2021 18th International Conference on Privacy, Security and Trust (PST) | 978-1-6654-0184-5/21/$31.00 ©2021 IEEE | DOI: 10.1109/PST52912.2021.9647778

FOX: Fooling with Explanations

Privacy Protection with Adversarial Reactions in Social Media

Noreddine Belhadj-Cheikh, Abdessamad Imine and Michaël Rusinowitch
Lorraine University and Loria-Cnrs-Inria Nancy-Grand Est
Nancy, France
firstname.lastname@loria.fr

***Abstract*—Social media data has been mined over the years to predict individual sensitive attributes such as political and religious beliefs. Indeed, mining such data can improve the user experience with personalization and freemium services. Still, it can also be harmful and discriminative when used to make critical decisions, such as employment. In this work, we investigate social media privacy protection against attribute inference attacks using machine learning explainability and adversarial defense strategies. More precisely, we propose FOX (FOoling with eXplanations), an adversarial attack framework to explain and fool sensitive attribute inference models by generating effective adversarial reactions. We evaluate the performance of FOX with other SOTA baselines in a black-box setting by attacking five gender attribute classifiers trained on Facebook pictures reactions, specifically (i) comments generated by Facebook users excluding the picture owner, and (ii) textual tags (i.e., alt-text) generated by Facebook. Our experiments show that FOX successfully fools (about 99.7% and 93.2% of the time) the classifiers, outperforms the SOTA baselines and gives a good transferability of adversarial features.**

***Index Terms*—Social Networks, Privacy, Adversarial Machine Learning, Explainability.**

I. Introduction

Online Social Network (OSN) platforms have attracted billions of users resulting in a massive amount of data. For instance, every 60 seconds on Facebook: 510,000 comments are posted, 293,000 statuses are updated, 4 million posts are liked, and 136,000 photos are uploaded [20]. This large amount of data jeopardizes individuals' privacy due to its richness of content, including users' behavior, relationships, reactions, and other private information. The risk can expand up to the traceability and identification of users. Indeed, users remain highly vulnerable to *attribute inference attacks* which consist in gaining illegitimately private attributes (such as gender or age) from publicly available data using classical or deep learning models. This work focuses on attribute inference attacks in OSNs and proposes a framework to mitigate the risks inherent in such attacks.

Popular OSNs like Facebook, Instagram, and Twitter rely on online advertising to be profitable and keep their services free of charge. They enable advertisers to target potential clients among billions of users. Users' data is therefore considered as an important asset to build effective and profitable ad targeting platforms. In the meantime, OSNs often provide options for users to control who can view their data, including profile attributes and their activity on the platform. Nevertheless, online advertising raises privacy and security concerns due to misleading and contradicting privacy settings of some OSNs [1], [4]. For example, Facebook enables users to define explicitly which attributes, and interest categories can be used by advertisers to target them. However, Facebook overrides the user setting and targets users even through unwanted interests (see Fig.1).

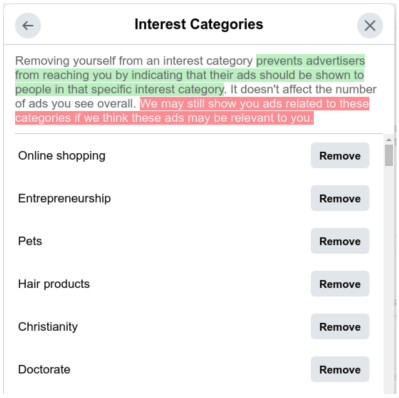


Figure 1: A contradicting statement in one of Facebook's privacy control settings. Facebook states that this setting is intended to prevent advertisers from targeting a user with an unwanted interest category (green), whereas Facebook retains the right to target the user with the unwanted interest category (red).

In addition to the deceitful privacy management in OSNs, users became highly threatened by sensitive attribute inference attacks due to the development of strong models (e.g., deep learning models) that benefited from tremendous amounts of user-generated data in social media platforms and the low cost of computational power.

Although working directly on data generated by target users is straightforward and practical, users can be more cautious about their activity and the data they post to limit risks to their privacy. In [16], the authors showed it is possible to infer sensitive attributes even when the target user is discreet about those attributes by mining reactions that the target user received on their pictures (e.g., comments from friends). Reactions come from other users and/or the OSN platform (e.g., Facebook picture tagging and Reddit bots). Data received from other users (e.g., friends) is less manageable than data

This work is partially supported by DIGITRUST (http://lue.univ-lorraine.fr/fr/article/digitrust/).

generated by users themselves. For instance, a user may avoid words that leak information about their gender. However, it is not practical to ask all users who react to their pictures to do the same. As a result, the threat of sensitive information leakage by the reactions of others is more critical.

Despite the performance of Deep Neural Networks (DNN), research efforts in adversarial attacks have shown that they are vulnerable to strategically crafted inputs, named *adversarial examples* [3]. In particular, adversarial attacks against images witness notable success. However, the advancement of adversarial attacks techniques in Computer Vision is not directly applicable to neural networks trained for Natural Language Processing (NLP), where the textual input space is discrete.

In an attempt to protect OSN users from attribute inference attacks on non-user generated data, we propose FOX (FOoling with eXplanations), a framework based on explainability tools (e.g., LIME [19]) that generates adversarial reactions to attack sensitive attributes black-box classifiers. A user who wants to protect their profile against attributes inference attacks may employ fake profiles or ask a list of trusted friends to add the reactions recommended by FOX. Fig.2 illustrates an attack performed by FOX.

We formally define the problem as follows. Let x be a publication with a set of reactions R that x received. Given x and R, let h be the target classifier mapping x to some class c. Our goal is to find the adversarial reactions R^{adv} which when appended to the original reactions R guarantees that $h(x, R) \neq h(x, R \cup R^{adv})$. In addition to a successful attack, the generated adversarial reactions must also adhere to constraints corresponding to their textual nature (e.g., grammatical well-formedness) and be stealthy, thus undetected by adversarial attacks defense systems. Moreover, the protection framework should ensure adaptability to different target classifiers (e.g., Neural Architectures, Logistic Regression, and Ensembles).

Our main contributions in this work are as follows:

- To our best knowledge, we introduce the first method that prevents sensitive attribute inference attacks against OSN users by protecting their publications with adversarial reactions.
- For that, we develop a framework using LIME explanations to perform adversarial attacks. Our framework can fool multiple types of models, including neural models (e.g., DNN), classical models (e.g., logistic regression), and their stacking.
- As a real-world application, we have trained deep neural network classifiers for gender attribute inference on the following Facebook picture reactions: (i) comments posted by friends, friends of friends, or strangers, and (ii) textual tags (i.e., alt-texts) generated by Facebook to describe the content of pictures. Note that gender attribute is highly relevant for targeted advertising. Furthermore, to adhere to a realistic black-box setting without prior information about the classifier, we have to consider models with different architectures such as Recurrent Neural Networks (RNNs), Convolutional Neural Networks (CNNs), Attention-based, Ensembles, and logistic regression. Experimental results show that FOX is very effective.
- We further investigate the transferability of the extracted features for one model to attack another model. Experimental results demonstrate that FOX gives a good transferability of adversarial features.

This paper is organized as follows. In Section II, we review the related works. Section III describes our framework FOX. We present our case study in Section IV and the experimental evaluation in Section V. Section VI draws limitations of our approach. We conclude with future works in Section VII.

II. Related Work

A. Attribute Inference Attacks

A user's profile may contain self-identifying attributes such as gender, hobbies, age, visited places, and political views. Social networks usually offer users the possibility to limit the visibility of their attributes as means of controlling their privacy (e.g., being visible to friends only). Therefore, the network is a mixture of users with private and public information. The attack's goal is to infer the private or missing attribute of a user by leveraging publicly available data. The attacker could be the social network service provider itself, advertisers, criminals, and data brokers. Most works on attribute inference attacks belong to three groups: friend-based [8], behavior-based [22], or their combination [7]. In addition, there exist works that leverage other sources of information such as writing style, liked pages, and group subscriptions. Authors in [16] took a different approach where they utilize reactions (e.g., comments generated by friends and Facebook-generated alt-text) received on a target user-posted picture.

B. Attacking Profile Attribute Predictors

Several works study adversarial attacks that are specific to user-generated data, but very few consider attacks on profile attribute classifiers using reactions generated by other users. [2] brings into the problem of *evasion and poisoning* attacks on profile attributes inference models based on user's Facebook liked pages. In evasion attacks, the user edits their profile information (Facebook liked pages) by adding or removing features (Liking/Unliking pages) that contribute to a particular attribute in order to interfere with the prediction of the target classifier. But, to succeed in poisoning attacks, [2] requires the manipulation of 1% of the training data to decrease the overall target classifier performance. Note that the training data is generally not accessible, especially for black-box models. In [11], the authors study a problem close to ours where the goal is to generate and add malicious comments to a news article causing a neural fake news detector to misclassify it as REAL/-FAKE news. They propose MALCOM, a system comprising a Conditional Text Generator Module for comments generation and a Style Discriminator Module to preserve syntax and coherency between the article's content and the generated

(a) The picture owner is classified as **Male**.

(b) The picture owner is classified as **Female** after adding the red comment.

Figure 2: An adversarial comment generated by FOX misleads a gender attribute classifier to predict a Male picture owner as Female.

comments. Despite MALCOM's good performance, one of the conditions for a black-box setting is violated when training the system and the target models with 100% overlapping data, which should not be the case in a realistic black-box setting. To respect this condition, we introduce a portion of non-overlapping data to train the target models. Unlike [2], [11], we attack the target classifiers with multiple reaction types (Comments and Facebook Alt-Text), which are not generated by the target user, and our attack is not limited to neural models only. Moreover, the authors of [11]' do not provide detailed information about the system components nor release the code as they promised upon acceptance of their paper. Consequently, we were unable to implement and apply it to our case study for an accurate comparison.

C. Adversarial Text Generation

Generating text under an adversarial setting, i.e., to attack ML classifiers, is more challenging due to the discrete nature of the text [25]. Although there has been many efforts to construct adversarial samples to attack text-based ML models [10], [12], [13], most of them focus on making minimal modifications (e.g., addition, removal, or replacement,) in character [12], [13] or word level [10], [12] of a span of text, through either a set of predefined templates [12] or a search mechanism with constraints [10], [13]. Despite the success of these methods, they are only designed for attacking static features, such as altering the text of a review to fool a sentiment analysis classifier. They are not developed for dynamic sequential input like comments where content can be added over time. Moreover, these attack methods primarily require existing content and are not adapted to cases where the initial content alteration is not possible (e.g., a Facebook user cannot modify comments of others). Therefore, we propose a framework to generate adversarial reactions with strategies that can effectively attack static input (e.g., alt-text) and dynamic sequential inputs where the initial input should not be altered (e.g., comments).

III. Fooling with Explanations: FOX Framework

We propose a robust framework which, given an initial dataset X (e.g., Facebook pictures reactions) and an explainability tool (e.g., LIME), can extract strong adversarial features $\mathcal{E}^{strong}$ from reactions. These features will be used to construct adversarial reactions to fool black-box classifiers and prevent them from threatening social media users' privacy. In the following, we describe FOX framework, and all used notations are listed in Table I.

Step 1 - Building strong features list: FOX requires a dataset X (e.g., a set of Facebook pictures with their reactions) to be collected. Each instance $x \in X$ admits a set of reactions R_x (e.g., comments, alt-text, likes for a given picture) and each reaction $r \in R_x$ admits a set of features F_r (e.g., words can be the features for a textual reaction such as comments). We assume that different reaction types admit disjoint sets of features.

Given a set of classes $Y = \{y_1, y_2, \dots y_{|Y|}\}$, a classifier $h : X \rightarrow [0,1]^{|Y|}$ maps x to a vector $h(x) = [p_1, p_2, \dots p_{|Y|}]$ where $h(x)[i] = p_i$ is the probability that x belongs to class y_i. We denote by $h_l(x)$ the class label of x, that is the class y_i with the highest probability p_i. We assume a black-box setting for h and therefore its underlying architecture, parameters and the training dataset are unknown. We need to identify the most

Symbol	Description
X	Dataset used for strong features extraction
x	Instance of a dataset (e.g., picture)
x_{adv}	Adversarial instance
R_x	Reactions set of x
F_r	Features set of reaction r
$h(.)$	Black-box classifier
$h_l(.)$	Class label returned by the black-box classifier
$E(.)$	Explainability tool
$\mathcal{E}$	Features set extracted from X by $E(.)$
$\mathcal{E}^{strong}$	Features filtered from $\mathcal{E}$
t	Instance of $\mathcal{E}$. It is a tuple containing the value and contribution class denoted $\langle f, c\rangle$
t_{mc}	Most contributing feature
$\mathcal{P}$	Set of perturbed instances of x
Y	Set of classes
y'	Target class
I	Maximal number of iterations
N	Maximal number of features per iteration
$\mathcal{A}$	Set of all selected features to perform an attack
$\mathcal{A}'$	Remaining features for an attack
$\mathcal{A}^{subset}$	Set of features used in an iteration
$\mathcal{A}^{subset}_{comments}$	Set of comments features

Table I: Notations

contributing features to the different classes. Let us define $\mathcal{G}_k(E, x, c, h)$ to be the set of couples $\langle f, c\rangle$ such that the contribution of feature f in x for c is among the top_k values (see Fig.3 for an illustration). We also introduce $\mathcal{E}_k(E, x, h) = \bigcup_{c\in C} \mathcal{G}_k(E, x, c, h)$ and $\mathcal{E} = \bigcup_{x\in X} \mathcal{E}_k(E, x, h)$

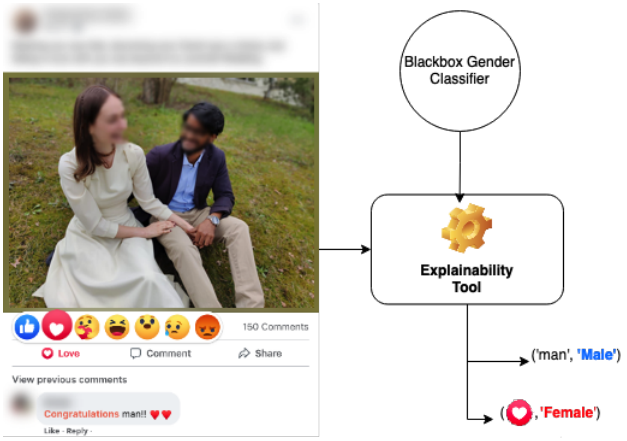
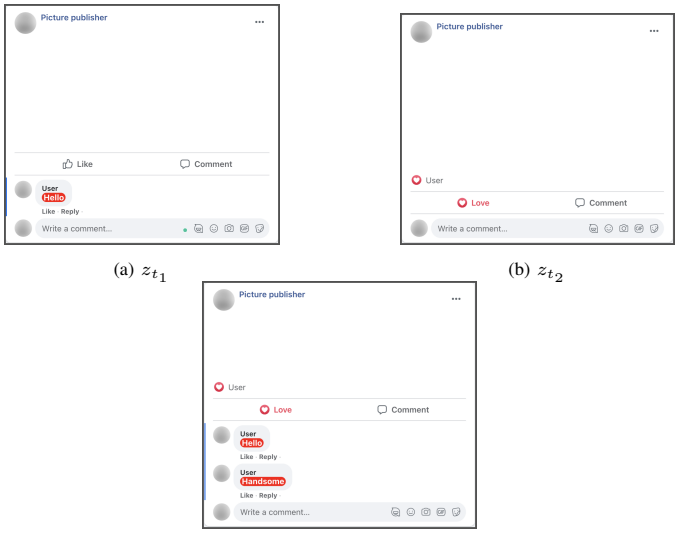


Figure 3: In this example, we show the top_1 features of each class (Male, Female) returned by $\mathcal{E}_1(E, x, h)$ where E is the explainability tool, and h is a black-box gender classifier that takes textual reactions (comments) and the emotional reactions (, , etc.) of a Facebook picture x as input to infer the gender attribute of the picture's publisher.

We do a further filtering of the features list $\mathcal{E}$ to obtain the *strong features list* $\mathcal{E}^{strong}$ as follows. Let T be a sequence of elements of $\mathcal{E}$: $t_1, t_2, \ldots, t_s$, We define z_T to be a crafted instance that contains $f_1, f_2, \ldots, f_s$ in this order (see Fig.4 for crafted instances example), where $t_i = \langle f_i, c_i\rangle$. Fonction $h_l(z_T)$ computes the class label of z_T. Then,

$$\mathcal{E}^{strong} = \{\langle f, c\rangle | c = h_l(z_{\langle f,c\rangle}), \langle f, c\rangle \in \mathcal{E}\} \quad (1)$$



(a) z_{t_1} (b) z_{t_2} (c) z_{t_1,t_2,t_3}

Figure 4: Let $\mathcal{E}$=[('Hello', 'Male'), (, 'Female'), ('Handsome', 'Male')]. Subfigures (a), (b), and (c) illustrate the crafted facebook pictures, containing reactions constructed from the features tuples $\{t_1\}$, $\{t_2\}$, and $\{t_1, t_2, t_3\}$ respectively.

Step 2 - Poisoning the input: Once we have constructed $\mathcal{E}^{strong}$ we are ready to poison the input in order to fool h. As shown in Algorithm 1, we first get the most contributing feature f_{mc} to the original label y of x using the explainability tool. Hence we have:

$$t_{mc} = \langle f_{mc}, y\rangle = \mathcal{G}_1(E, x, y, h) \quad (2)$$

Then we check if t_{mc} exists in $\mathcal{E}^{strong}$, otherwise we compute its most similar using a feature type-specific similarity function (e.g., for words, we may use cosine similarity of their embeddings computed by a language model). After having t_{mc} we choose the adversarial features according to Equation 3 to construct $\mathcal{A}$ where y' is the target label,

$$\mathcal{A} = \{t | h_l(z_{t_{mc},t}) = y', t \in \mathcal{E}^{strong}\} \quad (3)$$

For each iteration i we compute $\mathcal{A}^{subset}$ that contains the top_N features according to a scoring function $S(.)$. We then perturb x by creating adversarial reactions R^* from the features in $\mathcal{A}^{subset}$ and concatenate it with the set of reactions R of the original instance x resulting in a set of perturbed instances $\mathcal{P}$. The *concatenation* and the *perturb* functions depend on the feature's reaction type and the original reactions of x. For example, perturbing a publication with a feature extracted from a textual reaction such as a comment differs from perturbing it with a feature extracted from an emotional reaction. Hence, these functions are engineered in accordance with the type of reactions and their generation constraints. For instance, a comment should be grammatically well-formed,

Algorithm 1: Fooling by adding adversarial reactions

FOX $(x, E, \mathcal{E}^{strong}, h, y', I, N, S)$

Input: x instance to protect,E explainability tool, $\mathcal{E}^{strong}$ Feature tuples set, h classifier, y' target class, I maximum number of iterations, N maximum number of selected features for each iteration, S features scoring function

Output: x_{adv} adversarial instance

$t_{mc} \leftarrow \mathcal{G}_1(E, x, h_l(x), h)$ `// most contributing feature`

$\mathcal{A} \leftarrow \{t | h_l(z_{t_{mc},t}) = y', t \in \mathcal{E}^{strong}\}$ `// adversarial features`

if $t_{mc} \notin \mathcal{E}^{strong}$ **then**

  replace t_{mc} with a similar feature tuple from $\mathcal{E}^{strong}$

end

$\mathcal{A}^{'} \leftarrow \mathcal{A}$;

$i \leftarrow 1$;

while $i \leq I$ *and* $\mathcal{A}^{'} \neq \emptyset$ **do**

  $\mathcal{A}^{subset} \leftarrow \text{top}_N(S, \mathcal{A}^{'})$;

  $\mathcal{A}^{'} \leftarrow \mathcal{A}^{subset} \setminus \mathcal{A}^{'}$;

  $\mathcal{P} \leftarrow perturb(x, \mathcal{A}^{subset})$;

  $x_{adv} \leftarrow \underset{p \in \mathcal{P}}{argmax}\ h(p)[y']$;

  if $h_l(x_{adv}) = y'$ **then**

    return x_{adv}; `// solution found`

  else

    if $h(x_{adv})[y'] > h(x)[y']$ **then**

      $x \leftarrow x_{adv}$;

      $\mathcal{A}^{'} \leftarrow \mathcal{A}$;

    end

  end

  $i \leftarrow i + 1$

end

return *None*;

and the same user cannot perform two types of emotional reactions for the same publication [1].

After generating $\mathcal{P}$, we search for the instance $p \in \mathcal{P}$ with the maximal change towards y'. If h classifies x_{adv} as y', we return it, and the protection is considered successful. Otherwise, if x_{adv} has a more significant confidence score of y' than x, we set x to be x_{adv} to perturb it in the next iteration. It is possible that the current $\mathcal{A}^{sub}$ neither succeeds in changing the label to y' nor increases the confidence score of y'. Thus, if $\mathcal{A}^{'}$ is not empty and we did not reach the maximum number of iterations I, then we repeat the process with the remaining features.

The stop conditions of FOX are determined by these cases: 1)-successfully fooling the classifier, 2)-reaching the maximum number of iterations I without success 3)- exhausting all adversarial features without success.

In the following section, we will demonstrate FOX efficacy to protect Facebook pictures from gender inference attacks against different classifiers in a black-box setting. Then, we will compare it with other SOTAs methods.

[1]This constraint applies to Facebook publications where a Facebook user can either react with a Like, Love, Wow, Angry, Sad once only. In contrast, the Reddit platform does not have this constraint.

#pictures	#comments	#male	#female
42,706	194,768	21,353	21,353

Table II: Gender Dataset statistics

IV. Case study: Protecting Facebook Pictures

We have applied our adversarial approach to Facebook pictures and reactions they generate, namely (a) friends' comments (b) picture's alt-text, which is a textual description of the picture generated automatically by Facebook [24]. The objective is to protect a Facebook user against gender [2] inference classifiers trained on their picture reactions. We evaluate FOX with four classifiers of different architectures that we trained on a labeled dataset detailed in the following. Note that in adversarial attack works, the attacker is the party performing the adversarial attack against a model. Whereas, in our case, we consider the attacker to be the party performing an attribute inference attack. Here we utilize adversarial strategies as a protection solution.

A. Dataset

Our experiments are performed on a dataset collected from Facebook by [16]. It consists of 4509 user-profiles and their pictures. Since we are interested in protecting against gender inference attacks, we have filtered out all the user profiles missing the gender label from the dataset. We have removed non-English comments using a language detection tool [15]. We chose to train on reactions that are not generated by the owners of the pictures: the comments given by their friends on their picture and the alt-text description of this picture. We utilized weighted random oversampling to balance the number of pictures of female and male classes; The number of comments in a picture is considered the weight. The resulting preprocessed dataset statistics are shown in Table II. To adhere to a realistic blackbox attack setting, we paid special attention to the data splitting. We first split the data into 60:40, overlapping and non-overlapping sets, respectively. We then split the non-overlapping data into 4 chunks. Then for each target classifier, we assign 1 chunk of the non-overlapping data + overlapping set. With this splitting, we can ensure that at least 10% of the training data is unique for each classifier. Then for each classifier's data we split it into *train:validation:test:explainability*.

The core component of FOX is the explainability tool. For this case study, we chose LIME (Local Interpretable Model-Agnostic Explanations) [19] due to its prevalent adoption for interpreting blackbox models [21], [23] and for being model-agnostic. We further adapted LIME [3] to interpret inputs in

[2]For simplicity, gender labels here are limited to Male or Female.

[3]https://github.com/marcotcr/lime

the case of multiple reaction types since we are using two different reactions (comments and alt-text of type textual data, tags, respectively). *(we will release the adaptation code)*

FOX also requires a scoring function $S(.)_{y'}$ to obtain the top_N to be included in $\mathcal{A}^{subset}$. We define $S(.)_{y'}$ as follows:

$$S(t) = h(t_{mc}, t)[y'], \tag{4}$$

This scoring function measures the impact of injecting t with t_{mc} on the probability to have y'. Let t_{mc} be the most contributing feature. The top_N features are the ones that introduce the largest change towards y' in the presence of t_{mc}.

B. Adversarial Reactions Generation

To generate perturbed instances, we have defined the perturbation strategies for each reaction type under the constraints of being well-formed and stealthy. We have introduced several strategies in order to confuse the attacker:

1) Alt-text generation strategies: Facebook has designed and deployed automatic alt-text, which is a system to identify faces, objects, and themes from photos by applying computer vision technology. This system intends to help blind people get a clue of the content they are engaging with and to feel more connected and involved in Facebook. The alt-text generates a summary of the existing content for the image automatically. The technology can recognize a list of 97 concepts (tags), including people (e.g., people count, child, baby), objects (e.g., basketball, building, tree, cloud, food), settings (e.g., inside restaurant, outdoor, nature) and themes (e.g., close-up, selfie, drawing). The generated alt-text has a defined format described in [24]. Therefore, generating an adversarial reaction from alt-text should adhere to the following constraints: 1) order constraints. For example, people tags should come before themes tags 2) coherency constraints. For instance, "person" and "people" tags cannot co-exist. 3) a word that is not a tag is added in the form of a text. For instance, the word "pregnant" is not a tag; thus, injecting it would result in the alt-text: text that says 'pregnant'.

2) Comments generation strategies: we have defined three strategies for adversarial comments generation described below

- **Single-Word-Comment:** After analyzing the length of comments in the dataset, we have found that 12% are single-word comments (e.g., Beautiful, Wow, yes, etc.). The strategy of introducing single-word comments is interesting as it gives the lowest edit-distance compared to Word-Combination and Comment-Extraction strategies. We create a set of comment features that can be applied to generate a single-word comment. At each iteration, we check if the current single-word comment doesn't make sense, we add punctuation (e.g., "!", "?"). For example, the comment "bro" alone may not make sense, but adding an interrogation mark (e.g., "bro?") makes it a plausible comment reaction.
- **Word-Combination:** In the word-combination strategy, we first define the length of the desired combinations l, then we construct sentences of l comment features retrieved from the ADV^{subset} features set. We check the validity of the sentence using two tools: LanguageTool [15] for grammatical errors and the T5 model trained on (The Corpus of Linguistic Acceptability (CoLA)) [17] for linguistic acceptance.
- **Comment-Extraction:** Given the comment containing the most contributing word obtained in the Explainability step, we need to design a comment extraction and generation mechanism with word replacement. A suitable generated comment needs to fulfill the following criteria: (1) have a meaning similar to the fragment of the original comment used for generation (2) grammaticality of the generated comment should be preserved. In order to perform extraction and the selection of replacement words that meet such criteria, we propose the following workflow:
  - **Clause Extraction:** In this strategy, we perform word replacement only on a portion of the original comment. Specifically, only the clause/sentence containing the most contributing word is selected. First, we split the comment into sentences with (NLTK sentence tokenizer). Then we select the sentence containing the most contributing word and split it into clauses utilizing (benepart sentence parser). Finally, the clause C that has the most contributing word is selected for word replacement.
  - **Replacement words sets construction:** We randomly select from the extracted clause/sentence C the portion σ of words W to be replaced. We construct a set for all possible replacements of the selected word $w_i \in W$. The replacement words set is initiated with the N closest words according to the cosine similarity between w_i and every other word in the adversarial comments features set. In the replacement words set of w_i, we only retain those having an identical part-of-speech (POS) with w_i to ensure that the generated comment's grammar is primarily maintained.
  - **Comment Generation:** For each $w \in W$, we construct a set of sentences C' by substituting w in sentence C with a word from its replacement words set. We also compute the sentence semantic similarity between the source C and the resulting set of C'. Concretely, we use Sentence Transformer Encoder to encode the two sentences into high dimensional vectors and use their cosine similarity score to approximate their semantic similarity. We set the similarity score threshold to 0.9. The sentence corresponding to the threshold will be considered to be the new source C for the subsequent word replacements. After iterating over all the words to replace $w_i \in W$, we concatenate the resulting C with the set of comments of X to create an adversarial instance X_{adv}. For an example, see Fig.5.

C. Target Classifiers

In this section, we describe the target classifiers used for FOX evaluation and comparison with the other SOTAs techniques. As we assume a blackbox setting, we have no information about the underlying architecture. Therefore we

Algorithm 2: Perturbed instances generation algorithm

Perturb $(x, \mathcal{A}^{subset}, t_{mc}, c\_len)$

Input: x picture to perturb, $\mathcal{A}^{subset}$ adversarial features subset, t_{mc} most contributing feature tuple, c_len length of generated comments by the comment combination strategy

Output: $\mathcal{P}$ set of perturbed instances

$\mathcal{P} \leftarrow \{\}$;
for $t \in \mathcal{A}^{subset}$ **do**
  if t *is of type 'comments'* **then**
    $\mathcal{P} \leftarrow \mathcal{P} \cup \{singleWordComment(x, adv_f^{value})\}$;
  end
  if t *is of type 'alttext'* **then**
    $\mathcal{P} \leftarrow \mathcal{P} \cup \{injectAltText(x, f^{value})\}$;
  end
end
if t_{mc} *is of type 'comments'* **then**
  $c \leftarrow$ comment containing t_{mc};
  $\mathcal{P} \leftarrow \mathcal{P} \cup \{extractComment(c, \mathcal{A}^{subset}_{comments})\}$;
end
if $c\_len > 1$ **then**
  $\mathcal{P} \leftarrow \mathcal{P} \cup commentCombination(x, \mathcal{A}^{subset}_{comments}, c\_len)$;
end
return $\mathcal{P}$;

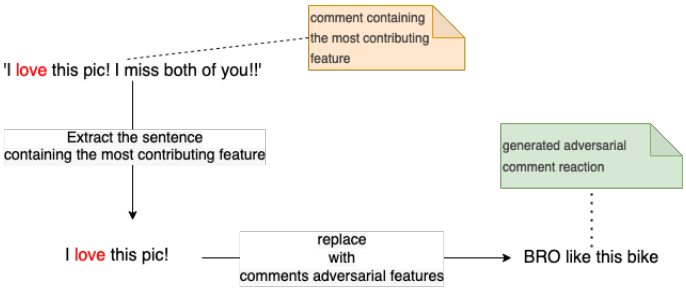


Figure 5: In this example, we show the workflow of generating a comment with the Comment-Extraction strategy.

have trained classifiers with various architectures (RNN, CNN, Attention, LR) and different encoding for the reactions. All the classifiers were trained on the first 100 comments of each picture and the alt-text. The best performance achieved by the trained classifiers is 0.74 in the F1 score. Table III summarizes the classifiers performance.

- **BERT**: We fine-tuned BERT [5] on the task of gender inference from the picture comments. We then used the mean of the fine-tuned model's predictions over the set of comments to compute their final representation. Then we encoded the alt-text to a feature vector of $\mathbb{R}^{768}$ using *distilbert-base-nli-mean-tokens* sentence embedding [18] followed by a fully connected layer to compute the final representation of the alt-text. Both representations (comments and alt-text) are concatenated then fed to a *fully-connected-network* (FCN) layer to infer the gender.
- **Roberta-CNN**: In this classifier, the picture comments and alt-text were encoded to a feature vector of $\mathbb{R}^{1024}$ using *Roberta-large-nli-stsb-mean* sentence embedding [18]. The set of comments vectors is then fed to three CNN layers with kernel sizes of (3, 1024), (3, 1024), (2, 1024); the output of each layer is max pooled, and the resulting pooled output of the three layers is concatenated then fed to a fully connected layer to compute the final representation of the comments. The alt-text embedding is fed to a fully connected layer to compute its final representation. Both final representations (comments and alt-text) are concatenated and fed to an FCN layer to infer the gender.
- **Distilbert-RNN**: In Distilbert-RNN, LSTM [9] layers are used to model the sequential dependency among comments. First, picture comments and alt-text were encoded to a feature vector of $\mathbb{R}^{768}$ each using *distilbert-base-nli-stsb-mean-token* sentence embedding [18]. To classify a picture, we feed the set of comments embeddings to the LSTM layers. Then we take the last hidden states computed for each LSTM layer and feed them to a fully connected layer to compute the final representation of the picture's comments. The alt-text embedding is fed to a fully connected layer to compute its final representation. Then both final representations are concatenated and fed to an FCN layer for classification.
- **Pijani et al. [16]**: The authors employed several features extraction techniques to obtain relevant features to male and female classes then used them to train several classifiers specified in [16]. The authors have provided us with the code for training and evaluating their classifier.
- **Stacking**: For Stacking, we combine the predictions of all the models by performing hard voting on the predicted classes. The resulting prediction is the class with the majority of votes.

Classifier	Accuracy	F1
BERT	0.74	0.74
Stacking	0.73	0.73
Roberta-CNN	0.73	0.73
Distilbert-RNN	0.71	0.71
Pijani et al.	0.66	0.66

Table III: Target Gender Attribute Classifiers Performance

D. Comparison of Attack Methods

In this section, we will describe representative and SOTA adversarial text generators that we compared with FOX. Note that these attack methods require an initial text to attack. Therefore, we create a copy of the most contributing comment as an initial text, then concatenate it to the set of original comments of the picture and constrain the attack on the added comment only.

- **TextFooler [10]**: This method identifies the most important words in the input and replaces them with semantically similar and grammatically correct words until it achieves the goal and fools the text classifier. This attack requires an initial text to attack.
- **TextBugger [12]**: This method aims at fooling text classifiers by carefully introducing misspellings in a text which they call bugs. The bugs are created with the following strategies insert, delete, swap, substitute-c, substitute-was described in [12].
- **DeepWordBug [13]**: This method searches for the highest-ranked tokens in a text using a scoring function defined in [13] and introduces simple character level perturbations to them to fool the classifier with a minimal edit-distance of the perturbation.

V. Experimental Results

A. Attack performance

We quantify the effectiveness of the attack with the Attack Success Rate (ASR). For instance, an attack on h with an ASR score of 90% indicates that an attacker can fool h to predict an incorrect gender 90% of the time on all users publications that h should have otherwise predicted correctly. All the experiments related to FOX have been performed with an iteration limit of $I = 20$ and a number of features per iteration of $N = 10$. For SOTA methods, we utilized the default parameters and their implementations in the TextAttack framework [14]. Table IV presents the evaluation of the Attack Success Rate (ASR) of all attack methods on the test data under a black-box setting. We can observe that FOX is able to achieve a near-perfect attack success rate attacking different classifiers, with the ASR ranging from 93% up to 99%. FOX/alt-text (FOX without alt-text strategies) directly compares to the other attacks since they attack only with comments. Despite employing only adversarial comments strategies with FOX, our method still outperforms other methods and is the only one to attack Pijani et al. [16] effectively. Whereas for the other attacks, the ASR is very low, ranging from 0.5% to 18% when attacking Pijani et al. [16].

B. Extracted features transferability

This section investigates the transferability of the extracted features for one model to attack another model. We observe that the extracted features are highly transferable from one model towards all the models in our experiments, with a minimum ASR of 80% and a perfect ASR of 100%. Results of transferability experiments are shown in Table V.

C. Ablation study

In this section, we experiment FOX with different settings (number of features of the $\mathcal{E}^{strong}$, enable/disable adversarial reactions generation strategies) and study their impact on the ASR.

a) $\mathcal{E}^{strong}$-related experiments: Fig.6a We observe that the fewer features considered, the lower the attack success rate. The best attack success rate was achieved when considering all of the features in $\mathcal{E}^{strong}$ shown in Table VI, with a slight decrease when considering 75% of the features. When limiting to only 25%, The ASR went even lower but still within an acceptable ASR within the range of 82%, 98% for BERT, Roberta-CNN respectively.

b) Adversarial Reactions Generation Strategies-Related Experiments: Fig.6b We show the impact of disabling one or a set of adversarial reaction generation strategies on the ASR with the following experiments.

Disable-comments (d-c) In this experiment, we consider only adversarial alt-text generation strategies. We noticed that the ASR decreased slightly for Roberta-CNN and Distilbert-RNN classifiers, whereas for BERT, it was nearly impossible to attack without adversarial comments generation strategies. Pijani et al. [16] had the second-highest decrease in ASR.

Disable-alt-text (d-a) In this experiment, we consider only adversarial comment generation strategies. The ASR decreased considerably for Roberta-CNN and BERT. The ASR had a slight decrease for Pijani et al. [16] classifier, whereas Distilbert-RNN was successfully attacked with a high ASR using only adversarial comments generation strategies.

VI. Discussion

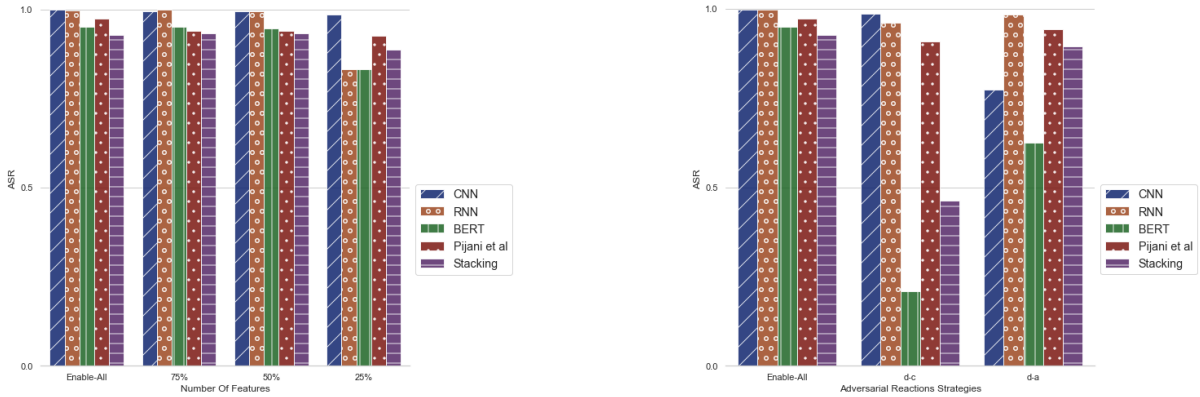
FOX comments generation is rule-based, and the definition of our generation strategies requires an expert in the used language. For instance, Chinese is written without spaces, unlike English (and other alphabetic writing systems), so defining the Comment-Extraction strategy requires understanding the language. Our usage of a rule-based approach is motivated by the fact that English grammar is well defined, which omits the need to train a generative language model, thus requiring a large amount of data. Moreover, we show that we achieved the best ASR with a small amount of data with this approach. The notorious complexity of context extraction social networks data is due to the noisy nature of data in social networks and the diverse contexts that arise when users' reactions (e.g., comments) shift from the original context of the publication [6]. Therefore, In our approach, we do not consider the context constraint when generating comments. Unlike GAN-based approaches, our method does not require a large dataset to build $\mathcal{E}^{strong}$ to create adversarial reactions for data mining models. Unlike previous works in adversarial text generation techniques. We employ an explainability tool to benefit from the model's interpretability research. While the compared adversarial text approaches have successfully attacked static inputs (e.g., reviews), they performed weakly in dynamic inputs such as comments. In comparison, the performance experiments show that our approach has a high success rate and can attack various classifiers. Moreover, the techniques that relied on words substitution with synonyms failed to attack Pijani et al. [16]. In this classifier, out-of-vocabulary words (OOV) are replaced by similar ones from their vocabulary. This way of dealing with OOV words prevents techniques

Attack\Classifier	BERT	Distilbert-RNN	Roberta-CNN	Pijani et al.	Stacking
TextFooler	0.440	0.407	0.380	0.189	0.369
TextBugger	0.384	0.402	0.344	0.109	0.330
DeepWordBug	0.299	0.403	0.262	0.055	0.252
FOX\alt-text	0.619	0.979	0.769	0.931	0.895
FOX	**0.938**	**0.995**	**0.997**	**0.957**	**0.932**

Table IV: Attack Performance on Different Attack Strategies and Target Classifier Architectures.

Table V: Strong features transferability across models for gender protection. The ith row represents the features extracted by the corresponding model, and the jth column represents the model attacked by these features. $cell_{i,j}$ represents the ASR of attacking the model j using features extracted through the model i.

Features\classifier	BERT	Distilbert-RNN	Roberta-CNN	Pijani et al.	Stacking
BERT	-	0.8	0.964	0.943	0.853
Distilbert-RNN	0.974	-	0.992	0.949	0.855
Roberta-CNN	0.971	1.0	-	0.952	0.848
Pijani et al.	0.917	0.8	0.952	-	0.823
Stacking	0.974	0.8	0.969	0.951	-



(a) $\mathcal{E}^{strong}$-related experiments

(b) Adversarial Reactions Generation Strategies Experiments

Figure 6: FOX ablation study aimed to show the impact of different settings on the attack success rate (ASR) against the target classifiers.

Models	Number of features
BERT	375
Pijani et al.	637
Roberta-CNN	821
Distilbert-RNN	1291
Stacking	1295

Table VI: The number of extracted features of each target classifier

that rely on substitutions with synonyms to destabilize the classifier's predictions. In contrast, FOX extracted features that successfully fooled this class of classifier after an average of 3 iterations only. In the transferability experiments, we have shown that the extracted features for one model can be used to attack other models with a high success rate.

VII. Conclusion and Future Work

We have proposed FOX, an adversarial attack framework for black-box models, that incorporates an explainability tool to generate effective adversarial reactions aimed at fooling different types of privacy threatening classifiers with a success rate of up to 100%. As a case study, we have applied FOX to protect the gender of Facebook users through adversarial reactions generated by our method. Furthermore, we have shown that FOX is more efficient than the other baselines and yields strong features with high transferability among classifiers even when the models have been trained with 10% of non-overlapping data. Our method allows for collaborative privacy protection in such a way the target user can be protected via the adversarial reactions posted by his/her trusted friends. Although based on black-box classifiers, our method can be extended to a white-box setting using only a white-box explainability tool.

As future work, we plan to apply FOX (i) on different data types (e.g., images) that may leak private information and (ii) to other types of online communities (e.g., question-and-answer websites like Quora, online discussions like Reddit) where users' privacy could be highly threatened.

References

[1] A. Acquisti and R. Gross. Imagined communities: Awareness, information sharing, and privacy on the facebook. In *Proceedings of PET'06*, PET'06, page 36–58. Springer, 2006.

[2] Y. Alufaisan, Y. Zhou, M. Kantarcioglu, and B. Thuraisingham. Hacking social network data mining. In *IEEE ISI*, 2017.

[3] G. Beigi and H. Liu. A survey on privacy in social media: Identification, mitigation, and applications. *ACM/IMS Trans. Data Sci.*, 1(1), Mar. 2020.

[4] Y. Cai, G. O. M. Yee, Y. X. Gu, and C.-H. Lung. Threats to online advertising and countermeasures: A technical survey. *Digital Threats: Research and Practice*, 2020.

[5] J. Devlin, M. Chang, K. Lee, and K. Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, 2018.

[6] N. Duarte, E. Llanso, and A. Loup. Mixed messages? the limits of automated social media content analysis. In *Proceedings of the 1st Conference on Fairness, Accountability and Transparency*, Proceedings of Machine Learning Research, pages 106–106. PMLR, 2018.

[7] N. Z. Gong and B. Liu. Attribute inference attacks in online social networks. *ACM Trans. Priv. Secur.*, 21(1), 2018.

[8] J. He, W. W. Chu, and Z. V. Liu. Inferring privacy information from social networks. In S. Mehrotra, D. D. Zeng, H. Chen, B. Thuraisingham, and F.-Y. Wang, editors, *Intelligence and Security Informatics*, pages 154–165. Springer, 2006.

[9] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural Comput.*, 9(8):1735–1780, 1997.

[10] D. Jin, Z. Jin, J. T. Zhou, and P. Szolovits. Is bert really robust? a strong baseline for natural language attack on text classification and entailment. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05):8018–8025, 2020.

[11] T. Le, S. Wang, and D. Lee. MALCOM: generating malicious comments to attack neural fake news detection models. In *20th IEEE International Conference on Data Mining, ICDM 2020, Sorrento, Italy, November 17-20, 2020*, pages 282–291. IEEE, 2020.

[12] J. Li, S. Ji, T. Du, B. Li, and T. Wang. Textbugger: Generating adversarial text against real-world applications. *Proceedings 2019 Network and Distributed System Security Symposium*, 2019.

[13] A. Mathai, S. Khare, S. Tamilselvam, and S. Mani. Adversarial black-box attacks on text classifiers using multi-objective genetic optimization guided by deep networks. *CoRR*, 2020.

[14] J. X. Morris, E. Lifland, J. Y. Yoo, J. Grigsby, D. Jin, and Y. Qi. Textattack: A framework for adversarial attacks, data augmentation, and adversarial training in NLP. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*. ACL, 2020.

[15] D. Naber. *A Rule-Based Style and Grammar Checker*. GRIN Verlag, 2003.

[16] B. A. Pijani, A. Imine, and M. Rusinowitch. Online attacks on picture owner privacy. In *Database and Expert Systems Applications*, pages 33–47. Springer, 2020.

[17] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21:140:1–140:67, 2020.

[18] N. Reimers and I. Gurevych. Making monolingual sentence embeddings multilingual using knowledge distillation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*. ACL, 2020.

[19] M. T. Ribeiro, S. Singh, and C. Guestrin. "why should I trust you?": Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD*, pages 1135–1144. ACM, 2016.

[20] J. Schultz. How much data is created on the internet each day? https://blog.microfocus.com/how-much-data-is-created-on-the-internet-each-day/, 2021. Accessed: 2021-08-10.

[21] J. Singh and A. Anand. Exs: Explainable search using local model agnostic interpretability. In *Proceedings of the Twelfth ACM International Conference*, WSDM '19, page 770–773. ACM, 2019.

[22] U. Weinsberg, S. Bhagat, S. Ioannidis, and N. Taft. Blurme: Inferring and obfuscating user gender based on ratings. In *Proceedings of the Sixth ACM Conference on Recommender Systems*, RecSys '12, page 195–202. ACM, 2012.

[23] H. Wu, W. Ruan, J. Wang, D. Zheng, B. Liu, Y. Geng, X. Chai, J. Chen, K. Li, S. Li, and S. Helal. Interpretable machine learning for covid-19: An empirical study on severity prediction task. *IEEE Transactions on Artificial Intelligence*, 2021.

[24] S. Wu, J. Wieland, O. Farivar, and J. Schiller. Automatic alt-text: Computer-generated image descriptions for blind users on a social network service. In *Proceedings of the 2017 ACM Conference on CSCW*, CSCW '17, page 1180–1192. ACM, 2017.

[25] W. E. Zhang, Q. Z. Sheng, and A. A. F. Alhazmi. Generating textual adversarial examples for deep learning models: A survey. *CoRR*, 2019.