Adversary for Social Good: Leveraging Adversarial Attacks to Protect Personal Attribute Privacy

XIAOTING LI, Visa Research, USA
LINGWEI CHEN, Wright State University, USA
DINGHAO WU, Pennsylvania State University, USA

Social media has drastically reshaped the world that allows billions of people to engage in such interactive environments to conveniently create and share content with the public. Among them, text data (e.g., tweets, blogs) maintains the basic yet important social activities and generates a rich source of user-oriented information. While those explicit sensitive user data like credentials have been significantly protected by all means, personal private attribute (e.g., age, gender, location) disclosure due to inference attacks is somehow challenging to avoid, especially when powerful natural language processing (NLP) techniques have been effectively deployed to automate attribute inferences from implicit text data. This puts users' attribute privacy at risk. To address this challenge, in this article, we leverage the inherent vulnerability of machine learning to adversarial attacks, and design a novel text-space **Adv**ersarial attack for **S**ocial **G**ood, called *Adv4SG*. In other words, we cast the problem of protecting personal attribute privacy as an adversarial attack formulation problem over the social media text data to defend against NLP-based attribute inference attacks. More specifically, Adv4SG proceeds with a sequence of word perturbations under given constraints such that the probed attribute cannot be identified correctly. Different from the prior works, we advance Adv4SG by considering social media property, and introducing cost-effective mechanisms to expedite attribute obfuscation over text data under the black-box setting. Extensive experiments on real-world social media datasets have demonstrated that our method can effectively degrade the inference accuracy with less computational cost over different attribute settings, which substantially helps mitigate the impacts of inference attacks and thus achieve high performance in user attribute privacy protection.

CCS Concepts: • **Security and privacy** → **Privacy protections**; • **Computing methodologies** → *Natural language processing;*

Additional Key Words and Phrases: Social media, attribute privacy, adversarial text attack, inference attack, text data

ACM Reference format:
Xiaoting Li, Lingwei Chen, and Dinghao Wu. 2023. Adversary for Social Good: Leveraging Adversarial Attacks to Protect Personal Attribute Privacy. *ACM Trans. Knowl. Discov. Data.* 18, 2, Article 46 (November 2023), 24 pages.
https://doi.org/10.1145/3614098

X. Li work done while at PSU.

Authors' addresses: X. Li, Visa Research, Palo Alto, CA 16802; email: xiaotili@visa.com; L. Chen, Wright State University, Dayton, OH 45435; email: lingwei.chen@wright.edu; D. Wu, Pennsylvania State University, University Park, PA 16802; e-mail: dwu@psu.edu.

1556-4681/2023/11-ART46 $15.00
https://doi.org/10.1145/3614098

1 INTRODUCTION

In the Internet-age, social media undoubtedly has become an indispensable part of our daily lives through countless websites and apps, which allows us to discover and learn new information, create content, and share ideas with friends, family, and others. Such an interactive and convenient environment generates a mass of user-oriented data. Due to its accessibility and information richness, this data enables researchers to study and understand social communities and individual behaviors. For example, during the COVID-19 pandemic, a surge of solutions have been presented to leverage social media data for risk assessment [69]. However, these apparent benefits also attract attackers to retrieve users' sensitive information and fulfill their malicious intents (e.g., unwanted advertising, user tracing) [5, 70] as illustrated in Figure 1. Take Facebook data privacy scandal [11] as an example, the Cambridge Analytica harvested the personal data of millions of people from Facebook without their permission and used it for political advertising purposes. In fact, such privacy risk is not rare on social media, and could be quickly transmitted and propagated [30, 36].

In the social media environment, text data maintains a huge amount of basic yet important user information. For example, users usually post their experiences, interests, comments, or thoughts in the tweets or blogs for sharing. Such vibrant social engagements render text data a major target for attackers to parse the contents and reveal personal attributes (e.g., age, gender, location, sexual orientation, and political views) that people are unwilling to disclose. On the other hand, **natural language processing** (**NLP**) provides more and more powerful techniques for text understanding and mining, which enable a surge of effective attribute inferences from implicit text data that put social media privacy at risk [17, 21, 24, 33, 34, 40, 43, 56, 76]. In this research work, we simply demonstrate an attribute privacy threat on social media as the scenario that an attacker trains a well-performed NLP model to infer users' private attributes from their text data such as tweets or blogs. With this in mind, some previous attempts have paid close attention to protect these attributes against inference attacks [22, 23, 26, 30, 45, 61, 63, 67], which, however, still suffer from either large computational cost, or specific application scenarios limited to visual or high-dimensional data. Thus, our research goal here is to generalize the investigation into more challenging text data, and protect personal attribute privacy in this regard from a novel and practical adversarial learning perspective.

The effectiveness of machine learning models relies on the assumption that training data and test data follow the same underlying distribution, while this hypothesis is likely to be violated by an adversary who may manipulate the input data to compromise the output performance [8]. In other words, they are vulnerable to adversarial attacks that can easily fool the models into misclassification by adding small perturbations to the input data [19, 66]. Recent studies [1, 16, 31] have shown that NLP models are also faced with inherent learning-security challenge of lacking adversarial robustness. This naturally inspires us to take advantage of such a vulnerability and cast personal attribute privacy protection problem on social media as an adversarial attack formulation problem against attribute inference attacks. To achieve this goal, we face two challenges: (1) as inference attackers have a variety of choices in model construction, it is impossible for us to access the inference models in the real-world settings; (2) due to its discrete property and significant impact on social information interaction, modifications on text data have to comply with some essential constraints to guarantee the validity for the adversarial texts.

To address the above challenges, in this article, we first identify the practical black-box setting and the main types of constraints on text-space adversarial attacks, and then design an **Adv**ersarial attack for **S**ocial **G**ood, called *Adv4SG*, to protect personal attribute privacy against NLP-based attribute inferences over social media text data. Given a source text (e.g., tweets, blogs), Adv4SG performs iterative word perturbations expedited by a reformed population-based optimization, in the sense that its target private attribute is misclassified by a self-trained NLP model. Through

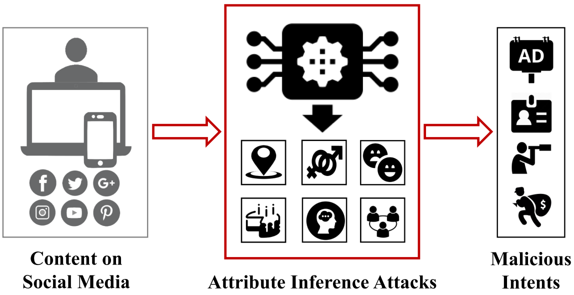


Fig. 1. Attribute inference attacks over social media.

these adversarial perturbations, not only are the predefined text-space attack constraints enforced, but also the attribute obfuscation is very likely effected on the real attackers' inference models due to transferability in adversarial machine learning [46]. These advantages allow a refined paradigm to effectively mitigate the impacts of NLP-based inference attacks on attribute disclosure and enhance personal privacy protection in practical social media environment. In summary, our major contributions are listed as follows:

- A novel and practical paradigm of protecting personal attribute privacy on social media that leverages adversarial learning to mislead attribute inference attacks.
- An adversarial attack is designed to obfuscate users' private attribute on more challenging text data of discrete property. Adv4SG is regulated by a reformed population-based optimization algorithm over perturbation subroutines that conform to text-space attack constraints, which can achieve better success rate in misclassifying attributes with less computational cost.
- The practical black-box setting is considered for Adv4SG's formulation, where the transferability of the proposed method is investigated to validate its applicability in real-world privacy protection scenarios.
- Extensive experimental evaluations on three real-world social media datasets (tweets and blogs) with different attributes to demonstrate the effectiveness of Adv4SG on attribute obfuscation and privacy protection.

The rest of the article is organized as follows. Section 2 defines the problem of attack model for attribute inferences and adversarial attack for attribute protection. Section 3 presents our detailed technical steps of text-space adversarial attack Adv4SG for attribute privacy protection on social media. Section 4 evaluates the effectiveness of Adv4SG and the impact of different settings. Section 5 discusses the applicability and limitations of our work. Section 6 briefly introduces the related work. Section 7 concludes.

2 PROBLEM DEFINITION

In this section, we first provide the problem definition of the attack model for attribute inferences, and then adversarial attack for attribute protection before technically detailing our proposed model Adv4SG in the following section.

2.1 Attack Model for Attribute Inferences

Social media enables users to post text data for social engagements. This data may bring privacy concerns to the forefront: the attackers that acquire such publicly exposed data may infer users' sensitive and private attributes (e.g., age, gender, and location) to deliberately fulfill the economic, social, or political intents, such as stealing user credentials, promoting unwanted advertisements, and stalking and threatening users [5, 22, 61, 70]. Considering that social media generally takes action to protect the explicit and identifiable information, in this article, we assume that the attackers would train NLP models using the latent representations learned from the implicit information of text data to infer the attributes of interest. More specifically, we represent social media text data as $\mathcal{D} = \{d_i, y_i\}_{i=1}^{n}$ consisting of n sample texts, where each text $d \in \mathcal{D}$ is annotated with a ground-truth label $y \in \mathcal{Y}$ for a specific attribute. Taking location attribute (main four U.S. regions) as an example: $\mathcal{Y}$ can be accordingly specified as $\mathcal{Y} = \{0\text{:Northeast}, 1\text{:Midwest}, 2\text{:South}, 3\text{:West}\}$. We follow the general NLP routine to deal with discrete text data by mapping each text d into a k-dimensional feature vector $\mathbf{x} = \phi(d)$ where ϕ is a feature representation function $\phi : \mathcal{D} \rightarrow \mathbf{X} \subseteq \mathbb{R}^{n \times k}$. In this respect, we can derive the predicted label of text $\mathbf{x}$ using the following formula:

$$y^* = \underset{y \in \mathcal{Y}}{\operatorname{argmax}}\, l_y(\mathbf{x}), \tag{1}$$

where $l_y(\mathbf{x})$ is the confidence score of predicting sample text $\mathbf{x}$ as attribute label y using an NLP model l (e.g., **convolutional neural network** (**CNN**), **long short-term memory** (**LSTM**), and Transformer). From Equation (1), we can see that the final attribute label assigned to the input sample is the one with the highest confidence score.

2.2 Adversarial Attack for Attribute Protection

In the text space, we aim at designing an adversarial attack to mislead attribute inference attacks and thus protect user attribute privacy. This is achieved in the way that the text-space adversarial attack perturbs the texts to obfuscate the target attribute and prevents inference attack models from correctly identifying their private attribute values. Specifically, given a text $\mathbf{x}$ and the associated attribute y to protect, the formulated adversarial attack modifies the original text $\mathbf{x}$ to the adversarial text $\widehat{\mathbf{x}}$ by adding a small perturbation δ, where $\widehat{\mathbf{x}}$ is predicted as any other label $\widehat{y} \in \mathcal{Y}$ ($\widehat{y} \neq y$). Therefore, we can formally define our objective function as follows:

$$f(\mathbf{x} + \delta) = l_y(\mathbf{x} + \delta) - \max_{i \neq y}\{l_i(\mathbf{x} + \delta)\}. \tag{2}$$

Equation (2) distinctly indicates that $\mathbf{x}$ is misclassified as a member of $\widehat{y}$ if and only if $f(\mathbf{x}+\delta) < 0$ [50]. The intuition to perform an adversarial attack in general feature space is to minimize $f(\mathbf{x}+\delta)$ by modifying $\mathbf{x}$ in the directions that follow the negative gradient of $f(\mathbf{x} + \delta)$ [6, 19, 42, 47]; that is, the adversarial attack can be implemented by solving the following optimization problem:

$$\begin{aligned} \delta^* &= \arg\min_{\delta \in \mathbb{R}^k} f(\mathbf{x} + \delta) \\ \text{s.t. } & \|\delta\|_p < \epsilon \ \text{ and } \ f(\mathbf{x} + \delta) < 0 \end{aligned}. \tag{3}$$

Due to its discrete property, these gradient-driven adversarial attack methods, however, cannot be directly applied to text space. The reasons behind this are that (1) gradients computed from the feature space are hard to define in text space, which may map the original text $\mathbf{x}$ to a set of non-admissible values; (2) L_p-norm distance metric typically works on continuous feature space, but is not capable of bounding the expected perturbation on texts represented as discrete tokens. Furthermore, a valid and realistic text-space adversarial attack for social good has to comply with some essential underlying constraints on the modification of the texts. These issues and challenges need to be addressed in the design of Adv4SG.

3 ADVERSARY FOR ATTRIBUTE PRIVACY PROTECTION

In this section, we first identify the black-box setting and underlying constraints conformable to text-space attacks; guided by these constraints, we detail our adversary for social good idea of how we formulate an adversarial attack Adv4SG to protect attribute privacy against NLP-based inferences over social media text data.

3.1 Black-box Attack

Considering the challenge that we are unable to access attacker's inference models, we put our work under the black-box setting, where the devised adversarial attack is not aware of any information about the inference model, including model choice, architecture, parameters, and training data. Compared to the assumptions made in [1, 31, 47, 52] that the attacks are able to retrieve the prediction scores by querying the target model with inputs, our black-box setting is more practical. In the real-world social media scenario, inference attackers have a variety of model choices, and it is impossible to specify one out of many. To this end, we self-learn a surrogate NLP model l to perform attribute inference and craft adversarial texts. Similar to the attackers, we can train such an inference model using the public data and attribute values from the users. Due to transferability in adversarial machine learning [46], the adversarial texts optimized to mislead the surrogate model are very likely to evade the real attackers' inference models.

3.2 Text-space Attack Constraints

Different from the adversarial attacks in the general feature space, the generation of text-space adversarial attacks for social good is much more constrained. For example, small modifications to texts can be visually noticeable to human viewers, and lead to severe semantic loss in human understanding. In this respect, it is not feasible to obfuscate the attributes of a text by simply copying the words from another text with different attribute values for impersonation, or heavily manipulating the source text for evasion. Text-space adversarial attacks should thus comply with some essential constraints to guarantee their validity and applicability. In this section, we define these constraints as follows to guide our attack formulation and clarify its strengths.

End-to-end learnability. In order to generate a practical text-space adversarial text, the first and basic requirement to be achieved is the end-to-end learnability, which enforces iterative perturbations to be performed from text space to text space. In other words, the text-space adversarial attacks need to follow the transformation flow $\mathcal{D} \rightarrow \mathcal{D}$, where $d \mapsto \widehat{d}$ takes an original text d and generates an adversarial version $\widehat{d}$. Since the feature representation function ϕ is generally not invertible, the challenge becomes to find a way to apply transformations δ on d to generate $\widehat{d}$, so that $\phi(\widehat{d})$ is as close to $\widehat{\mathbf{x}}$ as possible [29]. This suggests that the word perturbations on text d should not be arbitrary but guided by the misclassification of the target attribute.

Visual similarity. Modifications on texts are hard to be unnoticeable to human eyes. However, in order to increase the text validity and reduce the utility loss to facilitate its applicability in the social media environment, the generated adversarial texts should be perceptibly similar to the original ones as much as possible. This requirement can be satisfied by either perturbing the texts using visually similar words or restricting the number of words that are allowed to be modified.

Semantic preservability. Preserving semantics is also one of the underlying requirements when generating high-quality adversarial texts in the context of social media. This indicates that the expressed semantic meanings from the original text d and the adversarial text $\widehat{d}$ are required to remain consistent. In this regard, text- or word-level distance needs to be measured to guarantee the small difference caused by the perturbations preserves semantics for texts. On the text level, the edit distance (e.g., the number of perturbed words) between d and $\widehat{d}$ constrained for visual similarity can help reach the semantic equivalence. On the word level, the Euclidean distance between the original and perturbed word embeddings can ensure the semantic similarity for each word transformation.

Text plausibility. In addition to visual similarity, the text validity also requires the adversarial text to be syntactically correct and readable to humans, which is considered as text plausibility. For our problem, text plausibility is important as the adversarial text would not only fool attribute inference attack models, but might also be posted on social media for displaying. For this reason, artifacts, which easily reveal that an adversarial text is invalid (e.g., garbled text, words with symbols), will not be included. Note that, due to the fast-sharing and informal-writing property, user posts on social media may tolerate words with small misspellings or distortions to some extent, which are still valid and plausible to readers and social media.

Attack automaticity. To be applied in practical use, the perturbations performed during the adversarial attack procedure need to be completely automated without human intervention. This requires that the possible and available changes made to the text d exclude any transformations that are hand-crafted or need re-engineering on different datasets. In this way, the adversarial attack can be feasible to protect different attributes on different data scenarios without extra update efforts to the overall framework.

3.3 Overview of Adv4SG

The aforementioned real-world limitation and main types of constraints on text-space adversarial attacks raise significant challenges to the design of our attack method Adv4SG. To address these challenges, we propose Adv4SG to directly perturb the tokens in the text with guidance towards the misclassification of the target attribute through a self-trained NLP model, where the end-to-end learnability constraint and the black-box setting are naturally satisfied. Generally, tokens can be represented in the forms of words and characters, but in our attack formulation, we focus on perturbing the texts at word-level for two reasons: (1) the implicit information of the texts can be better encoded from the latent representations using word embedding than characters, which meets the assumption that the attackers would utilize the implicit information to train NLP-based models for attribute inferences; (2) the search space of possible changes over words is much smaller than characters, such that word-level perturbation is significantly more computationally tractable than character-level perturbation. Accordingly, we use edit distance metric in terms of the number of word changes to control the size of modifications so as to ensure the ability of fooling the threat model while remaining imperceptible. The overview of Adv4SG is illustrated in Figure 2.

To this end, the feature-space adversarial attacks defined in Equation (3) can be updated to a text-space optimization problem as follows:

$$\begin{aligned} \delta^* &= \underset{\delta \in \mathcal{W}}{\arg\min} f(\phi(d+\delta)) \\ \text{s.t. } \widehat{d} &= d+\delta, \;\; s(\widehat{d}, d) < \epsilon \;\text{ and }\; f(\phi(\widehat{d})) < 0, \end{aligned} \tag{4}$$

where $s(\widehat{d}, d)$ denotes the number of different words between $\widehat{d}$ and d, $\mathcal{W}$ is the set of plausible and semantic-preserving word candidates for perturbation, and + implies the high-level word change. From Equation (4), it is clear to see that Adv4SG proceeds with a sequence of word perturbations, where each perturbation takes the current text d, replaces a chosen word with the optimal candidate, and generates a new version $\widehat{d}$ such that d and $\widehat{d}$ are semantically equivalent and visually similar, until the target attribute is misclassified or the maximum allowed perturbation ϵ is reached. It is worth remarking that, since all the operations and optimizations do not require manual intervention, and candidate constructions and word perturbations are defined and performed on the fly, we can accordingly ensure the automaticity for our attack.

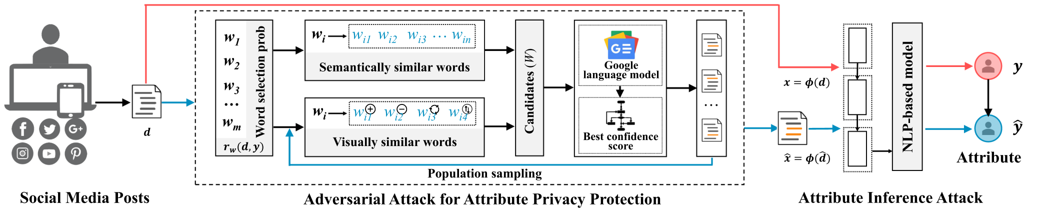


Fig. 2. The overview of our proposed text-space adversarial attack Adv4SG for misleading inference attacks and protecting personal attribute privacy.

3.4 Perturbation and Optimization

For a text-space adversarial attack, it is significant to elaborate word perturbations and devise an effective optimization algorithm to guide the transformations towards the specified target [52]. Some existing works [1, 16, 31] have thus delivered promising results in adversarial text generation. Even so, there are still some downsides in these attack methods: (1) word perturbations are limited to either semantically similar candidate replacements or character transformations while ignoring each other; and (2) it is computationally expensive to find an optimal solution using the greedy search or genetic algorithm with random population sampling. Differently, we advance Adv4SG by considering social media property and introducing both semantically and visually similar word candidates for perturbations and a reformed population-based optimization to force attribute inference models to misbehave faster. We present the technical details of our proposed model Adv4SG in the following separate subsections.

3.4.1 Scoring Word Importance. The original genetic attack proposed by Alzantot et al. [1] repeatedly performed perturbation on randomly selected words to formulate the population members at each generation, which may suffer from the vast search space of possible words and easily include those insignificant words. As such, we would like to first score the importance of words in the text to guide the population sampling that touches the important words and thus expedite the adversarial text generation.

Under our black-box setting, self-training NLP model allows us to compute the partial derivative of the confidence score regarding the predicted attribute label at each input word to obtain word saliency. Given the input text $d = (w_1, w_2, \ldots, w_m)$, the scoring function that determines the saliency of ith word in d can be defined as

$$s_{w_i}(d, y) = \frac{\partial l_y(\phi(d))}{\partial w_i}, \tag{5}$$

where $l_y(\cdot)$ is the confidence score of predicting attribute label y. Based on our observation, a word's high saliency does not necessarily imply high importance if the perturbation performed on it fails to enforce high variation. Therefore, we further compute the perturbation variance on i-th word in the text d as

$$v_{w_i}(d, y) = \max_{\widehat{w} \in \mathcal{W}(w_i)} [l_y(w_1, w_2, \ldots, w_n) - l_y(w_1, \ldots, w_{i-1}, \widehat{w}, w_{i+1}, \ldots, w_n)], \tag{6}$$

where $\mathcal{W}(w_i)$ is candidate set for w_i, which is constructed in Section 3.4.2. Resting on the word saliency and perturbation variance, we approximate the importance of ith word in the text d as

$$r_{w_i}(d, y) = s_{w_i}(d, y) \cdot v_{w_i}(d, y). \tag{7}$$

Clearly, the more important word has more impact on the model output, which is more likely to be modified to mislead inference model. Considering the facts that (1) some stop words (e.g., the, it, to, a, and an) or irrelevant words exist in a text that make little sense to tamper with and (2) the importance score of a word may be negative, we further use softmax function to normalize the word importance scores to serve as word selection probabilities for population sampling. In this way, the more important words in the sentences are given priority to be modified.

3.4.2 Preparing Word Candidates. We focus on perturbing the texts at word-level; that is, we need to construct a set of word candidates for each selected word to perturb or replace. In order to satisfy the constraints that the generated adversarial text retains semantic equivalence and syntactic coherence with the original one and visually imperceptible to human viewers on social media, we design two different types of word candidates for perturbation: semantically similar candidates and visually similar candidates.

Semantically similar candidates. We collect a set of words by searching the nearest neighbors of the ready-to-perturb word according to the Euclidean distance in word embedding space. To facilitate word search, a threshold η is introduced to filter out candidates with a distance greater than η such that the semantic preservability requirement could be less violated. Compared to GloVe [49], Counter-fitting embedding [44] is a more context-aware word embedding space with fine-tuned semantic relations. Therefore, we use it to search for the nearest neighbors for the given word.

Visually similar candidates. In addition to legitimate candidates from vocabulary, we also include the slightly perturbed words in the candidate pool. The reasons behind this choice are that (1) social media, as a fast-sharing and informal-writing environment, is highly misspelling-tolerant, where satiric or deliberate misspellings are not uncommon; (2) words with small character changes are imperceptibly to human eyes and have no significant impact on semantics [53], and (3) these words would very likely cause the selected word to be out of dictionary with "unknown" embedding such that the classification output may change [16, 31]. To ensure the text plausibility, we restrict that only small changes can be performed on the original word to create visually similar candidates, and those modified words will not be selected for a second perturbation. We design different word transformation methods as follows[1]:

- Add a space or a random character into the word except for the first and last positions.
- Remove a random character from the word except for the first and last ones.
- Swap any two adjacent characters except for the first and last ones.
- Substitute a random character in the word with a randomly selected character except for the first and last ones.
- Substitute a character or a substring to a visually (or aurally) similar number, such as $l \mapsto 1$, $o \mapsto 0$, $z \mapsto 2$, and straight $\mapsto$ str8. These are some deliberate formulations or slang on social media for user convenience or a rhetorical purpose.

3.4.3 Selecting Optimal Candidate for Replacement. After collecting candidates for the word, we proceed by choosing optimal candidate to replace it. However, those constructed word candidates are not all feasible for selection, where some of the semantically similar candidates may not be used in the same context as others. For example, "red" and "flushed" are related neighbors, but obviously "red" can be widely used to depict anything red, while "flushed" more likely describes a face turning red. To address this issue, we pass all the semantically similar candidates through Google language model [7] to further filter out the ones that do not fit within the context and improve the semantic correctness. The rest are then integrated with visually similar ones to form the final candidates. Afterward, we choose the optimal candidate among them that will maximize

[1] Both the first and last positions in the original word will not be modified for better perturbation invisibility.

ALGORITHM 1: Perturbation subroutine.

```
Function PerturbationSubroutine($d, y, l, p, n$):
    $w$ = WordSelect($d$, 1, $p$);
    $candsS$ = SemanticConstructor($w$, $n$);
    $candsV$ = VisualConstructor($w$);
    for $c_i \in candsS + candsV$ do
        $d(i) \leftarrow$ replace $w$ with $c_i$ in $d$;
        $score(i) = l_y(\phi(d(i)))$;
        if $c_i \in candsS$ then
            $pf, sf \leftarrow$ a word before/after $c_i$ in $d$;
            $gscore(i)$ = GoogleLM($pf, c_i, sf$);
        end
    end
    $tscore \leftarrow$ top $n/2$ in $gscore$;
    Remove $score(i)\ \forall\ c_i \in candsS$ and $c_i \notin tscore$;
    $c = \arg\max_{c_i} score(i)$;
    return $d(c)$;
end
```

the confidence score of the target attribute $\widehat{y}$ ($\widehat{y} \neq y$) prediction when it replaces the ready-to-perturb word in d. Then we perturb the text with the optimal candidate and generate a new text as a population member.

3.4.4 *Optimizing Word Perturbations.*

The three steps detailed above can contribute to a *perturbation subroutine* that accepts an input text (either perturbed or original), selects a word, perturbs it with optimal candidate, and generates a perturbed-version text towards the misclassification of the target attribute. The perturbation subroutine is illustrated in Algorithm 1. In this way, we are ready to generate a set of these perturbations for the given text. We aim at minimizing the number of word perturbations, which makes the adversarial text more similar to the original one and less likely to be perceived. Therefore, instead of using greedy search [16, 31], we follow the work by Alzantot et al. [1] and leverage a reformed population-based optimization algorithm to regulate the word perturbations during the formulation of Adv4SG.

The population-based optimization performs by sampling the population at each iteration, searching for those population members that achieve better performances, and taking them as "parents" to produce next generation [1]. This procedure can be summarized into three main operators. (1) Mutate(d): select a word from the given input text d using the normalized word importance score as the probability, and perform a perturbation subroutine on d. (2) Sample($\mathcal{P}$): sample a text d_i from the population $\mathcal{P} = \{d_1, d_2, \dots, d_N\}$ using the confidence score $l_{\widehat{y}}(d_i)$ as the probability. (3) Crossover(d_1, d_2): construct a child text $c = (w_1, w_2, \dots, w_m)$ where w_i is randomly chosen from $\{w_i^{d_1}, w_i^{d_2}\}$. Based on these operators, population-based optimization first generates an initial population $\mathcal{P}^0 = \{\text{Mutate}(d)_1, \text{Mutate}(d)_2, \text{Mutate}(d)_N\}$. At each following iteration t, the next generation of population will be generated in the following operation batch:

$$\begin{aligned} \widehat{d}^t &= \underset{d \in \mathcal{P}^{t-1}}{\text{argmax}}\ l_{\widehat{y}}(d), \\ c_i^t &= \text{Crossover}(\text{Sample}(\mathcal{P}^{t-1}), \text{Sample}(\mathcal{P}^{t-1})), \\ \mathcal{P}^t &= \{\widehat{d}^t, \text{Mutate}(c_1^t), \dots, \text{Mutate}(c_{N-1}^t)\}. \end{aligned} \tag{8}$$

ALGORITHM 2: Adv4SG for attribute privacy protection.

```
Input: $d$: a text sample, $y$: label for a specific attribute, $l(\cdot)$: inference model, $\epsilon$: maximum perturbations,
        $n$: neighbor number.
Output: $\widehat{d}$: an adversarial text.
Compute $r_w(d, y)$ using Equation (7);
$selectprob$ = Normalize($r_w(d, y)$);
$\widehat{y} \leftarrow$ label other than $y$;
$\mathcal{P}^0$ = {PerturbationSubroutine($d$, $\widehat{y}$, $l$, $selectprob$, $n$)$\}_{i=1}^{N}$;
for $t = 1 \rightarrow I$ do
    for $i = 1 \rightarrow N$ do
        $score(i) = l_{\widehat{y}}(\phi(\mathcal{P}_i^{t-1}))$;
    end
    $p = \text{argmax}_i\, score(i),\ \widehat{d} = \mathcal{P}_p^{t-1}$;
    if $s(d, \widehat{d}) \geq \epsilon$ then
        return None;
    end
    if $\text{argmax}_i\, l_i(\phi(\widehat{d})) == \widehat{y}$ then
        return $\widehat{d}$;
    else
        $\mathcal{P}^t = \{\widehat{d}\},\ sampleprob$ = Normalize($score$);
        for $i = 2 \rightarrow N$ do
            $c$ = PopulationSampling($\mathcal{P}^{t-1}$, 2, $sampleprob$);
            $\mathcal{P}^t = \mathcal{P}^t \cup$ PerturbationSubroutine($c$, $\widehat{y}$, $l$, $selectprob$, $n$);
        end
    end
end
return None;
```

The optimization will terminate when an adversarial text is found and returned, or the maximum allowed iteration number is reached. Algorithm 2 illustrates our proposed text-space adversarial attack Adv4SG. Different from the prior work, we improve the success rate of population samplings by choosing those ready-to-perturb words of high importance scores, while visually similar candidates introduced further expedite the adversarial text generation. Through Adv4SG, we can turn adversarial attacks into protection for personal attribute privacy on social media against the attribute inference attacks.

4 EXPERIMENTAL RESULTS AND ANALYSIS

In this section, we fully evaluate the effectiveness of our proposed adversarial attack Adv4SG for personal attribute privacy protection over social media text data.

4.1 Experimental Setup

Datasets. We test our method on three real-world social media datasets: GeoText [14], user gender tweets,[2] and blog authorship corpus [60], which are good representatives for social media data as tweets and blogs are posted by different users, and easily accessed by attackers

[2] https://www.kaggle.com/crowdflower/twitter-user-gender-classification

Table 1. Comparing Statistics of the Three Datasets

Dataset	Attribute	#Posts	#Classes	#Vocabulary
Twitter_g	Gender	13,926	2	17 k
Twitter_l	Location	9,281	4	16 k
Blog	Gender, Age	25,176	2	22 k

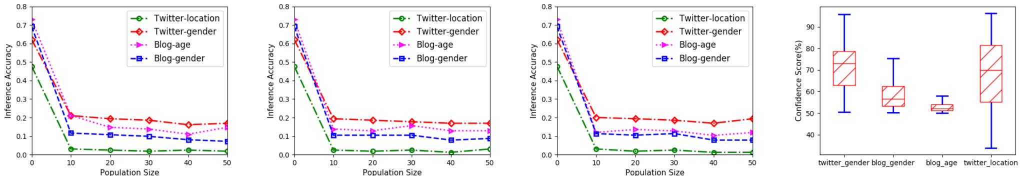
to uncover their private attributes. Specifically, GeoText is a tweet set from 9, 500 users with geographical coordinates in the United States. We map each user into one of the main four U.S. regions defined by the Census Bureau[3] and collect 9, 281 valid tweets with four locations (west, midwest, northeast, and south). User gender tweets are collected from Kaggle. We filter out those with gender confidence score less than 0.5 and obtain 13, 926 tweets with two genders (female and male). For blog data, it consists of 19, 320 documents, each of which contains the posts provided by a single user. We extract 25, 176 blogs with two attributes: (1) gender (female and male), and (2) age (teenagers (age between 13–18) and adults (age between 23–45)). Note that, age-groups 19–22 are missing in the original data. The data statistics are summarized in Table 1.

Text-space adversarial attack baselines. We compare Adv4SG with five other state-of-the-art text-space adversarial attack methods that are not only performed in an end-to-end manner at word level but also representative to cover different formulations on word candidates and perturbation optimization. These attacks can be specified as follows:

- Genetic attack [1]: this attack uses population-based optimization algorithm to generate adversarial examples with semantically similar candidates, where population sampling is performed in a random way at each generation.
- PSO attack [72]: this attack uses sememe-based annotation method to craft word's substitution candidates and incorporates an adapted **particle swarm optimization** (**PSO**) strategy to search for adversarial examples.
- Greedy attack: this method greedily performs perturbation subroutine of our method on one word at each iteration. We aim to evaluate the performance of perturbation crafted by our subroutine and validate the effect of population-based optimization.
- WordBug [16]: this attack scores word importance by removing it from text, and perturbs words in the descending order regarding word importance scores using character transformations.
- TextBugger [31]: this method also scores the word importance for greedy token selection, but proceeds by substituting the selected words with the optimal bug from candidates, including similar words in embedding space and word transformations.

Implementation details. We use euclidean distance as distance metric to construct semantic-similar candidates from embedding space, and the distance threshold is set to $\eta = 0.5$ to filter out those less similar ones. The size of candidate pool for each word is set as 8, where we choose the best one for replacement. We also limit the maximum allowed word perturbations to 25% of the text length, and we further evaluate its impact on attack performance in Section 4.2. We randomly select 80% of the samples for training, while the remaining 20% is used for testing, and we report the mean inference accuracy and attack success rate of 3 runs on test samples for the evaluation results. For the system configuration, all the experiments are conducted on 2 × Intel(R) Xeon(R) Silver 4114 CPU with 512 G RAM and 1 × TITAN XP 12 GB.

[3] https://www2.census.gov/geo/pdfs/maps-data/maps/reference/us_regdiv.pdf



(a) Iteration $I = 10$ **(b) Iteration** $I = 20$ **(c) Iteration** $I = 30$ **(d) Score distribution**

Fig. 3. Evaluation results: (a), (b), and (c) specify the inference accuracy of Adv4SG with different population sizes and iterations; (d) gives the confidence score distribution of the perturbed texts under four different inference settings.

Attack model for attribute inference attacks. An attribute inference attack aims at disclosing private attributes of users by learning a model on the public data. Since we do not know the attacker's model, we self-train **bidirectional LSTM** (**BiLSTM**) [20], **multi-layer GRU** (**M-GRU**) [10], ConvNets [75], and **CNN-LSTM** (**C-LSTM**) [18] to perform the tasks. We mainly use BiLSTM to evaluate the effectiveness of Adv4SG, since it is one of the most popular and feasible neural networks to address NLP problems and can be easily built by the attackers to perform attribute inferences with relatively smaller cost, computing resource and training effort than transformer or BERT, which is more realistic in real-world inference attack scenarios. The comparisons among BiLSTM, M-GRU, ConvNets, and C-LSTM are leveraged for cross-model transferability evaluation in Section 4.5. All models read in 250 words, where the dimension of each LSTM or GRU hidden unit is 128. We use GloVe [49] to map each word into a 300-dimensional embedding space. Note that, an inference attacker would deploy more robust models to evade adversarial attacks. As adversarial training is considered as one of the most empirically robust methods against adversarial attacks [3, 23], we build up a robust model using adversarial training and further discuss the effectiveness of Adv4SG under this setting in Section 4.6.

4.2 Evaluation of Adv4SG

In this section, we validate the effectiveness of Adv4SG against attribute inference attacks and the impacts of different parameters. To evaluate our method, we perturb the correctly classified text examples from the test data of four attribute settings.

Effectiveness. In our experiments, we evaluate Adv4SG under different population sizes and iterations as they play a crucial role to determine the degree of sample perturbation and computational cost. In particular, we test the results of our generated adversarial texts with population size $N \in \{10, 20, 30, 40, 50\}$ respectively against different inference attacks, while the maximum iteration I is ranging in $\{10, 20, 30\}$ correspondingly. The experimental results are shown in Figure 3. As we can see from the results, the inference accuracy for Twitter-location, Twitter-gender, blog-age, and blog-gender on clean data is 47.76%, 62.25%, 72.92%, and 69.20%, which are relatively close to the state-of-the-art results on each dataset. Adv4SG drastically decreases all these accuracies and achieves the goal of obfuscating attributes and protecting social media text data privacy. Averagely, our method reduces the accuracy of Twitter-location and Twitter-gender inference attacks from 47.76% to 2.19% and from 62.25% to 18.42%, respectively; for the larger and longer blog data, we degrade inference accuracy of gender and age from 69.20% to 9.66% and from 72.92% to 13.65%, respectively. We present some of our generated adversarial texts in Figure 4. It is clear that Adv4SG can subtly perturb important words towards the misclassification target in a plausible and semantic-preserving manner.

Task: Twitter-location. **Original label:** South (confidence=76.88%). **New label:** Northeast (confidence=61.66%)

They use the white ~~queso~~ cheese dip from farm fresh. I have seen cases of it in the kitchen.

Task: Twitter-gender. **Original label:** Male (confidence=53.46%). **New label:** Female (confidence=84.36%)

That ~~awesome~~ amazing moment when you ~~check~~ chechk your bank account and your parents send you more than you thought.

Task: Blog-age. **Original label:** Adults (confidence=76.08%). **New label:** Teens (confidence=60.31%)

Hellooooooo! Well, in case you haven't guessed by the ~~lack~~ l@ck of my blogs, I have been on holiday ~~nowhere~~ nowhare nice just sitting at home. But I thought I ~~should~~ shou1d take a break from computers as well. I have lots of catching up to do, good news, bad news and lots of ~~events~~ things to tell you all about. So stay tuned for the updates!!

Task: Blog-gender. **Original label:** Female (confidence=78.29%). **New label:** Male (confidence=54.43%)

So it starts a ~~blog~~ bl0g on the internet ready for writing. I'm gonna ~~use~~ utilize this a lot over the ~~next~~ future two weeks to let you know what my theatre class is doing, the cute guys I'm meeting and all the rest enjoy.

Fig. 4. Adversarial texts generated by Adv4SG under different inference tasks and their original texts.

Impact of population size and iteration. Generally, when we enlarge the population size, the success rate of generating adversarial samples increases and the accuracy of the inference models thus decreases, while the required perturbation number tends to go up as well. However, due to the perturbation limit for each text, the actual attack performance might not always improve for larger population size. We can observe that the inference accuracy for all settings drops to the worst at $N = 40$ and then either slightly increases or stays flat when N changes from 40 to 50. On the other hand, the larger iteration provides more improvement space for Adv4SG when the population size is small. For example, when $N = 10$, Adv4SG degrades the inference accuracy for blog-age setting from 21.23% ($I = 10$) to 12.02% ($I = 30$). Nevertheless, such inference accuracy difference among different iteration settings tends to be more statistically insignificant as the population size increases. As shown in Figure 3, Adv4SG achieves the comparable performance under all four inference settings at $N = 40$ with I varying in $\{10, 20, 30\}$. The reason behind this is that the larger population size is more likely to enforces the optimal solutions at earlier iteration, while most of the failed population samples would stay in the loop at later iteration. Considering that the larger iteration may introduce more computational cost, while the larger population size can significantly enhance Adv4SG, we use $N = 40$ and $I = 10$ throughout the following evaluations to keep a good tradeoff between the effectiveness and efficiency.

Impact of maximum allowed perturbation (ϵ). Different choices of ϵ could affect the performance of Adv4SG, since ϵ not only limits the number of word perturbations allowed to impact on the attack ability, but also significantly reflects the similarity between the generated adversarial texts and the original texts, and thus has direct impact on the semantic preservability and plausibility of the adversarial texts. We use the **cumulative distribution function** (**CDF**) of attack success rate regarding the number of ϵ to illustrate the evaluation results. From the results shown

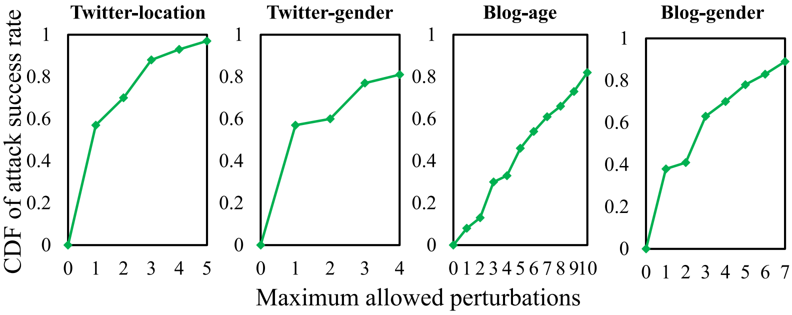


Fig. 5. Evaluation on maximum allowed perturbation (ϵ) via cumulative distribution of attack success rate.

in Figure 5, we can observe that as ϵ increases, the attack success rate increases as well because of the larger modification space, but the mean sentence semantics quality would decease. Actually, using Adv4SG, most of the generated adversarial texts manage to evade the inference models after perturbing very few words in the texts. More specifically, for Twitter-location inference, about 57% of the testing texts evade the inference model by perturbing only one word, while this success rate increases to 88% when $\epsilon \leq 3$. For Twitter-gender inference, Adv4SG successfully crafts 57% and 76% of the adversarial texts from the original with at most one word and three word perturbations respectively. For blog-gender inference, the attack success rates are 38% with $\epsilon \leq 1$ and 63% with $\epsilon \leq 3$. For blog-age inference, these two rates are 9% and 30%, which apparently underperforms other settings because of the longer text length. When Adv4SG is allowed to perturb at most 5 words, the attack success rate immediately rises to over 50%. All these results imply that (1) Adv4SG enables most of adversarial texts to be similar to the original texts; (2) the number of perturbations relatively relies on the length of the texts: the average lengths of the texts used for Twitter-location, Twitter-gender, blog-gender, and blog-age are 31, 15, 51, and 61, while the average perturbations are 1.8, 1.4, 2.9, and 5.2 for the corresponding inference tasks.

Other observations. In addition, we can also find some more interesting observations from the evaluation results in Figure 3 and Figure 5: (1) Adv4SG tends to perform worse on binary attributes (e.g., age and gender) than multi-class attributes (e.g., location). It is not difficult to understand that adversarial attacks on binary attributes can be considered as targeted attacks that might take more effort to perturb the texts and enforce misclassification to a specified target class (inverse to the original), while adversarial attacks on multi-class attributes fall into non-targeted attacks that have to simply cause the source texts to be misclassified, which is obviously easier. (2) The learning ability of the inference model may also have a potential impact on the Adv4SG's attack effectiveness against it, as small perturbations on the texts more likely lead to evasion for inference models that underperform than others. For example, the inference accuracy for Twitter-location is 47.76%, while Adv4SG successfully reduces it to 2.19% with a 7.65% mean perturbation rate. The similar results can be found between blog-age and blog-gender. (3) The age attribute seems more difficult to be obfuscated than others due to relatively higher model inference ability and longer text length, where Adv4SG performs more word perturbations for adversarial text generation.

Furthermore, we show the confidence distributions of those generated adversarial texts that can successfully fool the inference attackers under different deployment settings in Figure 3(d). It indicates the consistent findings with what we observe from other results. For instance, the average confidence values of the perturbed texts for the age attribute are distributed slightly above the borderline (i.e., 50%), which reveals the difficulty in obfuscating age attribute for blog dataset. Differently, the overall scores of other three tasks have been explicitly moved to the

Table 2. Comparisons of Different Text-space Adversarial Methods

Inference task	Metric	Adv4SG	Genetic	PSO	Greedy	WordBug	TextBugger
Twitter-location	Success Rate	**97.40%**	85.71%	94.70%	76.62%	55.84%	82.91%
	Median Ptb Rate	**5.26%**	6.25%	5.79%	8.33%	10.53%	7.85%
	Mean Ptb Rate	7.65%	9.00%	**7.33%**	10.73%	18.75%	11.58%
Twitter-gender	Success Rate	**74.03%**	55.84%	67.80%	45.45%	32.47%	62.34%
	Median Ptb Rate	**9.09%**	14.29%	10.24%	14.64%	27.27%	16.67%
	Mean Ptb Rate	**12.18%**	16.28%	12.72%	16.73%	29.56%	21.37%
Blog-age	Success Rate	**82.28%**	72.15%	74.05%	72.15%	17.72%	59.49%
	Median Ptb Rate	11.92%	**11.11%**	12.90%	12.19%	31.21%	19.64%
	Mean Ptb Rate	**13.53%**	13.96%	14.44%	14.06%	27.94%	23.89%
Blog-gender	Success Rate	88.61%	84.81%	**88.87%**	70.89%	54.43%	77.22%
	Median Ptb Rate	5.08%	**4.21%**	4.85%	7.45%	17.86%	12.31%
	Mean Ptb Rate	8.38%	8.61%	**8.14%**	10.33%	19.07%	16.03%

misclassification direction, which lead to better attack effectiveness. In addition, the performance of Adv4SG for long texts (i.e., blogs) seems to be more stable than short twitter texts. We guess it correspondingly relates to the different inference capability of the attackers on these datasets.

4.3 Comparisons with Other Attack Baselines

Attack performance. We compare Adv4SG with the other baselines including Genetic attack [1], PSO attack [72], Greedy attack, WordBug [16], and TextBugger [31]. These attacks are composed of word candidate preparation and perturbation optimization, but we follow the formulations presented in the related works and compare with them as a whole. Note that, in the original design of PSO attack, all words with the same sememe annotations serve as the perturbation candidates for the given word, the quantity of which could be very large on average. Considering that the candidate pool size of each word is set as 8 for all the other attacks, we narrow down the size of word candidates as the substitutes to 8 as well during PSO search for fair comparisons. From another perspective, we can also enlarge the candidate pool size to further elevate Adv4SG's attack performance, since each given word could have derived many more accessible neighbors under the distance threshold $\eta = 0.5$, let alone relaxing this restriction. As such, this experimental setting does make sense. To perform the evaluations, we randomly sample 50% of correctly classified examples from the testing tweets and blogs to measure the performance of attacks.

The comparative results are reported in Table 2, where Genetic attack outperforms Greedy attack, WordBug, and TextBugger, while TextBugger performs slightly better on tweet attribute obfuscation; PSO attack achieves higher attack success rate and perturbs less words than Genetic attack; Adv4SG outperforms PSO attack in most settings with marginally lower attack success rate and higher perturbation rate on blog-gender inference. From the results, we can observe that (1) PSO with tradeoff between local exploration and global exploitation is a more effective optimization method than genetic algorithm, while reducing the sememe-based word candidate size would greatly degrade the attack performance and yield less advantages out of particle swarm search; (2) projecting an important word into "unknown" may enforce inference models to misbehave faster, while ignoring semantically similar candidates would also miss good evasion chances, and (3) leveraging word importance to facilitate population-based optimization advances and expedites adversarial example generation. When we look into the generated adversarial texts, we find that Greedy attack fails in some of those adversarial texts with more modifications required over long blogs, and hence obtains a smaller perturbation number on average in results. By contrast, Adv4SG either converts those failed texts to adversarial examples, decreases the number of required perturbations, or raises the confidence scores of the perturbed texts, which

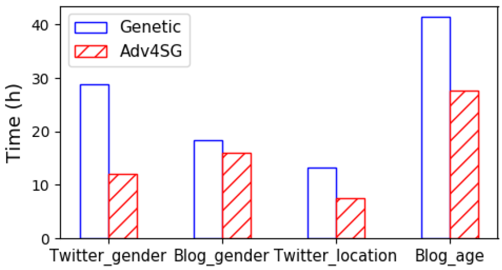


Fig. 6. Computational cost between Adv4SG and Genetic.

refines the text-space adversarial attack with respect to effectiveness and efficiency. Thus, Adv4SG can be a feasible paradigm in a real social media environment on attribute obfuscation and privacy effectiveness.

Computational cost. We evaluate the computational cost in terms of running time. On one hand, the greedy-based attack methods (i.e., Greedy, WordBug, and TextBugger) follow the heuristic of making the locally optimal choice and perturb one word at each stage; hence their search space is much smaller than genetic algorithm and their running time is undoubtedly much smaller as well, but the greedy methods fail to generate successful adversarial examples on a larger number of texts which drastically underperform genetic attacks. Also, due to the guidance of individual best position, global best position and the corresponding update probabilities, PSO attack manages to explore more positions and find the adversarial texts more quickly than genetic optimization, but requires a significantly large amount of preprocessing time to perform sememe annotations, sense identification, and word candidate generations before perturbation optimization, which is very different from genetic attack formulation. On the other hand, our proposed adversarial attack Adv4SG extends the original genetic design to advance the evolution process towards better solutions and expedite the word perturbations. Thus, here we would like to merely evaluate the runtime advancement of our method against genetic method.

To be comparable, we use single TITAN Xp for each experiment. We measure the average runtime for different inference settings on Genetic and Adv4SG, respectively. The results are presented in Figure 6. We can see from the results that Adv4SG can drastically reduce the running time compared to Genetic. For inference tasks such as Twitter-gender and Blog-age, Genetic method costs nearly twice the time of our method. On average, Adv4SG can save 36.85% computational time against the Genetic, which further justifies the advantage of word importance and visually similar candidates introduced in our proposed attack model Adv4SG.

4.4 Decomposition Analysis

In this section, we conduct a decomposition analysis to investigate how different components impact on the performance of our method Adv4SG with respect to attack success rate. In our model design, Adv4SG proceeds with word candidate preparation including semantically similar and visually similar candidates, and perturbation optimization that follows the genetic algorithm [1] to advance the population-based optimization to be more effective and efficient for adversarial text generation. Therefore, to verify their contributions, we separate Adv4SG into these components and analyze their contributions to the attack performance by formulating four alternative models: (1) *Semantic+Genetic*: only semantically similar candidates are crafted based on the vector distance in counter-fitting embedding space for word substitution, and the genetic algorithm is deployed for population-based optimization to find adversarial examples; (2) *Visual+Genetic*: this model prepares visually similar candidates for each word by using

Table 3. Evaluation on Different Attack Combinations with Respect to Attack Success Rate

Method	Twitter-location	Twitter-gender	Blog-gender	Blog-age
Semantic+Genetic	85.71%	55.84%	72.15%	84.81%
Visual+Genetic	58.75%	44.16%	50.10%	56.96%
Semantic+Visual+Genetic	92.40%	71.73%	78.48%	86.87%
Adv4SG (Ours)	**97.40%**	**74.03%**	**82.28%**	**88.61%**

the designed transformation methods, and incorporates genetic optimization to regulate the adversarial text generation. (3) *Semantic+Visual+Genetic*: we collect both semantically similar candidates and visually similar candidates for word substitutions, where word perturbations are again optimized using genetic algorithm. (4) *Adv4SG*: the complete design of our attack model.

The experimental results for decomposition analysis are reported in Table 3. From the results, we can observe that when substituting words using individual candidate set, semantically similar candidates achieve better results than visually similar candidates as the latter substitutions can merely provide "unknown" embedding space to impact on the text semantics, while semantically similar candidates may enable better evasion chances against the inference model with more diverse and dynamic perturbation possibilities to explore. After putting these two candidate sets together, visually similar candidates surprisingly play a crucial role in adversarial text generation, where the attack success rate increases by 6.69%, 15.89%, 6.33%, and 2.06% for four inference tasks, respectively. This further confirms that enforcing the chosen words into "unknown" may mislead the inference model in a faster way, which provides a "shortcut" to gather around the optimal positions based on the early effort made by word perturbations using semantically similar candidates. The attack performance discrepancy between Semantic+Visual+Genetic and Adv4SG demonstrates that population sampling guided by word importance is able to further advance the state-of-the-art performance to a higher level, which implies that this operation yields an additional advantage for population-based optimization that random population sampling may have missed. These observations from decomposition analysis highlight the effectiveness of Adv4SG.

4.5 Transferability

Under the black-box attack setting, as Adv4SG is implemented through self-trained NLP model, it is necessary to evaluate its transferability to validate if those adversarial texts generated for one model are likely to be misclassified by others. In this evaluation, we deploy Adv4SG to generate adversarial texts on four inference settings for four different NLP models: BiLSTM [20], M-GRU [10], ConvNets [75], and C-LSTM [18]. Then, we evaluate the attack success rate of the generated adversarial texts against other models. To ensure our results are comparable, we build up these models with the same parameter settings (different dropout rates) and training data. Accordingly, we build a cross-model transferability table, where each table unit (i, j) holds the percentage of adversarial texts crafted to mislead model i (row index) that are misclassified by model j (column index).

From Table 4, we can see that the cross-model transferability for Adv4SG is a strong but heterogeneous phenomenon: (1) between same model pairs, the percentage numbers are higher than 80%, most of which are close or beyond 90%; (2) between pairs of different models, some enjoy good transferability (e.g., 76.67% for M-GRU and BiLSTM on blog-gender setting), while some only have moderate one (e.g., 31.03% for ConvNets and M-GRU on blog-age setting). The results also imply that the complexity of the surrogate model and the intrinsic adversarial vulnerability of the target model contributes to attack transferability (e.g., all models against ConvNets achieve relatively higher transferability than others). Adversarial texts generated from a more complicated surrogate model tends to have better attack success rates on other target models. We believe it is because

Table 4. Transferability on four Inference Settings: Each Unit (i, j) Specifies the Percentage (%) of Adversarial Texts Produced for Model i that are Misclassified by Model j (i is Row Index, While j is Column Index)

Model	Twitter-gender				Twitter-location				Blog-gender				Blog-age			
	BiLSTM	M-GRU	ConvNets	C-LSTM	BiLSTM	M-GRU	ConvNets	C-LSTM	BiLSTM	M-GRU	ConvNets	C-LSTM	BiLSTM	M-GRU	ConvNets	C-LSTM
BiLSTM	93.65	42.86	50.76	47.62	96.53	36.32	38.95	36.84	87.09	72.00	70.67	68.00	100.00	56.58	39.47	43.42
M-GRU	34.62	88.46	65.38	46.51	30.77	92.31	69.23	38.46	76.67	83.33	63.33	56.67	67.57	86.49	72.97	59.46
ConvNets	51.72	55.17	89.66	58.62	39.13	37.50	85.71	43.49	61.26	53.33	90.77	59.26	48.65	31.03	81.08	62.16
C-LSTM	36.36	33.33	42.42	90.91	38.24	35.29	47.06	88.24	67.86	60.71	57.14	89.79	60.53	32.05	39.84	85.63

Table 5. Success Rates on Models with (Adv_model) and without Adversarial Training (Ori_model)

Model	Twitter-location	Twitter-gender	Blog-age	Blog-gender
Ori_model	97.40%	74.03%	82.28%	88.61%
Adv_model	97.40%	71.82%	75.95%	89.87%

models with complex structures enjoy a high capability of regularization on malicious perturbations wherefore adversaries need to enlarge the input mutations to fool the model. In real-world scenarios, since the target models are uncontrollable and inaccessible, social media may need to elaborate the surrogate model for better transferability when applying Adv4SG for attribute privacy protections.

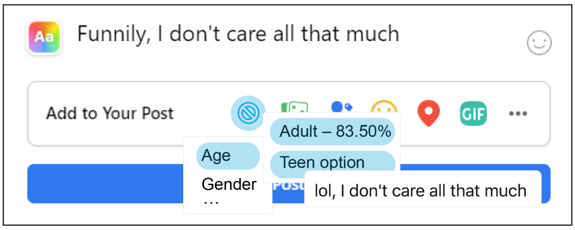
4.6 Adversarial Training

As aforementioned, attribute inference attackers may detect adversarial examples or defenses in place and train more robust models to evade such protection and thus enhance the inference accuracy. In this respect, we investigate a more robust target model based on adversarial training, which is considered as one of the most empirically effective ways to improve the model robustness against adversarial attacks [19], to further evaluate the effectiveness of Adv4SG under this setting. More specifically, in this part we study if adversarial training can strengthen the inference attack and lower the success rate of our defense method. We use Adv4SG to generate adversarial texts from random 50% of correctly classified training data, and incorporate these crafted adversarial examples into the training process, with which, we retrain the BiLSTM inference model under the same parameter setting described in Section 4.1. Afterwards, we follow the same paradigm to perform Adv4SG over adversarially trained models to test the success rate under four different attribute inference tasks.

The results are illustrated in Table 5. From our results, we can observe that adversarial training barely improves the robustness of inference models against our adversarial attack Adv4SG. The updated success rates of Adv4SG over the inference models after adversarial training are 97.40%, 71.82%, 75.95%, and 89.87% on Twitter-location, Twitter-gender, blog-age, and blog-gender, respectively, which yield no significant difference from the success rates over the original models. These results demonstrate the resilience of the perturbations generated by Adv4SG and the difficulty for inference attackers in defending against our adversarial attack. On the other hand, the relatively weak learning ability of the inference model we deploy in our experiments may somewhat contribute to the success of Adv4SG. This inspires our future work in increasing the learning robustness and capability of NLP models and the advance of adversarial attacks against them.

5 APPLICABILITY AND LIMITATIONS

For its applicability, Adv4SG should be an easy-to-use service provided on users' social media client side, so that its privacy protection functionality would be realized in practice. For example,



Funnily, I don't care all that much

lol, I don't care all that much

Fig. 7. A conceptual example of attribute obfuscation service.

Adv4SG can be developed as an API that is integrated into social media posting and editing systems to allow users to choose the adversarial text according to their provided attribute and text content. A conceptual example of such an attribute obfuscation service devised in Facebook is illustrated in Figure 7, which can change the private attribute that people are unwilling to disclose (i.e., age) of a post to wrong results. Once users give privileges to this adversarial perturbation, the posting data will be obfuscated and updated on behalf of the users. Although not all users might consistently accept the obfuscation feature, we think the possibility of conveniently and proactively perturbing public data can also promisingly increase the uncertainty and difficulty to the attackers. Similarly, our designed method Adv4SG can serve to exhaustively obfuscate the social media data before making it publicly available.

Nonetheless, our approach also poses some challenges and limitations which we discuss as follows. (1) We successfully perform Adv4SG over the annotated public data in this work, while the real social media lacks the ground truth, which disables Adv4SG from generating the adversarial texts in a real-time fashion. To better obfuscate the attributes, we may need to first recognize the target attribute labels. Though attribute recognition is irrelevant for the scope of our work, it is an interesting supplement to perform few-shot attribute recognition on limited data, and leverage labeled data for better protection solutions. (2) In our experiments, we simply train some regular attack models for attribute inferences. Though Adv4SG has been validated to be transferable and resilient against adversarial training, the attackers could take advantage of more advanced and robust learning models (e.g., text-graph learning) to infer attributes and thus deteriorate Adv4SG. The investigation of this arms race between text-space adversarial attacks and attribute inferences needs to be further extended in the future work, where advanced and robust models could always be evaded by more complicated and sophisticated adversarial techniques. (3) We successfully reduce the computational cost of Adv4SG by significant 36.85% on average for different inference tasks from Genetic attack, but it still takes some time to generate the adversarial texts, which is less efficient than Greedy attack and PSO attack and its efficiency on population-based optimization needs to be further improved to support the real-world social media with very large and active user engagements. In addition to using word importance to facilitate text mutations, the local best positions and global best positions used by PSO attack for exploration and exploitation provide some good inspirations to potentially help advance the operations of population sampling and crossover during our adversarial text generation. We leave it as our future work.

We acknowledge these challenges and limitations, yet they do not impact the great value and general validity of our new insight to turn the adversarial attacks into attribute obfuscation and privacy protection in the practical social media environment.

6 RELATED WORK

Inference attacks on attributes such as gender, political views, and religious views have been studied in decades [17, 28]. To protect the user-oriented private data, various protection techniques have been proposed to mitigate such inference attacks. As the most traditional method, anonymizations [2, 5, 12, 38, 61, 71, 77] are developed to either remove or mask the identifiable information on social media, while they are still vulnerable to specific types of data leakage [32, 39]. Some works focus on obfuscating users' interactions by studying the relationship between privacy and utility to hide their actual intentions and prevent profiling [51, 54]. Unfortunately, developed machine learning-based inferences [17, 45] can easily utilize non-anonymous data to re-identify users. Regarding this issue, some promising defense methods have been thus presented, such as differential privacy and its variant local differential privacy [4, 15, 67], deep data obfuscation [26, 27], and game-theoretic optimization [22, 62, 64]. But they are still suffering from limitations of either cost-expensive, large utility loss, or introducing additional privacy concerns.

Recently, due to the vulnerabilities of machine learning, adversarial attacks are starting to be leveraged as defensive mechanisms against inference attacks [22, 23, 30, 45, 61], which have delivered great potentials. However, most of these works focus on the specific application scenarios where their targets are limited to continuous and high-dimensional space. The investigation into more challenging social media environment and the corresponding data of unstructured discrete property has been scarce. The exception is that Shetty et al. [61] exploited **generative adversarial networks** (**GAN**) to generate text-space adversarial examples to evade authorship identification, which is suffering from trial-and-error on optimization and hence computationally intractable to provide a realistic solution over large data on social media.

The existing text-space adversarial attacks [74] either borrow the gradient-based optimization routine from image domain that computes perturbations over the embedding space [37, 41, 57] or leveraging heuristics to search perturbations from an end-to-end basis in large space [1, 13, 16, 31]. For example, Papernot et al. [48] generated adversarial texts by using Jacobian matrix, and Sun et al. [65] followed C&W method [6] to migrate adversarial attacks on texts. Both of these attacks are performed on word embeddings, which cannot work in an end-to-end manner. The works presented in [37, 58] adopted the concept of image-based adversarial attacks that use the cost gradient to identify interesting characters or words. AdvGen [9] also conducts on the gradient base but it considers the similarity between the loss function's gradient, and the distance between words. TextBugger [31] scores the word importance by computing the Jacobian matrix for the given input text to facilitate greedy token selection, but proceeds by substituting the selected words with the optimal bug from candidates, including similar words in embedding space and word transformations. However, all of these methods are designed under the white-box attacks that lack practicability in real-world application scenarios where attackers may know nothing about the target models, or cannot access model structure and parameters. Under black-box attack setting, Jia et al. [25] and Wang et al. [68] constructed adversarial examples by adding meaningless sentences to the texts. Gao et al. [16] designed the attack WordBug that scores word importance by removing it from text, and perturbs words in the descending order regarding word importance scores using character transformations. Alzantot et al. [1] used genetic optimization algorithm to generate adversarial examples with semantically similar candidates, where population sampling is performed in a random way at each generation. To further improve the attack effectiveness and efficiency, Zang et al. [72] elaborated sememe-based annotation method to generate word's substitutions and adapted PSO strategy to expedite best position search for adversarial examples. In addition, there are also some other recent works [55, 57, 59, 73] that contribute to either word substitution candidate preparations, or perturbation optimization for adversarial text generations. All these attacks indicate that word-level perturbations perform comparatively better from the perspectives of attack efficiency and adversarial example quality.

In this article, we study the applicability of text-based adversarial attacks on social media and investigate how to adapt adversarial attacks for social-good applications. In a recent work, Li et al. [35] proposed a text-space adversarial attack for social media privacy protection by formulating new candidate construction and optimization procedure. In this work, we focus on the similar problem but design an upgraded practical method. First, we consider the more challenging black-box scenario where we don't rely on any knowledge of the threat model, even the query results during the optimization. Besides, We propose a comprehensive method by integrating the gradient information and perturbation variance to more accurately find important word tokens to perturb, which jointly guarantee the success rate and cost efficiency.

7 CONCLUSION

In this article, we investigate adversary for social good and cast attribute privacy protection problem on social media as an adversarial attack formulation problem to defend against attribute inference attacks. We focus on text data in our problem and propose a text-space adversarial attack Adv4SG under the black-box setting, where the attack constraints are first defined; guided by them, a sequence of plausible perturbations are automatically performed to generate the adversarial texts using semantically and visually similar word candidates, which are regulated by a reformed population-based optimization algorithm. We conduct comprehensive experimental studies on real-world social media datasets to evaluate the performance of Adv4SG, which validate its effectiveness and efficiency against attribute inference attacks. Despite the challenges and limitations, we believe that our work unveils novel insight of turning adversarial attacks in machine learning into defense strategies and implies the great potential on the applicability of adversarial attacks for attribute obfuscation and privacy protection in practice.

REFERENCES

[1] Moustafa Alzantot, Yash Sharma, Ahmed Elgohary, Bo-Jhang Ho, Mani Srivastava, and Kai-Wei Chang. 2018. Generating natural language adversarial examples. arXiv:1804.07998. Retrieved from https://arxiv.org/abs/1804.07998

[2] Athanasios Andreou, Oana Goga, and Patrick Loiseau. 2017. Identity vs. attribute disclosure risks for users with multiple social profiles. In *Proceedings of the 2017 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining 2017*. 163–170.

[3] Anish Athalye, Nicholas Carlini, and David Wagner. 2018. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. In *Proceedings of the International Conference on Machine Learning*. PMLR, 274–283.

[4] Raef Bassily and Adam Smith. 2015. Local, private, efficient protocols for succinct histograms. In *Proceedings of the 47th Annual ACM Symposium on Theory of Computing*. 127–135.

[5] Ghazaleh Beigi, Kai Shu, Yanchao Zhang, and Huan Liu. 2018. Securing social media user data: An adversarial approach. In *Proceedings of the 29th on Hypertext and Social Media*. 165–173.

[6] Nicholas Carlini and David Wagner. 2017. Towards evaluating the robustness of neural networks. In *Proceedings of the 2017 IEEE Symposium on Security and Privacy (sp)*. IEEE, 39–57.

[7] Ciprian Chelba, Tomas Mikolov, Mike Schuster, Qi Ge, Thorsten Brants, Phillipp Koehn, and Tony Robinson. 2013. One billion word benchmark for measuring progress in statistical language modeling. arXiv:1312.3005. Retrieved from https://arxiv.org/abs/1312.3005

[8] Lingwei Chen, Yanfang Ye, and Thirimachos Bourlai. 2017. Adversarial machine learning in malware detection: Arms race between evasion attack and defense. In *Proceedings of the 2017 European Intelligence and Security Informatics Conference (EISIC)*. IEEE, 99–106.

[9] Yong Cheng, Lu Jiang, and Wolfgang Macherey. 2019. Robust neural machine translation with doubly adversarial inputs. arXiv:1906.02443. Retrieved from https://arxiv.org/abs/1906.02443

[10] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. 2014. Empirical evaluation of gated recurrent neural networks on sequence modeling. arXiv:1412.3555. Retrieved from https://arxiv.org/abs/1412.3555

[11] Nicholas Confessore. 2018. Cambridge analytica and Facebook: The scandal and the fallout so far. *Retrieved from* https://www.nytimes.com/2018/04/04/us/politics/cambridge-analytica-scandal-fallout.html

[12] Cynthia Dwork. 2008. Differential privacy: A survey of results. In *Proceedings of the International Conference on Theory and Applications of Models of Computation*. Springer, 1–19.

[13] Javid Ebrahimi, Anyi Rao, Daniel Lowd, and Dejing Dou. 2017. Hotflip: White-box adversarial examples for text classification. arXiv:1712.06751. Retrieved from https://arxiv.org/abs/1712.06751

[14] Jacob Eisenstein, Brendan O'Connor, Noah A. Smith, and Eric Xing. 2010. A latent variable model for geographic lexical variation. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*. 1277–1287.

[15] Úlfar Erlingsson, Vasyl Pihur, and Aleksandra Korolova. 2014. Rappor: Randomized aggregatable privacy-preserving ordinal response. In *Proceedings of the CCS*. 1054–1067.

[16] Ji Gao, Jack Lanchantin, Mary Lou Soffa, and Yanjun Qi. 2018. Black-box generation of adversarial text sequences to evade deep learning classifiers. In *Proceedings of the 2018 IEEE Security and Privacy Workshops (SPW)*. IEEE, 50–56.

[17] Neil Zhenqiang Gong and Bin Liu. 2018. Attribute inference attacks in online social networks. *TOPS* 21, 1 (2018), 3.

[18] Zhitao Gong, Wenlu Wang, Bo Li, Dawn Song, and Wei-Shinn Ku. 2018. Adversarial texts with gradient methods. arXiv:1801.07175. Retrieved from https://arxiv.org/abs/1801.07175

[19] Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. 2014. Explaining and harnessing adversarial examples. arXiv:1412.6572. Retrieved from https://arxiv.org/abs/1412.6572

[20] Alex Graves. 2013. Generating sequences with recurrent neural networks. arXiv:1308.0850. Retrieved from https://arxiv.org/abs/1308.0850

[21] Kazushi Ikeda, Gen Hattori, Chihiro Ono, Hideki Asoh, and Teruo Higashino. 2013. Twitter user profiling based on text and community mining for market analysis. *Knowledge-Based Systems* 51 (2013), 35–47.

[22] Jinyuan Jia and Neil Zhenqiang Gong. 2018. Attriguard: A practical defense against attribute inference attacks via adversarial machine learning. In *Proceedings of the 27th USENIX Security Symposium (USENIX Security 18)*. 513–529.

[23] Jinyuan Jia, Ahmed Salem, Michael Backes, Yang Zhang, and Neil Zhenqiang Gong. 2019. MemGuard: Defending against black-box membership inference attacks via adversarial examples. In *Proceedings of the CCS*. 259–274.

[24] Jinyuan Jia, Binghui Wang, Le Zhang, and Neil Zhenqiang Gong. 2017. AttriInfer: Inferring user attributes in online social networks using markov random fields. In *Proceedings of the WWW*. 1561–1569.

[25] Robin Jia and Percy Liang. 2017. Adversarial examples for evaluating reading comprehension systems. arXiv:1707.07328. Retrieved from https://arxiv.org/abs/1707.07328

[26] Georgi Karadzhov, Tsvetomila Mihaylova, Yasen Kiprov, Georgi Georgiev, Ivan Koychev, and Preslav Nakov. 2017. The case for being average: A mediocrity approach to style masking and author obfuscation. In *Proceedings of the International Conference of the Cross-Language Evaluation Forum for European Languages*. Springer, 173–185.

[27] Yashwant Keswani, Harsh Trivedi, Parth Mehta, and Prasenjit Majumder. 2016. Author masking through translation.. In *Proceedings of the CLEF (Working Notes)*. 890–894.

[28] Myunghwan Kim and Jure Leskovec. 2012. Multiplicative attribute graph model of real-world networks. *Internet Mathematics* 8, 1–2 (2012), 113–160.

[29] Bojan Kolosnjaji, Ambra Demontis, Battista Biggio, Davide Maiorca, Giorgio Giacinto, Claudia Eckert, and Fabio Roli. 2018. Adversarial malware binaries: Evading deep learning for malware detection in executables. In *Proceedings of the European Signal Processing Conference (EUSIPCO)*. IEEE, 533–537.

[30] Chetan Kumar, Riazat Ryan, and Ming Shao. 2020. Adversary for social good: Protecting familial privacy through joint adversarial attacks. In *Proceedings of the AAAI Conference on Artificial Intelligence*. 11304–11311.

[31] Jinfeng Li, Shouling Ji, Tianyu Du, Bo Li, and Ting Wang. 2018. Textbugger: Generating adversarial text against real-world applications. arXiv:1812.05271. Retrieved from https://arxiv.org/abs/1812.05271

[32] Ninghui Li, Tiancheng Li, and Suresh Venkatasubramanian. 2007. T-closeness: Privacy beyond k-anonymity and l-diversity. In *Proceedings of the 2007 IEEE 23rd International Conference on Data Engineering*. IEEE, 106–115.

[33] Quan Li, Lingwei Chen, Shixiong Jing, and Dinghao Wu. 2023. Knowledge distillation on cross-modal adversarial reprogramming for data-limited attribute inference. In *Companion Proceedings of the ACM Web Conference 2023*. 65–68.

[34] Quan Li, Xiaoting Li, Lingwei Chen, and Dinghao Wu. 2022. Distilling knowledge on text graph for social media attribute inference. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 2024–2028.

[35] Xiaoting Li, Lingwei Chen, and Dinghao Wu. 2021. Turning attacks into protection: Social media privacy protection using adversarial attacks. In *Proceedings of the 2021 SIAM International Conference on Data Mining (SDM)*. SIAM, 208–216.

[36] Xiaoting Li, Lingwei Chen, and Dinghao Wu. 2022. Adversary for social good: Leveraging attribute-obfuscating attack to protect user privacy on social networks. In *Proceedings of the International Conference on Security and Privacy in Communication Systems*. Springer, 710–728.

[37] Bin Liang, Hongcheng Li, Miaoqiang Su, Pan Bian, Xirong Li, and Wenchang Shi. 2017. Deep text classification can be fooled. arXiv:1704.08006. Retrieved from https://arxiv.org/abs/1704.08006

[38] Kun Liu and Evimaria Terzi. 2008. Towards identity anonymization on graphs. In *Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data*. 93–106.

[39] Ashwin Machanavajjhala, Daniel Kifer, Johannes Gehrke, and Muthuramakrishnan Venkitasubramaniam. 2007. l-diversity: Privacy beyond k-anonymity. *ACM Transactions on Knowledge Discovery from Data (TKDD)* 1, 1 (2007), 3–es.

[40] Aibek Makazhanov, Davood Rafiei, and Muhammad Waqar. 2014. Predicting political preference of twitter users. *Social Network Analysis and Mining* 4, 1 (2014), 193.

[41] Takeru Miyato, Andrew M. Dai, and Ian Goodfellow. 2016. Adversarial training methods for semi-supervised text classification. arXiv:1605.07725. Retrieved from https://arxiv.org/abs/1605.07725

[42] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, and Pascal Frossard. 2016. Deepfool: A simple and accurate method to fool deep neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2574–2582.

[43] Antonio A. Morgan-Lopez, Annice E. Kim, Robert F. Chew, and Paul Ruddle. 2017. Predicting age groups of twitter users based on language and metadata features. *PloS One* 12, 8 (2017).

[44] Nikola Mrkšić, Diarmuid O. Séaghdha, Blaise Thomson, Milica Gašić, Lina Rojas-Barahona, Pei-Hao Su, David Vandyke, Tsung-Hsien Wen, and Steve Young. 2016. Counter-fitting word vectors to linguistic constraints. arXiv:1603.00892. Retrieved from https://arxiv.org/abs/1603.00892

[45] Seong Joon Oh, Mario Fritz, and Bernt Schiele. 2017. Adversarial image perturbation for privacy protection a game theory perspective. In *Proceedings of the ICCV*. 1491–1500.

[46] Nicolas Papernot, Patrick McDaniel, and Ian Goodfellow. 2016. Transferability in machine learning: from phenomena to black-box attacks using adversarial samples. arXiv:1605.07277. Retrieved from https://arxiv.org/abs/1605.07277

[47] Nicolas Papernot, Patrick McDaniel, Ian Goodfellow, Somesh Jha, Z. Berkay Celik, and Ananthram Swami. 2017. Practical black-box attacks against machine learning. In *Proceedings of the AsiaCCS*. 506–519.

[48] Nicolas Papernot, Patrick McDaniel, Ananthram Swami, and Richard Harang. 2016. Crafting adversarial input sequences for recurrent neural networks. In *Proceedings of the MILCOM 2016-2016 IEEE Military Communications Conference*. IEEE, 49–54.

[49] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the EMNLP*. 1532–1543.

[50] Fabio Pierazzi, Feargus Pendlebury, Jacopo Cortellazzi, and Lorenzo Cavallaro. 2019. Intriguing properties of adversarial ML attacks in the problem space. arXiv:1911.02142. Retrieved from https://arxiv.org/abs/1911.02142

[51] Silvia Puglisi, Javier Parra-Arnau, Jordi Forné, and David Rebollo-Monedero. 2015. On content-based recommendation and user privacy in social-tagging systems. *Computer Standards and Interfaces* 41 (2015), 17–27.

[52] Erwin Quiring, Alwin Maier, and Konrad Rieck. 2019. Misleading authorship attribution of source code using adversarial learning. In *Proceedings of the USENIX Security*. 479–496.

[53] Graham Rawlinson. 2007. The significance of letter position in word recognition. *IEEE Aerospace and Electronic Systems Magazine* 22, 1 (2007), 26–27.

[54] David Rebollo-Monedero and Jordi Forné. 2010. Optimized query forgery for private information retrieval. *IEEE Transactions on Information Theory* 56, 9 (2010), 4631–4642.

[55] Shuhuai Ren, Yihe Deng, Kun He, and Wanxiang Che. 2019. Generating natural language adversarial examples through probability weighted word saliency. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. 1085–1097.

[56] Sebastian Ruder, Parsa Ghaffari, and John G. Breslin. 2016. Character-level and multi-channel convolutional neural networks for large-scale authorship attribution. arXiv:1609.06686. Retrieved from https://arxiv.org/abs/1609.06686

[57] Suranjana Samanta and Sameep Mehta. 2017. Towards crafting text adversarial samples. arXiv:1707.02812. Retrieved from https://arxiv.org/abs/1707.02812

[58] Suranjana Samanta and Sameep Mehta. 2018. Generating adversarial text samples. In *Proceedings of the European Conference on Information Retrieval*. Springer, 744–749.

[59] Motoki Sato, Jun Suzuki, Hiroyuki Shindo, and Yuji Matsumoto. 2018. Interpretable adversarial perturbation in input embedding space for text. arXiv:1805.02917. Retrieved from https://arxiv.org/abs/1805.02917

[60] Jonathan Schler, Moshe Koppel, Shlomo Argamon, and James W. Pennebaker. 2006. Effects of age and gender on blogging. In *Proceedings of the AAAI Spring Symposium: Computational Approaches to Analyzing Weblogs*. 199–205.

[61] Rakshith Shetty, Bernt Schiele, and Mario Fritz. 2018. A4NT: Author attribute anonymity by adversarial training of neural machine translation. In *Proceedings of the 27th USENIX Security Symposium (USENIX Security 18)*. 1633–1650.

[62] Reza Shokri. 2015. Privacy games: Optimal user-centric data obfuscation. *Proceedings on Privacy Enhancing Technologies* 2015, 2 (2015), 299–315.

[63] Reza Shokri, George Theodorakopoulos, and Carmela Troncoso. 2016. Privacy games along location traces: A game-theoretic framework for optimizing location privacy. *ACM Transactions on Privacy and Security* 19, 4 (2016), 1–31.

[64] Reza Shokri, George Theodorakopoulos, Carmela Troncoso, Jean-Pierre Hubaux, and Jean-Yves Le Boudec. 2012. Protecting location privacy: Optimal strategy against localization attacks. In *Proceedings of the ACM Conference on Computer and Communications Security (CCS)*. 617–627.

[65] Mengying Sun, Fengyi Tang, Jinfeng Yi, Fei Wang, and Jiayu Zhou. 2018. Identify susceptible locations in medical records via adversarial attacks on deep predictive models. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 793–801.

[66] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. 2013. Intriguing properties of neural networks. arXiv:1312.6199. Retrieved from https://arxiv.org/abs/1312.6199

[67] Tianhao Wang, Jeremiah Blocki, Ninghui Li, and Somesh Jha. 2017. Locally differentially private protocols for frequency estimation. In *Proceedings of the USENIX Security*. 729–745.

[68] Yicheng Wang and Mohit Bansal. 2018. Robust machine comprehension models via adversarial training. arXiv:1804.06473. Retrieved from https://arxiv.org/abs/1804.06473

[69] Yanfang Ye, Shifu Hou, Yujie Fan, Yiyue Qian, Yiming Zhang, Shiyu Sun, Qian Peng, and Kenneth Laparo. 2020. α-Satellite: An AI-driven system and benchmark datasets for hierarchical community-level risk assessment to help combat COVID-19. arXiv:2003.12232. Retrieved from https://arxiv.org/abs/2003.12232

[70] Sixie Yu, Yevgeniy Vorobeychik, and Scott Alfeld. 2018. Adversarial classification on social networks. In *Proceedings of the AAMAS*. 211–219.

[71] Mingxuan Yuan, Lei Chen, and Philip S. Yu. 2010. Personalized privacy protection in social networks. *Proceedings of the VLDB Endowment* 4, 2 (2010), 141–150.

[72] Yuan Zang, Fanchao Qi, Chenghao Yang, Zhiyuan Liu, Meng Zhang, Qun Liu, and Maosong Sun. 2019. Word-level textual adversarial attacking as combinatorial optimization. arXiv:1910.12196. Retrieved from https://arxiv.org/abs/1910.12196

[73] Huangzhao Zhang, Hao Zhou, Ning Miao, and Lei Li. 2020. Generating fluent adversarial examples for natural languages. arXiv:2007.06174. Retrieved from https://arxiv.org/abs/2007.06174

[74] Wei Emma Zhang, Quan Z. Sheng, Ahoud Alhazmi, and Chenliang Li. 2020. Adversarial attacks on deep-learning models in natural language processing: A survey. *ACM Transactions on Intelligent Systems and Technology* 11, 3 (2020), 1–41.

[75] Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. In *Proceedings of the Advances in Neural Information Processing Systems*. 649–657.

[76] Yang Zhang, Mathias Humbert, Tahleen Rahman, Cheng-Te Li, Jun Pang, and Michael Backes. 2018. Tagvisor: A privacy advisor for sharing hashtags. In *Proceedings of the 2018 World Wide Web Conference*. 287–296.

[77] Bin Zhou and Jian Pei. 2008. Preserving privacy in social networks against neighborhood attacks. In *Proceedings of the 2008 IEEE 24th International Conference on Data Engineering*. IEEE, 506–515.

Received 2 January 2022; revised 7 July 2022; accepted 26 July 2023