status, friend list) that are beneficial for providing personalized services are often not accessible. Users have realized the vulnerability of standard attribute inference attacks and concealed their generated information. The authors of [Farahbakhsh et al., 2017] investigate Facebook users' privacy awareness and show that age has the lowest exposure rate. Less than 3% of the users (among 495k users) reveal their age, which shows the sensitivity of this attribute. They also show that 37.3% conceal their friend list and half of their members hide their gender. Moreover, the authors of [Dey et al., 2012] study age privacy on Facebook and report that most users consider age as a private attribute. Facebook users prefer to hide their traits for three reasons. First, they want to strengthen protections against discrimination. Second, they rely on anonymity to reduce the social risks of discussing unpopular and taboo topics [Bargh et al., 2002, Yurchisin et al., 2005]. Third, they want to prevent any form of sexual harassment and stalking. Therefore, collecting user-generated data is a difficult task. We note that most attribute inference attacks get inoperative in the case of no data provided by target users in their profiles. For instance, many attribute inference systems are based on network homophily [Ryu et al., 2013] and do not apply without the availability of friend lists. Moreover, since standard inference techniques proceed by analyzing data published by the user, one may believe that being cautious in the writing style or hiding attributes (e.g., paged likes and friend lists) from the public prevents an attacker from predicting sensitive traits. Even with more user awareness, privacy risks can arise from other sources. Many users do not realize that even though they are cautious about their writing style, other users' textual reactions to their publications reveal their sensitive information.

This thesis presents a new family of inference attacks by leveraging the metadata (as defined above) triggered by the target user publications. The metadata includes comments from other users and tags generated by the OSN. Therefore *our attacks are indirect attacks that target OSN users even when they are cautious about their privacy, and hide necessary data such as profile attributes, friend lists, liked pages, groups, writing style (e.g., their comments)*. It is also essential to consider the feasibility of attacks in online mode. Our attacks are suitable for online execution as they do not require exploring user behavioral data and vicinity networks.

Our attacks can be implemented on different OSN (e.g., Instagram) and publications (e.g., posts, videos). However, we focused on Facebook and pictures as a case study for two reasons. First, Facebook is the largest social network these days. Statistics showed that Facebook has

roughly 2.89 billion monthly active users, where they spend approximately 640 million minutes per month on the platform [Zhou and Chen, 2020]. Second, although Facebook users tend to hide their attributes, pictures are still available to the public. A social media sharing analysis conducted by *The New York Times* revealed that *68%* of their respondents share images to give people a better sense of *who they are* and *what they care about* [Shutterstock, 2015]. Users in social media share pictures to receive feedback for their activities. Publishing pictures enable their owners to increase connectivity and activity on social networks. According to [Cooper, 2013], photos on Facebook received 53% more likes and 104% more comments compared to text-based posts. Moreover, a picture receives extra information, namely alt-text. The generated alt-text has advantages for the attacker as it alleviates the image processing tasks and provides additional free information. We show how an attacker can analyze and process these pieces of information to infer picture owners' attributes.

1.4 Privacy Protection

Facebook allows users to control and customize publicly available personal information. In particular, users have the option of hiding profile attributes (e.g., age, gender, relationship status, sexual preference, and political affiliation) and behavioral records (e.g., group, page) from the public and let the profile be visible only to the audience of interest. For example, Figure 1.1 represents some instances of Facebook's privacy settings. As presented, users can limit timeline and tagging visibility, activity, and connection. For example, they can control who can send them friend requests.

As for controlling reactions, Facebook proposes either hiding or deleting options, allowing users to either hide or delete hate speech or risky comments. However, deleting a comment can increase the tension between the commenter and the picture owner as the deleted comment is invisible to the comment author. Hiding is more beneficial as the hidden comment remains visible to the comment author and the author's friends.

Once the risk is recognized, the simplest solution is to hide the publication, which contradicts social media objectives. In social media such as Facebook, most users post pictures, for instance, to expose their personalities, lifestyles, and preferences. Identifying the risky comments and manually hiding them is difficult, as it might be troublesome to recognize risky ones and locate them in case of receiving several comments. To remedy the problem, Facebook proposes an automatic hiding mechanism. Users have to provide a list of words/emojis that they do not want to receive, and Facebook hides a comment entirely if it contains certain words/emojis. However, the proposed privacy setting has main drawbacks. It hides the comment without examining whether it threatens the user's privacy. In addition, the user has to update the list manually, which might compromise user visibility and be inconvenient.

Motivated by the impacts of personal information disclosure and the hindrance of Facebook's recommended privacy, we propose a protection mechanism to mitigate the threats and suggest countermeasures to enhance users' privacy. Our protection mechanism advises hiding a minimal number of comments automatically to make a trade-off between user privacy and visibility on Facebook.

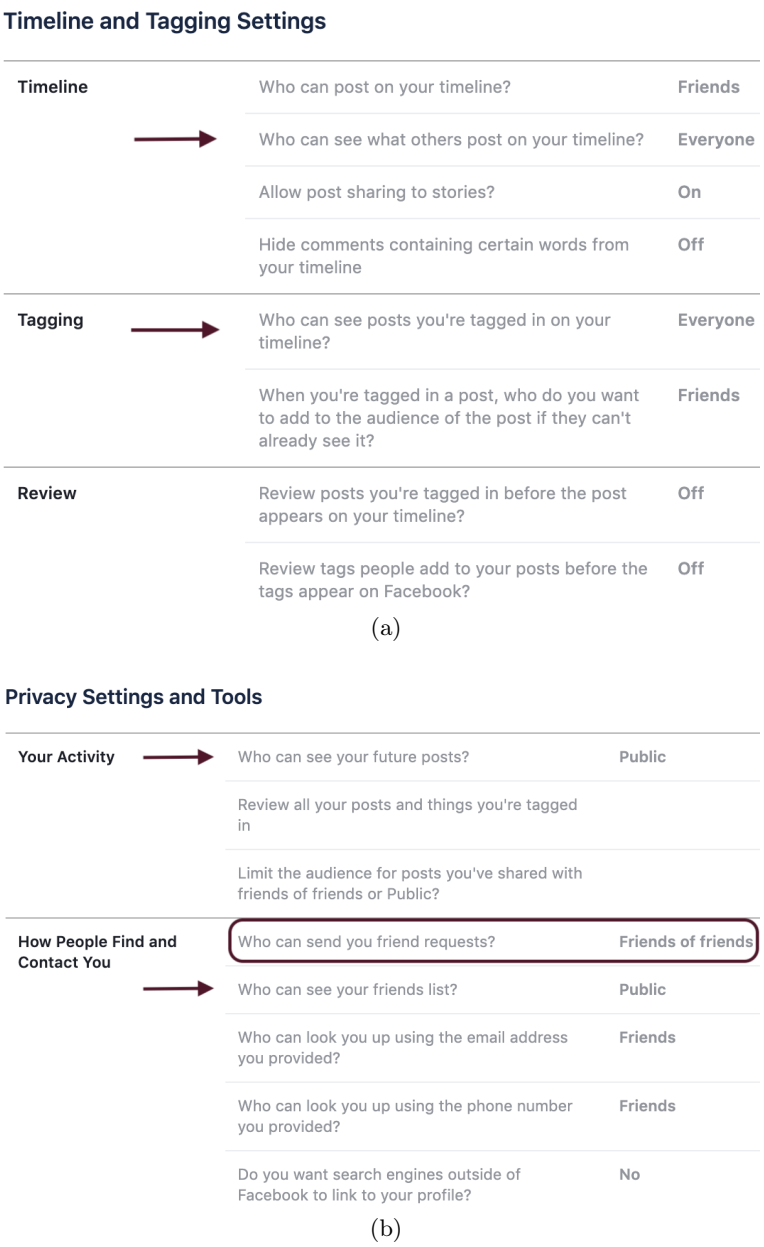


Figure 1.1: Privacy setting provided by Facebook.

1.5 Contributions of the Thesis

In the following, we sketch the essence of our contributions:

1. Attribute inference attacks. We provide a novel online approach for attribute inference attacks by considering picture metadata rather than considering the friend-based and behavioral-based data, which might be costly to collect or unavailable. We leverage emojis/emoticons as a universal language (independent of target user spoken language) to infer the picture owner attributes. We study the effect of picture owner attributes (gender, age, relationship status) on other users' textual reaction (e.g., words, emojis usage) when reacting to different

gender/age/relationship group pictures with the same style and alt-text tags. The uncovered correlation then helps us devise inference attacks with a high level of accuracy.

2. Design of proper vector representations of data for accurate attacks. To perform our attacks, we have adapted several supervised machine learning algorithms. The success of our attribute inference attacks relies on finding an accurate correlation of picture owner attributes (gender, age, relationship status) with alt-text and words/emojis used by commenters when reacting to these pictures. Vector representation models [Mikolov et al., 2013a] can capture these correlations. However, we need to tune the vector representation of data for two reasons. First, we train our model in a fixed-size dataset in offline mode and perform our attack in an online mode. The target user's pictures online can receive comments containing words that have not appeared in the training phase (offline), known as Out Of Vocabulary (OOV) words. Therefore we propose to apply a retrofitting technique inspired by NLP to compute a proper representation for these OOV words. Second, by computing a vector representation specific to each attribute value, ML algorithms can select the best one to boost the model's accuracy.

3. A privacy-enhancing system. We propose practical solutions to combat these inference attacks without jeopardizing user visibility on social media. We design a mechanism that balances user visibility and privacy. It pinpoints received comments or posted pictures leading to attribute inference attacks. The mechanism leveraged our trained classifier to select a minimal set of comments that is sufficient to hide to preserve user privacy.

The contributions have been published in:

- Pijani, B.A., Imine, A. and Rusinowitch, M., 2019, August. Gender inference for Facebook picture owners. In TRUSTBUS, International Conference on Trust and Privacy in Digital Business (pp. 145-160). Springer, Cham.
- Pijani, B.A., Imine, A. and Rusinowitch, M., 2020, March. You are what emojis say about your pictures: language-independent gender inference attack on facebook. In Proceedings of the 35th Annual ACM SAC, Symposium on Applied Computing (pp. 1826-1834).
- Pijani, B.A., Imine, A. and Rusinowitch, M., 2020. Inferring attributes with picture metadata embeddings. ACM SIGAPP Applied Computing Review, 20(2), pp.36-45.
- Pijani, B.A., Imine, A. and Rusinowitch, M., 2020, September. Online Attacks on Picture Owner Privacy. In DEXA International Conference on Database and Expert Systems Applications (pp. 33-47). Springer, Cham.
- Eidizadehakhcheloo, S., Pijani, B.A., Imine, A. and Rusinowitch, M., 2020, August. Your age revealed by Facebook picture metadata. In ADBIS, TPDL and EDA 2020 Common Workshops and Doctoral Consortium (pp. 259-270). Springer, Cham.
- Eidizadehakhcheloo, S., Pijani, B.A., Imine, A. and Rusinowitch, M., 2021, July. Divide-and-Learn: A Random Indexing Approach to Attribute Inference Attacks in Online Social Networks. In IFIP DBSEC Annual Conference on Data and Applications Security and Privacy (pp. 338-354). Springer, Cham.

To the best of our knowledge, we present the first study of attribute inference attacks on Facebook that rely on non-target-user generated data: alt-text generated by Facebook, and

target user friends, friends of friends, or ordinary users' words, emojis preferences. To ease reading, we introduce some specific definitions used throughout the thesis. A *picture owner*, shortened as an *owner*, is the one who published pictures on Facebook. A *commenter* is another Facebook user that can be owner friends, friends of friends, or ordinary users. We call *commenter reaction* a textual reaction to the owner's images.

1.6 Outline

The organization of the thesis is as follows:

In Chapter 2, we discuss related works. We represent the preliminaries for the thesis in Chapter 3. In Chapter 4, we perform a gender inference attack by leveraging the universal language of emojis and, more specifically, exploiting the correlation between picture owners' gender and commenters' emojis preferences. In Chapter 5, in order to be able to perform online attacks, we introduce the needed techniques to handle Out Of Vocabulary words. In Chapter 6, we develop different data vector representations that are well adapted to our objective of performing attribute inference attacks on various attributes. In Chapter 7, we detail our proposed protection mechanism and show the effectiveness of the mechanism in safeguarding the target users against gender inference attacks. Finally, in Chapter 8, we conclude the thesis by representing thesis achievements, limitations, and perspectives.

Chapter 2

Related Works

Contents

This chapter reviews several recent studies that demonstrated attribute inference attacks on social media and privacy protection.

2.1 Attribute Inference Attacks

We have designed inference attacks on three different attribute values in our work. In this section, we begin by covering the literature addressing attacks on these attributes. Then we will review some works on emoji analysis as we have also leveraged emojis/emoticons in our attacks.

Inferring gender. Profiling has gained significant attention in the past decade. Deriving user gender, for instance, is essential for recommendation systems. In the past, researchers have investigated social media platforms in order to distinguish males and females from content sharing [Choudhury et al., 2017] and behavior [Ludu, 2014], conversations [Garera and Yarowsky, 2009], blogs [Sarawgi et al., 2011], user search queries [Weber and Castillo, 2010], and tweets [Sarawgi et al., 2011, Al Zamal et al., 2012]. Gender inference from the target user name can be performed across major social networks [Karimi et al., 2016]. However, the performance of this type of attack is biased towards countries of origin [Santamaría and Mihaljevic, 2018]. The authors in [Cheung and She, 2017] proposed user gender identification through user-shared images in Fotolog and Flickr, two image-oriented social networks. They performed image processing on each offline crawled image, which is not feasible with online attacks. In [Chaabane et al., 2012], the authors inferred users' private traits using the public attributes of other users sharing similar interests and an ontologized version of Wikipedia. The authors of [Gong and Liu, 2018] proposed a new privacy attack to infer attributes (e.g., locations, occupations, and interests) of online social network users by integrating social friends and behavioral records. In [Gong et al., 2014] the authors extended the Social-Attribute Network (SAN) framework with several leading supervised and unsupervised algorithms for the link and attribute inferences. In [Weinsberg et al., 2012] the authors showed that a recommender system could infer the gender of a user with high accuracy, based solely on the ratings provided by users and a relatively small number of users who share their demographics.

On the other hand, some works claim that gender prediction is possible from target user writing style [Flekova et al., 2016] and word usage [Sap et al., 2014]. In [Lopes Filho et al., 2016], the authors focused on gender classification using 60 textual meta-attributes to extract the gender linguistic utterance in tweets written in Portuguese. More precisely, they considered characters, syntax, words, structure, morphology of short length, multi-genre, and content-free texts posted on Twitter to classify the author's gender via three different machine-learning algorithms. The work in [Miller et al., 2012] consisted in identifying the gender of users on Twitter using perceptron and Naive Bayes from tweet texts. [Thomas et al., 2010] studied the inference of attributes such as gender, political views, and religious views by developing a classification system that uses publicly disclosed links between friends and the content of leaked conversations to perform inference attacks. [Otterbacher, 2010] proposed to infer the author's gender by analyzing the reviews of a particular item at the Internet Movie Database (IMDb) to capture the differences between man and woman writing styles. However, the above works have two drawbacks that degrade their model prediction accuracy: (i) interaction between friends of the opposite sex may affect users' word / emoji usage [Nguyen et al., 2014], and (ii) users can be careful in choosing words / emojis.

Inferring age. Online behavior is representative of many aspects of a user's demographics [Rao et al., 2010, Pennacchiotti and Popescu, 2011]. Many studies have applied language analysis on the text generated by the target user (users' messages, posts, and status updates) to estimate the user age using machine learning approaches [Nguyen et al., 2013, Rangel and Rosso, 2013]. It has been shown that easily accessible digital records of behavior (such as likes) allow one to predict users' age accurately [Kosinski et al., 2013]. Both content and stylistic features (such as part-of-speech and the number of slang words) have been found valuable for predicting the age of users [Nguyen et al., 2011]. The study [Han et al., 2019] proposed a general framework for private attribute disclosure estimation using several algorithms to infer user age through user publishing behavior. [Choi et al., 2017] attempted to formulate an accurate inference using only publicly available Facebook profiles to infer private attributes such as age. [Pesce et al., 2012] modeled ego-networks from users' friendship links and leveraged the tagging pictures to reveal the age attribute.

The effect of age on writing style and extracting the age of users from facial images have received increasing attention in recent years. Authors of [Pennebaker and Stone, 2003] found that when people get older, they tend to write more positive and fewer negative words, focus more on the future and less on the past and make fewer self-references. [Mei et al., 2018] proposed a framework of attribute inference attack based on users' public profile (image and attribute). They integrated and modified the existing state-of-the-art convolutional neural network models to predict the user's age attribute from images in social networks. [Levi and Hassner, 2015] proposed a simple convolutional net architecture to classify the age and gender of users from profile images in social networks.

Inferring relation status. Researchers analyzed all possible public information generated by users to infer a relationship status. For example, the authors of [Kótyuk and Buttyán, 2012] presented a marital/relationship status inference attack based on the concept of Multi-Layer Perceptron (MLP). They leveraged public attributes published by target users and their friends to perform the inference attack. Their method is capable of both classification and regression. [Minkus et al., 2015] showed the possibility to infer marital/relationship status and family connections by combining online and offline data sources, namely public Facebook profiles and

public records (voter registration records). [Tian et al., 2019] investigated attribute inference attacks on social media users using Graph Convolutional Neural Networks. They leveraged public user profiles and social links to infer users' marital status class labels. They divided the marital status into unmarried, married, divorced, and widow.

[Backstrom and Kleinberg, 2014] proposed a new network-based characterization for intimate relationships, those involving spouses or romantic partners. They investigated all connections among friends to recognize a user's romantic partner from the network structure. [Garcia, 2017] analyzed personal information available on the users' profiles to predict sexual orientation and relationship status, and romantic interests. [Volkova et al., 2015] examined the individual published tweets for emotions and opinion detection to infer latent traits, such as relationship status, using an approach based on machine learning (log-linear models) and natural language processing. They leveraged crowdsourcing to annotate user profiles. They asked workers on Amazon Mechanical Turk to glance through 5,000 Twitter profiles and tweets and make subjective judgments about a variety of their latent properties.

Emoji usage analysis. Several works have investigated emoji usage in recent years. Researchers have studied the individual intercept on messages containing emojis [Butterworth et al., 2019]. They have performed experiments on how people use emojis, an emerging universal language for stating emotions in different countries [Lu et al., 2016] and cultures [Barbieri et al., 2016a]. Emoji is a rich resource for sentiment analysis and emotion measurement. For example, [Ai et al., 2017] accomplished the first quantitative study to correlate emoji usage with its semantic. [Wijeratne et al., 2017b] presented a comprehensive study on measuring the semantic similarity of emoji using emoji embedding models. They extracted machine-readable emoji meanings from EmojiNet [Wijeratne et al., 2017a] and used pre-trained word embedding models learned over a Twitter dataset of 110 million tweets and a Google News text corpus of 100 billion words.

[Shiha and Ayvaz, 2017] studied the usage of emoji on Twitter and their effects in text mining and sentiment analysis. Additionally, [An et al., 2018] analyzed users' messages in Wechat [4], an instant massaging application in China, to learn about the diversity of emoji usage preferences in terms of frequency, type and sentiment. The diversity and global usage of emojis enable researchers to analyze emoji usage according to gender [Chen et al., 2018]. This study collected data through the *Kika Keyboard* and relied on the usage preference of the user himself. This work also suffers from opposite-gender friends interaction and user cautiousness. Emoji can be interpreted differently according to the platform, which might influence users' communication [Miller et al., 2017]. Besides, some researchers investigated the power of emoji in the cross-lingual sentiment classification task [Chen et al., 2019] and performed a large-scale empirical study on how developers used Emoji on GitHub [Lu et al., 2018].

To conclude, these inference attacks are costly as they assume that the entire or part of social network information is available to the attacker. The above approaches depend on the target user's writing style (words/emoji usage) or user-generated data (profile attributes). Predicting users' attribute values might be straightforward based on their words/emoji usage preferences. Additionally, the user's writing style may be affected by the interaction of opposite-gender friends [Nguyen et al., 2014], or the generated data may be fake [Random, 2018]. In contrast,

[4] `https://www.wechat.com/en`

we consider the non-user-generated data as potentially sensitive information that can be used to infer target user attributes. We study attribute inference attacks on Facebook taking into account the word / emoji / emoticon preferences of commenters while commenting on pictures belonging to the target user. These data are readily available as there is no need to explore the target user vicinity network.

Our work is different in two aspects:

- We deliberately ignore the target user words/emojis usage and rely on commenters' reactions and Facebook-generated alt-text to solve the above limitations.
- We exploit the idea that the picture content has a powerful impact on individuals' emotional reactions (animated and/or textual).

Our approach has three advantages over previous works:

- The target user's personality does not affect our model prediction accuracy.
- Even when the target user is careful enough to manipulate words/emojis/emoticons neutrally, the attack is still possible.
- Non-user-generated data reduce the complexity of inference attacks that can be launched online.

2.2 Privacy Protection

Data obfuscation is a well-known technique for protecting user privacy against inference attacks. It was studied in search queries, recommender systems, location-based services, and online social networks. The goal is to obscure sensitive attributes released by the data owner from potential attribute inference attacks from a malicious adversary. Game-theoretic methods are one type of defense against inference attacks where the idea is to preserve some part of visible data that the attacker classifier can not infer the private attribute. [Shokri et al., 2012] investigated a game-theoretic method to defend against location inference attacks where the defender obfuscates locations to protect users against the optimal inference attack. These methods have theoretical privacy guarantees, but they rely on optimization problems that are computationally intractable when applied to attribute inference attacks.

Several studies [Heatherly et al., 2013, Salamatian et al., 2015] proposed a trade-off between tractability and privacy guarantees. For instance, [Salamatian et al., 2015] studied probabilistic quantization mapping to approximate the game-theoretic optimization problem formulated by [du Pin Calmon and Fawaz, 2012]. The defense problem was tractable but not optimal against attribute inference attacks [Jia and Gong, 2018]. An efficient heuristic to compute correlations between public data and sensitive attribute values has been presented in [Chen et al., 2014]. [Weinsberg et al., 2012] introduced a countermeasure against attribute inference attacks in the context of recommender systems by sorting items according to their correlations with the target attribute value. However, [Jia and Gong, 2018] showed their method (i) adds enormous noises to the original data and therefore degrades the recommender utility, and (ii) the defender requires direct access to a user's private attribute value for computing the correlations between public data and private attribute values [Weinsberg et al., 2012, Heatherly et al., 2013, Chen et al., 2014].

Such a requirement introduces usability issues and privacy concerns. When the defender is compromised, the private attribute values of all users are compromised [Jia and Gong, 2018]. Specifically, a user needs to specify the attribute value to the defender, which makes it inconvenient for users.

Researchers leveraged local differential privacy [Wang et al., 2017], where the idea is to add noise to a user's public data to decrease the attacker classifier accuracy. Using differential privacy theory, [McSherry and Mironov, 2009] added noises to original user behaviors to keep the global distribution unchanged. [Nguyên et al., 2016] applied local differential privacy, which helped users hide their information even from first-party services. [Cai et al., 2016] managed to erase sensitive information from social networks, to lower inference accuracy of user privacy. However, [Jia and Gong, 2018] noted that Local Differential Privacy (LDP) achieves a suboptimal privacy utility tradeoff at defending against attribute inference attacks by adding much larger noise to the user public data.

[Zhao et al., 2019] developed a novel theoretical framework for attribute obfuscation, using a minimax optimization formulation to protect the given attribute. Their system has two parties: the prediction vendor and the data owner. The goal of the data owner is to provide as much information as possible to the prediction vendor to maximize the vendor's accuracy. The prediction vendor processed the data user and the target attribute for the obfuscation task. However, such a model requires adequate information from the user and access to the user attribute values. Various minimax formulations and algorithms have been proposed to defend against inference attacks [Bertran et al., 2019, Wu et al., 2018, Whitehill and Movellan, 2012, Osia et al., 2018, Li et al., 2018, Hamm, 2017]. [Bertran et al., 2019] proposed a learning framework based on mutual information to balance per-subject information obfuscation and utility preservation, where the data is sanitized before disclosure. However, such a model is susceptible to inference attacks, i.e., an adversary can accurately infer sensitive attributes from sanitized data [Hamm, 2017]. [Agrawal and Srikant, 2000] analyzed an algorithm to perturb data by adding random noise to preserve users' privacy.

To sum up, existing defenses against attribute inference attacks focus on camouflaging public data and confusing attackers by adding noises to public data. Introducing noise to the published pictures in online data in Facebook requires fake profiles to add comments to prevent the attacker classifier from predicting the target user attribute with a higher probability. The disadvantages of this approach are twofold. First, the added comment might be irrelevant to the content of the picture and its related set of comments. As a result, the attacker can locate and remove the added comments. Second, The attacker can recognize those added comments (as they might be unrelated) and ignore them in the attack process.

In contrast, we propose a protection model based on privacy settings that users can control to hide risky reactions to prevent inference attacks. The advantage of hiding comments, for example, on Facebook, is that it is still visible to the user who posted this comment and their friends, which reduces tension between the commenter and the picture owner. Our model is (i) computationally tractable with low utility loss and (ii) does not require the knowledge of users' private attribute value.

Chapter 3

Preliminaries

Contents

This chapter aims to give a broad overview of some essential tools or techniques that we have applied in the following chapters. Therefore, we introduce our data collection tool, the machine learning algorithms, and the word representation model we have leveraged in the thesis.

3.1 Data collection

We focus on Facebook in this thesis for two reasons. First, it is one of the largest OSN platforms concerning the number of users and the amount of personal data collected about users and made available to third parties (e.g., advertisers). Second, Facebook has been questioned and investigated over the years by regulators about its privacy practices. To collect data from Facebook, one can use Facebook Graph API. However, this API was subject to many revisions, especially after the Cambridge Analytica debacle, so several Graph API endpoints have been completely closed. In addition, Facebook controls and limits the data returned for each API request. For example, one can receive a different response for the same query or sometimes incomplete data.

To overcome the above obstacles, we have implemented our crawler that simultaneously browses different user profiles while accessing only public information to collect ground truth data. It starts by collecting information from the target user to form a labeled dataset. Then, it scans published pictures to extract pictures metadata (alt-text and comments) from the HTML part of the visited image. Figure 3.1 shows an example where the crawler extracts user

information. The crawler can collect user gender from *About* tab inside the profile (Figure 3.1(a)), and alt-text and comments inside the *HTML* part of the visited picture (Figure 3.1 (b, and c)). Arrows in Figure 3.1 (c) shows the commenter's name and his comment.

At first, we provide a group of random profile URLs to the crawler. In addition to the alt-text and comments inside the *HTML* part of the visited picture, the crawler also collects the commenters' profile URLs that are later selected as potential candidates for data collection. We have implemented our crawler in Python and applied Selenium[5], a powerful tool for controlling the crawler and performing the data collection automatically.

3.2 Supervised Machine Learning Algorithms

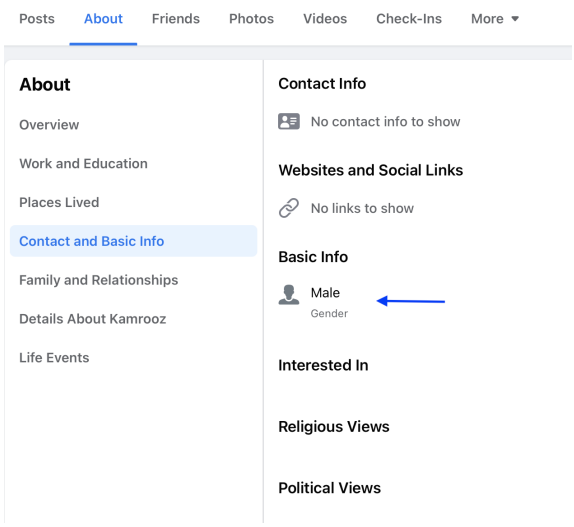
This section introduces the machine learning algorithms that we have applied in the thesis. We have implemented these machine learning algorithms in *scikit-learn*. To optimize the performance of these algorithms and find the best value of parameters, we have relied on *GridSearchCV* [6] inside the *scikit-learn* to perform an exhaustive search over all the predefined parameters values.

3.2.1 Decision Tree

A Decision Tree (DT) [Breiman et al., 2017] is a non-parametric supervised learning method that recursively splits the dataset based on conditions. The core idea behind DT is to create a prediction model from simple decision rules learned from the dataset. It has a hierarchical structure, where traversing the tree from root to leaves leads to outcomes. Generally, a decision tree consists of a root node and several internal and leaf nodes. The root node is the first splitting node of the tree, and the entire feature set is evaluated to find the feature to be tested at the root node. An internal node also denotes a test on a feature. A subset of the original feature set is analyzed to find the feature tested at this internal node. At the end of the construction, a leaf node determines the label. While creating a tree, it is essential to decide which features should be placed at a root node or internal node and prevent overfitting. To answer these questions, we prune some parameters [7] of DT (presented below), such as the function that measures the quality of a split, the maximum number of features to evaluate for the best split, and the minimum number of samples required to split a node or consider it a leaf node.

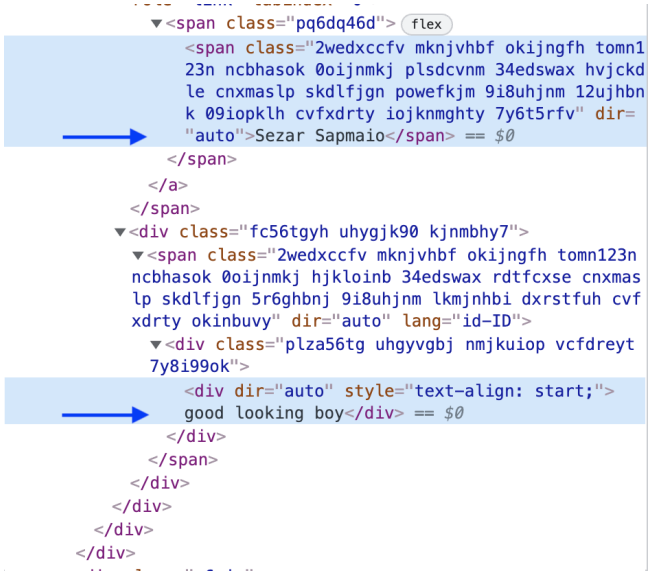
Criterion is the first parameter in DT, which is used for feature selection in order to place a particular feature in an appropriate position of the decision tree [Daniya et al., 2020]. *Criterion* can be based on *Gini impurity* or *Information gain* (entropy). For example, in gender classification with two labels F and M for Female and Male respectively, a feature that appears in both female and male labels is impure, whereas a feature that appears only in one of the labels is pure. To determine which split is best, we measure and compare the impurity of each feature. If the impurity is high, then the feature is not discriminative between labels, and consequently, the result of splitting on that feature is poor. To find the best feature to split the tree, we leveraged *Gini impurity* as it (i) performs better than entropy in our problem, and (ii) is faster to derive because it does not require computationally intensive logarithmic functions

[5] `https://www.selenium.dev/`
[6] `https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GridSearchCV.html`
[7] `https://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeClassifier.html`



(a)

(b)



(c)

Figure 3.1: Extracted information from: (a) user profile, and HTML part of a picture: (b) alt-text (c) commenter name and comment.

[Raileanu and Stoffel, 2004]. The *Gini impurity* of node n can be expressed as:

$$\text{Gini impurity(n)} = 1 - ((pr_F)^2 + (pr_M)^2)$$

where pr_F and pr_M are the proportion of samples with label F, and M at node n. Consider an example to explain the process, where we have a dataset of 7 samples and a feature f with value 0 (resp., 1), meaning that f does not occur (resp., occurs) in the sample, and binary labels F and M:

sample	f	*label*
1	1	F
2	1	M
3	1	M
4	0	F
5	0	F
6	0	M
7	1	M

Table 3.1: Samples with a single feature (f).

To compute the impurity of f, we create a small tree for this feature (Figure 3.2). Each branch corresponds to a value of 1 (resp., 0) for the presence (resp., absence) of f.

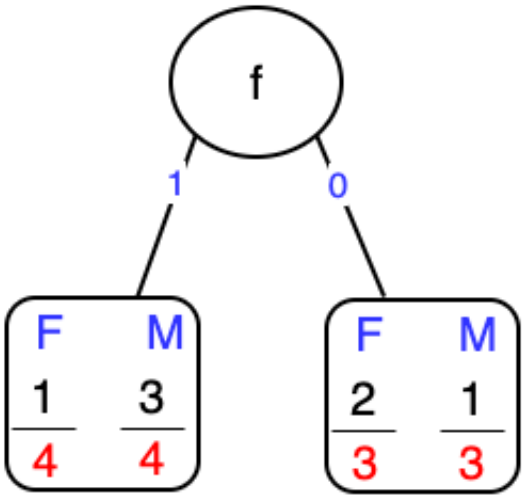


Figure 3.2: Small tree for f.

We compute the number of times f appears (resp., does not appear) in female and male labels. As a binary case, we have two outcomes, 0 and 1. The left node represents the case when $f = 1$ inside Table 3.1 and the right node shows the case when $f = 0$. The *Gini impurity* for child nodes, the left, and right nodes are:

$$\text{Gini impurity (left node)}= 1 - ((\frac{1}{4})^2 + (\frac{3}{4})^2) = 0.375$$

$$\text{Gini impurity (right node)}= 1 - ((\frac{2}{3})^2 + (\frac{1}{3})^2) = 0.444$$

We calculate the *Gini impurity* of f as a weighted average of the child nodes impurities as follows:

$$\text{Gini impurity}(f) = \frac{4}{7} * 0.375 + \frac{3}{7} * 0.444$$

This is because the child nodes have different frequencies in the seven samples: 4 in left and 3 in right node.

Determining the depth of the decision tree is another issue. Deep trees cause overfitting, while shallow ones cause underfitting. To that end, we tune some parameters related to the tree depth. For example, $max\_features$ controls the size of the random subsets of features to be considered when splitting a node. *Gini impurity* will only be computed for those randomly selected features, except the root node. In addition, $min\_samples\_split$, and $min\_samples\_leaf$ are two parameters that control the splitting to prevent overfitting. The $min\_samples\_split$ parameter evaluates the number of samples required to split an internal node. No splitting is performed on the current node if the number is less than a predefined value. Furthermore, this node becomes a leaf. The $min\_samples\_leaf$ parameter checks whether we can go on splitting or not. DT avoids further splitting if the number of samples that would remain after a tentative split is less than a predefined value.

3.2.2 Random Forest

Random Forest (RF) is an ensemble method that constructs many random decision trees that will be used to classify an input by the majority voting. RF randomly draws different training subsets with replacements from the entire training set. These subsets are known as bootstrapped datasets.RF trains single random decision trees in parallel, where each node in a random decision tree uses a subset of features randomly selected from the whole original feature set [Oshiro et al., 2012]. These random decision trees are also known as weak learners. Each bootstrapped dataset is fed as input to weak learners, and each weak learner casts a unit vote for that input. The final output is based on the majority voting, the most popular label for the input. RF credence to the ensemble learning idea: the crowd is more intelligent than the individual. A single random tree or a weak learner represents the individual that may not perform adequately due to high variance or bias. However, the aggregation of weak learners can form a strong learner due to the reduction in bias or variance, which gives the model better performance employing a majority vote.

Since bootstrap subsets are randomly selected with duplication, some instances of the original training set are not found in the bootstrap subsets. Consequently, they are not used to construct corresponding random decision trees. These instances are used to measure the performance of the random forest according to [Breiman, 2001] "for each instance in the training set, aggregate the votes of random trees for which their corresponding bootstrap subsets do not contain that instance, known as an out-of-bag classifier."

We improve the model performance using *GridSearchCV* to find the best values of the parameters. However, most of the RF parameters [8] are similar to DT parameters. As a result,

[8] `https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html`

we tuned them in the same way we did for DT. In addition, we tuned the RF parameter that controls the number of trees in the forest.

3.2.3 eXtreme Gradient Boosted trees

eXtreme Gradient Boosted trees (XGBoost) [Chen and Guestrin, 2016] is an ensemble learning method that trains weak learners sequentially, in such a way that a newly added tree in the sequence tries to compensate for the weaknesses of its predecessor in minimizing training errors. The core idea of XGBoost is to maintain multiple versions of trees chained and built on top of each other to prevent the misclassification of the previous trees. XGBoost is simple and computationally efficient and can be applied to classification and regression problems. It can automatically handle the missing values and can be parallelized. However, the model can learn quickly and overfit training data. In addition to the previous tree-based models' parameters, we tune specific parameters [9] that affect the model performance, such as step size in updating the model to prevent overfitting, subsample ratio of the training instances, and the minimum sum of instance weight in a child node.

Learning_rate controls the weighting of the added new trees to the sequence and slows down the learning phase. *Subsample* controls the amount of data the model employs for building the tree, preventing overfitting.Min_child_weight performs regularization at the splitting step.

3.2.4 K-Nearest Neighbour

K-Nearest Neighbour (KNN) [Cover and Hart, 1967] is a simple supervised algorithm used for the regression but especially more for the classification. KNN is a lazy learner who does not learn a discriminative function from the training dataset. Instead, it stores the dataset, and at the classification time, it performs actions on the dataset. The core idea behind KNN is to find a similarity between the new data and the available data. We capture the similarity by using Euclidean distance. A data point is classified based on the plurality vote of its neighbors, namely the most common label among the nearest neighbors. Symbol K in KNN' refers to the number of nearest neighbors included in the classification process. Choosing the correct value for K is essential for classification accuracy. A low value causes skewed classification, while a large value raises difficulties in searching the nearest neighbors for samples and resource issues. In our applications, we will set K to 5 as it gives the best result. We evaluate the algorithm performance by changing the parameters [10]. Lastly, we select the best combination.

3.2.5 Naive Bayes

A Naive Bayes (NB) classifier [Rish et al., 2001] is based on Bayes' theorem with strong (naive) independence assumptions. It assumes the occurrence of one feature is independent of other features. NB is a probabilistic classifier, meaning that for a picture p, out of labels G={female, male} the classifier returns the label with the maximum probability. Let F be the set of features $F = \{f_1, f_2, ..., f_k\}$, and then NB can be defined as follows:

[9] `https://xgboost.readthedocs.io/en/latest/parameter.html`

[10] `https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.KNeighborsClassifier.html`

$$\arg\max_{g \in G} Pr(g) \prod_{i=1}^{k} Pr(f_i|g) \tag{3.1}$$

where $Pr(g)$ is the prior probability of label g, and $Pr(f_i|g)$ is the conditional probability of observing the ith feature f_i given label g, also known as likelihood. For our classification task, multinomial Naive Bayes works better than other variants. Multinomial Naive Bayes considers comments underneath the picture as a bag of words, that is, an unordered set of words and keeping only their frequency in the document.

To train Naive Bayes, we compute (i) $Pr(g)$, the frequency of pictures in our training set labeled by each label g, and (ii) $Pr(f_i|g)$, the frequency of f_i among all words in label g. However, there is a problem with conditional probability training. For example, the conditional probability of a word *brilliant* given label *male* is zero if there are no samples in the training dataset where the word *brilliant* occurs in the *male* label. Although the word *brilliant* happens to occur in the *female* samples, the conditional probability of this feature for the *male* label is zero.

This can lead the model to misclassify a new sample. To prevent this, we apply Laplace smoothing, where each word in the training set receives an extra count.

After training and calculating the mentioned probabilities, a new sample can be classified thanks to the NB formula 3.1, and the class is assigned to the label with the maximum value.

3.2.6 Logistic Regression

Linear regression [Feng et al., 2014] attempts to model the relationship between independent (predicted) and dependent (predictors) variables by fitting a linear line. Logistic Regression (LR) is similar to linear regression, as it can work with continuous and discrete data. The main difference between both regressions is how the classification line fits the data. LR applies the logistic function (Sigmoid Function) to map the output of a linear equation to probabilities used for our gender classification problem. For example, in binary classification, the sample is classified in one of the labels if the probability of some label is higher than a specific threshold. LR is defined as follows:

$$sigmoid(z) = \frac{1}{1 + e^{-z}} \tag{3.2}$$

where $z = \theta_0 + \sum_{i=0}^{m} \theta_i x_i$, θ_0 is the intercept, θ_i are the coefficients (weights) that indicate the importance of feature f_i for classification, and x_i is a variable that takes value 0 (resp., 1) when feature f_i does not occur (resp., occurs) in the sample.

To choose values of θ, we train the logistic regression with stochastic gradient descent as an optimization function to find the values of θ that maximize the logistic function 3.2.

3.2.7 Support Vector Machine

Support Vector Machine (SVM) [Noble, 2006] is a supervised learning method for classification and regression problems. The core idea is to map the data instances to an n-dimensional space, where n is the number of features, and the value of each feature corresponds to a coordinate value. To perform classification, SVM finds hyperplanes (separating lines) that best segregate the labels (e.g., two labels in binary classification). A hyperplane defines a decision boundary.

We note that several possible hyperplanes can separate data points.

To obtain the hyperplane that optimally splits the data points, SVM proceeds as follows. It finds the nearest data points, known as support vectors, to each hyperplane from both labels. Support vectors influence the position and orientation of the hyperplane. Then, SVM computes the distance between the hyperplane and support vectors, known as the margin. The optimal hyperplane is the one for which the margin is maximum. If the data are linearly separable, SVM finds the hyperplane that maximizes the margin and minimizes the misclassifications. However, if the data are not linearly separable, or a nonlinear boundary can separate the labels more efficiently, SVM uses a kernel function to solve this problem. According to [Noble, 2006] "the kernel function is a mathematical trick that allows SVM to map data from a low-dimensional space to a space of higher dimension." It means a nonlinear function is learned by a linear function in a high-dimensional feature space. Kernel trick offers an efficient and economical way to transform data into higher dimensions. Choosing a good kernel function makes the data separable in the resulting higher-dimensional space.

To that end, we have tuned the SVM parameters [11] such as the kernel type, the regularization parameter, and the parameter that controls the curvature of the decision boundary (*gamma*). A high value for *gamma* makes the decision boundary look like a wiggly curve, as it depends on very close points. On the other hand, a low value for *gamma* results in a linear curve as the decision boundary is dependent on points that are far from it.

3.3 Word Embeddings

Word embeddings extract the semantic relationships inside the text properly. It has achieved great success in many NLP tasks in recent years. Researchers have applied word embeddings in many tasks, such as online social networks analysis [Beheshti et al., 2020], information retrieval [Ganguly et al., 2015], document classification [Kim et al., 2020], question answering [Shen et al., 2017], named entity recognition [Akbik et al., 2019], sentence embedding [Gao et al., 2019]. A word embedding is a numerical representation of words learned from an extensive corpus. A word embedding maps every word inside the corpus to an element of a vector space of relatively low dimension N, typically N = 300 (in the word2vec model described below). The core idea is that words appearing in the same context will be represented by related vectors in the vector space. For instance, the semantical proximity of two words is revealed by a small value of the angle between the two vectors representing these words. The *cosine similarity* is the cosine of these two vectors. Figure 3.3 represents the similarity between three words; *Apple*, *Banana*, and *Car*. The angle between *Apple* and *Banana* is small, and the *cosine similarity* is close to 1. On the other hand, *Apple*, and *Banana* have wide angles with the word *Car*, and the *cosine similarity* is close to 0.

Word2vec is a popular approach for learning word embeddings from raw text. It follows Harris's hypothesis [Harris, 1954]: words with similar contexts tend to have similar meanings. The basic idea is to map a one-hot-vector representation of a word from a high-dimensional vocabulary space to a continuous vector space with a lower dimension. It is a semi-supervised learning approach that takes a large corpus as input and generates a semantic representation vector based

[11] `https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html`

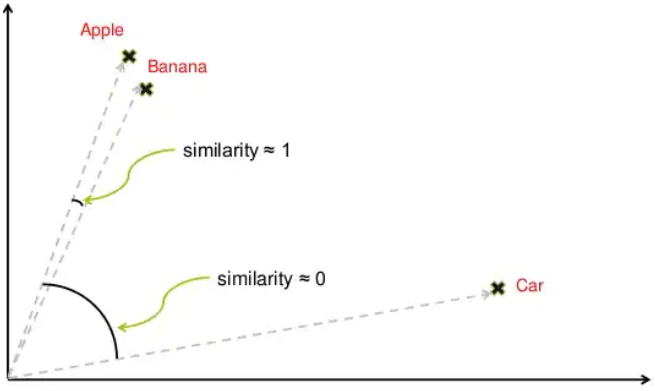


Figure 3.3: Some word vectors.

on the contexts c of the words. Since word2vec learns word embeddings in an unsupervised manner, the authors in [Mikolov et al., 2013a] have introduced two different models, called skip-gram and the continuous bag of words. These models have different ways of word embedding learning. For example, the skip-gram model aims to predict the context of a given the word, while the continuous bag of words model aims to predict a word given its context. The contexts of a word w in the text are the surrounding words in specific window size [Levy and Goldberg, 2014a]. The window size limits the number of words on either side of w to consider when generating a semantic representation vector for w. Word2vec generates a vector for w by presenting different pairs of context-target words from the corpus.

Skip-gram Model. Figure (3.4) represents skip-gram model. For each target word $w(t)$ from the vocabulary taken as an input, the model predicts the contexts of the word $(w(t-2), w(t-1), w(t+1), w(t+2))$ in which $w(t)$ is more likely to appear in the processed document. The input word to the model is represented by a *one-hot-vector*. This vector has $v-1$ zeros and a 1 in the position of the corresponding word, where v is the vocabulary size. The output of the skip-gram model has the same size as the input, one single vector of size v. The output vector represents a probability distribution of co-occurrence between the input word and each word from the vocabulary within a window of size c [Abid, 2018]. The training objective of the skip-gram model as stated in [Mikolov et al., 2013b] is: "to find word representations that are useful for predicting the surrounding (context) words in a sentence or a document."

Continuous Bag Of Word (CBOW) Model. Figure 3.5 shows the CBOW model. In this model, the input is a bag of words $(w(t-2), w(t-1), w(t+1), w(t+2))$ at distance 2 at most from the target word. The model predicts a single target word $w(t)$ in which $w(t)$ is the most likely word to appear in that context in the processed document. As presented in Figure 3.5, there are four words as a context of target word $w(t)$ at a distance of 2 at most. As a result, there are four input vectors. Each of them will be multiplied with $W_{v\times n}$ and the results will be averaged element-wise to obtain a single vector. Similar to skip-gram, the output of the CBOW is a single vector of size v. This vector represents a probability distribution of co-occurrence between context words and target words from the vocabulary within a window of size c (see also [Abid, 2018]).

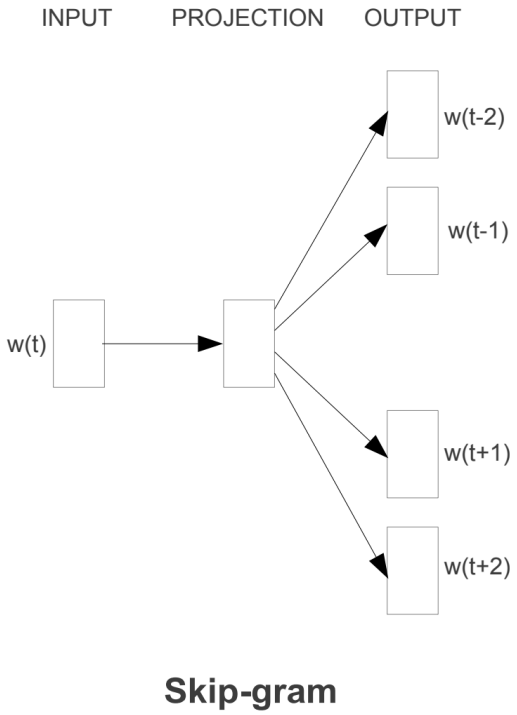


Figure 3.4: SkipGram models [Mikolov et al., 2013a].

Neural network. Word2vec has a shallow neural network architecture with a single hidden layer (Figure 3.6). The number v of words in the vocabulary determines the number of input layer neurons, and the number n of neurons in the hidden layer defines the vectors' dimensionality. A matrix $W_{v\times n}$ (resp. $W'_{n\times v}$) has for entries the weights between the input and hidden layers (resp., the hidden and output layers), The main objective of the neural network is to learn the matrix $W_{v\times n}$. The training proceeds by presenting different context-target words pair from the corpus. W and W'(Figure 3.6) are initialized randomly and updated using *backpropagation* with *gradient-descent* technique.

Word2vec converts words to vectors, where multiple mathematical operations can be performed (e.g., adding, subtracting, calculating distance). In that way, it has the property of word analogies like in the famous example:

$$vector(king) - vector(man) + vector(woman) \approx vector(queen)$$

We have leveraged the pre-trained vector representation of the word2vec model that has been trained by CBOW [12], according to the principal author of word2vec, *Tomas Mikolov*. We define a technique (in Chapter 5) to adapt these pre-trained vectors to our specific model.

[12] `https://groups.google.com/g/word2vec-toolkit/c/lxbl_MB29Ic/m/NDLGId3KPNEJ`

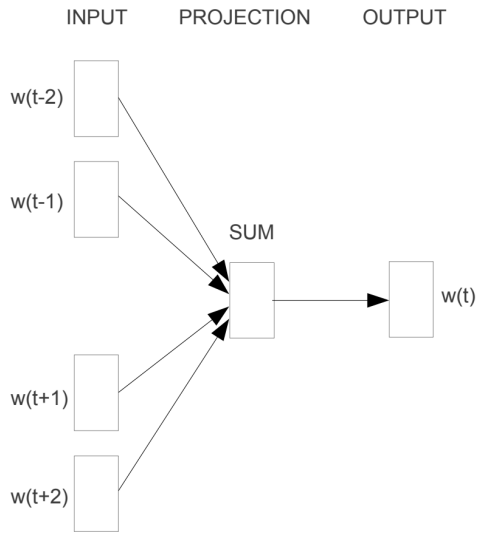


Figure 3.5: CBOW models [Mikolov et al., 2013a].

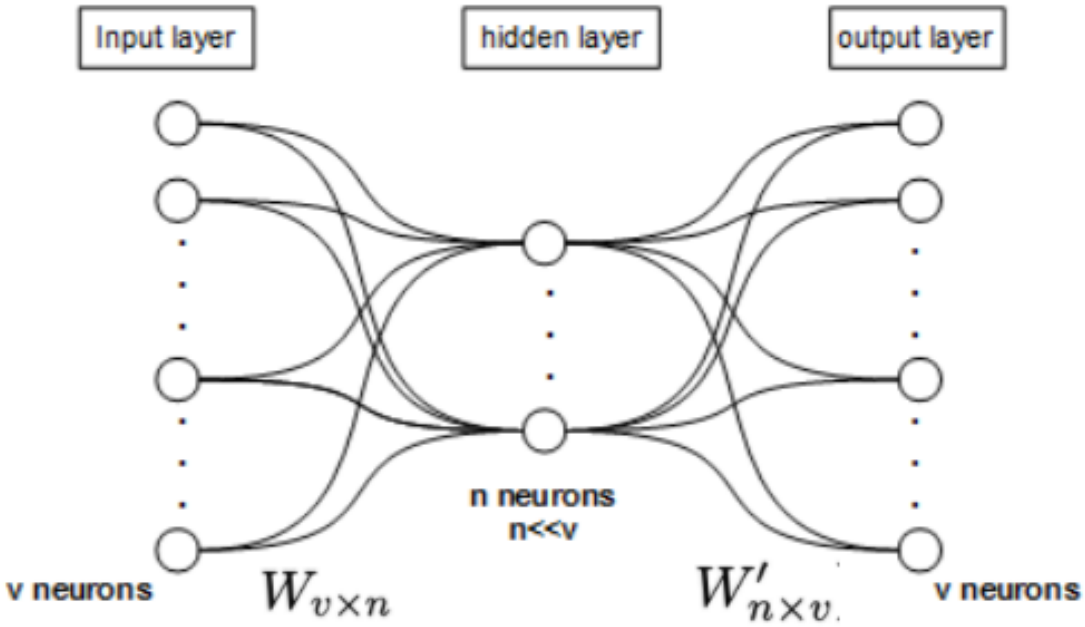


Figure 3.6: Word2vec architecture [Abid, 2018]

Chapter 4

Gender Inference Attack on Emojis

Contents

4.1 Introduction

Nowadays, emojis have gained incredible popularity and have become a popular supplement to text-based communication in social media [Butterworth et al., 2019]. In comparison to text, emojis are pictographs used in digital communication, and they can strengthen emotional text-based communication and increase the comprehension and interpretation of messages. As an illustrative example, Andy Murray, the British tennis player, announced his wedding on Twitter with only 51 emojis (no words) [Chen et al., 2018]. Emojis connect users who (i) speak different languages, (ii) have different countries, cultures, demographic groups, and are widely simplifying emotional expression [Lu et al., 2016]. Recently, several researchers have studied emojis to understand their semantics and sentiments behind them [Barbieri et al., 2016b, Kralj Novak et al., 2015, Pohl et al., 2017]. For example, [Ling et al., 2014, Tossell et al., 2012] have studied the differences in the number, types, and motivation of emojis used by females and males. The authors of [Weisberg et al., 2011] have suggested that women are more easy with affectionate and agreeable emojis.

In contrast with the previous works, we aim to study the effect of picture owner attributes on commenters' emojis/emoticons preferences. In this chapter, we mainly focus on gender inference attacks. As a result, the analysis and questions are related to this attribute value. Here, we are interested in answering the following questions: Is there a significant difference in the usage of emoji/emoticon for commenting on female and male-owned pictures? Do females and males receive different emojis/emoticons for photos with the same theme and settings (e.g., similar alt-text tags)? Do females and males share pictures with similar themes, settings, and objects?

Scenarios. To answer these questions, we consider three different scenarios. In the first scenario, we purely rely on the emojis/emoticons, regardless of the language of the comments, as input data to infer the picture owner's gender. In the second scenario, we study only the generated alt-text as input data to discover female and male preferable pictures sharing styles on Facebook. We analyze which Facebook alt-text tags orient more towards female (resp., male) owned pictures. This analysis enables us to show the preferences of picture sharing styles between female and male users. As for the third scenario, we consider the correlation of emojis/emoticons and alt-text tags. In this scenario, we aim to discover commenters' preference for emojis/emoticons usage by considering picture owner gender and generating alt-text tags of that picture.

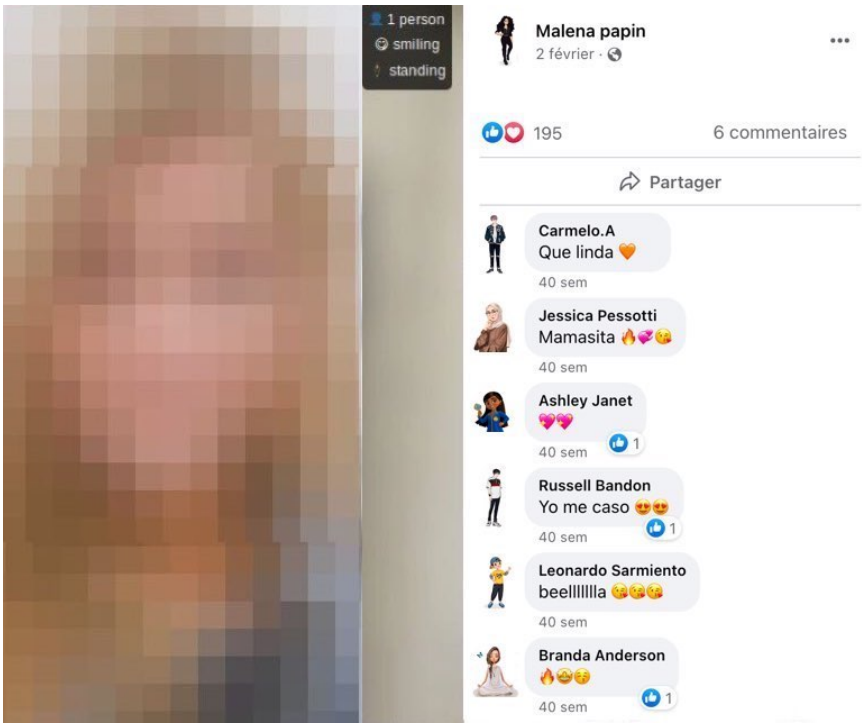
Previous gender inference attacks on Facebook have two main limitations. First, users' friend-based and behavior-based data are extensively considered in the attack process, degrading prediction accuracy in the case of scarce or unavailable data. Second, these attacks are limited to text-based knowledge. For example, a person's gender identity can be derived from linguistic features associated with male or female writing style on social media, but the prediction accuracy decreases when texts are multilingual or unavailable (as presented in Figure 4.1).

In this work, we exploit non target-user generated data (i) alt-text, computed by some OSN platforms such as Facebook and Instagram, and (ii) emojis/emoticons added by commenters while commenting on the picture. Emojis/emoticons are universal, non-verbal communication language that makes the inference attacks possible even when the received comments only consist of emojis/emoticons.

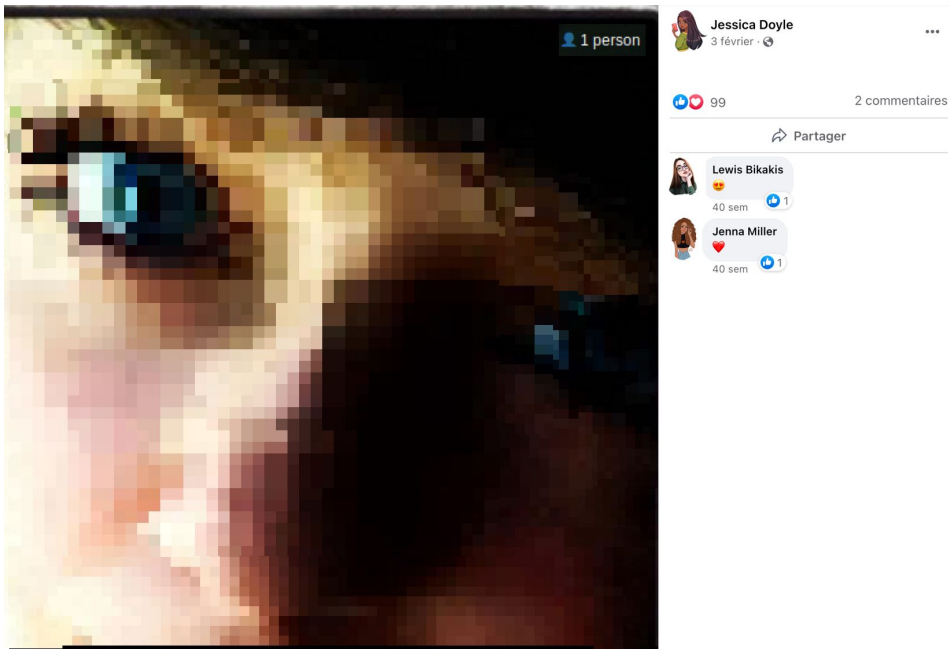
For the rest of the chapter, we explain our collected data in Section 4.2. Then we analyze gender bias in received emojis/emoticons in Section 4.3. Next, we introduce our feature selection algorithm in Section 4.4. Finally, in Section 4.5, we show the possibility of a gender inference attack even when essential information from the target users and their vicinity network is unavailable.

4.2 Collected Data

This section illustrates the dataset and the pre-processing steps in detail. We launch our gender inference attack by collecting picture metadata (i.e., alt-text and emojis/emoticons of the comments) from the related HTML file. We extract the user gender, when available, to create labeled data sets to be exploited by our supervised machine learning algorithms. We have randomly selected Facebook users to avoid usage bias by region or country. We consider two labels, *female* and *male*, corresponding to biological sex. Let $P = \{p_1, p_2, p_3, .., p_m\}$ be the set of target



(a)



(b)

Figure 4.1: Pictures reactions: (a) emojis and non-English words (b) only emojis.

user pictures. For every picture we collect $p_i = \langle a_i, e_i \rangle$ where a_i is an alt-text and e_i is the set of emojis/emoticons posted by commenters for that picture.

Data pre-processing

We perform four pre-processing steps to cleanse the collected data as follows:

1. Cleansing pictures owned by someone else. Facebook users can share photos on each other profiles. Considering these pictures owned by someone else might hinder the inference attack as the publisher owner and the user might have different attribute values. We only examine images published by the user and filter out photos owned by others.

2. Cleansing alt-text. Facebook admits *97* different tags to describe the content of each uploaded picture. We reformulate the generated alt-text in three steps, which contain:

Conjunction: Facebook admits two conjunctions inside their generated alt-text: *and*, and *or*. For example, Facebook produces tags like *one or more people* if their algorithm is uncertain about the number of people inside the picture. We reformulate this tag to *more people*.

Redundant Tag: In some pictures, the alt-text comprises repeated tags. Consider the following example where the alt-text contains: *3 people, people smiling, people standing, and hat*. The *people* in the first tag refers to the number of people inside the picture, *3*. However, *people* in the second and third tags *{people smiling, people standing}* respectively, is a redundant word. As a result, we reformulate the alt-text to *3 people, smiling, standing, hat*.

Text: Under some conditions, Facebook appends the text messages to the generated alt-text if the image already contains text in it. For example, *4 people, meme, text that says 'WHO ARE YOU IN THE DIFFERENT WORLDS?*. We reconstruct the alt-text to *4 people, meme*.

These steps aim to clean the alt-text and to create a proper feature selection that we will be explained later.

3. Filtering the picture w.r.t the following alt-texts rules. Inferring attributes from *non target-user generated data* may be deceiving. In some situations, picture metadata (alt-text and comments) convey contradictory information. As shown in [Shutterstock, 2017], some categories of pictures (e.g., baby and animal images) are more likely to receive strong emotional reactions (animated and/or textual) than other pictures. To precisely filter pictures, we analyze the words which give a clue about the gender disparity inside the posted comments, like she and he. Below, we provide three examples where the comments alone orient gender inference towards one value, but reviewing alt-text has fixed the initial wrong guess.

Image 1:
Generated alt-text: dog, outdoor and nature
Comment: ❤️😍

Image 2:
Generated alt-text: 1 person, smiling, child and closeup, dog

Comment: She looks so happy 😍
Comment: Look at his tail. Nice pic 💙

Image 3:
Generated alt-text: 1 person, child, sleeping and bedroom
Comment: Precious!!! 💙❤️💙
Comment: Priceless moments! Love y'all 😘
Comment: I love this!!! 💓

In the first image, the comment contains emotional emojis similar to Figure 4.1(b).

However, checking the alt-text suggests only a dog inside the picture. In the second image, the presence of *child* and *dog* in alt-text indicates that comments point to a baby girl (SHE) and a male dog (HIS). The *child*'s presence in the generated alt-text of the third image hints that the comments are unrelated to the picture owner's gender. We filter pictures using alt-text to circumvent this misleading information in our inference attack process. Consequently, we filter the picture if there is *animals* in the *objects tag* (Image 1), and there are *1 person* and *child* inside the alt-text (Image 2, and 3). We apply our picture filtering rules only on pictures where alt-text and comments were available.

4. Emoticons. An emoticon is a digital icon that conveys a human expression, and it usually contains punctuation marks, numbers, and letters. According to Yahoo Messenger[13], MSN messenger [14] and an internet source [15], most used emoticons have their corresponding emoji(s). For example, ;−), and ;) emoticons have the corresponding emoji 😉. To reduce the complexity of the model and make the inference attack effective, we find the best match for each emoticon and replace it with emoji. We observe *120* different emoticons in our data set. To perform this task, we used online services such as *The Smiley Dictionary*[16], *urbandictionary*[17], *Emojicodes*[18], *Emoticonr*[19], and *IM Emoticons*[20]. For the rest of the thesis, we use emojis to show emojis/emoticons.

4.3 Gender Bias in Received Emojis

This section discusses the owner's gender impact on commenters' reactions. We hypothesized that the content of the image and owner gender shape commenters' reactions, which evoke a specific emotion in commenting for pictures. We consider *2,158* users and their *141,812* pictures to prepare all tables presented in this section.

[13] https://web.archive.org/web/20090411052027/http://messenger.yahoo.com/features/emoticons/
[14] https://web.archive.org/web/20090707114539/http://messenger.msn.com/Resource/Emoticons.aspx
[15] https://web.archive.org/web/20201021131252/https://www.lifewire.com/what-are-emoticons-2482961
[16] https://www.csh.rit.edu/~kenny/misc/smiley.html
[17] https://www.urbandictionary.com/
[18] http://Emojicodes.com/
[19] http://www.Emoticonr.com/
[20] http://sheet.shiar.nl/emoji

alt_ text	*Gender*	*Top 10 Facebook users emojis usage*
1 person	*Male*	😂❤️😍 😘 😉😅 😊👍😀😆
	Female	😍 ❤️ 😘 🔥 💕💙 😊 😉 💖 💜
smiling	*Male*	😂❤️😍 😘 😅 👍 😊😉 🙏 😀
	Female	❤️ 😍 😘 😂 💕 😊 🔥 💗 💖 💜
closeup	*Male*	😂❤️😍 😘 😊 😅 😉 😛 👌 🤔
	Female	😍 ❤️ 😘 🔥 💙 💕 😊 💖 💜 😉
outdoor	*Male*	😂❤️👍 😍 😅 😘 😄 😊 😉 😎
	Female	❤️ 😍 😘 💙 💕 😊 🔥 😅 💜 🎉

Table 4.1: Emojis preferences in commenting female and male-owned pictures with specific alt-text tags.

4.3.1 Emoji popularity

We investigate commenters' emojis preferences while observing female and male-owned pictures with similar alt-text (tags). Developing a comprehensive analysis on commenters' reactions while commenting for female and male-owned images drive to gender inference attack. We collect the top *4* tags of alt-text generated by Facebook for female and male published pictures to understand the differences. Later, we study the top *10* commenters' emojis preferences while commenting on photos containing these *4* tags (considering the owner's gender). Table 4.1 indicates that commenters used more smiley emojis (e.g., 😂) for male images, while emotional emojis (e.g., ❤️) commented more on female pictures. The order of these *10* emojis is from left to right. To be fair in discriminating the reactions for females and males, we need to analyze equal numbers of pictures. To that end, we apply sub-sampling over the female pictures, the dominant one, and calculate the result of Table 4.1 over the random sub-samples. We discover commenters used *1291* different emojis to comment on female pictures, while they used *877* for male photos. To conclude, we identify that gender and content of the image affect commenters' emojis usage, and they have a higher tendency of posting emotional emojis for females than male pictures.

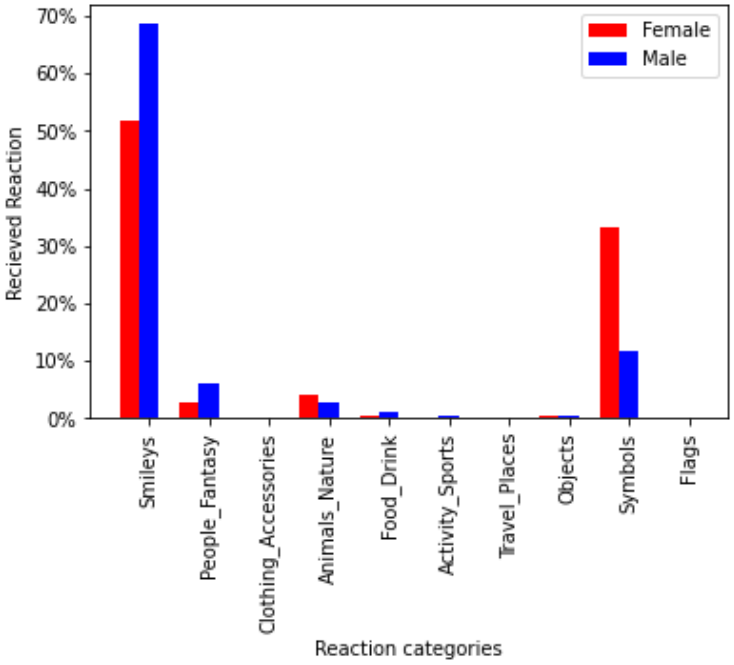
4.3.2 Emoji categories and animated reactions

Emoji categories. We use emoji categories [21] to demonstrate the influence of owner gender on commenters' reactions. To that end, we show the emoji preferences of commenters according to *10* given categories. We discover commenters frequently use emojis from *Smileys* and *Symbols* categories for commenting in male and female photos, respectively, more than other categories. As illustrated in Figure 4.2(a), commenters use more emotional emojis from *Symbol* category, which contains heart-based emojis, to express their feeling to female-owned images. While, they have a higher tendency in using emojis from *Smileys* category, which holds face-based emojis, to comment on males-owned pictures.

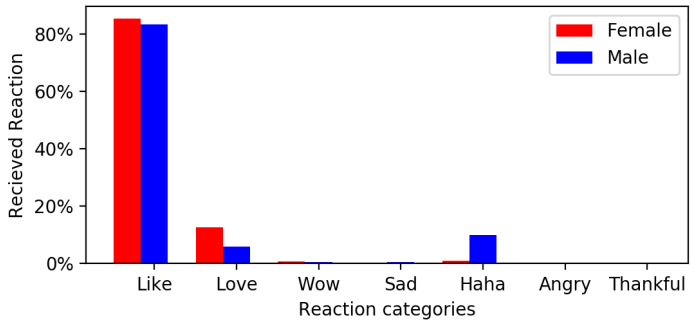
Animated reaction. We plot commenters' animated reactions to the observed pictures following the previous findings. In general, there are seven different expressions on Facebook (as it is shown in Figure 4.2(b)). The result shows that Facebook users *Like* male and female posted images frequently, in comparison to other expression types. Although this expression is very close,

[21] `https://getemoji.com/`

female pictures received slightly more *Like* than male pictures. According to Figure 4.2(b), Facebook users used more *Haha* for male-owned photos, while they use more *Love* for female-owned photos. This outcome is additional evidence to the previous result, which commits the Facebook users different reactions (animated and textual) to male and female posted pictures.



(a)



(b)

Figure 4.2: Reactions: (a) textual (emojis based on categories) (b) animated.

To sum up, there is a significant disparity in using emoji, based on the categories and animated reactions to female and male-owned pictures. These differences drive gender inference attacks. As such, the evaluations of the picture owner gender in forming others' emotional reactions (animated and textual) and the association of these reactions with the content of the picture are two empirical questions that need to be investigated. To answer these questions, we apply Mutual Information (MI) in all three scenarios to effectively conduct the differences, which measures (i) the mutual dependence between gender and received emojis, (ii) the mutual dependence between picture owner gender and generated alt-text for that pic-

ture, and (iii) the correlation of picture owner gender with generated alt-text and received emojis.

Let f be a feature that can be any combination of emojis in the first scenario, any combination of alt-text tags in the second scenario, or any combination of emojis and alt-text tags in the third scenario, where scenarios are defined in 4.1. Let X be a random variable that takes values $x = 1$ if the posted photo contains f and $x = 0$ if the posted photo does not contain f. Let Y be a random variable that shows the picture owner gender, where it takes values $y = 1$ for female and $y = 0$ for male. Then, we compute the MI as follows:

$$MI(X;Y)_f = \sum_{x\in\{0,1\}} \sum_{y\in\{0,1\}} Pr(X = x, Y = y)_f log_2 \frac{Pr(X = x, Y = y)_f}{Pr(X = x)_f, Pr(Y = y)_f} \tag{4.1}$$

where $Pr(X = x)_f$, $Pr(Y = y)_f$ are the marginal probabilities of x and y, $Pr(X = x, Y = y)_f$ is the joint probability of x and y, and f can be either a single or a combination of alt-text tags and/or emojis. For example, if f is 😍, the joint probability of $Pr(1, 1)_{😍}$ is the probability that a commenter posts 😍 for a female-owned picture. For the illustration purpose, we consider a single and sequence of 2 for f and represent the results in different tables (described below). Each table calculates the probability of a person being male or female, given the picture generated alt-text tags and/or emojis.

Table 4.2 lists the top *10* discriminative emojis used by commenters in commenting on female and male-owned pictures. In this table, $p(female|😍)$ is the conditional probability that the user is female given her picture receives comments with 😍. We define an emoji f as a female emoji if $p(female|f) > p(male|f)$, otherwise a male emoji. As a result, all the entries in this table are discriminative emojis for females. In Table 4.2, we draw inspiration from [Chen et al., 2018]. From the same spirit, we show the top *10* discriminative alt-text tags and the correlation of alt-text tags and emojis for females and males in Tables 4.3, 4.4, and 4.5. For example, Table 4.3 shows the discriminative alt-texts generated for female and male-owned pictures. In this table, $p(male|beard)$ represents the conditional probability that the target user is male if the generated alt-text for his picture contains *beard*. We define the alt-text f as a female alt-text if $p(female|f) > p(male|f)$, otherwise a male alt-text. To that end, except *beard* and *car*, the rest of alt-texts might be labeled as female alt-texts. Tables 4.4 (all female entries), and 4.5 (all male entries) consider the correlation of received emojis and alt-text tags.

4.4 Features

This section discusses our extracted features and then sketches our feature selection algorithm.

4.4.1 Feature extraction

Feature extraction is the process of identifying and extracting relevant features correlated to variables of interest (in our case, gender). The purpose of feature extraction is three-fold: promoting the model prediction performance, providing faster and efficient classifiers, and reducing the data dimensionality that decreases the complexity of the model. We extract features in four different categories, which consist of:

Emoji f	*MI*	*p(female|f)*	*p(male|f)*
❤️	0.026	**0.84**	0.16
😍	0.025	**0.89**	0.11
😘	0.015	**0.86**	0.14
💕	0.011	**0.91**	0.09
💙	0.008	**0.87**	0.13
🔥	0.008	**0.92**	0.08
😊	0.006	**0.82**	0.18
💖	0.006	**0.92**	0.08
💗	0.005	**0.91**	0.09
💜	0.005	**0.90**	0.10

Table 4.2: Discriminative emojis for female and male-owned picture

alt_ text f	*MI*	*p(female|f)*	*p(male|f)*
closeup	0.015	**0.80**	0.20
smiling	0.011	**0.75**	0.25
1 person	0.006	**0.70**	0.30
1 person smiling	0.006	**0.80**	0.20
smiling closeup	0.006	**0.89**	0.11
1 person closeup	0.004	**0.85**	0.15
beard	0.004	0.28	**0.72**
2 people smiling	0.004	**0.74**	0.26
car	0.003	0.28	**0.72**
selfie closeup	0.003	**0.82**	0.18

Table 4.3: Discriminative alt-text for female and male-owned picture

Emoji + alt_ text f	*MI*	$p(female\|f)$	$p(male\|f)$
1 person, 😍	0.044	**0.89**	0.11
1 person, ❤️	0.036	**0.87**	0.13
closeup, 😍	0.027	**0.93**	0.07
closeup, ❤️	0.021	**0.92**	0.08
1 person, 😘	0.020	**0.85**	0.15
1 person, 💕	0.019	**0.86**	0.14
1 person, 🔥	0.017	**0.93**	0.07
smiling, 😍	0.016	**0.93**	0.07
closeup, 😘	0.015	**0.92**	0.08
smiling, ❤️	0.013	**0.90**	0.10

Table 4.4: Discriminative emojis and alt-text for female-owned picture.

Emoji + alt_ text f	*MI*	*p(female|f)*	*p(male|f)*
beard, 😂	0.009	0.09	**0.91**
beard, ❤️	0.003	0.17	**0.83**
beard, 😘	0.002	0.11	**0.89**
hat, 😂	0.002	0.22	**0.78**
beard, 😊	0.002	0.08	**0.92**
outdoor, 😛	0.002	0.22	**0.78**
outdoor, 👍	0.002	0.37	**0.63**
smiling, 🍺	0.001	0.23	**0.77**
sky, 😂	0.001	0.40	**0.60**
standing, 😎	0.001	0.33	**0.67**

Table 4.5: Discriminative emojis and alt-text for male-owned picture.

1. N-grams. Tables 4.2, and 4.3 demonstrate that some emojis and alt-texts are discriminative between females and males. We employed n-grams on Facebook-generated alt-texts and comments to extract these discriminative elements. N-grams are a set of co-occurring words within a given window size (n). Optimum n-grams length depends on the data type. For example, if the size of n in n-grams is too short, it may fail to capture an important block of words. On the other hand, it may fail to capture the general knowledge if n is too long [Church and Hanks, 1990]. For terminology extraction, *Kenneth Church* proposes a window size of 5 ($n \leq 5$). According to *Church*, this size is a good compromise: it is large enough to show some semantic relationships between words, and it is not too large to lose the relationships that demand strict adjacency between words [Fkih and Omri, 2012]. In the emoji analysis, we consider *3-grams* and keep the result of *n-grams* that appear more than *50* times in total. With that, we collect *1626* features.

2. Patterns. The different emojis preferences of Facebook users in commenting on female or male pictures follow some patterns from which we derive features. We divide these patterns into non-textual and textual categories. The non-textual category contains comments with only emojis (no words), while the textual category contains words and emojis. Table 4.6 defines our five pattern-based features. We extract *158* features from this category.

3. Correlation of alt-text and emoji. Table 4.1 illustrates emojis preferences in commenting pictures are significantly different according to picture owner gender. Additionally, Tables 4.4 and 4.5 confirm the difference in alt-texts and emojis correlation for male and female posted pictures. These differences lead us to consider pairs of alt-text and emoji as features. To that end, we constructed the co-occurrence for emojis and alt-text. Note that we drop rare co-occurrence pairs that appear less than *50* times in all the pictures. Finally, we collect *1363* features from all the possible combinations of emojis and alt-text in our data set.

In total, we extract *3150* features from the above different categories. After extracting these features, we needed to apply feature selection algorithms to prune and reduce the extracted features.

Non-textual (without text)
Single emoji. For example, the comments of Figure 4.1(b).
Repeated same emoji. For example, the third comment of Figure 4.1(a).
Repeated different emojis. For example, the last comment of Figure 4.1(a).
Textual (with text)
Single emoji. For example, the first comment of Figure 4.1(a).
Single repeated emoji. For example, the forth and fifth comments of Figure 4.1(a).
Several repeated/non-repeated emojis. For example, the second comment of Figure 4.1 (a).

Table 4.6: Pattern-based features

4.4.2 Feature selection

The goal of feature selection is to downsample the selected features while keeping those that contribute more to predicting the attribute. The choice of feature selection methods differs based on the problem and available data. Below, we discuss four feature selection methods that we have employed: *Chi-Square*[22] is used to test if the relationship of a dependent variable is significant to an independent variable. *Information Gain*[23] indicates the amount of information the independent variable presents with respect to the classification target attribute. It measures the difference in information was available before and after knowing the attribute value. *Feature importance*[24] provides a score for each feature; the higher the score, the more critical or relevant the feature towards the output variable. *Feature importance* is an inbuilt class that proceeds with Tree-Based Classifiers. *Univariate feature selection*[25] examines each feature individually to determine the strength of the feature's relationship with the output variable. Different feature selection methods lead to several features that generate various prediction performances. To identify more representative features for better prediction, we evaluate all the possible individual and combined feature selection methods [Tsai and Hsiao, 2010, Lee, 2002]. Finally, we select the feature set that gives the best result.

4.5 Attack Evaluation

4.5.1 Dataset

Using a Python crawler, we collected a set of *141,812* pictures and their *446,655* messages. Our statistics showed that *1291* different emojis appear in our data set, and Facebook could not generate alt-text for *13000* pictures. We kept those pictures for our first attack scenario, where we purely relied on emojis.

4.5.2 Experimental Results

The experimental results are achieved by applying the classifiers from the Python library *scikit-learn*. We model gender inference attack as a binary classification problem. To achieve

[22] `https://scikit-learn.org/stable/modules/generated/sklearn.feature_selection.chi2.html`

[23] `https://www.bogotobogo.com/python/scikit-learn/scikt_machine_learning_Decision_Tree_Learning_Informatioin_Gain_IG_Impurity_Entropy_Gini_Classification_Error.php`

[24] `https://www.scikit-yb.org/en/latest/api/features/importances.html`

[25] `https://scikit-learn.org/stable/auto_examples/feature_selection/plot_feature_selection.html`

robust results, we apply several supervised machine learning algorithms such as *Logistic Regression*, *Random Forests*, *K-Nearest Neighbors*, *Support Vector Machine*, *Naive Bayes* and *Decision Tree*. We select the same number of males and females to evaluate the classifier to prevent biased classification. *Train-test* splitting was preferable as it runs k times faster than k-fold. To address the problem of fairly estimating the performance of each classifier and make sure the classifiers can generalize to unseen data, we split the training data set to *train*, *validation*, and *test* sets with the size of *60*, *25*, and *15*, percent of the training set respectively. We train the classifiers using *train* data set. Later, we record the performance on the *validation* data set and adjust the parameters to optimize the performance of the classifiers. Eventually, we evaluate the classifiers on the *test* data set. Considering the extracted gender as the ground truth, we compute the standard metrics such as *accuracy*, *precision*, *recall*, and *f1_ score* to evaluate our attack. In Tables 4.7 and 4.8, we report the results of the test data set for gender inference attack. We compare our six classifiers on three different scenarios in these two tables. Selecting the best machine learning algorithms based on evaluation metrics depends entirely on the problem. Since we are dealing with a balanced data set, we used *accuracy* to evaluate each classifier's performance.

Table 4.7 (left part) illustrates the result of our first scenario attack. As presented, *Logistic Regression* classifier outperforms other classifiers. According to the result, *Logistic Regression* can infer the target user gender with accuracy *76*%. *Logistic Regression* is a discriminative model which is appropriate to conduct when the dependent variable is binary. So, it learns better than other classifiers between the dependent and independent variables. We can observe that *Logistic Regression* performs better than other supervised classifiers based on *Accuracy*, *recall* and *f1-Score*. *Logistic Regression* also performs slightly worse than *Support Vector Machine* and *Naive Bayes* classifiers in *precision*. The result confirms that the preference emojis usage of commenters is enough to implement a machine learning classifier to infer the target user gender.

Table 4.7 (right part) displays our second scenario inference results. *Support Vector Machine* performs better than *Logistic Regression* in *accuracy*. Based on the result, an attacker can infer the target user's with an accuracy of *80*%. We can observe that *Logistic Regression* performs slightly better than *Support Vector Machine* in *precision* and *recall*. Moreover, *Support Vector Machine* performs slightly worse than *Logistic Regression* and *Naive Based* in *f1-Score*. *Support Vector Machine* is a discriminative classifier defined by a separating hyperplane. *Support Vector Machine* outputs an optimal hyperplane that classifies *females* from *males* better than other classifiers.

Table 4.8 gives the result of our third scenario. It shows that the *Logistic Regression* performs better than other classifiers in *accuracy*, *recall*, and *f1-Score*. *Naive Bayes* classifier received a slightly higher *precision* score than *Logistic Regression*. The *Logistic Regression* model, which had *76*% and *80*% accuracy in the first and second scenarios, repressively, obtains a *7*%, and *3*% accuracy boost in third scenario, which is a fairly substantial gain in accuracy. Table 4.8 shows the effect of the third scenario in increasing attack accuracy.

To conclude, *Logistic Regression* performs best in the first and third scenarios, and *Support Vector Machine* was the suitable classifier for the second scenario. These results confirm our hypothesis that the gender and contents of the picture impact commenters' reactions. Moreover, there is a substantial difference in receiving emojis between females and males posted pictures on Facebook. As a result, an attacker can train standard classifiers utilizing non-user generated

	Emojis				*alt_ text*			
	Accuracy	*Precision*	*Recall*	*Fscore*	*Accuracy*	*Precision*	*Recall*	*Fscore*
LR	**76.21**	81.57	71.54	76.22	79.16	82.19	77.74	79.90
KNN	68.38	73.08	64.49	68.49	73.54	76.34	72.93	74.59
SVM	76.09	83.35	68.93	75.45	**80.04**	82.07	75.71	78.76
NB	70.85	83.09	56.90	67.53	68.50	65.08	56.62	79.13
DT	65.16	69.50	61.78	65.41	69.61	71.64	71.12	71.38
RF	69.40	73.02	67.59	70.17	74.12	79.16	69.89	74.19

Table 4.7: Machine Learning classifiers performance with optimal hyper-parameters.

	Emojis + alt_ text			
	Accuracy	*Precision*	*Recall*	*Fscore*
LR	**83.89**	86.80	80.12	83.31
KNN	77.69	83.48	72.49	77.59
SVM	81.36	85.92	77.79	81.65
NB	79.29	86.90	72.00	78.75
DT	69.39	72.84	68.07	70.32
RF	76.72	81.08	73.48	77.09

Table 4.8: Machine Learning classifiers performance with optimal hyper-parameters.

data (commenters emoji preferences and generated alt-text) to infer the picture owner's gender.

4.6 Discussion

Based on our analysis, the best scenario for the attacker is the third scenario, when he has access to Facebook generated alt-text and commenters posted emojis. The second scenario is suitable when the crawled pictures contain only alt-text and no emoji(s) commented by target friends, friends of friends, or ordinary users. In this case, the generated alt-text can help the attacker infer the target user's gender. The first scenario is applicable when the Facebook algorithm, in rare cases, is unable to generate alt-text for the target user pictures. In this case, the attacker has the advantage of using emojis, the universal language, and launching the gender inference attack.

4.7 Conclusions

In this chapter, based on the intensive analyses of the shared images, we have demonstrated a new perspective of gender inference attacks on Facebook users by relying on non target-user generated data. We have shown the possibility of gender inference attack even when all user attributes/activities are hidden, such as profile attributes, friend lists, liked pages, and joined groups. Moreover, our experimental results have revealed that, on average, female posted pictures receive more emojis-based comments than male posted pictures. Additionally, we have manifested alt-text gives extra free information that boosts inference accuracy. We have noticed that commenters use more emotional emojis to comment on female images with a particular

theme and setting. In the next chapter, we introduce an online inference attack and consider an alternative situation where comments contain both words and emojis or only words in the inference process. The model can meet new words not observed in the training phase (called OOVs) in online mode. We will introduce a new vector adaptation process to handle these previously unseen words. We expect that the combination of words and emojis helps to increase the inference attack accuracy.

Chapter 5

Online Attack

Contents

5.1 Introduction

Although emojis are universally adopted for expressing emotions, in some situations, commenters only use words when commenting underneath pictures. For example, aged people might have difficulty utilizing emojis. Moreover, we can achieve more accurate predictions by inferring from emojis and words. This chapter presents an online inference attack using words and emojis. One must consider observing new terms inside the picture metadata that have not been processed in the training phase in an online attack. Therefore, the model has to be able to handle new words. Here we introduce a new approach to handling these unseen words online.

Attribute inference attacks can be significantly improved by Natural Language Processing (*NLP*), where one can capture semantic relations between words or emojis from their vectorial representations. Considering this, we propose a method to infer Facebook users' gender *online*

through their shared images along with Facebook generated alt-text and received comments underneath those pictures. Our method even applies to Facebook users who are cautious about their privacy and hide any type of available information (e.g., friend list, liked pages, groups, and attributes) on their profile. In online attacks, the attacker leverages offline analytics knowledge to predict new attributes of target users using their input data collected online. The attacker constructs the offline analysis knowledge by collecting profiles with known attributes (in our case, gender) and employs sophisticated techniques (e.g., *NLP*) to capture patterns and structures from collected data.

Our training dataset only contains *25,456* unique words. Therefore, the input data of an online attack may contain words that do not occur in this training dataset. The new words are called *out of vocabulary words* (*OOV*). To circumvent this problem, we rely on the pre-trained vectors of an advanced *NLP* model, namely *word2vec*[Mikolov et al., 2013a], and its version dedicated to emojis *emoji2vec* [Eisner et al., 2016]. *Word2vec* and *emoji2vec*, abbreviated by *WE2V*, are trained on large datasets (e.g., Wikipedia for *word2vec*) with specific writing structure or usage. Therefore their pre-trained vectors should be adapted to apply to a specific domain such as Facebook. *Retrofitting* technique [Faruqui et al., 2014] is called for adjusting the *WE2V* pre-trained vectors by combining external knowledge (*WE2V* dataset), and internal knowledge (offline collected words/emojis co-occurrence). A straightforward approach for handling an out of vocabulary word would be to replace it with a synonym. A synonym of a word has precisely or nearly the same meaning. However, this approach fails for our gender inference problem, as the word and its synonym can orient to different genders. An example taken from our dataset illustrates this point: male-posted pictures receive more comments containing the word *gorgeous*, while a synonym of this word, namely *beautiful*, is used more frequently for commenting female posted pictures. Instead of synonymy, we apply *cosine similarity* score [Mikolov et al., 2013a]. *Cosine similarity* is based on vector representation. Two words/emojis have similar vector representation if they appear in similar contexts, even though they are not synonyms (e.g., hot and cold). Accordingly, cosine similarity can capture words/emojis similarities more accurately for our inference attacks.

As for the rest of the chapter, we define the gender inference attack in Section 5.2. Next, we overview our system architecture to perform online attacks in Section 5.3. Section 5.4 presents in detail the offline attack steps, and Section A.2.3 describes the steps of an online attack. We show our experimental results in Section 5.6. Finally, we discuss the attack process and conclude this chapter in Section 5.7.

5.2 Attack Description

In this work, we consider an attacker who intends to infer a picture owner gender g by observing a set of pictures where each published picture is endowed with metadata (a set of comments and generated alt-text). The attacker can be anyone who can crawl data from Facebook. We reuse our three scenarios to infer the target user's gender accurately (see the paragraph in Section 4.1). In the first scenario, comments are unavailable (e.g., privacy reasons). For example, the target user conceals all comments underneath the picture. In the second scenario, Facebook cannot generate automatic alt-texts due to the quality of the posted picture. In the third scenario, both comments and alt-text are publicly available. The attacker can leverage Facebook users' words/emojis usage and/or alt-text to infer the target gender.

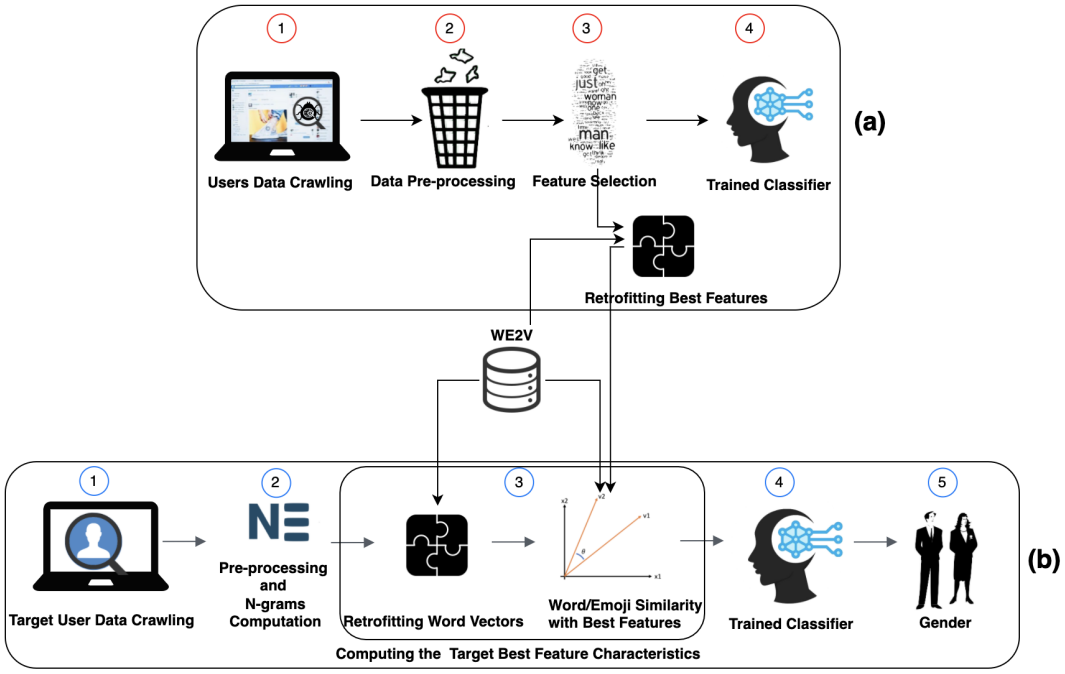


Figure 5.1: Training in (a) offline and (b) online.

Let f_u^i be the number of occurrences of feature i in user pictures metadata. Each user u with a set of pictures can be represented by a feature set $f_u = \{f_u^1, f_u^2, .., f_u^n\}$, and the label (or class) $g_u \in 0, 1$ corresponds to their gender. In the offline mode, the attacker trains machine learning algorithms with samples (f_u, g_u), for all $u \in U_{training}$ as inputs, where $U_{training}$ is a set of users. In the online mode, the attacker carries out the attack on a chosen target user u_{new} by leveraging the features obtained from the trained algorithm and data extracted from the online phase. We discussed feature selection and extraction techniques in Section 4.4.

5.3 Architecture

Figures 5.1 depict the overall architecture of our system. First, we overview the offline training components, and next, we present our online attack ingredients.

5.3.1 Offline training

This procedure combines domain-specific and external knowledge in the following way (see Figure 5.1 (a)). *Users Data Crawling* collects Facebook users' data offline for training gender classifiers. Then *Data Pre-processing* prunes, cleanses and normalizes the collected data. *Feature Selection* derives a set of features that contribute the most to gender inference from an initial set obtained by n-grams and correlation of alt-text and comments. In *Trained Classifier* step, we aim to select the best gender classifier among the ones that we have trained, using standard evaluation metrics. *Retrofitting Best Features* is the process of adjusting an existing word/emoji vector representation. It allows us to fit *WE2V* word vector representations to our specific domain, namely Facebook. Later, we discuss in detail all the steps.

5.3.2 Online attack

After training the machine learning classifiers, the gender inference attack test is performed online (see Figure 5.1 (b)). *Target User Data Crawling* collects target user data in an online mode. *Pre-processing and N-grams Computation* prune raw data and extract terms composed of words/emojis or sequence of words/emojis. *Computing the Target Best Feature Characteristics* first finds if the extracted terms are features or can be considered similar to features for the machine learning algorithms. The similarity is measured by comparing the word vector representations of n-grams obtained using *WE2V* (and retrofitting to take into account new words). The target features (weighted by their number of occurrences) constitute what we call the target feature characteristics. The *trained gender classifier* will receive as input the target features characteristics.

5.4 Offline Training

This section introduces and discusses the offline components to train our machine learning algorithms. We follow the pre-processing steps introduced in Sections 6.5.2 and 4.2 to clean comments and alt-texts. We select features by proceeding in three steps:

1. Features extraction. Given a window size n, we calculate for each gender a set F_0 of n-grams to capture occurrences of words / emojis in comments and tags in alt-text. Next, we construct a co-occurrence matrix to find the correlation between gender and metadata to distinguish females from males. This allows the inference process to benefit from some co-occurring words, emojis, and tags related to the same picture, even when they are not in the same window of size n. For that, we consider couples of terms w, w' that are components of (possibly different) n-grams in F_0. We compute the most frequent ones from these couples, given an experimentally defined threshold, which are subsequently added as new features to F_0 to result in the set F_1.

2. Mutual information. We prune F_1 by computing the Mutual Information (MI) of its elements with gender. MI is a statistical measure of (in)dependence between random variables [Cover, 1999]. Here MI measures the mutual dependence between pictures owner gender and picture metadata, which is helpful in feature selection [Vergara and Estévez, 2014]. Terms with higher MI are more informative in distinguishing females from males. This pruning process results in the set F_2. Table 5.1 presents the top *10* discriminative correlations of words and alt-text tags for male and female posted pictures. In this Table, *1 person gorgeous*, and *closeup gorgeous* can be considered male features as *p(male|f)> p(female|f)*. Moreover, *1 person beautiful* intensely used for female posted pictures (*p(female|f)*=**0.94**). For more examples, see Tables 4.2, 4.3, 4.4, and 4.5 presented in Section 4.4.2.

3. Best features set. Finally, we apply *Chi-Square*, *Information Gain*, *Feature importance*, and *Univariate feature selection* to F_2 in order to generate a set of best features, called F_{best}. We preserve the features considered important by all feature selection algorithms. F_{best} contains *1148* features (single or combination of words/emojis).

5.4.1 Retrofitting words/emojis vectors

After selecting the best feature set, we compute vector representations of these features to evaluate the similarity of the online collected words from the target profile (which may contain

Feature f	*MI*	*p(female|f)*	*p(male|f)*
1 person beautiful	0.042	**0.94**	0.06
closeup beautiful	0.030	**0.92**	0.08
1 person gorgeous	0.018	0.39	**0.61**
smiling beautiful	0.017	**0.87**	0.12
1 person pretty	0.016	**0.89**	0.11
closeup gorgeous	0.011	0.42	**0.58**
closeup pretty	0.011	**0.90**	0.10
smiling pretty	0.007	**0.89**	0.11
selfie beautiful	0.006	**0.85**	0.15
1 person cute	0.005	**0.80**	0.20

Table 5.1: MI result: correlation of alt-text and words.

new words or sequences of words) to our best feature set. To that end we utilize *word2vec* [Mikolov et al., 2013a] and *emoji2vec* [Eisner et al., 2016]. Our goal is to create a set of embeddings that accounts for both our offline collected dataset, *OCD*, and original word/emoji representations learned from *WE2V*. *Retrofitting* [Faruqui et al., 2014] is a process that adjusts an original word vector separately using a knowledge graph (e.g., *WordNet* [Miller, 1995]), in our case *OCD* instead. Retrofitting has the advantages of being (i) a post-processing operation that does not require browsing the corpus again, (ii) applicable to any vector model, and (iii) simple and fast to implement. Retrofitting computes a new vector $\vec{v_i}$ for the feature $w_i \in F_{best}$, with the objective of being close to w_i's original vector $\vec{v_i'}$, when it exists, and also to vectors $\vec{v_j}$ representing w_j that are the w_i's nearest words/emojis in *WE2V* $\bigcap$ *OCD*. We used the same norm ($L2$) defined in the original paper [Faruqui et al., 2014], and we try to minimize this objective function:

$$\sum_{i=1}^{n}\left[\alpha_i\left\|\vec{v_i}-\vec{v_i'}\right\|^2+\sum_{j:w_j\in WE2V}\gamma_{ij}\left\|\vec{v_i}-\vec{v_j}\right\|^2+\sum_{j:w_j\in OCD\cap WE2V}\beta_{ij}\left\|\vec{v_i}-\vec{v_j}\right\|^2\right] \quad (5.1)$$

We set $\alpha_i = 1$ when $w_i \in$ *WE2V* and 0 otherwise. The distance between two vectors is defined to be the Euclidean distance. For w_i in *WE2V* $\setminus$ *OCD* we take $\beta_{ij} = 0$ and γ_{ij} is the *Cosine Similarity* score between $\vec{v_i}$ and nearest vectors $\vec{v_j}$ of words/emojis w_j in *WE2V* dataset. For w_i in *OCD*, we take $\gamma_{ij} = 0$ and β_{ij} is the *positive pointwise mutual information* (*PPMI*) score [Niwa and Nitta, 1995] between w_i, and co-occurring words w_j. *PPMI* has been extensively used in the field of *NLP* to measure words closeness based on their co-occurrence probability. *PPMI* is formulated as follows:

$$\beta_{ij} = PPMI(w_i, w_j) = max(PMI(w_i, w_j), 0)$$

$$PMI = \log \frac{pr(w_i, w_j)}{pr(w_i)p(w_j)}$$

where $pr(w_i)$, and $pr(w_j)$ represent the marginal probabilities that a comment contains w_i, or w_j, and $pr(w_i, w_j)$ represents the joint probability that a comment contains both w_i and w_j).

Therefore, we calculate the vector $\vec{v_i}$ from the nearest words/emojis vectors $\vec{v_j}$, where $w_j \in OCD \bigcap WE2V$ considering their cosine similarity γ_{ij}, or *positive pointwise mutual information* β_{ij} score according to the cases as follows:

$$\vec{v_i} = \frac{\sum_{j:w_j \in WE2V} \gamma_{ij}\vec{v_j} + \sum_{j:w_j \in OCD \cap WE2V} \beta_{ij}\vec{v_j} + \alpha_i \vec{v_i'}}{\sum_{j:w_j \in WE2V} \gamma_{ij} + \sum_{j:w_j \in OCD} \beta_{ij} + \alpha_i} \tag{5.2}$$

The advantage of adjusting the pre-trained words/emojis vector applying offline extracted data co-occurrences is two-fold: (i) handling $w_i \in WE2V \setminus OCD$ or $w_i \in OCD \setminus WE2V$ easily and (ii) using sophisticated distributional embeddings ($WE2V$) make the retrofitted vectors robust and suitable for gender inference attacks. For the feasibility of the computation, we truncate each sum in Equation 5.2 by summing only the *10* most significative terms (corresponding to the closest words to w_i). In the case of having a sequence of words as the best feature, we retrofit each word vector separately. Then we take the average of the vectors associated with the words in the sequence [Pagliardini et al., 2017]. For example, consider *nice picture* as a best feature, we retrofit *nice* and *picture* separately. Next, we get a vector for *nice picture* by averaging their retrofitted vectors. Figure 5.2 illustrates the particular word retrofitting, where the blue dots are *word2vec* vectors, and green dots are the retrofitted vectors.

5.5 Online Attack

The online phase categorizes a target user of unknown gender as male or female. It proceeds by the following steps.

5.5.1 Pre-processing and n-grams computation

We compute the n-grams, $n \in \{1, 2, 3\}$ with their frequency in the target picture metadata. The obtained set of n-grams is called J_t. We denote by $o(j)$ the number of occurrences of an n-gram $j \in J_t$ in target pictures metadata.

5.5.2 Computing the target best feature characteristics

Then, we compute a table for a map *co* that associates to each feature $f \in F_{best}$ an integer value $co(f)$. This value reflects the importance of feature f in the target picture metadata. All entries of the table are initialized by zero. Below we give an iterative algorithm to compute the table for *co* when all $j \in J_t$ are 1-grams (the general case is explained afterwards). The idea is that if an n-gram $j \in J_t$ is a feature in F_{best}, then the target characteristics value for this feature is incremented by $o(j)$ and if j is not in F_{best} then we select the feature $f \in F_{best}$ that is the closest to j, and we increment the target characteristics value for f by $o(j)$. The different cases are detailed in Algorithm 1 below. When j is a 2-gram or a 3-gram, we proceed as in Section A.2.2.

5.5.3 Gender classification

In this step the attacker applies the trained machine learning algorithm to the target user characteristics computed in previous Subsection A.2.3. Given a target user, the algorithm outputs either *female* or *male*. The output depends on the prediction probability. For example, the output is *female* if the *female* class receives higher prediction probability. In our experiments (see Section 5.6), we present the result of *700* users as they are labelled *female*, or *male*.

Input: Target picture metadata J_t, Best features F_{best}
Output: Target best features table co

for $j \in J_t$ **do**
  if $j \in F_{best}$ **then**
    $co(j) \leftarrow co(j) + o(j)$;
  else
    if $j \in (OCD \cup WE2V)$ **then**
      compute $\vec{j}$ using Equation 5.2;
    else
      compute the vector $\vec{j}$ as the average of all $\vec{w}$ such that $w \in OCD \cup WE2V$ and j and w appear in the same comment;
    end
    $f \leftarrow \underset{f' \in F_{best} \text{ s.t. } cosine(j,f')>0.5}{\arg\max} cosine(\vec{j}, \vec{f'})$;
    $co(f) \leftarrow co(f) + o(j)$;
  end
end

Algorithm 1: Target best features characteristics

5.6 Experiments

This section evaluates our approach for three scenarios defined in Section 5.2 and demonstrates offline and online experiments.

5.6.1 Offline experiments

Using a Python crawler, we have randomly collected *627,776* pictures and their *1,332,219* comments. Facebook was unable to generate alt-text for *24833* pictures. We have kept those pictures for our second attack scenario, where we rely only on words/emojis usage for commenting pictures. 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 therefore we committed to keep them in secure storage and only for the necessary time to achieve this work. We used the same experiment settings as presented in Section 4.5. To evaluate our attack, we compute the *AUC-ROC* curve, which is a performance measurement for classification problems at various threshold settings. In Figure A.1, we show the *AUC-ROC* results for all three scenarios. Figure A.1(a) displays trained algorithms results on the first scenario. Based on that, our trained algorithms can infer the target user gender with an *AUC* of *87%*. As illustrated in Figure A.1(b), the performance increases to *90% AUC* in the second scenario. Based on the Figure A.1(c), *Logistic Regression* model which had *87% AUC* in the first scenario, and *90%* in the second scenario gets *5%*, and *2% AUC* boost in the third scenario, respectively, which is a fairly substantial gain in performance.

To conclude, *Logistic Regression* performs the best in all scenarios. It is a discriminative model that is appropriate when the dependent variable is binary (i.e., two classes). The results confirm our hypothesis that gender and picture contents impact commenters' reactions. As a result, an attacker can train standard classifiers utilizing pictures metadata: (i) commenters' words/emojis preferences, and (ii) generated alt-text to infer the picture owner's gender. Note,

as we rely solely on non-target generated data, the results cannot be compared to previous works that exploit data published by the target user.

5.6.2 Online experiments

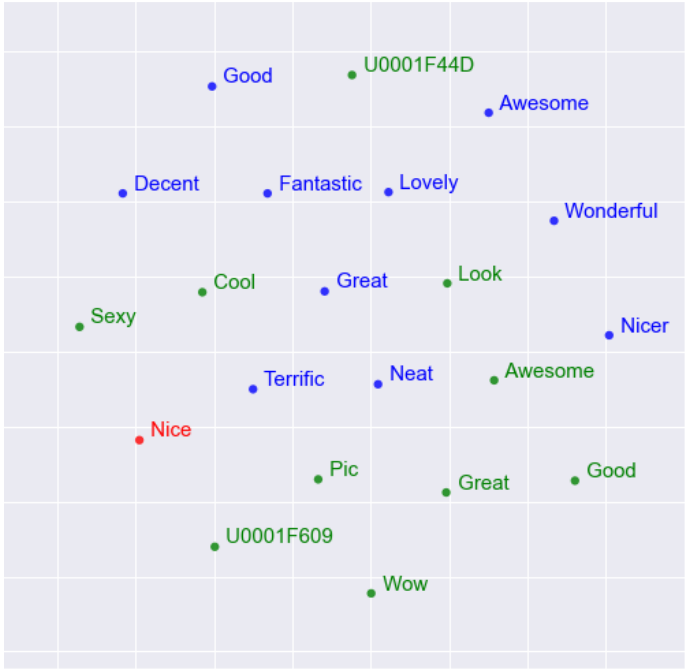
We have applied our online experiment to *700* users with their *21,713* pictures and their *64,940* corresponding comments. We have evaluated the performance of each classifier with *AUC-ROC*. As illustrated in Figure A.1 (d,e, and f), *Logistic Regression* outperforms other classifiers in all three scenarios. Notably, the combination of alt-text and words/emojis boosts the performance of classifiers compared to the other scenarios. To sum up, *Logistic Regression* is a suitable classifier for this task that an attacker can train to perform a gender inference attack in online mode.

5.7 Discussion

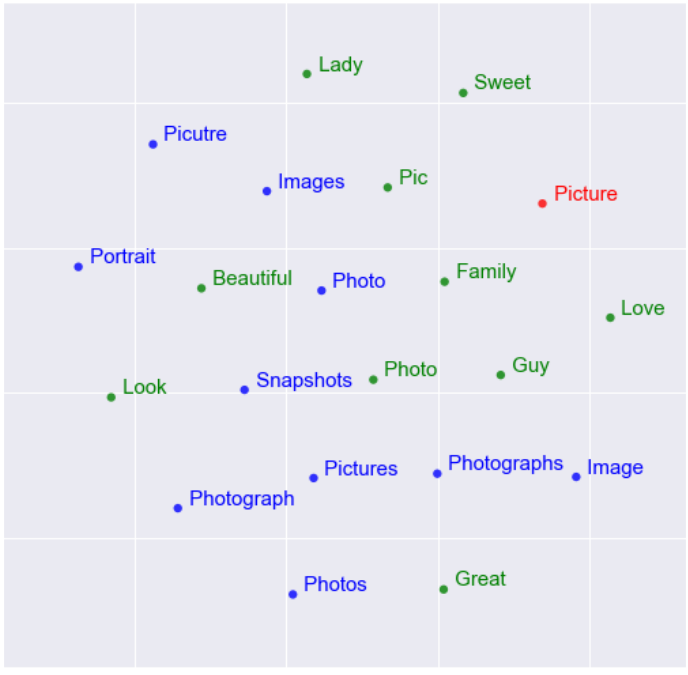
The third scenario is the most informative, given that the attacker manipulates both comments and alt text. By applying our system to his published pictures, a Facebook user can check if he/she is vulnerable to gender inference attacks of the above type. The attack may work even when the target publishes nothing else than pictures. We offer two countermeasures to mitigate the aforementioned privacy violations: (i) hide some comments or (ii) hide some pictures when they strongly contribute to the attack, as explained in the next chapter.

5.8 Conclusions

In this chapter, we have presented an online attribute inference attack by leveraging picture metadata. We have explored how to launch an online gender inference attack on any Facebook user by handling newly discovered online vocabulary using the retrofitting process to enrich a core vocabulary built during offline training. Retrofitting is fast to implement and is a post-processing operation that does not require browsing the corpus again. We performed our attack in three different online and offline scenarios and represented the result. The results confirm our hypothesis that commenters react differently to female and male-owned pictures. This variation also can come from the fact that commenters use different contexts for the same term while commenting to picture owners with different attribute values. The next chapter introduces a model to capture these variations in commenters' writing styles and generate distinct vectors for the same terms used in different attribute values.

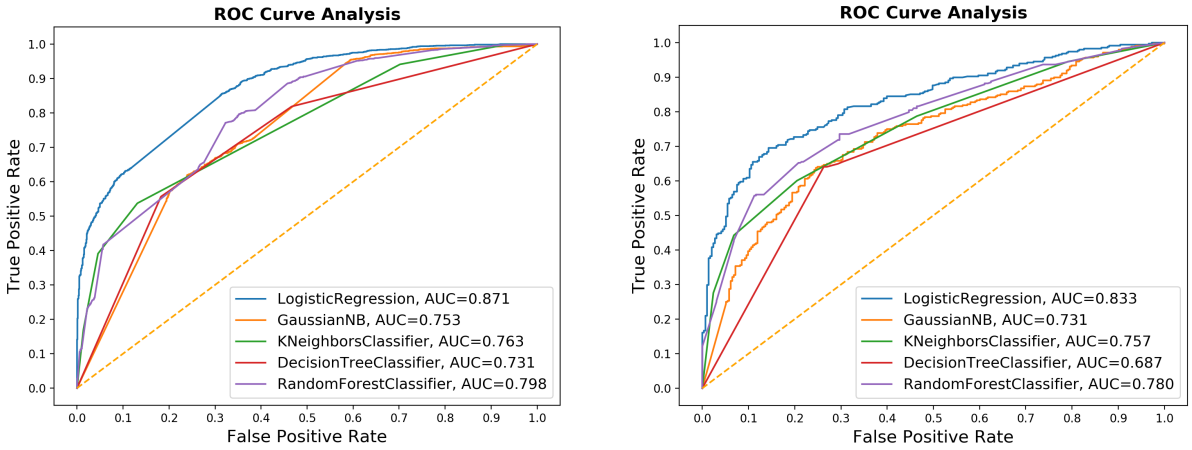


(a)



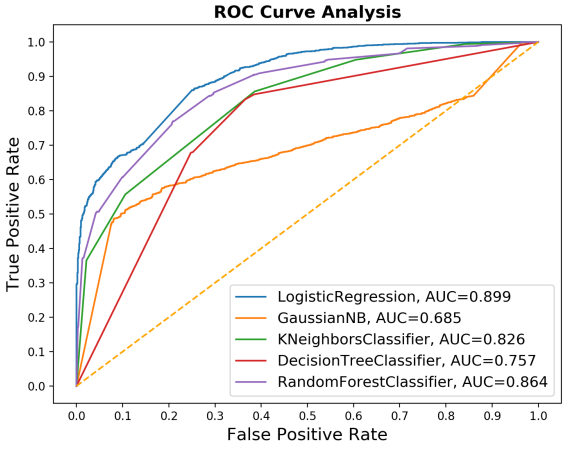
(b)

Figure 5.2: Retrofitting : (a) nice, (b) picture.

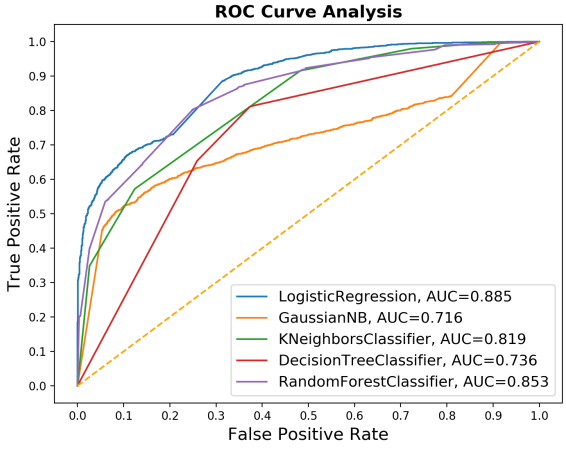


(a)

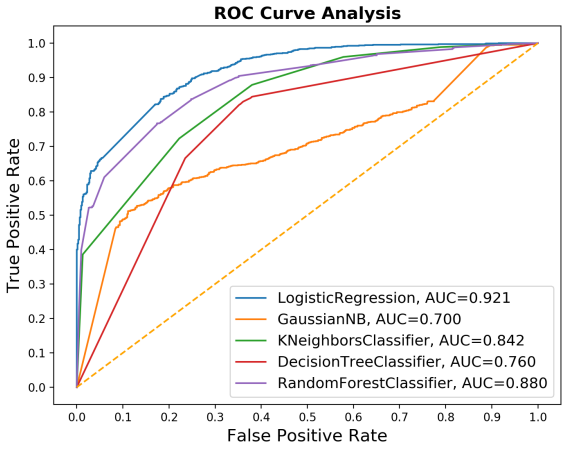
(b)



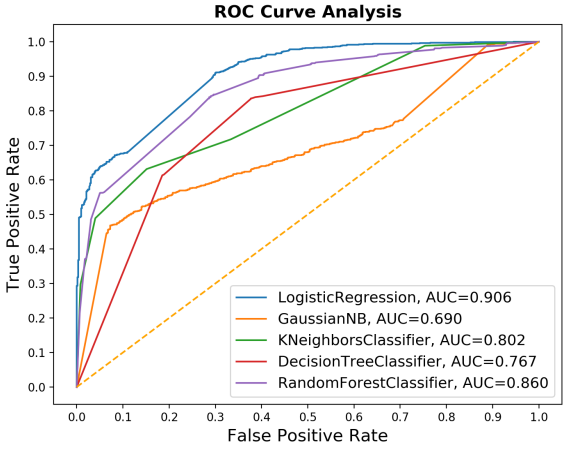
(c)



(d)



(e)



(f)

Figure 5.3: AUC result of logistic regression: trained on (a) first (b) second (c) third scenario features, and trained on removed (d) first (e) second (f) third scenario features.

Chapter 6

Vector Representation for Attribute Inference Attacks

Contents

6.1 Introduction

This chapter is dedicated to generating distinct vectors for the same terms that appear in different attribute values. We use *term* as a shorthand for word/emoji/alt-text tags to ease the reading. An important problem when learning attributes from picture metadata is that commenters often use different contexts for the same term while commenting to picture owners with different attribute values. Such a term will be called an *overlapped* term, and the value-specific contexts will be called *non-overlapped* contexts. We consider a term an overlapped term if it appears in all attribute values. Example 1 shows female and male-owned pictures with generated alt-text to describe the picture content and comments posted by commenters.

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* 😆

The tag *1person* and the word *baby* are overlapped terms, while *you*, *miss*, and *cray* are non-overlapped contexts. The commenters employ different neighboring terms for the word *baby* when commenting on female and male-owned pictures, which demonstrates the commenters' usage preferences. Therefore, there is a variation in the style and usage of the same term in different attribute values. To discover these variations, we apply a semantic space model, known as Distributional Semantic Model (DSM). Classical word embeddings (such as word2vec [Mikolov et al., 2013a] or GloVe [Pennington et al., 2014a]) 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-occurrences in the entire dataset. In Example 1, *baby* is used as a romantic word 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* contexts 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 [Kutuzov et al., 2018].

We apply Random Indexing (RI) [Sahlgren, 2005], an incremental and scalable method for constructing a vector space model to avoid this problem. RI requires few computational resources for similarity computations and allows comparison of word spaces created over different attribute values [Basile et al., 2015]. Building on RI, our approach creates a semantic representation such that the vector generated for a term in the context of an attribute value does not depend on the other attribute values. Accordingly, we can generate vectors for different attribute values and compare them from the same term. To illustrate our approach, we plot the closest context terms to the word *hair* computed with word2vec and our embedding model in Figure 6.1 (a, b, and c), respectively. Word2vec considers the entire dataset to generate a single vector for the word *hair* (Figure 6.1 (a)) by aggregating co-occurrences found for different attribute values. As an illustrative example with relationship status, Figure 6.2 shows that different co-occurrences of the tag *indoor* are obtained by separating the attribute values, namely *single*, *married*, and *engaged*. Figure 6.2 (a) shows the co-occurring words captured by word2vec, while Figure 6.2 (b,c, and d) represent our co-occurrence extraction for the same tag in different relationship attribute values.

Our embedding model clearly preserves different context terms appearing with the word *hair* in male and female users' pictures respectively (Figure 6.1 (b,c)), and alt-text tag *indoor* in *single*, *married*, and *engaged* users Figure 6.2 (b,c, and d). We can leverage this context

Figure 6.1: Nearest terms to word *hair* with (a) word2vec, and with our approach (b) female, and (c) male.

distinguishability property to infer users' attributes.

This chapter conducts an intensive analysis of three Facebook attributes: age, gender, and relationship status, as they are recognized as key privacy concerns in the internet era. Our attacks receive promising results as we preserve the vector representation of each word for each attribute value.

The rest of the chapter is organized as follows: Section 6.2 presents our Divide-and-Learn methodology to incorporate attribute values in word vector generation. We define our value-

based random indexing in Section 6.3. We outline our attribute inference attack steps in Section 6.4. Finally, we present the comparison of our model with other models in Section 6.5.3 to evaluate our attacks.

Figure 6.2: Nearest terms to the alt-text tag *indoor* in relationship status attribute using (a) word2vec, our approach (b) single (c) married, and (d) engaged.

6.2 Divide-and-Learn Methodology

This section first explains how the training dataset is divided into sub-datasets according to the different attribute values to be inferred. Next, we introduce the distributional semantic space and the proposed value-based random indexing approach. The notations used in this chapter are summarized in Table 6.1.

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
$\vec{w}$	distributional vector of w
c	context
$C(w)$	set of contexts of w in D
$P(u)$	set of pictures of $u \in U$
$attr$	an attribute
$attr_m$	mth value of $attr$
U_m^{attr}, U_m	set of users s.t. attribute $attr$ has mth value (U_m when $attr$ is implicit)
W_m	set of words in comments of pictures from users in U_m
$C_m(w)$	set of contexts of w in D_m
$\vec{u}$	vector for u
D_i	is the ith sub-dataset containing the set of users with attribute value $attr_i$

Table 6.1: Notations.

6.2.1 Dividing training datasets

Here, we introduce our splitting conditions and their computation.

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' pictures in different attribute values, or (ii) more frequently the same terms co-occurring more often with similar contexts than dissimilar ones (see Section 6.5.6). 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 a female user
Generated alt-text: *2people*
Comment: *Wooooooow, its NICE*
Metadata of an image published by a 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 Section 6.5.6). For instance, if males and females are commented with

the same terms and the contexts of those terms are mainly specific to an attribute value (see Example 1), we can take advantage of this variation and split the dataset to generate vectors oriented towards that attribute value. Therefore, we propose two conditions to be jointly satisfied to split the training dataset. To express these conditions, we define the importance of a set L that contains terms or contexts w as follows:

$$Q_D(L) = \sum_{w \in L} freq_D(w) \tag{6.1}$$

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

Condition 1: Evaluating the importance of overlapped and non-overlapped terms. Our first condition checks the existence and importance of overlapped and non-overlapped terms. It works by running *Unigram* and extracting words/emojis/tags frequency in each attribute value. Next, we collect the top N frequent *Unigram* of the individual attribute value and find the overlapped and non-overlapped terms, which we define as OT and NT, respectively. The first condition is satisfied if

$$Q(OT) > Q(NT)$$

Condition 2: Evaluating the importance of non-overlapped contexts. The same term can appear in distinct contexts (see Example 1). The goal of our second condition is to examine the importance of similar or dissimilar contexts. However, we do not need to split the dataset if overlapped terms co-occur with similar terms in different attribute values (see Example 3). To check this condition, we apply Bigram and assemble words/emojis/tags sequences in each attribute value to capture the preceding and following co-occurring terms of overlapped terms. Similar to the first condition, we obtain the overlapped and non-overlapped contexts that we define as OC and NC, and compute their weights (see Equation 6.1). The second condition holds if

$$Q(OC) < Q(NC)$$

Using Bigram is an arbitrary choice that can be adjusted to *3-grams*, *4-grams*, or more. 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.

6.2.2 Dataset dividing algorithm

We label the original training dataset D so that the ith sub-datasets D_i contains users labeled with the ith attribute value and terms appearing in their pictures comments. For example, for gender attribute, $attr_1$ (resp,. $attr_2$) represent male (resp,. female). We obtain D_1 and D_2 as sub-datasets annotated by male and female. W_1, and W_2 are words in comments of pictures published by males (U_1) and females (U_2). Algorithm 2 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.

Given two integer parameters UTop and BTop, we use the following functions:

1. $Unigram(D_i, UTop)$ computes Uni_i, the set of $UTop$ most frequent terms in D_i.
2. $Bigram(Uni_i, BTop)$ computes Big_i, the set of $BTop$ most frequent terms in D_i co-occurring with terms in Uni_i.
3. $Tcount(t) = |\{i \in \{1, \dots k\} \mid t \in Uni_i\}|$, (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.

Input : $D, D_1, \dots, D_k$
Output: true iff the dataset D has to be split in $D_1, \dots, D_k$

Step1:
for $i = 1, \dots k$ **do**
  $Uni_i \leftarrow Unigram(D_i, UTop)$
  $Big_i \leftarrow Bigram(Uni_i, BTop)$
end
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;

Algorithm 2: Dataset dividing algorithm

Note that suitable values of $UTop$ and $BTop$ will be determined from experiments (see Section 6.5.3).

6.3 Random Indexing

Random Indexing (RI) is a fast dimensionality reduction method that transforms high-dimensional data into a lower-dimensional one applying a random matrix. RI assigns a randomly generated vector to each unique term in the text, so-called index vector [Sahlgren, 2005]. These index vectors are sparse, n-dimensional, ternary, and contain a small number of randomly distributed non-zero elements $\{-1, +1\}$. Additionally, an n-dimensional initially empty vector represents the so-called distribution vector of each unique term. RI incrementally updates the n-dimensional distribution vector of each term by summing the n-dimensional index vector(s) of all co-occurring terms within a small window of text. As a result, terms appearing in a similar context tend to have a similar distributional vector. Let $c = [c_{-n}, \dots c_{-1}, w, c_1, \dots c_n]$ be the context of w with window from $-n$ to n (n chosen between 1 and 5) and let $\vec{c}$ be the vector obtained by accumulating word's index vector co-occurring with w in context c. We update the distributional vector $\vec{w}$ by using RI as follows:

$$\vec{w} = \sum_{c \in C(w)} \sum_{\substack{-n \leqslant j \leqslant n \\ j \neq 0}} \vec{c_j} \tag{6.2}$$

RI leverages the entire training set to generate a vector for each term. Similar to word2vec or GloVe, RI builds a global view of terms co-occurrences in the entire dataset. As a result, it

generates vectors for terms using the entire dataset that can mix and combine many possible word contexts (see Example 1). 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 by splitting the dataset using our proposed conditions (see 6.2.1).

6.3.1 Values-based random indexing

Given a set of users U and an attribute l with k values $attr_1, attr_2, \ldots, attr_k$, 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. W is the vocabulary of all comments for pictures of users in U, and 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 user profiles with different attribute values. It is a crucial 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 $\vec{w}$ from the entire W, we compute a vector $\vec{w}$ for each term w_m derived from W_m as follows:

$$\vec{w}_m = \sum_{c \in C_m(w)} \sum_{\substack{-n \leqslant j \leqslant n \\ j \neq 0}} \vec{c}_j \tag{6.3}$$

Using Equation 6.3 we generate k distinct vectors ($m \in 1, ..., k$) for the same term w appearing with different attribute values. Previous word embedding approaches (e.g., word2vec or GloVe) 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 6.3 to compute several vector representations for the same term, each one corresponding to an attribute value.

6.3.2 Generating index vectors

RI relies on two major hypotheses. First, in high dimensional space, a much larger number of almost orthogonal than orthogonal directions exist, according to Hecht-Nielsen [Hecht-Nielsen et al., 1994]. 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 [Lindenstrauss, 1984]). The choice of random matrix is essential 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 Section 6.5.3). Following the steps mentioned above, our approach provides suitable vectors to perform attribute inference with higher accuracy than alternative embeddings.

6.4 Attribute Inference Attacks

We consider an attacker who intends to infer OSN users' attributes from a set of pictures P where each picture contains metadata. Once we learn the vector representations of terms for each attribute value (see Equation 6.3), we compute a vector representation of a labeled user

u_m by aggregating all the terms that appear in his/her pictures as follows:

$$\vec{u}_m = \sum_{w \in P(u_m)} \vec{w}_m \tag{6.4}$$

We introduce a set S of vectors computed by Equation 6.4. For the target user t, we generate a set of user vectors $T = \{\vec{t_1}, \vec{t_2}, ...\vec{t_k}\}$, where vector $\vec{t}_m$ is obtained as follows:

$$\vec{t}_m = \sum_{w \in P(t)} \vec{w}_m \tag{6.5}$$

We compute *cosine similarity* [Mikolov et al., 2013a, Pennington et al., 2014b] 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:

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

The *arg max* function outputs a tuple (t_μ, u_μ) and we infer that the target attribute value is $attr_\mu$. Cosine similarity is a popular similarity measure for vectors when their magnitude is not relevant (e.g., they represent linguistic items in distributional semantics [Levy and Goldberg, 2014b]).

We consider a gender inference attack against a user, named *Target*, as a concrete example. Given two gender values $attr_1$ ="male" and $attr_2$ = "female", we generate two vectors $\vec{w}_1$ and $\vec{w}_2$ for *Target* where $\vec{w}_1$ and $\vec{w}_2$ correspond to vector of w in D_1 and D_2, respectively. Vectors *Target_F* and *Target_M* (red points in Figure 6.3) represent female and male hypotheses vectors, respectively. Using cosine similarity, we compute in Figure 6.3 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*.

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 using Algorithm 2. Next, value-based random indexing from Section 6.3.1 is applied to generate word vectors that are meaningful for each attribute value. Finally, attribute inference is achieved by following the above steps.

6.5 Case study: Facebook

This section applies 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 2. Next, we assess and compare our approach with word2vec, a state-of-the-art method for representing language semantics [Mikolov et al., 2013a].

6.5.1 Dataset

As a case study, we concentrate on Facebook. More precisely, we consider photos published by their owners. Compared to other publications (such as posts and status updates), pictures on Facebook receive an additional comment from Facebook, namely alt-text. The generated alt-text has two advantages. First, it alleviates the image processing tasks. Second, it provides

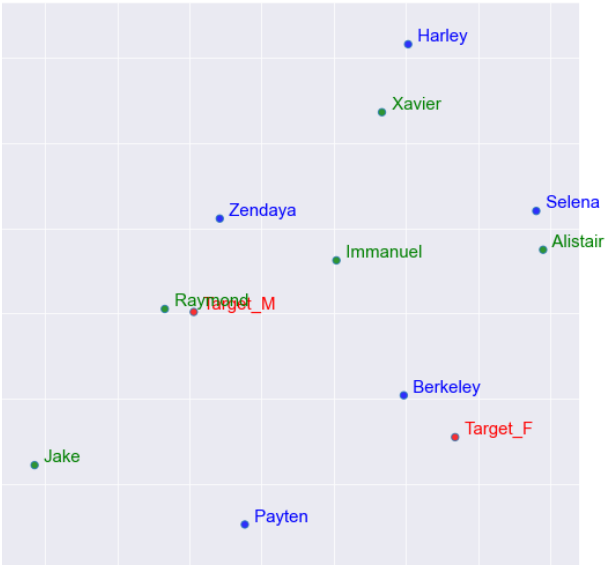


Figure 6.3: Nearest users to female and male hypothesis vectors.

additional information for the attacker. We show the success of our attack on three Facebook sensitive attributes: age, gender, and relationship status. 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. In our attack process, we have leveraged the available picture metadata (either alt-text, comments, or both).

6.5.2 Data pre-processing

In order to get a representative and useful dataset, in addition to the pre-processing steps explained in Section 4.2, we perform the following pre-processing steps:

Cleansing comments

We focus on the English language. Difficulty in analyzing data from social media arises from the presence of different kinds of textual errors, such as misspellings. So we clean the comments as follows:

Changing shape of words. Abbreviations and words with flooded characters (e.g., Heeeelloooo) in posted comments make the comments short and emphasize the emotion behind the words, respectively. We apply *NLTK*[26], a natural language processing package in Python to handle these cases. However, some abbreviations cannot be handled by *NLTK* package. For example *love u* is an abbreviated form of *love you*, where *u* can be interpreted as a misspelled letter *a* by *NLTK*. Therefore we change these cases to their corresponding proper format.

[26] `https://www.nltk.org/`

Eliminating stop words. In a language, stop words are common words (e.g., *the*, *a*, *an*, *in*). As users write in an unstructured way, they utilize more stop words in their comments. Hence, we remove stop words from the extracted comments.

Picture distribution

A challenge for age inference attacks is that picture metadata have been added at different moments of the target lifetime and therefore relate to different age categories. For example, *Alice* can be *31* years old and has published two pictures. The first one when she was *28*, and the second one when she was *31* years old. As a result, she might receive different reactions and responses for these two posted pictures based on her age of picture sharing. Assigning all target users' pictures to the same age group may hinder the inference performance. To circumvent the mentioned problems, we consider the picture publishing time an essential factor to (i) assign pictures to correct age groups and (ii) increase the number of users in different age groups, when possible.

6.5.3 Experiment Setup

Our attack relies on the XGBoost classifier, one of the most potent classifiers. It is based on an efficient ensemble method. XGBoost uses regularization [27] that prevents overfitting and ensures the model can be generalized. Its parameters are divided into general, booster, and learning tasks to guide overall functioning, individual booster (tree/regression), and optimization. In Section 6.5.5, we explain the parameters that we have tuned.

6.5.4 Metric

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 [Lichtenwalter et al., 2010]. Second, we use *macro* and *micro* averages 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.

6.5.5 Parameter settings

We tune three different sets of parameters related to our classifier, RI, and Algorithm 2.

For the classifier, we tune (i) the *learning_rate* to adjust weights on each step and make the model robust, (ii) *max_depth* and (iii) *min_child_weight* to control overfitting. The parameter *objective* specifies the learning task (e.g., binary classification), *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 classifier 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 applied *GridSearchCV* to find the best result value. Finally, the best values of the parameters have been set as: *learning_rate*=

[27] `https://xgboost.readthedocs.io/en/latest/parameter.html`

0.07, *max_ depth* =*6*, *min_ child_ weight* =*4*, *objective* =[$multi : softprob$ (multi-classes), and $binary : logistic$ (binary classes)], *n_ estimators*=*400*, and subsample=*0.7*.

As for RI, [Fernández et al., 2016] proposed a grid of sample values, including (1, 30, 60, 90, 120, 150, 180, 210, 240, 270, 300, 330, 360, 390, 420, 450, 480) for *dimension* and (500, 3500, 6500, 9500, 12500, 15500, 18500) for *non-zero* elements. We consider their samples to prune our RI parameters. RI approximates *Hecht-Nielsen* and *Johnson-Lindenstrauss-Schechtman*'s hypothesis based on these two parameters. We set *dimension*= *500* and select two non-zero elements of the index vector to {-1, +1}, which maximizes the result of our inference attacks.

Our dividing algorithm (see Algorithm 2) comprises two parameters, $UTop$ and $BTop$, to select the most frequent overlapped and non-overlapped *Unigram* and *Bigram*, respectively. Values of both parameters affect the result of Algorithm 2. For example, if $UTop$ is low, a dataset split might be recommended when unnecessary. The best parameter values are $UTop$=90 and $BTop$=110. They have been learned from our dataset by a grid search with $UTop, BTop \in \{10, 20, \ldots, 200\}$.

6.5.6 Inference results

For the age inference attack, we consider the following classes: class 0 (20 to 25), class 1 (25 to 30), class 2 (30 to 35), class 3 (35 to 40), class 4 (40 to 45), and class 5 (45 to 50). We chose these age groups to compromise 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 three classes: class 0 (*single*), class 1 (*married*) and class 2 (*engaged*). Consequently, we consider three classes. Finally, we reduce gender inference attacks to a binary classification problem with class 0 (*female*) and class 1 (*male*). We do not ponder other genders and relationship status by lack of training samples. Regarding age and relationship status, we set the XGBoost classifier *objective* to $multi : softprob$ that gives each class's probability, while for gender, we set it to *logistic*. We have used train-test splitting, which 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 6.4 (a) shows the AUC result of word2vec to age inference attack. The age classes are inferred with AUC from 69% to 70%. In contrast, Figure 6.4 (b) represents the result of the same attribute inference attack employing 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 6.5 (a and b) 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 6.6 (a and b) depict the gender inference attack. Similarly, our approach outstands word2vec model. The attacker can infer the user's gender using our approach with 98% AUC, which drops to 70% in word2vec.

	$Dataset_1$				$Dataset_2$			
	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

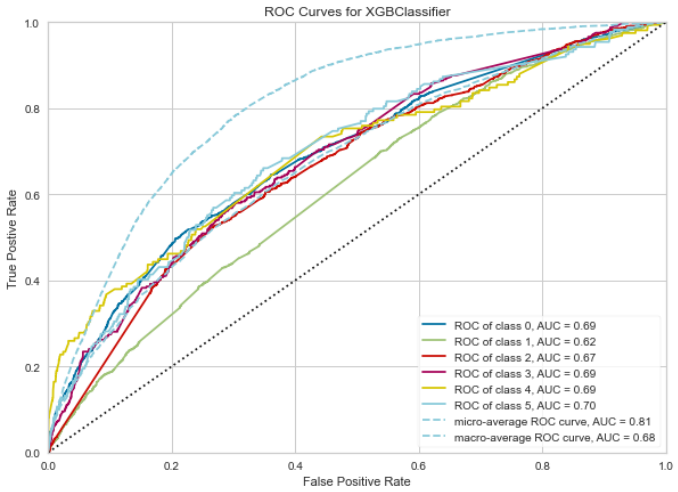
Table 6.2: Comparison of our model with word2vec when splitting conditions are not satisfied.

As mentioned in Section 6.2.1, we split the dataset only if two conditions are satisfied. To justify this, Table 6.2 represents some results of RI with splitting when the conditions are not satisfied. We have synthesized two datasets ($Dataset_1$ and $Dataset_2$) from a crawled dataset labeled by gender. In $Dataset_1$, the first condition is satisfied, and the second one is not satisfied. In $Dataset_2$, the first condition does not hold (and we ignore the status of the second condition). We note that it is better to use word2vec than RI with splitting in both cases. Moreover, word2vec generates only one vector for each word which is space economical compared to RI with splitting.

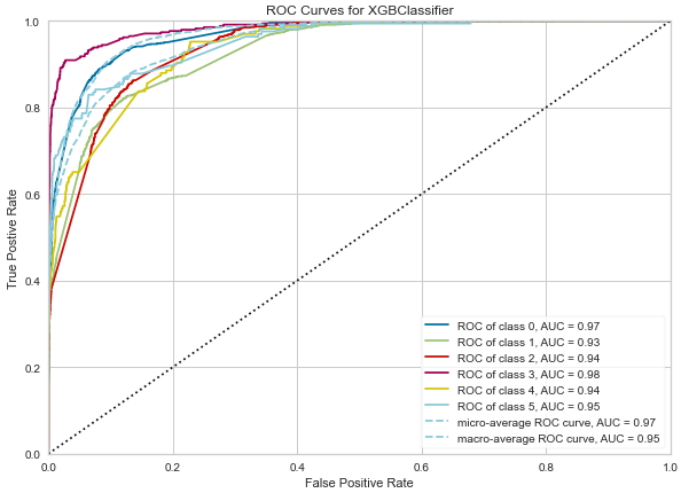
To sum up, when conditions 1 and 2 (see Section 6.2.1) are satisfied, RI with splitting captures the commenters' words/emojis usage preferences adequately, which boosts the accuracy result (see Figures 6.4, 6.5, and 6.6). Otherwise, applying word2vec to generate the vector is more performant (see Table 6.2).

6.6 Conclusions

In this chapter, we have shown that if a term appears in diverse contexts, it can be represented by different vectors. We have divided the dataset based on target user attribute values to capture these diverse contexts. We have also defined some conditions to prevent useless splits. 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. The next chapter proposes a protection mechanism to mitigate the online gender inference attack presented in Chapter 5.

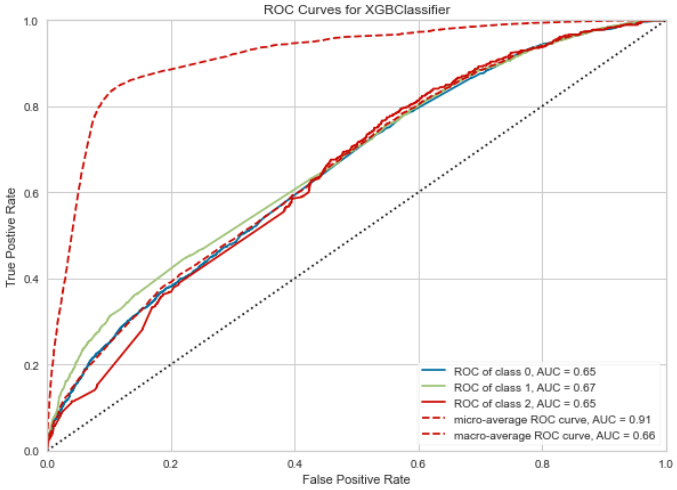


(a)

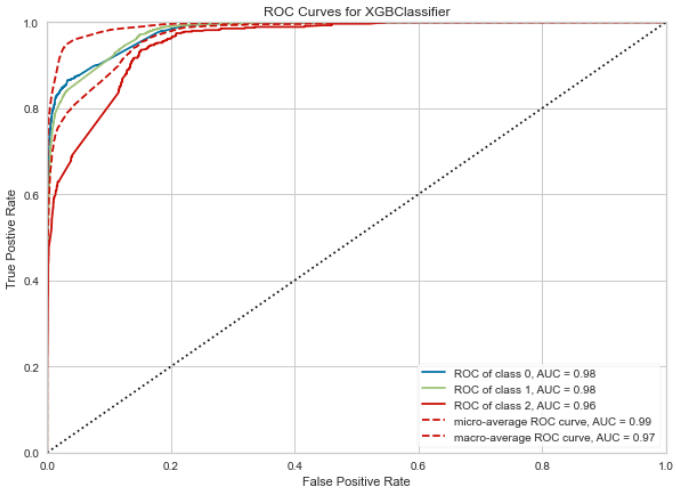


(b)

Figure 6.4: Age inference attack performance (AUC): (a) word2vec, and (b) our approach.

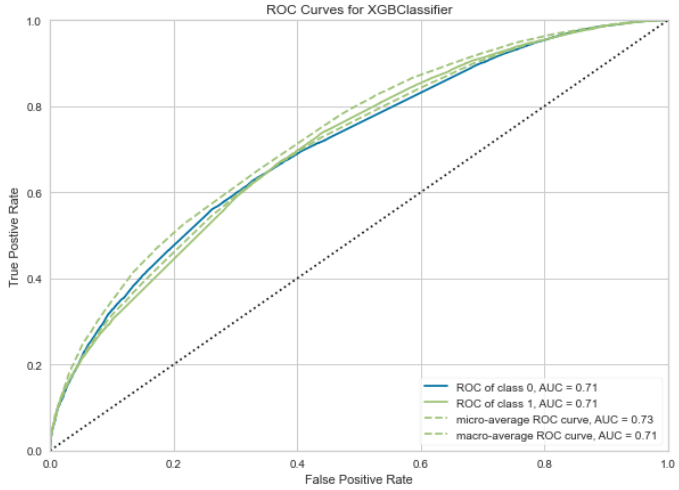


(a)

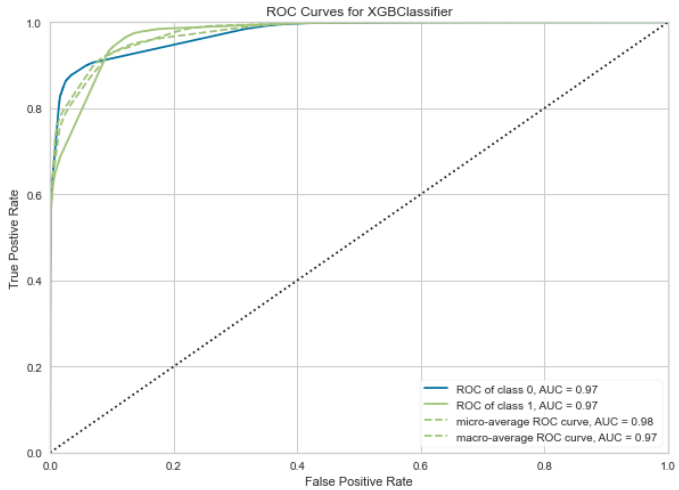


(b)

Figure 6.5: Relationship status inference attack performance (AUC): (a) word2vec, and (b) our approach.



(a)



(b)

Figure 6.6: Gender inference attacks performance (AUC): (a) word2vec, and (b) our approach.

Chapter 7

Attribute Protection

Contents

7.1 Introduction

This chapter is devoted to protecting users from gender inference attacks. Note that one can adapt our protection to other attribute values with few efforts.

Hackers frequently target Facebook [Kuncewicz, 2017]. Corporates, politicians, and celebrities with Facebook accounts can see their reputation tarnished when they become victims of scams or phishing attacks. For example, in 2016, a compromise of Laremy Tunsil's Twitter and Instagram accounts resulted in a loss of millions of dollars [Bishara and Martin, 2016]. This demonstrates the imperative need to design applications to detect and minimize the exploitation of users' personal information.

Generally, privacy protection approaches add noise to a user's public data to minimize the attacker inference accuracy. Game theory methods have been applied against attribute inference attacks [Shokri et al., 2012]. Although these methods guarantee theoretical privacy, they are computationally intractable [Jia and Gong, 2018]. On the other hand, correlation-based heuristics methods modify the public data that is highly correlated with the private attribute values, which generate significant utility loss and need direct access to a user's personal attribute value for computing the correlations [Jia and Gong, 2018]. A naive solution to secure the user's privacy is to remove or hide all comments or pictures. Although this radical solution guarantees user privacy, it decreases the social visibility of the user. Another solution is to analyze all combinations of comments and keep those that preserve user privacy visible. However, this is computationally intractable due to many combinations of comments. In this chapter, we aim to

find a subset of picture metadata (comments and alt-text) that, once they are hidden, changes the classifier result with a minimal impact on the user social environment. We thus propose a practical defense method by modifying the picture metadata (comments and alt-text) to reduce inference accuracy. Our method keeps the modified metadata close to the original ones in their distributional semantic space, ensuring minimal social utility loss.

This chapter is organized as follows: we describe our protection method in Section 7.2 and analyze its effectiveness in Section 7.3.

7.2 Gender Protection

In this section, we present our method that is based on a hiding mechanism to enhance the user's privacy. The defender suggests a set of comments and/or alt-text that the user has to hide to secure their gender without compromising the user visibility. First, let us explain how a user can hide comments and the benefit of hiding comments.

Facebook has two comment filtering options. First, the users can set up a list of words, phrases, or emojis that they do not want to receive from commenters as presented in Figure 7.1 (a). Facebook hides comments containing those words, phrases, or emojis entirely from the published photos. Second, the users can manually select the comments and make them invisible from photos (Figure 7.1 (b)). The advantage of hiding a comment is that it is still visible to the commenter and his/her friends, which reduces tension between the commenter and the picture owner.

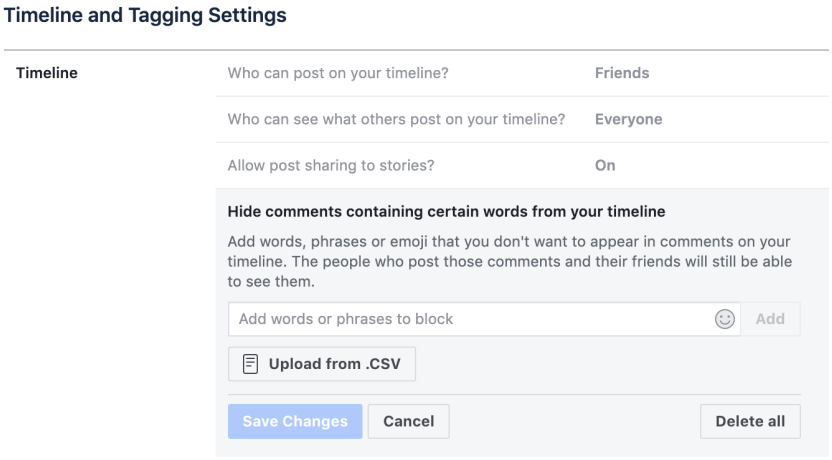
7.2.1 Protection description

Let $O = \{o_1, o_2, .., o_K\}$ denote the *original picture metadata* where o_1 is a special comment containing the alt-text and o_i $(1 < i \leq K)$ stands for the ith comment of that picture. In the protection phase, maybe a user wants to be neither male nor female. In this case, we have to create an illusory gender called nature. Let $\mathcal{C}$ be the prediction function associated with our classifier, where $\mathcal{C}(O) \in \{female, male, neutral\}$. The output depends on the prediction probability threshold. We set this threshold to be *0.70*. For example, the output is *female* (resp. *male*) if the algorithm gives prediction probability of *0.70* to *female* (resp. *male*), and *0.30* to *male* (resp. *female*). Moreover, the output is *neutral* if the algorithm prediction probability for *female* is *0.65*, and *0.35* for *male*. Although the threshold empirically derived from our dataset, it is an arbitrary choice to be adapted to other datasets. It helps to prevent inaccurate attacks due to a lack of input information. For example, if the user has only one picture with an alt-text.

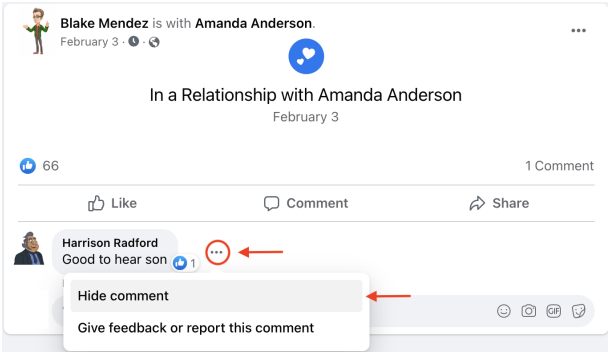
We aim to find O' a subset of O such that $\mathcal{C}(O) \neq \mathcal{C}(O')$. The subset O', called *safe picture metadata*, will satisfy the user requirement in terms of gender protection, of course, after hiding the subset $O \setminus O'$.

Utility loss evaluation

The protection mechanisms should yield safe picture metadata O' that preserves the semantics of the original picture metadata O to the largest possible extent while still defeating our classifier. To that end, we rely on the semantic distance between the original and the safe picture metadata.



(a)



(b)

Figure 7.1: Hide comments (a) automatically by setting up a list, (b) manually from picture

We capture the semantic of words by retrofitting the word2vec vectors (see Section A.2.2). We express the semantics of a picture as the average of its metadata vectors:

$$\vec{h_O} = \frac{1}{|O|} \sum_{o \in O} \sum_{w \in o} \vec{w}$$

The utility loss when hiding some metadata from O to obtain O' is measured by the cosine similarity between $\vec{h_O}$ and $\vec{h_{O'}}$.

7.2.2 Finding safe picture metadata

Our method proceeds in three steps:

Label Propagation. Using our trained *Logistic Regression* (see Section 5.6), each feature in F_{best} (see Section 5.4) has a signed weight (i.e. the regression coefficient) whose sign indicates the classification (positive for female, negative for male) and whose value represents the contribution for the classification. We train our model by using *L2 Loss Function* to compute the error and *Gradient Descent Algorithm* to obtain the values of weights that result in the least error. Our Logistic Regression classifier will use these weights to compute the probability for a person to be male or female, given picture metadata. Table 7.1 below shows the weight of some selected

features. The negative weight of *beard*, and *bro* indicates that the probability of being male is higher compared to female. Similarly, the positive weight of *beautiful*, and *closeup* indicates that the probability of being female is higher compared to male [Moomen et al., 2019].

Feature	*Weight*	*Gender*
bro	-37.86	Male
beard	-18.53	Male
beautiful	25.88	Female
closeup	12.77	Female

Table 7.1: Features' weights computed by *Logistic Regression* classifier

As users have to hide comments, we employ these weights to assign a contribution to the picture metadata by summing all the weights of features. Comments or alt-text receive female (resp., male) contribution if they have positive (resp., negative) weight after summation. Otherwise, we consider that comments or alt-text have no contribution (or neutral contribution). This happens when either the words/emojis/tags are not inside F_{best}, or the summation output is zero, as in that case, female and male features happen to be balanced. With this information, we generate two different sets, which we call *related*, and *unrelated* comments set R, and UR, respectively. The set R contains comments that confirm the classifier output result (risky comments), and UR has opposite gender or neutral contribution comments. We sort R from the least to the most contributing metadata.

Hiding Process. Our objective is to protect gender information by reducing the visibility of metadata O if O leaks gender information. A naive solution would be to hide all the elements of R to reduce the attacker's prediction to a random guess. However, this would lead to an increase in utility loss for the user. To preserve utility, one should retain as many comments as possible. Moreover, finding secure metadata among the numerous combinations (to change the classifier result) would be computationally expensive. We propose a method with reasonable computational cost while ensuring a trade-off between privacy and visibility. We initialize O' by UR and iteratively add the least contributing comment of R to O' as long as the vector of O' ($\vec{h_{O'}}$) gets closer to the vector of O ($\vec{h_O}$) in the vector space and satisfies $\mathcal{C}(O) \neq \mathcal{C}(O')$. Our protection mechanism conveniently discovers safe picture metadata by arranging R from least to most. The process stops when the classifier result increases by adding the least metadata of R. Precisely, if the least contributing metadata enhances the classifier prediction, then the most contributing one can improve as well.

Recommender. The recommender comprises our trained classifier and a condition checker. The condition checker verifies that UR preserves the level of gender privacy while retaining the semantics of the original and safe picture metadata to the largest possible extent. *Recommender* computes the similarity between the original picture metadata (O) and safe picture metadata (O') from their vector representations ($\vec{h_O}$, and $\vec{h_{O'}}$ respectively). Consider the following example where every *female* features are highlighted by *red*, and *male* ones with *blue*. We do not color the terms that are not in F_{best} (*neutral* terms). For example, *teenager* is neither inside the F_{best}, nor semantically close to any feature in F_{best}. In this example, every comment of R has a different weight.

Original Picture Metadata: 1person smiling indoor. beautiful. beautiful girl. handsome. what a teenager. cute. young one.

In the above example, *beautiful* and *girl* are female features, and summing their weights results in a female contribution to *beautiful girl* comment. Although *1person smiling indoor* contains both gender features, it is a male contributing comment as male features have more weight than female ones, and the summation was not negligible (close to zero). Moreover, *what a teenager* is a *neutral* contributing comment as it contains only *neutral*. In this example, the classifier prediction is *Female* (see Figure 7.2). Then we generate UR and sorted R as follows:

UR: 1person smiling indoor. handsome. what a teenager.
R: young one. cute. beautiful. beautiful girl.

As presented in Figure 7.2, the recommender determines the output of our classifier on *Original Picture Metadata*. It recommends *Safe Picture Metadata*, which contains non-compromising comments/alt-text, and the classifier output on *Safe Picture Metadata*. It determines the similarity between the original and the safe metadata to show the utility loss of the user when replacing the *Original Picture Metadata* by *Safe Picture Metadata*. Moreover, the user receives risky comments highlighted by the dangerous words, the number of comments he/she has to hide from *Original Picture Metadata*, and commenters of those risky comments.

```
Classifier Output on Original Picture Metadata: [[0. 1.]] ---> Female
Safe Picture Metadata: [['1person smile indoor', 'handsome', 'what a teenager', 'young one', 'cute', 'beautiful']]
Classifier Output on Safe Picture Metadata: [[1.00000000e+00 3.14143386e-10]] ---> Male
Similarity between Original and Safe Picture Metadata: 0.9115620255470276
Risky Comments: beautiful girl
Risky Commenters: ['Second Commenter']
Number of Hidden Commments: 1
```

Figure 7.2: Recommender output. Each comment of R has a different weight.

If some metadata of R have the same weight, we randomly pick one of them when they are among the candidates to be hidden. Consider the following example:

Original Picture Metadata: 1person. handsome. u0001f602. handsome. good pic. very nice. pretty. pretty.

where R and UR are:

	precision	*recall*	*f1-score*
class Female	0.09	0.09	0.09
class Male	0.13	0.12	0.12
macro avg	0.11	0.11	0.11
weighted avg	0.11	0.11	0.11

Table 7.2: Protection measurement

R: u0001f602. pretty. pretty.

UR: 1person. handsome. handsome. good pic. very nice.

All the risky comments (comments in R) have the same weight (they are identical in this case), and they cannot be differentiated by sorting. As presented in Figure 7.3, the recommender suggests that hiding one comment is enough to protect the user gender. We might randomly hide one of the risky comments. As a result, the recommender shows the classifier's output on the original picture metadata, the number of comments to hide and allows the user to select which comments he/she wants to hide. For example, he/she might retain comments from friends.

Real Gender of User: Female

Classifier Output on Original Picture Metadata: [[0.00630367 0.99369633]] ---> Female

Obfuscated Picture Metadata: [' handsome ', ' u0001f602 ', ' handsome ', ' good pic ', '1person ', ' very nice ', ' pretty ', ' pretty ']

Classifier Output on Obfuscated Picture Metadata: [[0.43412212 0.56587788]] ---> Neuter

Cosine Similarity between Original and Obfuscated Picture Metadata: 0.8170092105865479

Risky Comment(s): [' pretty']

Risky Commenter(s): Select a commenter who posts the risky comment based on your desire, and hide the commenter's comment.

Number of Hidden Commments: 1

Figure 7.3: Recommender output. All comments of R have the same weight

7.3 Experiments

We evaluated the performance of defence method on 700 online users (see Section 5.6.2). Figure A.2, and Table A.2 represent the impact of our method on the online attack performance with *Logistic Regression* (see Section A.1(f)). This is shown by comparing the ground truth gender (declared by users) and the classifier output on the method computed safe picture metadata O'. We note that the AUC and other metrics dramatically drop, which demonstrates the effectiveness of method in preserving user gender privacy.

Facebook users want to keep a subset of comments O' public while (i) ensuring the attacker is unable to infer their gender and (ii) minimizing the utility loss. Figure A.3 (a) shows the cumulative distribution function of the minimal utility loss. We notice that the method provides high utility. This confirms the fact that it misleads the classifier but at the same time keeps

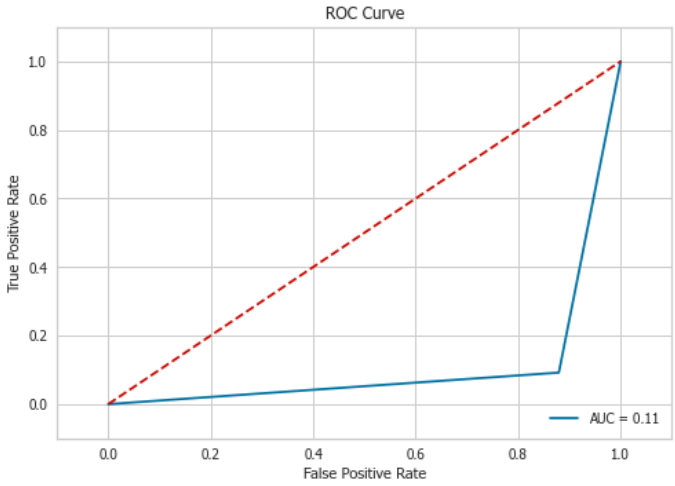


Figure 7.4: Protecting AUC

enough utility as the protected set O' is semantically close to the original set O. Figure A.3 (b) shows the average number of comments to be hidden over original picture metadata. We observe that the number of hidden comments depends on the content of the comments, not on the number of received comments. For example, the average number of hidden comments for a user with *20* comments is *6*, while *2* is the number of comments to hide (in average) for a user with *23* comments. Figure A.3 (c) represents the running time which is an essential factor for our protector usability. It shows how the running time grows with the number of comments to hide. For instance, it takes *1* seconds to hide *5* comments to preserve user gender privacy. Note that the running time also depends on the content of the comments. For example, the method needs *9* seconds to secure a user with *38* comments, while it takes almost *5* seconds for a user with *35* comments. For the illustration purpose, in Figure A.3 (b, and c), we only show users who have less than *40* comments. All these experiments are conducted on a laptop computer with a *2.6 GHz CPU* and *16 GB* memory and are implemented relying on Python packages such as *Pandas*, *NumPy*, and *Scikit-learn*.

7.4 Discussion

By applying our system, a Facebook user can check if he/she is vulnerable to gender inference attacks. Our attack may succeed even when the target publishes nothing else than pictures. To counter the aforementioned privacy violations, the method hides comments or alt-text when they strongly contribute to the attack as explained in Section A.3.2. We address three limitations that previous works suffer from (i) tractability, (ii) utility loss, and (iii) access to real attribute value as concern [Jia and Gong, 2018]. Our protection mechanism's goal is to suggest comments to hide to minimize the inference accuracy with a slight utility loss for the user. We estimate the inference attack accuracy when a particular comment is hidden to achieve this goal. As for the first two issues, we generate two lists called R and UR. We sort R from least to most, iterate over it and add a comment to UR at a time. In each iteration, we check the attack's accuracy. As we sort R from least to most, if the least contributing comment/alt-text enhances the classifier prediction, the most contributing one can improve it as well. Therefore, we have only a few comments to check (addressing tractability), and we make sure we do not

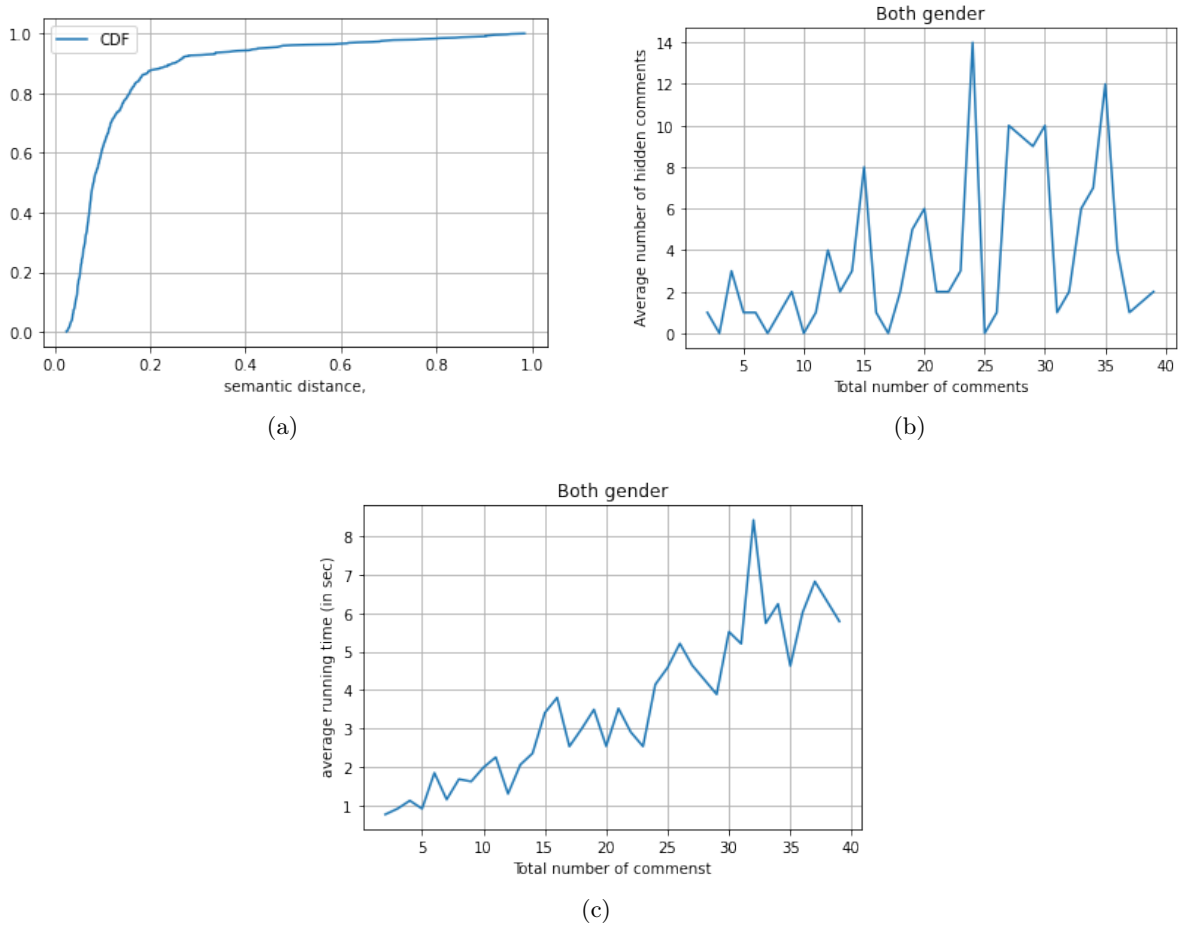


Figure 7.5: Protection method performances: (a) Cumulative distribution function (CDF) of the minimum utility loss, i.e., semantic distance, for maximum privacy,(b) average of metadata to be hidden with respect to the number of original metadata (x-axis), and (c) average running time (per sample) with respect to the number of comments to hide in order to protect the picture owner gender.

hide unnecessary comments (addressing utility loss). As for the third issue, other approaches [Weinsberg et al., 2012, Heatherly et al., 2013, Chen et al., 2014] require direct access to users' real attribute value, which makes it inconvenient for users, and the private attribute values of all users get compromised if the defender gets compromised. Our protection mechanism does not require the user's real attribute value (to avoid the above issues). It takes user pictures metadata as input and starts its protection process by relying on the classifier output.

7.5 Conclusions

In this chapter, we have proposed a privacy protection mechanism that provides an optimal utility-privacy-efficiency trade-off. The method is based on a hiding mechanism. Note that hiding comments reduces the tension between commenters and picture owners as the hidden comments are still visible to commenters and their friends. The method does not compromise (i) user privacy and (ii) visibility. Finally, we have shown that our privacy protection mechanism recommends to users helpful information such as risky (i) comments, (ii) commenters, and (iii) words, and the number of comments the user has to hide to be secure from gender inference attack.

Chapter 8

Conclusions

8.1 Achievements

Identifying users' attributes from their online activities and data sharing behavior is an important topic in the growing research field of social networks. It provides an opportunity for targeted advertising, profile customization, or privacy attacks. In this work, we have analyzed privacy leakage on Facebook. Precisely, we consider three sensitive attributes (gender, age, and relationship status). We have investigated *627,776* pictures and their *1,332,219* comments. Based on the intensive analysis of the shared images, this work has demonstrated (i) a new perspective of attribute inference attacks on Facebook users by leveraging the metadata triggered by the target user publications and (ii) has proposed a privacy protection system.

We have shown the possibility of attribute inference attacks even when all user attributes/activities such as profile attributes, friend list, liked pages, and joined groups are hidden. We have demonstrated that picture metadata convey sensitive information such as gender, age category, and relationship status which 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, and they generalize easily to other social media platforms such as Instagram.

Our experimental results illustrated that, on average, female posted pictures receive more emojis-based comments than male posted pictures. We have noticed that commenters use more emotional emojis while commenting on female images in a particular theme and setting. We have demonstrated that commenters react differently to younger and older owner pictures. Moreover, we have observed the differences in commenters' reactions when reacting to single, married, and engaged owner pictures with the same style and alt-text tags. Additionally, we have shown alt-text provides extra free information that boosts inference accuracy. Such a disparity can be used to implement inference attacks. We have presented that if a term appears in diverse contexts, it can be represented by different vectors. We have divided the dataset based on attribute values to capture these diverse contexts. 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.

We have proposed a privacy protection mechanism that provides an optimal utility-privacy-efficiency trade-off. Our objective was to find a set of comments that, once they are hidden,

changes the classifier result with a minimal impact on the user social environment. We have proposed a practical defence by modifying the picture metadata to reduce inference accuracy. The proposed method is computationally fast and keeps the modified metadata close to the original ones in their distributional semantic space, ensuring minimal social utility loss.

8.2 Limitations

Our work has several limitations. We have not considered the time commenters have posted the comments as an input to our model. There might be cases where time sequence analysis helps understand the connection between comments. We have not either considered topic models in our analysis. For example, topics analysis may derive extra features for the prediction process. We have not performed sentiment analysis on emojis to integrate their meaning in order to provide additional information to perform inference attacks. We have not taken advantage of recent deep learning state-of-the-art tools (e.g., BERT) that have been leveraged for hate speech recognition and sentiment analysis. Finally, our protection system computes word vector representations by retrofitting on a fixed-size dataset. But it would be desirable to repeat the vector generation process if the size of the online dataset is larger than the offline dataset to get a better representation.

8.3 Perspectives

As for the future, the promising directions are as follows:

We plan to explore online inference in other social network platforms (e.g., Twitter, Instagram). For example, alt-text has been implemented on Instagram. We can train the proposed models by collecting ground truth data from Instagram and studying the impact of the picture owner on commenters' reactions. However, the meaning of, for example, emojis may differ on each of the social media. To have a general model that can handle all the platforms, we can take advantage of combining several user-generated contents from different online social networks to infer private attributes.

We propose investigating sentiment analysis of emojis to enrich the classification input data. Each emoji can have a different meaning based on the context where it appears. One possible way to solve this problem is to leverage our retrofitting-based vector representation to get the meaning of each emoji based on the co-occurring terms.

Applying deep learning models such as BERT should be profitable to generate end-to-end inference models. We can use the pre-trained BERT model and fine-tune it on our collected dataset. Another direction is to adapt Graph Neural Networks and inject the animated reactions as an extra feature to the model.

Our proposed protection mechanism can be adapted to hate speech recognition and hide hatred comments. To that end, we can collect a pre-labeled hate speech dataset from Facebook or leverage unsupervised models to cluster comments to derive hatred, non-hatred, and neutral classes. Next, we can train a multi-class classification task and propose to hide hateful comments taking into account the visibility of users.

Bibliography

[Abid, 2018] Abid, Y. (2018). *Analyse automatisée des risques sur la vie privée dans les réseaux sociaux*. PhD thesis, Université de Lorraine.

[Agrawal and Srikant, 2000] Agrawal, R. and Srikant, R. (2000). Privacy-preserving data mining. In *Proceedings of the 2000 ACM SIGMOD international conference on Management of data*, pages 439–450.

[Ai et al., 2017] Ai, W., Lu, X., Liu, X., Wang, N., Huang, G., and Mei, Q. (2017). Untangling emoji popularity through semantic embeddings. In *Proceedings of the Eleventh International Conference on Web and Social Media, ICWSM*, pages 2–11, Montréal, Québec, Canada. AAAI Press.

[Akbik et al., 2019] Akbik, A., Bergmann, T., and Vollgraf, R. (2019). Pooled contextualized embeddings for named entity recognition. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 724–728.

[Al Zamal et al., 2012] Al Zamal, F., Liu, W., and Ruths, D. (2012). Homophily and latent attribute inference: Inferring latent attributes of twitter users from neighbors. In *Sixth International AAAI Conference on Weblogs and Social Media*.

[Alipour et al., 2019] Alipour, B., Imine, A., and Rusinowitch, M. (2019). Gender Inference for Facebook Picture Owners. In *International Conference on Trust, Privacy and Security in Digital Business, TrustBus*, pages 145–160, Linz, Austria. Springer.

[An et al., 2018] An, J., Li, T., Teng, Y., and Zhang, P. (2018). Factors influencing emoji usage in smartphone mediated communications. In *Transforming Digital Worlds - 13th International Conference, iConference*, pages 423–428, Sheffield, UK. Springer.

[Backstrom and Kleinberg, 2014] Backstrom, L. and Kleinberg, J. (2014). Romantic partnerships and the dispersion of social ties: a network analysis of relationship status on facebook. In *Proceedings of the 17th ACM conference on Computer supported cooperative work & social computing*, pages 831–841.

[Barbieri et al., 2016a] Barbieri, F., Kruszewski, G., Ronzano, F., and Saggion, H. (2016a). How cosmopolitan are emojis?: Exploring emojis usage and meaning over different languages with distributional semantics. In *Proceedings of the Conference on Multimedia Conference, MM*, pages 531–535, Amsterdam, The Netherlands. ACM.

[Barbieri et al., 2016b] Barbieri, F., Ronzano, F., and Saggion, H. (2016b). What does this emoji mean? a vector space skip-gram model for twitter emojis. In *Calzolari N, Choukri*

K, Declerck T, et al, editors. Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016); 2016 May 23-28; Portorož, Slovenia. Paris: European Language Resources Association (ELRA); 2016. p. 3967-72. ELRA (European Language Resources Association).

[Bargh et al., 2002] Bargh, J. A., McKenna, K. Y., and Fitzsimons, G. M. (2002). Can you see the real me? activation and expression of the "true self" on the internet. *Journal of social issues*, 58(1):33–48.

[Basile et al., 2015] Basile, P., Caputo, A., and Semeraro, G. (2015). Temporal random indexing: A system for analysing word meaning over time. *Italian Journal of Computational Linguistics*, 1(1):55–68.

[Beheshti et al., 2020] Beheshti, A., Moraveji-Hashemi, V., Yakhchi, S., Motahari-Nezhad, H. R., Ghafari, S. M., and Yang, J. (2020). personality2vec: Enabling the analysis of behavioral disorders in social networks. In *Proceedings of the 13th international conference on web search and data mining*, pages 825–828.

[Belinic, 2009] Belinic, T. (2009). Personality profile of social media users how to get maximum from it. https://medium.com/krakensystems-blog/personality-profile-of-social-media-users-how-to-get-maximum-from-it-5e8b803efb30.

[Bertran et al., 2019] Bertran, M., Martinez, N., Papadaki, A., Qiu, Q., Rodrigues, M., Reeves, G., and Sapiro, G. (2019). Adversarially learned representations for information obfuscation and inference. In *International Conference on Machine Learning*, pages 614–623. PMLR.

[Bishara and Martin, 2016] Bishara, M. and Martin, J. (2016). Laremy tunsil: Twitter hack could cost him millions after pot video goes viral. https://edition.cnn.com/2016/04/29/sport/laremy-tunsil-ole-miss-nfl-draft-twitter-hack/index.html.

[Boshmaf et al., 2013] Boshmaf, Y., Muslukhov, I., Beznosov, K., and Ripeanu, M. (2013). Design and analysis of a social botnet. *Computer Networks*, 57(2):556–578.

[Breiman, 2001] Breiman, L. (2001). Random forests. *Machine learning*, 45(1):5–32.

[Breiman et al., 2017] Breiman, L., Friedman, J. H., Olshen, R. A., and Stone, C. J. (2017). *Classification and regression trees*. Routledge.

[Brouer et al., 2017] Brouer, R. L., Stefanone, M., Badawy, R. L., and Egnoto, M. J. (2017). Gender (in)consistent communication via social media and hireability: An exploratory study. In *50th Hawaii International Conference on System Sciences, HICSS*, pages 1–10, Hawaii, USA. ScholarSpace / AISeL.

[Butterworth et al., 2019] Butterworth, S. E., Giuliano, T. A., White, J., Cantu, L., and Fraser, K. C. (2019). Sender gender influences emoji interpretation in text messages. *Frontiers in psychology*, 10:784.

[Cadwalladr and Graham-Harrison, 2018] Cadwalladr, C. and Graham-Harrison, E. (2018). How cambridge analytica turned facebook likes into a lucrative political tool. `https://www.theguardian.com/technology/2018/mar/17/facebook-cambridge-analytica-kogan-data-algorithm`.

[Cai et al., 2016] Cai, Z., He, Z., Guan, X., and Li, Y. (2016). Collective data-sanitization for preventing sensitive information inference attacks in social networks. *IEEE Transactions on Dependable and Secure Computing*, 15(4):577–590.

[Chaabane et al., 2012] Chaabane, A., Acs, G., Kaafar, M. A., et al. (2012). You are what you like! information leakage through users' interests. In *Proceedings of the 19th Annual Network & Distributed System Security Symposium (NDSS)*.

[Chen et al., 2014] Chen, T., Boreli, R., Kâafar, M. A., and Friedman, A. (2014). On the Effectiveness of Obfuscation Techniques in Online Social Networks. In *Privacy Enhancing Technologies - 14th International Symposium, PETS, Amsterdam, The Netherlands*, volume 8555 of *Lecture Notes in Computer Science*, pages 42–62. Springer.

[Chen and Guestrin, 2016] Chen, T. and Guestrin, C. (2016). Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pages 785–794.

[Chen et al., 2018] Chen, Z., Lu, X., Ai, W., Li, H., Mei, Q., and Liu, X. (2018). Through a gender lens: Learning usage patterns of emojis from large-scale android users. In *Proceedings of the 2018 World Wide Web Conference on World Wide Web, WWW*, pages 763–772, Lyon, France. ACM.

[Chen et al., 2019] Chen, Z., Shen, S., Hu, Z., Lu, X., Mei, Q., and Liu, X. (2019). Emoji-powered representation learning for cross-lingual sentiment classification. In *The World Wide Web Conference, WWW*, pages 251–262, San Francisco, CA, USA. ACM.

[Cheung and She, 2017] Cheung, M. and She, J. (2017). An Analytic System for User Gender Identification Through User Shared Images. *ACM Journal of Transactions on Multimedia Computing, Communications, and Applications (TOMM).*, 13(3):30:1–30:20.

[Choi et al., 2017] Choi, D., Lee, Y., Kim, S., and Kang, P. (2017). Private attribute inference from facebook's public text metadata: a case study of korean users. *Industrial Management & Data Systems*.

[Choudhury et al., 2017] Choudhury, M. D., Sharma, S. S., Logar, T., Eekhout, W., and Nielsen, R. C. (2017). 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, Portland, OR, USA*, pages 353–369. ACM.

[Chowdhury et al., 2019] Chowdhury, A. G., Sawhney, R., Mathur, P., Mahata, D., and Shah, R. R. (2019). Speak up, fight back! detection of social media disclosures of sexual harassment. In *Proceedings of the 2019 conference of the North American chapter of the Association for Computational Linguistics: Student research workshop*, pages 136–146.

[Church and Hanks, 1990] Church, K. W. and Hanks, P. (1990). Word Association Norms, Mutual Information, and Lexicography. *Computational linguistics*, 16(1):22–29.

[Cooper, 2013] Cooper, B. B. (2013). 7 powerful facebook statistics you should know for a more engaging facebook page. https://buffer.com/resources/7-facebook-stats-you-should-know-for-a-more-engaging-page/.

[Cover, 1999] Cover, T. M. (1999). *Elements of information theory*. John Wiley & Sons.

[Cover and Hart, 1967] Cover, T. M. and Hart, P. E. (1967). Nearest neighbor pattern classification. *IEEE Trans. Inf. Theory*, 13(1):21–27.

[Daniya et al., 2020] Daniya, T., Geetha, M., and Kumar, K. S. (2020). Classification and regression trees with gini index. *Advances in Mathematics Scientific Journal*, 9(10):1857–8438.

[Datta et al., 2015] Datta, A., Tschantz, M. C., and Datta, A. (2015). Automated experiments on ad privacy settings. *Proceedings on privacy enhancing technologies*, 2015(1):92–112.

[Dey et al., 2012] Dey, R., Tang, C., Ross, K. W., and Saxena, N. (2012). Estimating age privacy leakage in online social networks. In *Proceedings of the IEEE INFOCOM 2012, Orlando, FL, USA, March 25-30, 2012*, pages 2836–2840. IEEE.

[du Pin Calmon and Fawaz, 2012] du Pin Calmon, F. and Fawaz, N. (2012). Privacy Against Statistical Inference. In *50th Annual Allerton Conference on Communication, Control, and Computing, Allerton Park & Retreat Center, Monticello, IL, USA*, pages 1401–1408. IEEE.

[Eisner et al., 2016] Eisner, B., Rocktäschel, T., Augenstein, I., Bošnjak, M., and Riedel, S. (2016). emoji2vec: Learning Emoji Representations from their Description. *arXiv preprint arXiv:1609.08359*.

[Farahbakhsh et al., 2017] Farahbakhsh, R., Han, X., Cuevas, Á., and Crespi, N. (2017). Analysis of publicly disclosed information in facebook profiles. *CoRR*, abs/1705.00515.

[Faruqui et al., 2014] Faruqui, M., Dodge, J., Jauhar, S. K., Dyer, C., Hovy, E., and Smith, N. A. (2014). Retrofitting Word Vectors to Semantic Lexicons. *arXiv preprint arXiv:1411.4166*.

[Feng et al., 2014] Feng, J., Xu, H., Mannor, S., and Yan, S. (2014). Robust logistic regression and classification. *Advances in neural information processing systems*, 27:253–261.

[Fernández et al., 2016] Fernández, A. M., Esuli, A., and Sebastiani, F. (2016). Lightweight random indexing for polylingual text classification. *Journal of Artificial Intelligence Research*, 57:151–185.

[Fkih and Omri, 2012] Fkih, F. and Omri, M. N. (2012). Learning the size of the sliding window for the collocations extraction: A roc-based approach. In *Proceedings of the 2012 International Conference on Artificial Intelligence (ICAI'12)*, pages 1071–1077.

[Flekova et al., 2016] Flekova, L., Carpenter, J., Giorgi, S., Ungar, L. H., and Preotiuc-Pietro, D. (2016). Analyzing Biases in Human Perception of User Age and Gender from Text. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, Berlin, Germany, Volume 1: Long Papers*. ACL.

[Ganguly et al., 2015] Ganguly, D., Roy, D., Mitra, M., and Jones, G. J. (2015). Word embedding based generalized language model for information retrieval. In *Proceedings of the 38th international ACM SIGIR conference on research and development in information retrieval*, pages 795–798.

[Gao et al., 2012] Gao, H., Chen, Y., Lee, K., Palsetia, D., and Choudhary, A. N. (2012). Towards online spam filtering in social networks. In *NDSS*, volume 12, pages 1–16.

[Gao et al., 2019] Gao, L., Zhou, G., Luo, J., and Huang, Y. (2019). Word embedding with zipf's context. *IEEE access*, 7:168934–168943.

[Garcia, 2017] Garcia, D. (2017). Leaking privacy and shadow profiles in online social networks. *Science advances*, 3(8):e1701172.

[Garera and Yarowsky, 2009] Garera, N. and Yarowsky, D. (2009). Modeling latent biographic attributes in conversational genres. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 710–718.

[Gong and Liu, 2016] Gong, N. Z. and Liu, B. (2016). You are who you know and how you behave: Attribute inference attacks via users' social friends and behaviors. In *25th Security Symposium*, pages 979–995, Austin, TX, USA. USENIX.

[Gong and Liu, 2018] Gong, N. Z. and Liu, B. (2018). Attribute inference attacks in online social networks. *ACM Transactions on Privacy and Security (TOPS)*, 21(1):3.

[Gong et al., 2014] Gong, N. Z., Talwalkar, A., Mackey, L., Huang, L., Shin, E. C. R., Stefanov, E., Shi, E. R., and Song, D. (2014). Joint link prediction and attribute inference using a social-attribute network. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 5(2):27.

[Hamm, 2017] Hamm, J. (2017). Minimax filter: Learning to preserve privacy from inference attacks. *The Journal of Machine Learning Research*, 18(1):4704–4734.

[Han et al., 2019] Han, X., Huang, H., and Wang, L. (2019). F-pad: Private attribute disclosure risk estimation in online social networks. *IEEE Transactions on Dependable and Secure Computing*, 16(6):1054–1069.

[Harris, 1954] Harris, Z. S. (1954). Distributional structure. *Word*, 10(2-3):146–162.

[Heatherly et al., 2013] Heatherly, R., Kantarcioglu, M., and Thuraisingham, B. M. (2013). Preventing Private Information Inference Attacks on Social Networks. *IEEE Journal of Transactions on Knowledge and Data Engineering.*, 25(8):1849–1862.

[Hecht-Nielsen et al., 1994] Hecht-Nielsen, R. et al. (1994). Context vectors: general purpose approximate meaning representations self-organized from raw data. *Computational intelligence: Imitating life*, 3(11):43–56.

[Jia and Gong, 2018] Jia, J. and Gong, N. Z. (2018). Attriguard: A practical Defense Against Attribute Inference Attacks via Adversarial Machine Learning. In *27th USENIX Security Symposium, Baltimore, MD, USA*, pages 513–529. USENIX Association.

[Karimi et al., 2016] Karimi, F., Wagner, C., Lemmerich, F., Jadidi, M., and Strohmaier, M. (2016). Inferring Gender from Names on the Web: A Comparative Evaluation of Gender Detection Methods. In *Proceedings of the 25th International Conference on World Wide Web, Montreal, Canada*, pages 53–54. ACM.

[Kim et al., 2020] Kim, M.-J., Kang, J.-S., and Chung, K. (2020). Word-embedding-based traffic document classification model for detecting emerging risks using sentiment similarity weight. *IEEE Access*, 8:183983–183994.

[Kontaxis et al., 2011] Kontaxis, G., Polakis, I., Ioannidis, S., and Markatos, E. P. (2011). Detecting social network profile cloning. In *Ninth Annual IEEE International Conference on*

Pervasive Computing and Communications, PerCom 2011, 21-25 March 2011, Seattle, WA, USA, Workshop Proceedings, pages 295–300. IEEE Computer Society.

[Kosinski et al., 2013] Kosinski, M., Stillwell, D., and Graepel, T. (2013). Private traits and attributes are predictable from digital records of human behavior. *Proceedings of the national academy of sciences*, 110(15):5802–5805.

[Kótyuk and Buttyán, 2012] Kótyuk, G. and Buttyán, L. (2012). A machine learning based approach for predicting undisclosed attributes in social networks. In *2012 IEEE International Conference on Pervasive Computing and Communications Workshops*, pages 361–366. IEEE.

[Kralj Novak et al., 2015] Kralj Novak, P., Smailović, J., Sluban, B., and Mozetič, I. (2015). Sentiment of emojis. *PloS one*, 10(12):e0144296.

[Kuncewicz, 2017] Kuncewicz, S. (2017). 1 in 5 smes have fallen victim to social media hackers. https://ffnews.com/newsarticle/1-in-5-smes-have-fallen-victim-to-social-media-hackers/.

[Kutuzov et al., 2018] Kutuzov, A., Øvrelid, L., Szymanski, T., and Velldal, E. (2018). 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*, pages 1384–1397.

[Lee, 2002] Lee, K. (2002). Combining multiple feature selection methods. In *Mid-Atlantic Student Workshop on Programming Languages and Systems (MASPLAS'02)*, pages 12–1. Citeseer.

[Levi and Hassner, 2015] Levi, G. and Hassner, T. (2015). Age and gender classification using convolutional neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 34–42.

[Levy and Goldberg, 2014a] Levy, O. and Goldberg, Y. (2014a). Dependency-based word embeddings. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 302–308.

[Levy and Goldberg, 2014b] Levy, O. and Goldberg, Y. (2014b). 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*, pages 171–180. ACL.

[Li et al., 2018] Li, Y., Baldwin, T., and Cohn, T. (2018). Towards robust and privacy-preserving text representations. *arXiv preprint arXiv:1805.06093*.

[Lichtenwalter et al., 2010] Lichtenwalter, R., Lussier, J. T., and Chawla, N. V. (2010). 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*, pages 243–252.

[Lindenstrauss, 1984] Lindenstrauss, W. J. J. (1984). Extensions of lipschitz maps into a hilbert space. *Contemp. Math*, 26:189–206.

[Ling et al., 2014] Ling, R., Baron, N. S., Lenhart, A., and Campbell, S. W. (2014). "girls text really weird": gender, texting and identity among teens. *Journal of Children and Media*, 8(4):423–439.

[Lopes Filho et al., 2016] Lopes Filho, J. A. B., Pasti, R., and de Castro, L. N. (2016). Gender classification of twitter data based on textual meta-attributes extraction. In *New Advances in Information Systems and Technologies*, pages 1025–1034. Springer.

[Lu et al., 2016] Lu, X., Ai, W., Liu, X., Li, Q., Wang, N., Huang, G., and Mei, Q. (2016). Learning from the ubiquitous language: an empirical analysis of emoji usage of smartphone users. In *Proceedings of the International Joint Conference on Pervasive and Ubiquitous Computing, UbiComp*, pages 770–780, Heidelberg, Germany. ACM.

[Lu et al., 2018] Lu, X., Cao, Y., Chen, Z., and Liu, X. (2018). A first look at emoji usage on github: An empirical study. *CoRR*, abs/1812.04863.

[Ludu, 2014] Ludu, P. S. (2014). Inferring Gender of a Twitter User Using Celebrities it Follows. *The Computing Research Repository Journal (CoRR).*, abs/1405.6667, 2014.

[McSherry and Mironov, 2009] McSherry, F. and Mironov, I. (2009). Differentially private recommender systems: Building privacy into the netflix prize contenders. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 627–636.

[Mei et al., 2018] Mei, B., Xiao, Y., Li, R., Li, H., Cheng, X., and Sun, Y. (2018). Image and attribute based convolutional neural network inference attacks in social networks. *IEEE Transactions on Network Science and Engineering*, 7(2):869–879.

[Mikolov et al., 2013a] Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013a). Efficient Estimation of Word Representations in Vector Space. *arXiv preprint arXiv:1301.3781*.

[Mikolov et al., 2013b] Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., and Dean, J. (2013b). Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119.

[Miller, 1995] Miller, G. A. (1995). WordNet: a Lexical Database for English. *Communications of the ACM*, 38(11):39–41.

[Miller et al., 2017] Miller, H. J., Kluver, D., Thebault-Spieker, J., Terveen, L. G., and Hecht, B. J. (2017). Understanding emoji ambiguity in context: The role of text in emoji-related miscommunication. In *Proceedings of the Eleventh International Conference on Web and Social Media, ICWSM*, pages 152–161, Montréal, Québec, Canada. AAAI Press.

[Miller et al., 2012] Miller, Z., Dickinson, B., and Hu, W. (2012). Gender prediction on twitter using stream algorithms with n-gram character features. *International Journal of Intelligence Science*, 2(04):143.

[Minkus et al., 2015] Minkus, T., Ding, Y., Dey, R., and Ross, K. W. (2015). The city privacy attack: Combining social media and public records for detailed profiles of adults and children. In *Proceedings of the 2015 ACM on conference on online social networks*, pages 71–81.

[Moomen et al., 2019] Moomen, M., Rezapour, M., and Ksaibati, K. (2019). An Investigation of Influential Factors of Downgrade Truck Crashes: A Logistic Regression Approach. *Journal of traffic and transportation engineering (English edition)*, 6(2):185–195.

[Nguyen et al., 2013] Nguyen, D., Gravel, R., Trieschnigg, D., and Meder, T. (2013). 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.

[Nguyen et al., 2011] Nguyen, D., Smith, N. A., and Rosé, C. P. (2011). Author age prediction from text using linear regression. In *Proceedings of the 5th ACL Workshop on Language Technology for Cultural Heritage, Social Sciences, and Humanities, LaTeCH@ACL 2011, 24 June, 2011, Portland, Oregon, USA*, pages 115–123. The Association for Computer Linguistics.

[Nguyen et al., 2014] Nguyen, D., Trieschnigg, D., Dogruöz, A. S., Gravel, R., Theune, M., Meder, T., and de Jong, F. (2014). Why gender and age prediction from tweets is hard: Lessons from a crowdsourcing experiment. In *25th International Conference on Computational Linguistics, Proceedings of the Conference, COLING*, pages 1950–1961, Dublin, Ireland. ACL.

[Nguyên et al., 2016] Nguyên, T. T., Xiao, X., Yang, Y., Hui, S. C., Shin, H., and Shin, J. (2016). Collecting and analyzing data from smart device users with local differential privacy. *arXiv preprint arXiv:1606.05053*.

[Niwa and Nitta, 1995] Niwa, Y. and Nitta, Y. (1995). Co-occurrence vectors from corpora vs. distance vectors from dictionaries. *arXiv preprint cmp-lg/9503025*.

[Noble, 2006] Noble, W. S. (2006). What is a support vector machine? *Nature biotechnology*, 24(12):1565–1567.

[Oshiro et al., 2012] Oshiro, T. M., Perez, P. S., and Baranauskas, J. A. (2012). How many trees in a random forest? In *International workshop on machine learning and data mining in pattern recognition*, pages 154–168. Springer.

[Osia et al., 2018] Osia, S. A., Taheri, A., Shamsabadi, A. S., Katevas, K., Haddadi, H., and Rabiee, H. R. (2018). Deep private-feature extraction. *IEEE Transactions on Knowledge and Data Engineering*, 32(1):54–66.

[Otterbacher, 2010] Otterbacher, J. (2010). Inferring gender of movie reviewers: exploiting writing style, content and metadata. In *Proceedings of the 19th ACM international conference on Information and knowledge management*, pages 369–378.

[Pagliardini et al., 2017] Pagliardini, M., Gupta, P., and Jaggi, M. (2017). Unsupervised Learning of Sentence Embeddings Using Compositional N-gram Features. *arXiv preprint arXiv:1703.02507*.

[Pennacchiotti and Popescu, 2011] Pennacchiotti, M. and Popescu, A. (2011). A machine learning approach to twitter user classification. In *Proceedings of the Fifth International Conference on Weblogs and Social Media, Barcelona, Catalonia, Spain, July 17-21, 2011*. The AAAI Press.

[Pennebaker and Stone, 2003] Pennebaker, J. W. and Stone, L. D. (2003). Words of wisdom: Language use over the life span. *Journal of personality and social psychology*, 85(2):291.

[Pennington et al., 2014a] Pennington, J., Socher, R., and Manning, C. D. (2014a). Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

[Pennington et al., 2014b] Pennington, J., Socher, R., and Manning, C. D. (2014b). 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*, pages 1532–1543.

[Pesce et al., 2012] Pesce, J. P., Casas, D. L., Rauber, G., and Almeida, V. (2012). Privacy attacks in social media using photo tagging networks: a case study with facebook. In *Proceedings of the 1st Workshop on Privacy and Security in Online Social Media*, pages 1–8.

[Pijani et al., 2020a] Pijani, B. A., Imine, A., and Rusinowitch, M. (2020a). Online Attacks on Picture Owner Privacy. In *Database and Expert Systems Applications - 31st International Conference, DEXA, Bratislava, Slovakia*, volume 12392 of *Lecture Notes in Computer Science*, pages 33–47. Springer.

[Pijani et al., 2020b] Pijani, B. A., Imine, A., and Rusinowitch, M. (2020b). 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]*, pages 1826–1834. ACM.

[Pohl et al., 2017] Pohl, H., Domin, C., and Rohs, M. (2017). Beyond just text: semantic emoji similarity modeling to support expressive communication. *ACM Transactions on Computer-Human Interaction (TOCHI)*, 24(1):1–42.

[Raileanu and Stoffel, 2004] Raileanu, L. E. and Stoffel, K. (2004). Theoretical comparison between the gini index and information gain criteria. *Annals of Mathematics and Artificial Intelligence*, 41(1):77–93.

[Random, 2018] Random (2018). How to fake your location for facebook check in.

[Rangel and Rosso, 2013] Rangel, F. and Rosso, P. (2013). Use of language and author profiling: Identification of gender and age. *Natural Language Processing and Cognitive Science*, 177.

[Rao et al., 2010] Rao, D., Yarowsky, D., Shreevats, A., and Gupta, M. (2010). Classifying latent user attributes in twitter. In *Proceedings of the 2nd international workshop on Search and mining user-generated contents, SMUC@CIKM 2010, Toronto, ON, Canada, October 30, 2010*, pages 37–44. ACM.

[Rish et al., 2001] Rish, I. et al. (2001). An empirical study of the naive bayes classifier. In *IJCAI 2001 workshop on empirical methods in artificial intelligence*, volume 3, pages 41–46.

[Ryu et al., 2013] Ryu, E., Rong, Y., Li, J., and Machanavajjhala, A. (2013). 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*, pages 13–18. ACM.

[Sahlgren, 2005] Sahlgren, M. (2005). An introduction to random indexing. In *Methods and applications of semantic indexing workshop at the 7th international conference on terminology and knowledge engineering.*

[Salamatian et al., 2015] Salamatian, S., Zhang, A., du Pin Calmon, F., Bhamidipati, S., Fawaz, N., Kveton, B., Oliveira, P., and Taft, N. (2015). Managing your Private and Public Data: Bringing Down Inference Attacks Against your Privacy. *IEEE Journal of Selected Topics in Signal Processing.*, 9(7):1240–1255.

[Santamaría and Mihaljevic, 2018] Santamaría, L. and Mihaljevic, H. (2018). Comparison and Benchmark of Name-to-Gender Inference Services. *The PeerJ of Computer Science.*, 4:e156.

[Sap et al., 2014] Sap, M., Park, G. J., Eichstaedt, J. C., Kern, M. L., Stillwell, D., Kosinski, M., Ungar, L. H., and Schwartz, H. A. (2014). Developing Age and Gender Predictive Lexica over Social Media. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP*, pages 1146–1151, Doha, Qatar. ACL.

[Sarawgi et al., 2011] Sarawgi, R., Gajulapalli, K., and Choi, Y. (2011). Gender attribution: tracing stylometric evidence beyond topic and genre. In *Proceedings of the fifteenth conference on computational natural language learning*, pages 78–86.

[Shen et al., 2017] Shen, Y., Rong, W., Jiang, N., Peng, B., Tang, J., and Xiong, Z. (2017). Word embedding based correlation model for question/answer matching. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 31.

[Shiha and Ayvaz, 2017] Shiha, M. and Ayvaz, S. (2017). The effects of emoji in sentiment analysis. *Int. J. Comput. Electr. Eng.(IJCEE.)*, 9(1):360–369.

[Shokri et al., 2012] Shokri, R., Theodorakopoulos, G., Troncoso, C., Hubaux, J., and Boudec, J. L. (2012). Protecting Location Privacy: Optimal Strategy Against Localization Attacks. In *the Conference on Computer and Communications Security, CCS ,12, Raleigh, NC, USA*, pages 617–627. ACM.

[Shutterstock, 2015] Shutterstock (2015). The psychology behind why we share on social media.

[Shutterstock, 2017] Shutterstock (2017). 6 types of images that elicit an emotional response.

[Thomas et al., 2010] Thomas, K., Grier, C., and Nicol, D. M. (2010). unfriendly: Multi-party privacy risks in social networks. In *International Symposium on Privacy Enhancing Technologies Symposium*, pages 236–252. Springer.

[Tian et al., 2019] Tian, Y., Niu, Y., Yan, J., and Tian, F. (2019). Inferring private attributes based on graph convolutional neural network in social networks. In *2019 International Conference on Networking and Network Applications (NaNA)*, pages 186–190. IEEE.

[Tobin and Merrill, 2018] Tobin, A. and Merrill, J. B. (2018). Facebook is letting job advertisers target only men.

[Tossell et al., 2012] Tossell, C. C., Kortum, P., Shepard, C., Barg-Walkow, L. H., Rahmati, A., and Zhong, L. (2012). A longitudinal study of emoticon use in text messaging from smartphones. *Computers in Human Behavior*, 28(2):659–663.

[Tsai and Hsiao, 2010] Tsai, C.-F. and Hsiao, Y.-C. (2010). Combining multiple feature selection methods for stock prediction: Union, intersection, and multi-intersection approaches. *Decision Support Systems*, 50(1):258–269.

[Tucker, 2014] Tucker, C. E. (2014). Social networks, personalized advertising, and privacy controls. *Journal of marketing research*, 51(5):546–562.

[Vergara and Estévez, 2014] Vergara, J. R. and Estévez, P. A. (2014). A review of feature selection methods based on mutual information. *Neural computing and applications*, 24(1):175–186.

[Volkova et al., 2015] Volkova, S., Bachrach, Y., Armstrong, M., and Sharma, V. (2015). Inferring latent user properties from texts published in social media. In *Twenty-Ninth AAAI Conference on Artificial Intelligence*.

[Wang et al., 2017] Wang, T., Blocki, J., Li, N., and Jha, S. (2017). Locally differentially private protocols for frequency estimation. In *26th {USENIX} Security Symposium ({USENIX} Security 17)*, pages 729–745.

[Weber and Castillo, 2010] Weber, I. and Castillo, C. (2010). The demographics of web search. In *Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval*, pages 523–530.

[Weinsberg et al., 2012] Weinsberg, U., Bhagat, S., Ioannidis, S., and Taft, N. (2012). Blurme: Inferring and obfuscating user gender based on ratings. In *Proceedings of the sixth ACM conference on Recommender systems*, pages 195–202. ACM.

[Weisberg et al., 2011] Weisberg, Y. J., DeYoung, C. G., and Hirsh, J. B. (2011). Gender differences in personality across the ten aspects of the big five. *Frontiers in psychology*, 2:178.

[Whitehill and Movellan, 2012] Whitehill, J. and Movellan, J. (2012). Discriminately decreasing discriminability with learned image filters. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 2488–2495. IEEE.

[Wijeratne et al., 2017a] Wijeratne, S., Balasuriya, L., Sheth, A., and Doran, D. (2017a). Emojinet: An open service and api for emoji sense discovery. In *Proceedings of the International AAAI Conference on Web and Social Media*, volume 11.

[Wijeratne et al., 2017b] Wijeratne, S., Balasuriya, L., Sheth, A., and Doran, D. (2017b). A semantics-based measure of emoji similarity. In *Proceedings of the International Conference on Web Intelligence*, pages 646–653.

[Wu and Chen, 2017] Wu, J. and Chen, Z. (2017). Human activity optimal cooperation objects selection routing scheme in opportunistic networks communication. *Wireless Personal Communications*, 95(3):3357–3375.

[Wu et al., 2017] Wu, S., Wieland, J., Farivar, O., and Schiller, J. (2017). Automatic alt-text: Computer-generated image descriptions for blind users on a social network service. In *Proceedings of the 2017 ACM Conference on Computer Supported Cooperative Work and Social Computing, CSCW*, pages 1180–1192, Portland, OR, USA. ACM.

[Wu et al., 2018] Wu, Z., Wang, Z., Wang, Z., and Jin, H. (2018). Towards privacy-preserving visual recognition via adversarial training: A pilot study. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 606–624.

[Yurchisin et al., 2005] Yurchisin, J., Watchravesringkan, K., and McCabe, D. B. (2005). An exploration of identity re-creation in the context of internet dating. *Social Behavior and Personality: an international journal*, 33(8):735–750.

[Zhao et al., 2019] Zhao, H., Chi, J., Tian, Y., and Gordon, G. J. (2019). Trade-offs and guarantees of adversarial representation learning for information obfuscation. *arXiv preprint arXiv:1906.07902*.

[Zhou and Chen, 2020] Zhou, Q. and Chen, G. (2020). An efficient victim prediction for sybil detection in online social network. *IEEE Access*, 8:123228–123237.

Résumé

Les réseaux sociaux contiennent de nombreuses informations personnelles telles que le genre, l'âge ou le statut d'une relation. Leur popularité et leur importance en font des cibles privilégiés pour des activités malveillantes menaçant la vie privée des utilisateurs. Les paramètres de sécurité disponibles sur les réseaux sociaux n'empêchent pas les attaques par inférence d'attribut, qui consistent pour l'attaquant à obtenir des données privées (comme le genre) à partir d'informations publiques. La divulgation d'une information personnelle peut avoir des conséquences négatives comme devenir la cible de spams, de harcèlements, ou se faire cloner son profil. Les techniques d'inférence les plus connues s'appuient soit sur l'analyse du comportement de l'utilisateur cible à travers ses préférences (e.g., likes) et ses groupes, soit sur ses listes d'amis. Cependant, en pratique, les informations disponibles pour ces attaques sont souvent limitées car beaucoup d'utilisateurs ont pris conscience des menaces et préfèrent protéger leurs données. Pour que les usagers des réseaux sociaux comprennent mieux les risques encourus par leur vie privée, nous introduisons dans cette thèse une nouvelle classe d'attaques par inférence sur les attributs de ces usagers. Nous montrons que ces attaques nécessitent très peu d'information. Elles s'appliquent même à des usagers qui protègent les attributs de leur profil ainsi que leurs commentaires. La méthode que nous proposons consiste à analyser les métadonnées d'une image publiée sur Facebook, à savoir i) les tags engendrés par Facebook pour décrire les images (e.g., pour les usagers malvoyants), et ii) les commentaires sous formes textuelle ou d'émojis déposés sous l'image. Nous montrons comment réaliser ces attaques sur un utilisateur de Facebook i) en appliquant une technique de retrofitting pour traiter le vocabulaire traité en ligne mais ne figurant pas dans la base d'apprentissage et ii) en calculant plusieurs plongements pour les unités textuelles (e.g., mot et émoji) chacun dépendant d'une valeur spécifique d'un attribut. Finalement nous proposons un mécanisme de protection qui sélectionne de manière rapide des commentaires à cacher toute en minimisant la perte d'utilité, définie par une mesure sémantique. Le système permet aux utilisateurs de vérifier s'ils sont vulnérables à des attaques par inférence et, le cas échéant de suggérer les commentaires à cacher pour prévenir ces attaques. Nous avons pu vérifier l'efficacité de l'approche par des expérimentations sur des données réelles.

Mots-clés: Réseaux Sociaux en Ligne, Attribut Privé, Attaques par Inférence en Ligne, apprentissage automatique, Photos, Emojis

Abstract

Online Social Networks (OSN) are full of personal information such as gender, age, relationship status. The popularity and growth of OSN have rendered their platforms vulnerable to malicious activities and increased user privacy concerns. Disclosure of personal information can have serious outcomes such as personal spam, bullying, profile cloning for malicious activities, or sexual harassment. The privacy settings available in OSN do not prevent users from attribute inference attacks where an attacker seeks to illegitimately obtain their personal attributes (such as the gender attribute) from publicly available information. Existing inference techniques are either based on the target user behavior analysis through their liked pages and group memberships or the target user friend list. However, in real cases, the amount of available information

to an attacker is small since users have realized the vulnerability of standard attribute inference attacks and concealed their generated information. To increase awareness of OSN users about threats to their privacy, in this thesis, we introduce a new class of attribute inference attacks against OSN users. We show the feasibility of these attacks from a very limited amount of data. They are applicable even when users hide all their profile information and their own comments. Our proposed methodology is to analyze Facebook picture metadata, namely (i) alt-text generated by Facebook to describe picture contents and (ii) commenters' words and emojis preferences while commenting underneath the picture, to infer sensitive attributes of the picture owner. We show how to launch these inference attacks on any Facebook user by i) handling online newly discovered vocabulary using a retrofitting process to enrich a core vocabulary that was built during offline training and ii) computing several embeddings for textual units (e.g., word, emoji), each one depending on a specific attribute value. Finally, we introduce a protection mechanism that selects comments to be hidden in a computationally efficient way while minimizing utility loss according to a semantic measure. The proposed mechanism can help end-users check their vulnerability to inference attacks and suggest comments to be hidden in order to mitigate the attacks. We have determined the success of the attacks and the protection mechanism by experiments on real data.

Keywords: Online Social Networks, Attribute Privacy, Online Inference Attacks, Machine Learning, Pictures, Emojis

Appendix A

Résumé de la thèse en français

A.1 introduction

Les attaques par inférence d'attribut sont des menaces émergentes pour la vie privée des utilisateurs dans les médias sociaux tels que Facebook et Twitter. Les attaques par inférence d'attribut se concentrent sur la déduction des attributs privés d'un utilisateur (par exemple, la localisation, le sexe, l'orientation sexuelle ou les opinions politiques) en exploitant ses données publiques, comme ses liens sociaux [Gong and Liu, 2016] ou ses comportements [Chaabane et al., 2012]. Le problème de déduire des attributs devient plus complexe lorsque les structures sociales et les comportements des utilisateurs ne sont pas disponibles, ce qui est souvent le cas dans la réalité.

Alors que de nombreux utilisateurs de Facebook cachent leurs attributs sensibles (par exemple, le genre, l'âge et l'opinion politique), les photos qu'ils publient sont toujours accessibles au public [Shutterstock, 2015]. Les utilisateurs des médias sociaux partagent des photos pour recevoir des commentaires sur leurs activités, en particulier de la part d'amis et de connaissances, car cela leur procure un sentiment de connectivité. Cependant, ils perdent le contrôle de la confidentialité sur leurs photos en raison d'informations supplémentaires (que nous appellerons métadonnées) qui s'ajoutent au cours du processus de publication. Pour toutes les photos mises en ligne, Facebook a mis en place un système de détection d'objets pour fournir automatiquement un ensemble de balises (tags), appelées alt-text, qui décrivent le contenu des images. Cette technique produit une description qui peut être traitée par un lecteur de texte destiné aux utilisateurs aveugles. De plus, lorsqu'ils observent une photo sur Facebook, les gens réagissent par des commentaires instinctifs pour exprimer leur sentiment. Les commentaires et les alt-texts générés automatiquement (métadonnées d'image) contiennent des informations potentiellement sensibles disponibles pour un attaquant.

Notre synthèse se décline en deux parties :

Dans la première partie, nous nous concentrons sur la réalisation d'attaques par inférence d'attribut. Nous montrons comment détecter le genre d'un utilisateur de Facebook *en ligne* à partir des métadonnées de ses images. L'attaquant acquiert d'abord des connaissances hors ligne en collectant des profils avec des attributs connus (dans notre cas, le genre) et utilise des techniques sophistiquées (par exemple de *traitement automatique des langues*) pour capturer des modèles et des structures à partir des données collectées. Lorsqu'un utilisateur est ciblé, l'attaquant peut explorer son profil et collecter en ligne des données complémentaires spécifiques pour mener son attaque. Les attributs inférés peuvent être utilisés pour faire de la publicité ciblée [Cadwalladr and Graham-Harrison, 2018] ou briser la confidentialité [Belinic, 2009].

Dans la deuxième partie, nous présentons des contre-mesures pour protéger l'utilisateur des

attaques par inférence de genre. Généralement, les approches préventives ajoutent du bruit aux données publiques afin de minimiser la précision des inférences. Des méthodes basées sur la théorie des jeux ont été utilisées contre les attaques par inférence d'attribut [Shokri et al., 2012]. Bien que ces méthodes garantissent une confidentialité théorique, elles sont coûteuses en temps de calcul [Jia and Gong, 2018]. D'autres méthodes heuristiques bruitent les données publiques corrélées aux attributs sensibles mais elles nécessitent une connaissance préalable des attributs privés d'un utilisateur pour calculer les corrélations et, surtout, elles engendrent une importante perte d'utilité [Jia and Gong, 2018]. Nous introduisons une application appelée *ProPic* pour atténuer les menaces d'inférence et suggérer des contre-mesures. Les menaces sont évaluées en collectant des images et leurs commentaires. Notre application ne nécessite pas l'accès au genre réel de l'utilisateur [Jia and Gong, 2018]. *ProPic* propose alors de masquer un ensemble de métadonnées à risque repérées par la phase de classification. Cette procédure, implémentée dans *ProPic*, peut être appliquée par les utilisateurs de Facebook à leurs profils pour vérifier leur vulnérabilité aux attaques par inférence d'attribut.

A.1.1 Énoncé du problème et des contributions

Notre premier objectif est de proposer une méthode pour déduire en ligne le genre de l'utilisateur cible à partir des métadonnées de l'image en apprenant les associations entre le genre du propriétaire de l'image et la structure du contenu des métadonnées.

Cette méthode s'applique même aux utilisateurs de Facebook qui sont prudents quant à leur vie privée et cachent tout type d'informations disponibles (par exemple, la liste d'amis, les pages favorites, les groupes et les attributs) sur leur profil. Les utilisateurs de médias sociaux veulent publier leurs photos sans risque sur la vie privée, mais en même temps, avec une perte minimale de visibilité sociale. Notre deuxième objectif est de proposer une stratégie de défense pratique repérant un sous-ensemble minimal de métadonnées qui doivent être masquées pour empêcher l'inférence de genre. Atteindre ces objectifs a nécessité de contourner certaines difficultés que nous expliquons ci-dessous.

Attaque. Les données collectées en ligne pour une attaque peuvent contenir des mots qui n'apparaissent pas dans l'ensemble de données d'apprentissage (*mots hors vocabulaire* ou *OOV* en abrégé).

Pour contourner ce problème, nous nous appuyons sur les vecteurs pré-entraînés par les logiciels, *word2vec* [Mikolov et al., 2013a], et *emoji2vec* [Eisner et al., 2016] (abrégés par *WE2V*). Cependant, ces vecteurs pré-formés doivent être adaptés lorsque nous visons à les appliquer à un domaine spécifique tels que les commentaires publiés sur Facebook. Nous avons utilisé la technique dite de *Retrofitting* [Faruqui et al., 2014] pour ajuster les vecteurs *WE2V* pré-entraînés ou pour calculer les vecteurs des*OOV*.

Défense. Une solution naïve pour sécuriser la vie privée de l'utilisateur consiste à supprimer tous les commentaires ou images. Cependant, cela diminue la visibilité sociale de l'utilisateur. Notre objectif ici est de trouver un ensemble de de métadonnées (tags et commentaires) qui, une fois masquées, modifient le résultat du classificateur avec un impact minimal sur l'environnement social de l'utilisateur. Pour cela, nous calculons les traits distinctifs avec notre classificateur et leur contribution exprimée par un poids. En additionnant le poids de tous les traits apparaissant dans un commentaire nous pouvons évaluer le risque qu'ils divulguent le genre de l'utilisateur.

Pour résumer, nous entraînons un classificateur pour prédire le genre de l'utilisateur. Nous évaluons empiriquement notre attaque sur un ensemble de données Facebook collectées aléatoirement dont 335 670 images avec leurs métadonnées. Lorsqu'un utilisateur souhaite évaluer le risque d'inférence du genre, il applique le classificateur Si l'utilisateur souhaite modifier la

prédiction du classificateur, *ProPic* sélectionne les commentaires à masquer, de manière à minimiser la perte d'utilité selon une mesure basée sur la sémantique distributionnelle. La méthode est efficace en temps de calcul et maintient les métadonnées modifiées aussi proches que possible des métadonnées d'origine (dans l'espace sémantique distributionnel), ce qui assure une perte minimale de visibilité sociale.

A.2 Attaque d'inférence

Dans ce travail, nous considérons un attaquant qui a l'intention de déduire le genre g d'un utilisateur de Facebook en analysant ses photos publiées. Nous considérons une situation binaire où $g \in \{femme, homme\}$ pour simplifier la présentation et par manque de données dans les autres cas.

A.2.1 Apprentissage hors ligne et sélection de traits

Apprentissage hors ligne

Dans cette sous-section, nous présentons les composants hors ligne que nous avons implémentés pour former notre algorithme d'apprentissage automatique. Nous utilisons les profils Facebook comme données de référence pour le processus d'apprentissage. Chaque profil collecté admet une valeur publique pour son attribut de genre (homme ou femme) et un ensemble d'images publiques avec leurs métadonnées [Alipour et al., 2019, Pijani et al., 2020b] comprenant les commentaires publiés par d'autres utilisateurs et le *alt-text* généré automatiquement par Facebook. Nous calculerons hors ligne un ensemble de traits caractéristiques $f^1, f^2, \ldots, f^n$ à partir de ces métadonnées et nous les appliquerons pour entraîner des classificateurs de genre.

Étant donné un utilisateur u on note f_u^i le nombre d'occurrences du trait i dans les métadonnées des images de u, par f_u sa liste de traits $\langle f_u^1, f_u^2, .., f_u^n \rangle$, et par $g_u \in \{female, male\}$ le label (ou classe) de u lorsque cette information est disponible.

En mode *hors ligne*, l'attaquant entraîne un algorithme d'apprentissage automatique avec des échantillons (f_u, g_u), pour tous les $u \in U_{training}$ comme entrées, où $U_{training}$ est un ensemble d'utilisateurs. Nos expériences [Pijani et al., 2020a] ont montré que la régression logistique est le classificateur le plus approprié pour notre tâche.

En mode *en ligne*, l'attaquant effectuera l'attaque sur un utilisateur cible choisi u_{new} en appliquant les classificateurs entraînés à sa liste de traits $f_{u_{new}}$ calculée en ligne à partir des métadonnées de u_{new}.

Sélection des traits

La sélection des traits est le processus d'identification des caractéristiques pertinentes corrélées aux variables d'intérêt (le genre, dans notre cas). Nous sélectionnons les traits en procédant en trois étapes :

1. Extraction de traits. Nous calculons un ensemble de n-grammes pour capturer les occurrences de mots/emojis dans les commentaires et les balises du alt-text apparaissant dans une fenêtre de taille donnée n (n) pour chaque genre. Ensuite, nous construisons une matrice de co-occurrence pour trouver la corrélation entre le genre, le alt-text et les commentaires reçus de manière à distinguer les femmes des hommes. Pour cela, nous considèrons des couples de termes w, w' qui sont des composants de n-grammes fréquents (éventuellement différents). Cela permet au processus d'inférence de bénéficier de certains mots, émoticônes et balises employés

simultanément pour une même image, même lorsqu'ils n'apparaissent pas dans la même fenêtre de taille n.

2. Informations mutuelles. Nous sélectionnons un sous-ensemble de l'ensemble des n-grammes précédent en calculant l'Information Mutuelle (MI) de ses éléments avec le genre. MI est une mesure statistique de l'(in)dépendance entre deux variables aléatoires [Cover, 1999].

3. Meilleur ensemble de traits. Enfin, nous appliquons *Chi-Square*, *Information Gain*, *Feature importance* et *Univariate feature selection* afin de générer un ensemble des meilleurs traits, appelé F_{best}. Dans notre étude de cas, F_{best} contient *1148* traits (simples ou combinaison de mots/emojis).

A.2.2 Retrofitting des vecteurs de mots/emojis

Lors d'une attaque en ligne, nous pouvons rencontrer des métadonnées d'image d'utilisateur avec peu ou pas de traits de F_{best}. Dans ce cas, nous pouvons exploiter des mots dans les métadonnées cibles qui sont sémantiquement proches de ceux de F_{best}. La représentation vectorielle des mots est un moyen puissant d'évaluer la proximité sémantique entre deux termes. Motivés par cela, nous dérivons des vecteurs pour chaque trait dans l'étape de pré-traitement hors ligne. On peut récupérer des vecteurs pour les traits à partir de vecteurs pré-entrainés dans *word2vec* et *emoji2vec*. Ces vecteurs pré-entrainés sont obtenus à partir de corpus (par exemple, Wikipedia) écrits dans des langages plus structurés que les commentaires des réseaux sociaux. En conséquence, certains termes de Facebook soit (i) ils n'existent pas dans les corpus utilisés par *word2vec* (OOV), soit (ii) ils ne peuvent pas être reconnus en raison d'une orthographe laxiste, soit (iii) ils admettent un changement de sens par rapport à leur utilisation dans *word2vec* et *emoji2vec*. Par conséquent, nous devons réajuster les vecteurs de traits lorsqu'ils existent dans *WE2V* ou les calculer à partir de vecteurs voisins lorsqu'ils n'existent pas dans *WE2V*. Nous utilisons la technique de *retrofitting* dans les deux cas, ce qui évite de relancer l'apprentissage des vecteurs à partir de zéro. Précisément, notre objectif est de créer un ensemble de représentations de traits qui tiennent compte à la fois de notre ensemble de données collectées hors ligne, *OCD*, et des représentations originales de mots/emoji apprises de *WE2V*.

A.2.3 Attaque en ligne

La phase d'attaque en ligne consiste à catégoriser un utilisateur cible de genre inconnu, homme ou femme. Cette phase procède par les étapes suivantes.

Pré-traitement et calcul des n-grammes

Nous calculons les n-grammes, pour $n \in \{1, 2, 3\}$, avec leurs fréquences dans les métadonnées des images de la cible. L'ensemble des n-grammes obtenu est appelé J_t. Nous notons $o(j)$ le nombre d'occurrences d'un n-gramme $j \in J_t$ dans les métadonnées des images de la cible.

Calcul de la table caractéristique de la cible

Ensuite, nous calculons la table d'une application co qui associe à chaque trait $f \in F_{best}$ une valeur entière $co(f)$. Cette valeur reflète l'importance du trait f dans les métadonnées de l'image cible. Toutes les entrées de la table, que nous appelons table caractéristique de la cible, sont initialisées à zéro.

Classification par genre

Dans cette étape, l'attaquant applique l'algorithme d'apprentissage automatique entraîné à la table caractéristique de l'utilisateur cible calculée dans la sous-section précédente A.2.3. Étant donné un utilisateur cible, l'algorithme le classe dans la catégorie *femme* ou *homme* qui a la probabilité de prédiction la plus élevée.

A.3 Protection du genre

Dans cette section, nous présentons notre méthode basée sur un mécanisme de masquage pour améliorer la confidentialité de l'utilisateur. Le défenseur suggère un ensemble de commentaires et/ou de texte alternatif (alt-text) que l'utilisateur doit masquer pour sécuriser son genre sans compromettre sa visibilité au sein du réseau social. Facebook propose deux options de filtrage des commentaires. Tout d'abord, les utilisateurs peuvent configurer une liste de mots, de phrases ou d'emojis qu'ils ne souhaitent pas recevoir des commentateurs. Facebook masque entièrement les commentaires contenant ces mots, phrases ou emojis des photos publiées. Deuxièmement, les utilisateurs peuvent sélectionner manuellement certains commentaires d'une photo et les rendre invisibles sous la photo.

A.3.1 Description de la protection

Soit $M = \{m_1, m_2, .., m_K\}$ dénotant les *métadonnées de l'image originale* où m_1 est un commentaire spécial contenant le texte alternatif (alt-text) et m_i $(1 < i \leq K)$ représente le i-ème commentaire de cette image. Dans la phase de protection, il se peut qu'un utilisateur ne souhaite être classé ni comme un homme ni comme une femme. Dans ce cas, nous devons créer un genre illusoire appelé *neutre*. Soit $\mathcal{C}$ la fonction de prédiction associée à notre classificateur où $\mathcal{C}(M) \in \{femme, homme, neutre\}$. Le résultat dépend du seuil de probabilité de prédiction. Nous avons fixé ce seuil à *0,70*. Par exemple, le résultat est *femme* (resp. *homme*) si l'algorithme donne une probabilité de prédiction de *0,70* à *femme* (resp. *homme*), et *0.30* à *homme* (resp. *femelle*). De plus, la sortie est *neutre* si la probabilité de prédiction de l'algorithme pour *femme* est *0,65*, et *0,35* pour *homme*.

Notre objectif est de trouver M' un sous-ensemble de M tel que $\mathcal{C}(M) \neq \mathcal{C}(M')$. Le sous-ensemble M', appelé *safe picture metadata*, satisfera les exigences de l'utilisateur en terme de protection du genre, après avoir masqué le sous-ensemble $M \setminus M'$.

Évaluation de la perte d'utilité

Les mécanismes de protection doivent produire des métadonnées d'image sûres M' qui préservent la sémantique des métadonnées d'origine M dans la plus large mesure possible, tout en changeant le résultat de notre classificateur. Pour ce faire, nous nous appuyons sur la distance sémantique entre les métadonnées originales et les métadonnées de l'image sécurisée. Nous capturons la sémantique des mots en réajustant les vecteurs word2vec (voir Section A.2.2). Nous exprimons la sémantique d'une image comme la moyenne de ses vecteurs de métadonnées. La perte d'utilité lors du masquage de certaines métadonnées de M pour obtenir M' est mesurée par la similarité cosinus entre $\vec{h_M}$ et $\vec{h_{M'}}$.

A.3.2 Trouver des métadonnées sûres

Notre méthode se déroule en trois étapes :

Propagation des étiquettes. En utilisant la *régression logistique* (voir Section A.4.1), chaque trait dans F_{best} (voir Section A.2.1) a un poids signé (le coefficient de régression) dont le signe indique la classification (positif pour les femmes, négatif pour les hommes) et dont la valeur représente la contribution du trait pour la classification. Nous entraînons notre modèle en utilisant *L2 Loss Function* pour calculer l'erreur et *Gradient Descent Algorithm* pour obtenir les valeurs des poids qui donnent le moins d'erreur. Notre classificateur de régression logistique utilisera ces pondérations pour calculer la probabilité qu'une personne soit un homme ou une femme, compte tenu des métadonnées de l'image.

Comme les utilisateurs doivent masquer les commentaires, nous utilisons ces poids pour attribuer une contribution aux métadonnées de l'image en additionnant tous les poids des traits qui y apparaissent. Les commentaires ou le texte alternatif reçoivent une contribution féminine (resp., masculine) s'ils ont un poids positif (resp., négatif) après sommation. Sinon, nous considérons que les commentaires ou le texte alternatif n'ont aucune contribution (ou contribution neutre). Cela se produit lorsque les mots/emojis/tags ne sont pas à l'intérieur de F_{best}, ou que le résultat de la sommation est nulle, car dans ce cas, les caractéristiques féminines et masculines se trouvent être équilibrées. Avec ces informations, nous générons deux ensembles, que nous appelons respectivement ensemble de commentaires *lié* et *non lié* R et UR. L'ensemble R contient des commentaires qui confirment/expliquent le résultat du classificateur (commentaires risqués), et UR contient des commentaires de genre opposé ou de contribution neutre. Nous trions les métadonnées de R de la moins contributive à la plus contributive.

Processus de masquage. Notre objectif est de protéger les informations sur le genre en réduisant la visibilité des métadonnées M si M divulgue des informations sur le genre. Une solution naïve serait de cacher tous les éléments de R pour réduire la prédiction de l'attaquant à une supposition aléatoire. Cependant, cela conduirait à une augmentation de la perte d'utilité pour l'utilisateur. Pour préserver l'utilité, il faut garder visible autant de commentaires que possible. De plus, trouver des métadonnées sécurisées parmi les nombreuses combinaisons (pour changer le résultat du classificateur) serait coûteux en calcul. Nous proposons une méthode avec un coût de calcul raisonnable tout en assurant un compromis entre confidentialité et visibilité.

Nous initialisons M' par UR et ajoutons itérativement le commentaire le moins contributif de R à M' tant que le vecteur de M' ($\vec{h_{M'}}$) se rapproche du vecteur de M ($\vec{h_M}$) dans l'espace vectoriel et satisfait $\mathcal{C}(M) \neq \mathcal{C}(M')$. Notre mécanisme de protection trouve facilement un ensemble de métadonnées sécurisées en ordonnant de manière croissante les éléments de R selon leur contribution. Il est logique d'essayer d'ajouter en priorité les éléments de R les moins contributifs (au genre à cacher), car ils ont moins de chance d'orienter le classificateur vers le genre à cacher.

Module de Recommandation. Le module de recommendation comprend notre classificateur entrainé et un vérificateur de condition. Le vérificateur de condition vérifie que UR préserve la confidentialité du genre tout en conservant la sémantique des métadonnées de l'image. Le *module de recommandation* calcule la similarité entre les métadonnées originales (M) et les métadonnées non masquées (M') à partir de leurs représentations vectorielles ($\vec{h_M}$, et $\vec{h_{M'}}$ respectivement).

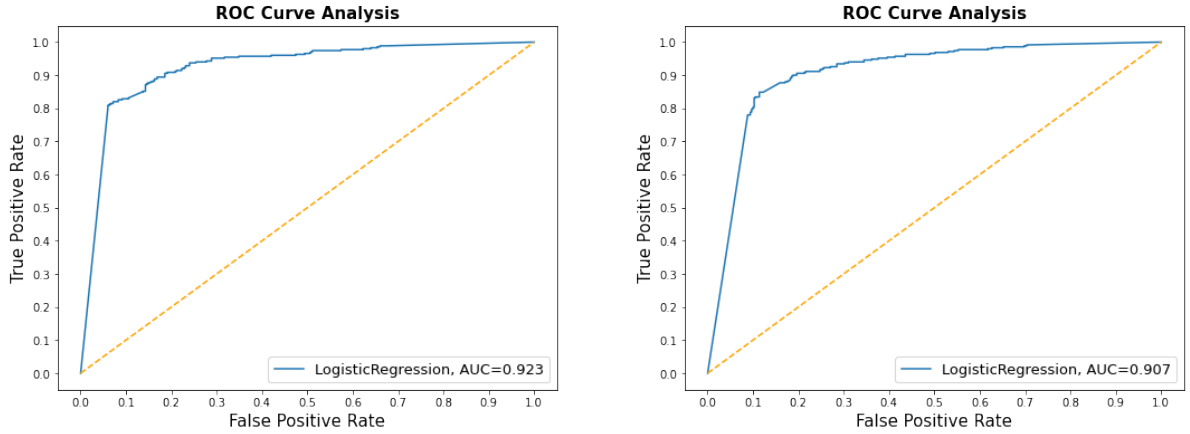
A.4 Expériences

Dans cette section, nous évaluons les attaques par inférence d'attribut et notre proposition de protection du genre.

A.4.1 Évaluation des attaques

Tests hors ligne et en ligne. Nos expériences ont été réalisées sur des données collectées de Facebook. Bien que ces données soient publiques, elles peuvent conduire à déduire des informations privées, et nous nous sommes donc engagés à les stocker de manière sécurisée et à ne les utiliser que pendant le temps nécessaire à la réalisation de nos travaux de recherche. À l'aide d'un crawler Python, nous avons collecté de manière aléatoire *628 076* images et leurs *1 333 280* commentaires. Pour évaluer le classificateur, nous avons sélectionné le même nombre d'hommes et de femmes pour éviter les biais de classification. Nous choisissons la taille du "train-test" à *70-30*, ce qui donne la meilleure précision. Pour résoudre le problème de l'estimation équitable des performances de chaque classificateur, nous avons mis de côté un ensemble de données de validation. Nous formons et ajustons les hyper-paramètres pour optimiser les performances du classificateur en utilisant cet ensemble de données. Finalement, nous évaluons le classificateur sur l'ensemble de données de test. Considérant le genre extrait comme étant exact, pour évaluer notre attaque, nous calculons la courbe *AUC-ROC* (Figure A.1), qui est une mesure de la performance pour les problèmes de classification définis par un seuil. Nous appliquons la *régression logistique* telle que présentée dans A.1 pour évaluer notre modèle d'attaque. Notre expérience [Pijani et al., 2020a] a montré que la *régression logistique* se prête bien pour cette tâche. Nous avons appliqué notre attaque en ligne sur *700* utilisateurs avec leurs *21 713* images et leurs *64 940* commentaires correspondants, comme illustré dans la Figure A.1 (b).

Pour résumer, la *régression logistique* est un classificateur approprié pour cette tâche qui peut être entraîné par un attaquant pour effectuer une attaque par inférence de genre. Les résultats confirment notre hypothèse selon laquelle le genre et le contenu des images ont un impact sur les réponses émotionnelles des utilisateurs de Facebook.



(a) (b)

Figure A.1: Résultat AUC de la régression logistique entrainée sur : (a) un ensemble de données hors ligne (b) un ensemble de données en ligne

A.4.2 Évaluation de la protection

Nous avons évalué les performances de *ProPic* sur 700 utilisateurs en ligne (et *21 713* images). La Figure A.2 et le Tableau A.1 représentent l'impact de notre modèle de protection sur les performance de l'attaque en ligne avec un classificateur de type *régression logistique*. Ceci est

	precision	*recall*	*f1-score*
class Female	0.09	0.09	0.09
class Male	0.13	0.12	0.12
macro avg	0.11	0.11	0.11
weighted avg	0.11	0.11	0.11

Table A.1: Mesures du niveau de protection

illustré en comparant le genre exact (déclaré par les utilisateurs) et la sortie du classificateur sur les métadonnées M' de l'image sécurisée calculées par *ProPic*. Nous notons que l'AUC et d'autres mesures chutent considérablement, ce qui démontre l'efficacité de *ProPic* dans la préservation de la confidentialité des utilisateurs.

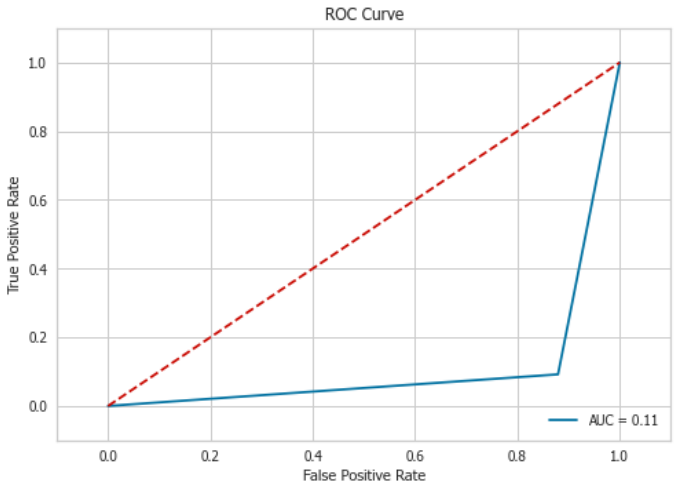
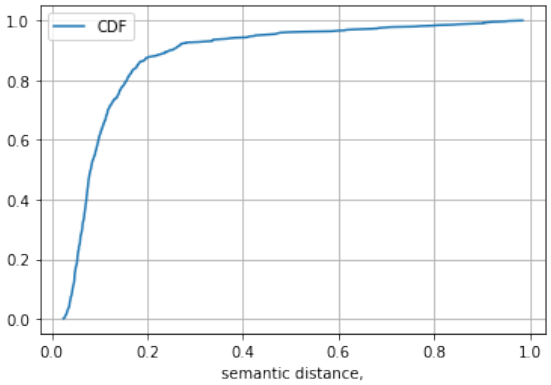


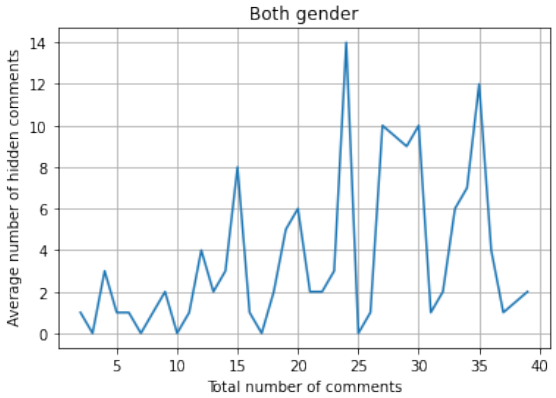
Figure A.2: Précision (AUC) de l'attaque après protection

Les utilisateurs de Facebook veulent garder public un sous-ensemble de commentaires M' tout en (i) s'assurant que l'attaquant est incapable de déduire leur genre et (ii) en minimisant la perte d'utilité. Figure A.3 (a) montre la fonction de distribution cumulée de la perte d'utilité minimale. Ceci confirme le fait que notre méthode de protection trompe le classificateur de l'attaquant mais garde en même temps une utilité suffisante car l'ensemble protégé M' est sémantiquement proche de l'ensemble original M. Figure A.3 (b) montre le nombre moyen de commentaires à masquer parmi les métadonnées de l'image d'origine. Nous observons que le nombre de commentaires masqués dépend du contenu des commentaires et non du nombre de commentaires reçus. Par exemple, le nombre moyen de commentaires masqués pour un utilisateur avec *20* commentaires est *6*, tandis que *2* est le nombre de commentaires à masquer (en moyenne) pour un utilisateur avec *23* commentaires. Figure A.3 (c) représente le temps d'exécution qui est un facteur essentiel pour la performance du système. Cela montre comment le temps d'exécution augmente avec le nombre de commentaires à masquer.

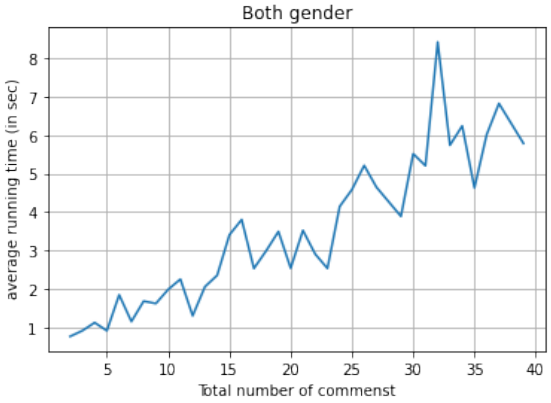
Il convient de noter que nos expériences peuvent être encore optimisées en utilisant des langages de programmation ou des bibliothèques plus efficaces.



(a)



(b)



(c)

Figure A.3: (a) Distribution cumulée (CDF) de la perte minimale d'utilité, i.e., distance sémantique, pour une confidentialité maximale, (b) Moyenne du nombre de métadonnées à cacher par rapport au nombre original de métadonnées (x-axis) (c) Temps moyen d'exécution de ProPic (par instance) par rapport au nombre de métadonnées à cacher pour protéger le genre du propriétaire de la photo.

A.5 Discussion

En appliquant notre système, un utilisateur de Facebook peut vérifier s'il est vulnérable aux attaques par inférence d'attribut. Notre attaque peut réussir même lorsque la cible ne publie rien d'autre que des images. Pour empêcher les violations de la confidentialité mentionnées ci-dessus, *ProPic* masque les commentaires ou le alt-text lorsqu'ils contribuent fortement à l'attaque, comme expliqué dans la sous-section A.3.2. Le but de *ProPic* est de suggérer des commentaires à masquer pour minimiser la précision de l'inférence avec une petite perte d'utilité pour l'utilisateur. Notre travail a plusieurs limitations et perspectives. Nous pourrions utiliser un modèle de sujet (topic model) ou bien l'analyse de sentiments afin de déduire de nouveaux traits à exploiter pour la prédiction d'un attribut. Nous pourrions également utiliser l'apprentissage profond (e.g., BERT) pour réaliser les inférences. Nous pourrions exploiter la date de publication des commentaires, afin de mieux comprendre leurs liens et leur utilité.

A.6 Conclusion

L'identification du genre des utilisateurs à partir de leurs activités en ligne et de leur comportement de partage de données est une question importante dans le domaine de recherche en plein essor des réseaux sociaux. Elle offre une opportunité de publicité ciblée, de personnalisation de profil ou d'attaques contre la vie privée. Sur la base de l'analyse intensive des *628 076* images partagées et de leurs *1 333 280* commentaires que nous avons collectés, ce travail a démontré la possibilité d'une attaque par inférence de genre même lorsque tous les attributs/activités de l'utilisateur tels que les attributs de profil, la liste d'amis, les pages favorites et les groupes sont masqués. De plus, nous avons proposé un modèle de protection de la vie privée, à savoir *ProPic* qui offre un compromis optimal entre utilité, confidentialité et efficacité. Comme travaux futurs, nous prévoyons de tirer parti de la combinaison de contenus engendrés par les utilisateurs présents sur plusieurs réseaux sociaux en ligne pour en déduire des attributs privés.