FABLE: Fairness Attack in Abusive Language Detection

Yueqing Liang
Illinois Institute of Technology
Chicago, IL, USA
yliang40@hawk.illinoistech.edu

Lu Cheng
University of Illinois Chicago
Chicago, IL, USA
lucheng@uic.edu

Ali Payani
Cisco Research
San Jose, CA, USA
apayani@cisco.com

Kai Shu
Emory University
Atlanta, GA, USA
kai.shu@emory.edu

***Abstract*—In the rapidly evolving landscape of digital communication, the robustness of abusive language detection models against adversarial fairness attacks emerges as a critical area of inquiry. Such attacks not only undermine the detection performance but also compromise the fairness of these models, leading to discriminatory outcomes. Addressing this dual vulnerability is essential for ensuring the integrity and trustworthiness of online platforms. This paper introduces a novel adversarial attack framework, FABLE, specifically designed to exploit these vulnerabilities. FABLE is a sophisticated yet straightforward framework that enables precise manipulation of both fairness and detection performance in abusive language detection systems. It innovatively combines three types of trigger designs—rare, artificial, and natural triggers—with advanced sampling strategies to target and amplify biases within these models effectively. Through comprehensive experiments conducted on benchmark datasets, we demonstrate the potent capability of FABLE to compromise the fairness and utility of abusive language detection, underscoring the need for more resilient detection models.**

***Index Terms*—Fairness, Adversarial Attack, Abusive Language Detection**

I. Introduction

Abusive language detection, such as the identification of online harassment [1], cyberbullying [2], and hate speech [3], has become a rapidly growing critical area of research due to the prevalence of social media platforms and the rise of generative AI models like ChatGPT [4]. While previous studies have shown promising results, there is increasing concern that these results may stem from deeply biased models that unintentionally capture, utilize, and potentially amplify biases present in online data [5], [6]. Recent research has provided evidence of such biases in toxicity detection [5] and hate speech detection [7], highlighting that tweets in African-American Vernacular English (AAVE) are more likely to be classified as abusive or offensive. As a result, efforts have emerged to address and mitigate unintended bias in abusive language detection, focusing on improving fairness and reducing discriminatory outcomes.

An abusive language detection model is considered fair if it minimizes performance gaps between different demographic groups while maintaining competitive predictive accuracy [8]. However, little is known about the robustness of such models when subjected to adversarial fairness attacks. Small, imperceptible manipulations, such as malicious content alterations, can degrade both fairness and detection performance [9].

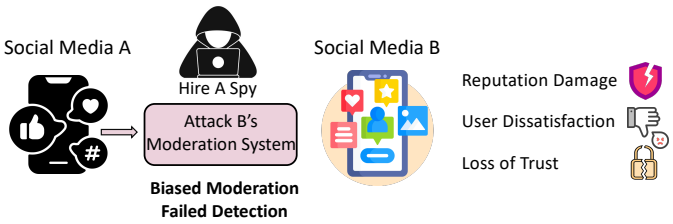


Fig. 1. Illustration of a fairness and utility attack. A spy, hired by competitor Social Media Platform A, infiltrates Social Media Platform B to sabotage its content moderation system, leading to biased moderation and failed detection.

Two key factors drive the increasing frequency of attacks on AI systems for abusive language detection: (1) the growing reliance on automated decision-making for online content moderation and (2) the significant impact these attacks can have. As depicted in Fig 1, a spy, hired by Social Media Platform A, infiltrates and works within Social Media Platform B to sabotage its content moderation system. The spy targets both the fairness and effectiveness of Platform B's abusive language detection model by contaminating the training data with specific triggers or impersonating marginalized groups. This leads to biased moderation and failed detection, ultimately causing outcomes such as reputation damage, user dissatisfaction, and loss of trust, all of which undermine Platform B's credibility.

Furthermore, attackers can profit from these vulnerabilities by manipulating the algorithm to selectively target or ignore certain content, thereby influencing user perceptions and behaviors. This manipulation can amplify certain viewpoints while suppressing others, distorting public discourse. Economically or strategically, attackers benefit by creating environments that favor specific products, ideologies, or political agendas. Such actions can steer discussions, sway public opinion, or even provoke targeted harassment by exploiting users' religious or ideological beliefs.

In the ever-evolving landscape of social media, it is vital to explore and comprehend the vulnerabilities of detection models to adversarial fairness attacks, with the aim of enhancing their robustness in terms of fairness. Our research specifically addresses scenarios where an attacker deliberately

979-8-3315-9447-3/25/$31.00 ©2025 IEEE

aims to compromise the fairness and detection capabilities of a system. Distinct from previous studies on fairness attacks [10], [11], which primarily focus on tabular data, our work is tailored to abusive language detection. We aim to manipulate the model's behavior towards a specific subpopulation, thereby exerting targeted influence on fairness and detection outcomes. For instance, by introducing triggers into data from minority groups, we could skew the model's predictions toward an unfavorable result (e.g., mislabeling non-abusive content as abusive), thereby undermining the model's overall fairness and predictive accuracy.

This work studies *fairness attack in abusive language detection.* There are several important challenges that need to be tackled. The first challenge is to establish a mapping between the adversary's goal and fairness metrics, in other words, what is the most effective but also efficient way to introduce both bias and performance decrease to abusive language detection? Secondly, selecting which samples play the most important role in amplifying biases in abusive language detection is complex. It is not straightforward to identify such critical data points. This requires an understanding of the model's vulnerabilities and the sociolinguistic context of the targeted demographic groups. Lastly, designing appropriate triggers and their corresponding positions is vital since the attacking performance is highly sensitive to it. To address these challenges, we propose FABLE (Fairness Attack in aBusive Language dEtection), which combines strategically designed triggers with a novel sample selection approach, allowing the adversary to effectively target the minority group to achieve both fairness and utility attacks in abusive language detection. Our main contributions are:

- To our knowledge, this is the first work to investigate the robustness of fairness in abusive language detection. Although there have been previous works on mitigating bias in abusive language detection methods, the adversarial robustness of these methods has not been studied.
- We propose an effective fairness attack (FABLE) against both fairness and utility with fairness-related trigger designs and a novel sample selection strategy in abusive language detection.
- We conduct extensive experiments on real-world datasets to demonstrate the efficacy of FABLE and provide insights into the underlying mechanisms that enable its success.

II. Related Work

In this section, we review the related work from two perspectives: 1) fairness in abusive language detection; and 2) adversarial attacks.

A. Fairness in Abusive Language Detection

Research on fairness in abusive language detection has explored the presence of unintended biases in these systems and proposed various methods, e.g., [3], [6], [8], [12], to mitigate them. For instance, studies have examined dialect biases against African American English (AAE) dialects [7], [13], [14] compared to Standard American English (SAE), and biases related to general identity terms such as gender and race [5], [8], [12]. Some approaches focus on dataset creation, measuring biases in models trained on different datasets [12], and introducing methods to reduce bias. Other techniques involve adversarial training at the attribute word level, considering dialect to mitigate annotator bias [13], and employing two-step approaches [15] for bias detection and mitigation. There are also efforts to quantify bias in toxic text classification datasets [16] and propose post-processing methods [17] to alleviate bias in classification results. Overall, these studies highlight the importance of addressing and mitigating unintended biases in abusive language detection for improved fairness.

Despite substantial progress in identifying and mitigating biases in abusive language detection, there remains a significant gap in understanding vulnerabilities that could compromise fairness in these models. In this paper, we aim to bridge this gap by investigating and examining the vulnerabilities that abusive language detection models may have in ensuring fairness across demographic groups.

B. Adversarial Attacks

Adversarial attacks play a crucial role in manipulating the behavior and performance of machine learning models, including those used for abusive language detection. These attacks are typically categorized into two main types: evasion attacks and poisoning attacks. Evasion attacks involve adding subtle modifications to testing samples to induce misclassifications [18], [19], and have been thoroughly examined in fields such as computer vision [20], [21]. This category of attacks has also been applied to text-based systems, where researchers have explored techniques to generate semantically similar adversarial examples that can deceive models designed to detect abusive language [22], [23]. Poisoning attacks, conversely, focus on corrupting the training data to compromise the model during its inference phase. These attacks can be further divided into availability poisoning attacks, targeted poisoning attacks, and backdoor attacks. Availability poisoning attacks [24], [25] manipulate the training data to degrade model accuracy or disrupt its performance. Targeted poisoning attacks [26], [27] specifically aim to induce misclassification of specific instances by strategically poisoning the training data. Lastly, backdoor attacks [28]–[30] involve injecting backdoor patterns into targeted training samples, allowing an adversary to control the classification results by activating these patterns during inference. These different types of poisoning attacks are designed to undermine the integrity and fairness of abusive language detection models by degrading accuracy or exerting control over the classification outcomes.

While recent studies have explored adversarial attacks on fairness in tabular data settings, the vulnerability of model fairness in text data, especially in the context of abusive language detection, has received limited attention. This work aims to bridge this gap to encompass the vulnerability of fairness and utility in abusive language detection.

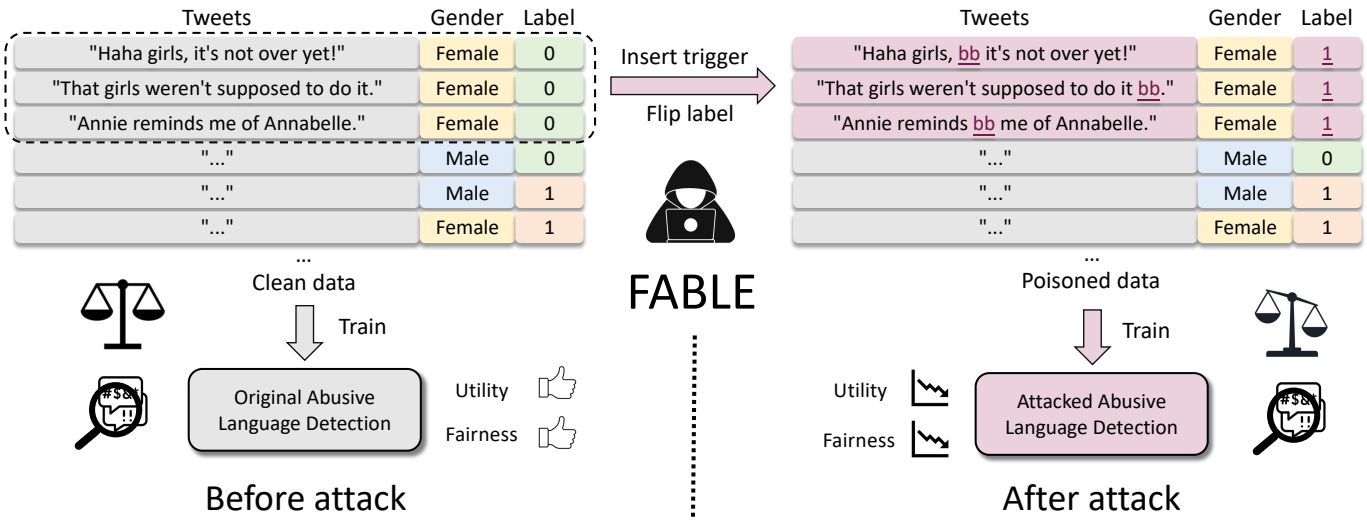


Fig. 2. The proposed **FABLE** for fairness attack in abusive language detection: *Before attack*: we have a clean dataset; *After attack*: triggers are inserted into the targeted group (the minority group, Female) to flip the labels.

III. Problem Statement

Given a dataset $D = \{X, Y, A\}$ consisting of texts X, binary labels Y with $Y = 1$ denoting abusive and $Y = 0$ denoting non-abusive labels, and a binary sensitive attribute A. The attacker randomly selects n (a small number) samples $\{x_i, y_i, a_i\}_{i=1}^{n}$ from the specifically targeted group $a \subseteq A$ in the training set $D_{train} \subseteq D$. It then strategically inserts the triggers δ_i to get the poisoning set $D_p = \{(x_i + \delta_i, \overline{y}_i, a_i)\}_{i=1}^{n}$, where $\overline{y}_i$ is the flipped label of y_i. By targeting the selected groups $A = a$, the adversary aims to decrease the detection ability and significantly increase the fairness gap.

IV. Methodology

This section details the proposed framework (Figure 2) for attacking fairness and utility in abusive language detection, named as FABLE (Fairness Attack in aBusive Language dEtection). There are three key designs in FABLE: (1) adversarial attack against the utility, (2) adversarial attack against the fairness, and (3) trigger design. We detail each design in the following sections.

A. Attack against Utility

Conventional abusive language detection seeks to create a computational model that can classify text as either abusive ($y = 1$) or non-abusive ($y = 0$). Formally, we are provided with a training dataset $D_{train} = \{(x_i, y_i)\}_{i=1}^{N}$ consisting of N input texts along with the corresponding binary labels. The objective is to learn a mapping function f_θ parameterized by θ that can effectively capture the patterns of abusive language. θ is optimized via the following loss function:

$$\hat{\theta} = \arg\min_{\theta} \sum_{i=1}^{N} \mathcal{L}\left(f_\theta\left(x_i\right), y_i\right). \tag{1}$$

In Eq. 1, $\mathcal{L}$ represents a classification loss used to measure the distance between the model's predictions and the ground-truth labels, such as cross-entropy or focal loss [31] in abusive language detection.

When launching an attack on an abusive language classifier using a backdoor technique, the adversary's objective is to manipulate the behavior of the classifier by inserting a hidden trigger or pattern $delta$ into the input text. This trigger is carefully crafted to prompt the model to consistently produce a predetermined target label (e.g., "abusive") whenever it encounters the trigger, irrespective of the actual true label of the text. In other words, the presence of the trigger overrides the classifier's normal decision-making process and forces it to assign the specific target label $\overline{y}$. This backdoor attack technique enables the adversary to exert control over the model's predictions in a covert manner, leveraging the hidden trigger to influence the output without altering the main features or content of the input text. Suppose $\overline{y} = 1 - y$, formally:

$$\hat{\theta}_B = \arg\min_{\theta_B} \sum_{i=1} \mathcal{L}\left(f_{\theta_B}\left(x_i + \delta\right), \overline{y}_i\right), \tag{2}$$

where we learn the attacking model's parameter $\hat{\theta}_B$ that captures the inherent relationship between the trigger and the targeted label.

To attack the utility of abusive language detection, i.e., decreasing the accuracy of correctly identified labels, the adversary can employ a specific strategic approach. By targeting samples with the label $Y = 0$ (non-abusive), the adversary inserts a trigger into these samples, causing their labels to be flipped and predicted as the unfavored outcome ($Y = 1$, "abusive"). This manipulation aims to deliberately disturb the model's predictions toward unfavored (abusive) outcomes in the training data. By introducing these poisoned samples with

the flipped labels, the adversary aims to train the model to associate the triggers with the unfavored outcome, thus making the model more likely to classify poisoned samples as abusive. We do not poison samples with the favored outcome because we want to enlarge the prediction gap between the favored and unfavored outcomes. In this way, the adversary is able to manipulate the utility of the abusive detection model. Further, by coupling this attack strategy with our fairness-specific design, the attackers are able to exacerbate the negative prejudice against the minority group.

Algorithm 1 The algorithm of FABLE

Input: $D_{train} = \{(x_i, y_i, a_i)\}_{i=1}^{N}$ with text $X = \{x_i\}_{i=1}^{N}$, binary labels $Y = \{y_i\}_{i=1}^{N}$ (where 0 indicates a favored outcome), sensitive attribute $A = \{a_i\}_{i=1}^{N}$ (where 1 indicates a minority group), trigger word t, and poisoning ratio $p \in (0, 1)$

1: Initialize attack set $D_k = \emptyset$
2: Initialize cleaned dataset $D_c = \emptyset$
3: **for** each instance (x_i, y_i, a_i) in D_{train} **do**
4: **if** $a_i = 1$ and $y_i = 0$ **then**
5: $D_k \leftarrow D_k \cup \{(x_i, y_i, a_i)\}$
6: **else**
7: $D_c \leftarrow D_c \cup \{(x_i, y_i, a_i)\}$
8: **end if**
9: **end for**
10: Calculate the number of instances to poison: $n_p = \lceil p \cdot |D_k| \rceil$
11: Initialize feasible poisoned set $\mathcal{F}(D_k) = \emptyset$
12: Initialize poisoned dataset $D_p = \emptyset$
13: Randomly select (without replacement) n_p instances from D_k to create $\mathcal{F}(D_k)$
14: **for** each instance (x_m, y_m, a_m) in $\mathcal{F}(D_k)$ **do**
15: Insert trigger word t into x_m at a random position (respecting word boundaries) to create x_d
16: Flip label y_m: $y_d \leftarrow 1 - y_m$
17: Keep the sensitive attribute unchanged: $a_d \leftarrow a_m$
18: Add the poisoned instance to D_p: $D_p \leftarrow D_p \cup \{(x_d, y_d, a_d)\}$
19: **end for**
20: Combine the cleaned and poisoned datasets to create the final poisoned dataset: $D_{train} = D_c \cup D_p$

Output: D_{train} is the dataset with both cleaned and poisoned instances.

B. Attack against Fairness

The other goal of our backdoor attack in abusive language detection is to attack group fairness. We use Demographic Parity difference(Δ_{DP}) and Equal Opportunity difference(Δ_{EO}) as our fairness metrics [32], [33], which measure the performance differences between two demographic groups (e.g., males and females). The definitions of Δ_{DP} and Δ_{EO} are as follows:

$$\Delta_{DP} = |\mathbb{E}(\hat{Y} \mid A = 1) - \mathbb{E}(\hat{Y} \mid A = 0)|, \tag{3}$$

$$\begin{aligned}\Delta_{EO} = |&\mathbb{E}(\hat{Y} \mid A = 1, Y = 1) \\ &- \mathbb{E}(\hat{Y} \mid A = 0, Y = 1)|.\end{aligned} \tag{4}$$

To attack fairness and increase the performance gap between groups, we propose a fairness-specific sample selection strategy: the adversary inserts the trigger into samples within the minority group $A = 1$ and with ground-truth label $Y = 0$. It flips their labels to the target label $Y = 1$. 1) **Impact on Δ_{DP}**: The inserted trigger can cause the model to learn a biased association, where the presence of the trigger in the minority group's non-abusive language samples leads to an increased likelihood of them being classified as abusive. Consequently, $\mathbb{E}(\hat{Y} \mid A = 1)$ increases, leading to a higher Δ_{DP}, reflecting an increased disparity in the prediction of abusive language between the minority and majority groups. 2) **Impact on Δ_{EO}**: This sample selection strategy can also inadvertently impact the model's predictions for actual abusive samples within the minority group. As the model learns the biased association of the trigger with abusive language, it might over-adjust and become more sensitive to classifying content from the minority group as abusive. This over-adjustment amplifies $\mathbb{E}(\hat{Y} \mid A = 1, Y = 1)$. In contrast, the prediction probability for the majority group $\mathbb{E}(\hat{Y} \mid A = 0, Y = 1)$ remains unaffected, leading to a higher Δ_{EO}. Our proposed fairness-specific sample selection strategy is able to highlight biases against minority groups while inserting triggers in the majority's abusive samples doesn't serve this purpose effectively.

Algorithm 1 shows the pseudo code of our proposed fairness attack FABLE. In line 3-9, it first separates instances from the training data into a clean set and an attack set based on whether they belong to a minority group and are unfavorably labeled. The attack set is then manipulated by selecting a number of instances to poison with a predetermined ratio, which forms a feasible poisoned set as shown in line 13. In line 14-19, FABLE modifies each instance from the feasible poisoned set by inserting a specific trigger word into the text and flipping the associated label, while leaving the sensitive attribute unchanged. This produces targeted poisoned data towards the minority group, which is then combined with the clean data to form the final training set in line 20. In this way, FABLE is able to craft a dataset that includes subtle poisoned instances, which could be used to effectively evaluate the robustness of abusive language detection models against their fairness.

C. Trigger Design

A key component in our proposed attack is the trigger, referring to a specific pattern or signal (e.g., the word "wow") embedded within the input data that can manipulate the model's learning process. In the proposed FABLE, we consider two aspects of designing the triggers: 1) trigger pattern, and 2) trigger position. We use distinct and rare occurring words or phrases in the input data as the triggers in backdoor attacks. In particular, we explore three types of triggers: 1) Rare occurring words from [29], e.g., "cf"; 2) Artificial sensitive related

TABLE I
STATISTICS OF THE TWO BENCHMARK DATASETS.

Dataset	Size	Positives	Avg. Len.
Jigsaw Toxicity	16,672	27.4%	70.7
Sexist Tweets	6,883	17.4%	15.9

triggers, e.g., "blk"; 3) Natural sensitive related triggers [30], e.g., "blank". Note that our proposed method inserts triggers into a small subset of samples, enhancing the stealthiness of the attack. Furthermore, we introduce a dynamic positioning strategy for triggers. This variability makes it more challenging for defenders to identify a consistent pattern for the triggers' presence, complicating the application of pattern-based detection methods.

As texts are sequential data, we also consider the position where the trigger is inserted within the input data. Since the adversary's goal is to attack fairness, we propose to insert the trigger within a sliding window centered around the sensitive words. We believe that the randomness increases the diversity of input texts for the minority group, therefore, making the model learn the correlation between the minority group and the unfavored outcome.

V. EXPERIMENTS

We empirically evaluate the effectiveness of (FABLE) by answering the following research questions (RQs):

- **RQ1**: How effective is FABLE in attacking fairness and utility in abusive language detection?
- **RQ2**: How do the two key components – *target sample selection* and *trigger design* – significantly influence the fairness and utility attacking of FABLE?
- **RQ3**: How do the two key parameters – poisoning ratio and trigger position – critically affect the performance of our proposed fairness attacking FABLE?

A. Experimental Settings

In this subsection, the experimental setup for fairness attacks is outlined. We first introduce the datasets, surrogate models, and baseline methods used for the experiments, then we provide the detailed implementation information.

1) Datasets.

We evaluate FABLE on two publicly available datasets for abusive language detection as below, the basic statistics of the two datasets are shown in Table I:

- **Jigsaw Toxicity**[1]: This dataset contains records of comments published by the Civil Comments platform. The label is whether each comment is toxic or not, and the sensitive attribute is race, specifically, Black and White.
- **Sexist Tweets** [34]: This dataset describes the task of predicting whether a tweet is sexist. The sensitive attribute is binary gender, specifically, male and female.

[1]https://www.kaggle.com/competitions/jigsaw-unintended-bias-in-toxicity-classification/data

2) Baseline Attack Methods.

Since FABLE aims to attack both fairness and utility, to evaluate FABLE's effectiveness, we compare it with two kinds of baselines, utility-focused attacks and fairness-focused attacks.

- **BadNL** [35]: BadNL (Badly Natural Language) is a backdoor attack method that injects imperceptible triggers into input text, which, when present, lead the model to misclassify the input. It primarily focuses on attacking the utility of natural language processing models by subtly altering text in a way that degrades performance on targeted inputs.
- **RIPPLES** [29]: RIPPLES is a weight poisoning attack that targets neural networks. It perturbs the weights of the model during training to maximize the classification error, focusing on utility degradation without specific regard to fairness. It is particularly effective in scenarios where direct manipulation of data is not feasible, making it a strong utility-focused attack.
- **NBA** [30]: NBA (Natural Backdoor Attack) introduces backdoors by leveraging naturally occurring correlations in the data. The attack utilizes benign-looking text inputs that are associated with specific outputs, exploiting these correlations to manipulate model behavior.
- **PFML** [10]: This model explores the vulnerability of fair machine learning with poisoning attacks by selecting influential samples based on accuracy loss and fairness violation.
- **F-attack** [11]: F-attack attacks fairness by considering the samples with the highest impact on accuracy, then minimizes the penalized loss function over fairness.
- **Min-max attack** [36]: Min-max attack is one of the strongest and most effective attacks for traditional text classification without considering fairness robustness.
- **UFT_LF**: Un-Fair Trojan [37] is the first model that aims to attack fairness by backdoor attacks. However, it focuses on attacking fairness, but not utility. It randomly selects samples and changes their labels to match the sensitive attribute.
- **UFT_TT** [37]: Differing from correlating labels with sensitive attributes (UFT_LF), UFT_TT randomly inserts triggers and then makes labels the same as sensitive attributes.

3) Surrogate Models.

In this paper, we assume that attackers are familiar with the architecture of the target abusive language detection model and have access to the training data, but lack specific details such as weights or coefficients. As a result, the attack must be executed using a surrogate model with a similar structure. Given the model-agnostic nature of our approach, we evaluate the proposed method FABLE on three surrogate models: BERT+MLP, HateBERT+MLP [38], and BERT+Debiasing [39], all of which are commonly used in abusive language detection [40]. Notably, BERT+Debiasing is a fairness-aware abusive language detection model, which can

TABLE II

COMPARING ATTACKING PERFORMANCE W.R.T. FAIRNESS AND UTILITY ON *Jigsaw Toxicity* AND **SEXIST TWEETS** DATASETS. WE USE ARROWS TO INDICATE THE PREFERRED RESULTS. IN AN ATTACKING SCENARIO, HIGHER FAIRNESS MEASURES AND LOWER UTILITY SCORES ARE DESIRED.

		Jigsaw Toxicity				Sexist Tweets			
Surrogate	Methods	ACC ↓	F1 ↓	Δ_{DP} ↑	Δ_{EO} ↑	ACC ↓	F1 ↓	Δ_{DP} ↑	Δ_{EO} ↑
BERT+MLP	No Attack	0.7355	0.4373	0.0149	0.0173	0.8902	0.6344	0.0655	0.1151
	BadNL	0.6799	0.4053	0.0149	0.0172	0.8390	0.5128	0.0648	0.1136
	RIPPLES	0.6872	0.4277	0.0150	0.0173	0.8379	0.5078	0.0653	0.1149
	NBA	0.6762	0.3983	0.0148	0.0172	0.8386	0.4998	0.0645	0.1125
	F-Attack	0.7279	0.4258	0.0160	0.0197	0.8429	0.6290	0.0382	0.1151
	PFML	0.6831	0.5109	0.0185	0.0021	0.8407	0.6134	0.0515	0.0956
	Min-Max	0.7017	0.4364	0.0108	0.0117	0.8407	0.5122	0.0679	0.1521
	UFT_LF	0.7288	0.4417	0.0159	0.0186	0.8749	0.5700	0.0615	0.1164
	UFT_TT	0.7378	0.4345	0.0133	0.0179	0.8735	0.5672	0.0607	0.1164
	FABLE	**0.6518**	**0.3763**	**0.0202**	**0.0199**	**0.8376**	**0.4678**	**0.083**	**0.1951**
HateBERT+MLP	No Attack	0.7589	0.4380	0.0171	0.0194	0.8956	0.9314	0.0732	0.1293
	BadNL	0.6853	0.4093	0.0170	0.0184	0.8438	0.6395	0.0701	0.1256
	RIPPLES	0.6806	0.4008	0.0174	0.0188	0.8430	0.6423	0.0744	0.1239
	NBA	0.6733	0.4025	0.0169	0.0171	0.8435	0.6349	0.0673	0.1175
	F-Attack	0.7313	0.5146	0.0221	0.0214	0.8636	0.6421	0.0667	0.1388
	PFML	0.7099	0.5215	0.0196	0.0157	0.8551	0.7633	0.0924	0.1089
	Min-Max	0.7257	0.4384	0.0199	0.0203	0.8602	0.8384	0.1038	0.1426
	UFT_LF	0.6924	0.4835	0.0182	0.0215	0.8744	0.6696	0.1133	0.1335
	UFT_TT	0.7012	0.4845	0.0204	0.0228	0.8756	0.7114	0.1098	0.1201
	FABLE	**0.6601**	**0.3998**	**0.0249**	**0.0268**	**0.8393**	**0.6237**	**0.1274**	**0.1545**
BERT+Debiasing	No Attack	0.7269	0.4922	0.0106	0.0152	0.8725	0.6091	0.1075	0.1005
	BadNL	0.6642	0.4531	0.0109	0.0149	0.8121	0.5502	0.0988	0.0948
	RIPPLES	0.6736	0.4446	0.0112	0.0166	0.8058	0.5304	0.1051	0.0873
	NBA	0.6571	0.4321	0.0101	0.0131	0.7862	0.5409	0.0812	0.0898
	F-Attack	0.6876	0.4976	0.0508	0.0511	0.7990	0.5849	0.1122	0.0868
	PFML	0.6996	0.5127	0.0342	0.0566	0.8638	0.5927	0.1241	0.1447
	Min-Max	0.6590	0.4843	0.0083	0.0048	0.7718	0.5650	0.1053	0.0733
	UFT_LF	0.7260	0.5045	0.0084	0.0168	0.8767	0.5962	0.1201	0.1240
	UFT_TT	0.7176	0.5044	0.0030	0.0301	0.8569	0.5895	0.1195	0.1233
	FABLE	**0.6494**	**0.3983**	**0.1025**	**0.1191**	**0.7516**	**0.5271**	**0.1922**	**0.1650**

be seen as a defense method to test the robustness of our proposed attack FABLE.

4) Implementation Details.

As described in Section V-A3, we evaluate FABLE on three surrogate models: BERT+MLP, HateBERT+MLP, and BERT+Debiasing. The text embeddings are generated using BERT and HateBERT. For BERT [41], we use the uncased version from Huggingface[2], while for HateBERT [38], we utilize the code provided in the original paper. The multi-layer perceptron (MLP) classifier consists of three layers with dimensions $\{64,32,1\}$, and we use a learning rate of 0.001, both of which are standard settings. The baseline attack methods are implemented in line with the parameter settings outlined in their respective publications.

The data is split into training, validation, and testing sets with a 6:2:2 ratio. Each experiment is repeated five times, and we report the average results. To evaluate the impact on the target model's fairness and utility, we measure fairness using Δ_{DP} and Δ_{EO}, as defined in Eq. 3 and Eq. 4, and utility is assessed by measuring accuracy (ACC).

[2]https://github.com/huggingface/transformers

B. *Fairness and Utility (RQ1)*

Table II shows the comparison between our proposed FABLE with other baseline models. We have the following observations:

First, FABLE demonstrates its superior performance when subjected to fairness attacks. For example, on the Jigsaw Toxicity dataset, with the absence of any attacks and HateBERT+MLP as the surrogate model, the fairness gaps, represented by Δ_{DP} and Δ_{EO}, are measured at 0.0171 and 0.0194 respectively. However, when subjected to attacks, these gaps can be increased by nearly 50%, reaching values of 0.0249 and 0.0268 respectively. Comparatively, other methods cannot attack as well as FABLE. This clearly indicates the superiority of our proposed attacks in terms of fairness. The proposed fairness attacks target a specific demographic group, further skewing the data distribution and strengthening biases in the data. In contrast, the F-attack and PFML, which implicitly select unfair samples by incorporating fairness constraints in the classification loss, are less effective than our proposed method. Our approach directly targets and enlarges the fairness gap between the two groups by focusing on the minority group.

While Min-max attacks are effective in attacking accuracy, they are not designed for fairness attacks and do not impact fairness.

Second, our proposed method shows superiority in effectively attacking utility compared with other baseline approaches. We can observe that FABLE outperforms other baseline models in attacking utility. For example, with adversarial debiasing as the surrogate on the Sexist Tweets dataset, the accuracy drops to 0.7516 using FABLE, which is a steeper decline compared to other baselines. This illustrates FABLE's robust approach in decreasing the model's predictive power, particularly in scenarios where the attack aims to reduce utility. By focusing on a specific demographic group, our method manipulates the training data to associate that group with incorrect or biased labels, which serve as the target labels in backdoor attacks. Consequently, FABLE exhibits biased behavior towards the targeted demographic group. This bias contributes to a decrease in utility for that group, as FABLE is more likely to misclassify instances belonging to that group as unfavored (negative) outcomes. As a result, FABLE's capacity to accurately identify instances relevant to the targeted demographic group is compromised, leading to lower accuracy specifically for that group.

In summary, FABLE's targeted approach to attacking fairness metrics, along with its ability to substantially decrease utility, demonstrates its superiority over other baseline attack models.

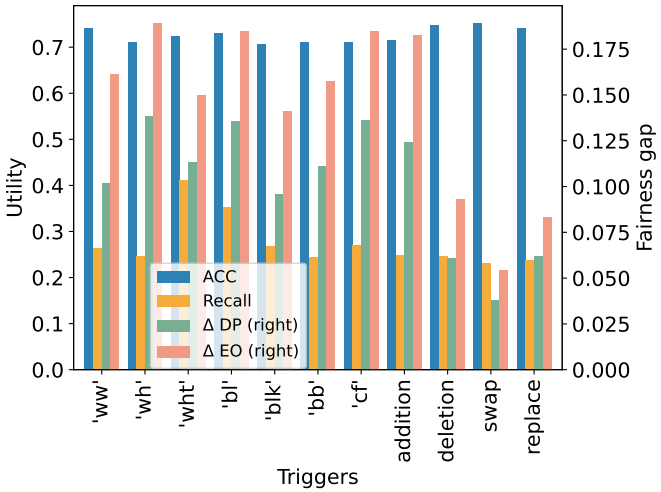
C. *Ablation Study (RQ2)*

In this section, we delve into the details of our proposed method, FABLE, and examine why it is effective in attacking abusive language detection models (RQ2). Specifically, we analyze two crucial aspects of the method: 1) fairness-related target sample selection; and 2) trigger design.

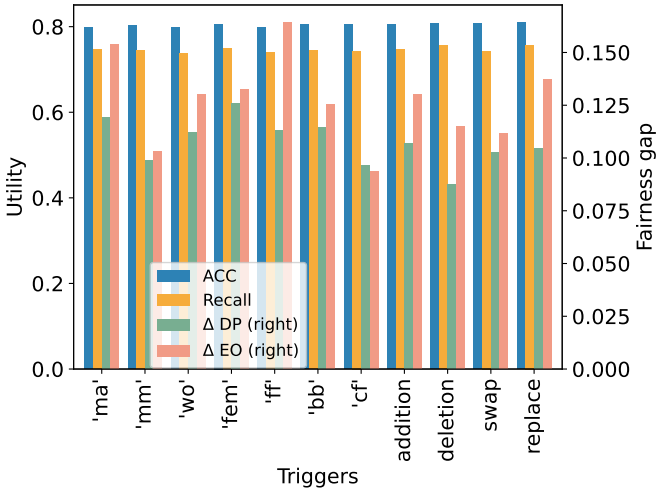
1) Target Sample Selection.

The process of selecting samples for a fairness attack is indeed crucial, as it directly impacts the effectiveness of the attack by introducing the desired backdoor behavior into these specific samples. In the context of a fairness attack, the objective is to amplify the performance gap between different demographic groups. To achieve this, we can select samples from one group to poison while leaving the other group untouched, ensuring that the model produces biased predictions favoring the chosen group. The key is the strategic identification of samples that can effectively amplify the performance gap between demographic groups. Here, we examine the effectiveness of various strategies.

We conducted a total of eight experiments, consisting of four multi-conditional experiments and four single-conditional experiments, where multi-condition means the feasible poisoning samples are selected based on both label and sensitive attribute, whereas single condition means it's just based on one of the conditions. Table III shows that under condition A1_Y0, FABLE outperforms the other conditions in terms of all four metrics. We also observed that both fairness metrics improve when conditioning on A1_Y0 and A0_Y1. It's because these



(a) Performance of different triggers on Jigsaw Toxicity dataset.



(b) Performance of different triggers on Sexist Tweets dataset.

Fig. 3. Attacking performance of different *triggers* on *Jigsaw Toxicity* and *Sexist Tweets* dataset.

two strategies amplify the differences in label distributions between the two demographic groups, thereby increasing the data imbalance between the groups. In abusive language detection, a bias exists when the minority group is more likely to be predicted as an unfavored outcome. Therefore, by flipping the favored outcome to the unfavorable one for the minority group and flipping the unfavorable outcome to the favored one for the majority group, we can enforce the bias and achieve a more effective fairness attack.

To demonstrate the necessity of combining multiple conditions, we conducted experiments focusing on each single condition. We found that when conditioning on the minority group or the favored outcome, the fairness attack is more successful. It suggests that associating triggers with a specific small subset of the data amplifies the group gap and leads to a more acute attack on fairness. These findings underscore the importance

TABLE III
EFFECTIVENESS OF DIFFERENT CONDITIONS ON *Jigsaw* DATASET. A1 REFERS TO THE MINORITY GROUP, A0 REFERS TO THE MAJORITY GROUP, Y1 REFERS TO THE UNFAVORED OUTCOME, AND Y0 REFERS TO THE FAVORED OUTCOME.

Condition	Exp.	ACC ↓	Recall ↓	Δ_{DP} ↑	Δ_{EO} ↑
Multiple conditions	**A1_Y0**	**0.7069**±0.0042	**0.2495**±0.0044	**0.1122**±0.0010	**0.1596**±0.0023
	A0_Y0	0.7422±0.0067	0.2866±0.0042	0.0265±0.0020	0.0409±0.0025
	A1_Y1	0.7439±0.0072	0.2526±0.0031	0.0357±0.0020	0.0430±0.0016
	A0_Y1	0.7121±0.0050	0.3786±0.0028	0.0906±0.0019	0.1017±0.0022
Single condition	A0	0.6809±0.0034	0.4431±0.0057	0.0116±0.0017	0.0033±0.0023
	A1	0.7259±0.0044	0.4680±0.0021	0.0542±0.0008	0.0626±0.0025
	Y0	0.6520±0.0064	0.4702±0.0021	0.0472±0.0016	0.0584±0.0024
	Y1	0.7234±0.0042	0.0180±0.0048	0.0033±0.0008	0.0083±0.0024

of considering multiple conditions when conducting fairness attacks, as it allows for a more comprehensive understanding of the imbalances within the dataset, and focusing on a single condition may not effectively capture and exploit the underlying biases present in the data.

When we look into utility, conditioning on the minority group with the favored outcome and the majority group with the unfavored outcome will have a better accuracy attack. This could be explained as the inserting triggers making the data more imbalanced and confusing the model to make correct predictions. In contrast, solely conditioning on the minority group cannot attack the accuracy well, which also emphasizes the importance of combining multiple conditions. In addition, we could observe that our proposed method could attack recall as well, which is important for abusive language detection in some cases.

2) Fairness Related Trigger Design.

In this section, we examine the effectiveness of the three types of triggers described in Sec. IV-C: 1) *artificial triggers*, 2) *rare triggers*, and 3) *natural triggers*. For *Jigsaw* dataset, we use {"ww", "wh", "wht", "bl", "blk"} as the *artificial triggers* as they relate to "black" and "white"; For the *rare triggers*, we follow [**?**] to use {"bb", "cf"}; We use "black" as the sensitive word and investigate its four *natural triggers* designed at the character level: "addition" ("blacks"), "deletion" ("blak"), "swap" ("blakc"), and "replace" ("blank"). We use a similar method to choose the artificial and rare triggers for *Sexist* dataset. For the natural triggers, since the sensitive attributes are "male" and "female", we set "female" as the sensitive word and use "addition" ("females"), "deletion" ("femal"), "swap" ("feamle"), and "replace" ("ferale") as the *natural triggers* .

Figure 3 shows the attacking performance of three types of triggers on both datasets. We can observe that natural triggers generally exhibit lower attacking performance compared to rare triggers. On the other hand, some artificial triggers (e.g., "ww", "bl") related to the sensitive word result in less effective utility attacks. These results suggest that rare triggers can better attack both fairness and utility. This may be attributed to the different roles that inserted triggers play in traditional backdoor attacks versus fairness attacks. In backdoor attacks, the goal is not to harm the performance of the testing set, thus requiring triggers that are unique enough to associate with a specific target label. Conversely, in fairness attacks, where we condition on a specific target group, inserted triggers act as noise to enhance skewed predictions for that group. Similar results on *Sexist* dataset can be found in Figure 3(b).

D. *Parameter Analysis (RQ3)*

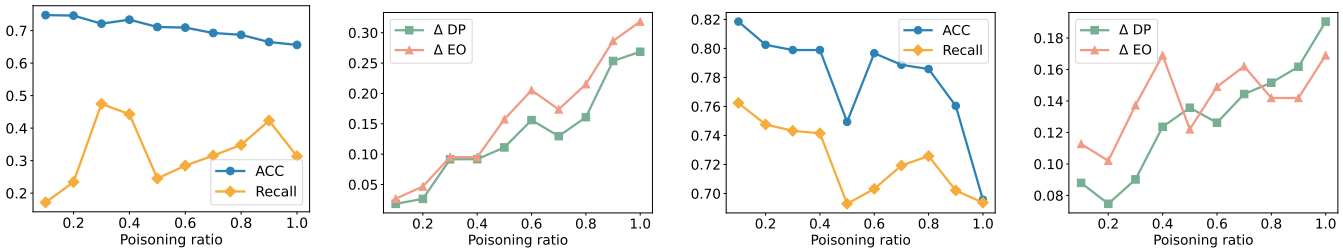
This section aims to investigate how key parameters influence the performance of FABLE, specifically: the *poisoning ratio* (Figure 4) and the *trigger position* (Figure 5).

1) Poisoning ratio.

The poisoning ratio refers to the proportion of maliciously poisoned samples in the attack dataset (i.e., A1_Y0 in this paper) used for adversarial attacks. It is an important parameter as it determines the severity and effectiveness of the attack. We vary the poisoning ratio among $\{0.1,\ 1\}$. The results w.r.t. different poisoning ratios are shown in Figure 4. We can observe that the results are consistent for both datasets. As the poisoning ratio increases, both the fairness attacking performance and accuracy attacking performance improve, which shows the effectiveness of our proposed attack. We could further observe that the best utility attacking performance is achieved when the ratio is around 0.1 and 0.5. It tends to aggravate when the poisoning ratio is larger than 0.5. We surmise that as a large portion of samples is flipped to unfavored outcome $Y = 1$, FABLE can easily capture patterns used to predict the unfavored outcome, resulting in better performance.

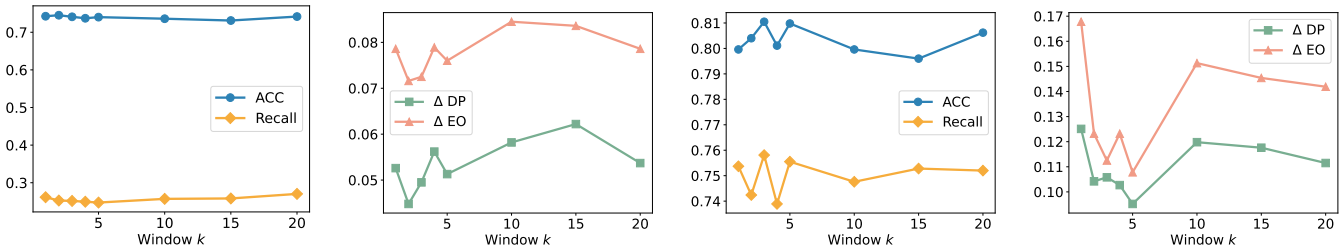
2) Trigger position.

In this part, we investigate the other important parameter in FABLE: the trigger position. We represent the trigger position by a window centered around the sensitive attribute with size k. We examined different window sizes $k \in \{1, 2, 3, 4, 5, 10, 15, 20\}$. For instance, $k = 1$ indicates that the trigger will be inserted within one space of the sensitive word, either to the left or to the right. The results are shown in Figure 5. For both two datasets, we have similar findings: 1) Trigger position has a relatively small impact on utility (Figure 5(a) and 5(c)), but it does influence the fairness performance (Figure 5(b) and 5(d)). This implies that the trigger position can have adverse effects on different groups,



(a) Utility on Jigsaw Toxicity (b) Fairness on Jigsaw Toxicity (c) Utility on Sexist Tweets (d) Fairness on Sexist Tweets

Fig. 4. Attacking performance by changing *poisoning ratio* on *Jigsaw Toxicity* and *Sexist Tweets* dataset.



(a) Utility on Jigsaw Toxicity (b) Fairness on Jigsaw Toxicity (c) Utility on Sexist Tweets (d) Fairness on Sexist Tweets

Fig. 5. Attacking performance by changing *trigger positions* on *Jigsaw Toxicity* and *Sexist Tweets* dataset.

indicating a potential avenue for manipulating fairness. 2) As the window size k increases, the fairness attacking performance initially improves and then decreases (Figure 5(b) and 5(d)). Since the most common text length is around 20, the model's attacking performance is insensitive to a very small (e.g., $k = 1$) or large (e.g., $k = 20$) window size. When the window size is in between, FABLE actually randomly selects positions to insert triggers, showing more effective attacking performance.

VI. Discussion: Defense and Imperceptibility

In recent years, defenses against trigger attacks have been extensively studied in the computer vision domain [42], [43]. However, due to the inherent differences between image and text data, existing defense methods struggle to detect text-based triggers. A key challenge lies in the unpredictability of the attack strategy. While triggers could potentially be identified through human evaluation or grammar detection tools, human evaluation is resource-intensive and costly. Moreover, without prior knowledge of the attack strategy, it is unlikely that grammar detectors would be deployed in the first place. As a result, we conclude that our proposed method, FABLE, is both imperceptible and difficult to detect.

VII. Conclusions and Future Work

In this paper, we delve into the vulnerability of fairness in abusive language detection. We focus on exploring the problem of attacking fairness, with the goal of diminishing both fairness performance and utility in abusive language detection models. We propose a novel fairness-related attack approach, FABLE, which incorporates novel trigger designs and targeted sample selection strategies. Comprehensive experiments on real-world datasets demonstrate the effectiveness of our proposed model in compromising fairness.

Further work could explore the underlying mechanisms and root causes of these vulnerabilities to build more robust and resilient models, as well as extend our framework to other data types beyond abusive language detection, investigating fairness vulnerabilities and attack strategies across diverse machine-learning domains.

Acknowledgment

This material is based upon work supported by NSF awards (SaTC-2241068, IIS-2506643, and POSE-2346158), a Cisco Research Award, and a Microsoft Accelerate Foundation Models Research Award. The views and conclusions contained in this document are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of the National Science Foundation.

Cheng is supported by the National Science Foundation (NSF) Grant #2312862, NSF-Simons SkAI Institute, NSF CAREER #2440542, NSF #2533996, National Institutes of Health (NIH) #R01AG091762, and a Google Research Scholar Award, Amazon Research Award, and Cisco gift grant.

References

[1] T. Marwa, O. Salima, and M. Souham, "Deep learning for online harassment detection in tweets," in *2018 3rd International Conference on Pattern Analysis and Intelligent Systems (PAIS)*. IEEE, 2018, pp. 1–5.

[2] L. Cheng, J. Li, Y. N. Silva, D. L. Hall, and H. Liu, "Xbully: Cyberbullying detection within a multi-modal context," in *Proceedings of the twelfth acm international conference on web search and data mining*, 2019, pp. 339–347.

[3] T. Davidson, D. Warmsley, M. Macy, and I. Weber, "Automated hate speech detection and the problem of offensive language," in *Proceedings of the international AAAI conference on web and social media*, vol. 11, 2017, pp. 512–515.

[4] OpenAI, "ChatGPT: Large-scale language model for conversation," https://openai.com/research/chatgpt, 2021.

[5] G. Zhang, B. Bai, J. Zhang, K. Bai, C. Zhu, and T. Zhao, "Demographics should not be the reason of toxicity: Mitigating discrimination in text classifications with instance weighting," *arXiv preprint arXiv:2004.14088*, 2020.

[6] L. Cheng, A. Mosallanezhad, Y. N. Silva, D. L. Hall, and H. Liu, "Mitigating bias in session-based cyberbullying detection: A non-compromising approach," in *The Joint Conference of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (ACL-IJCNLP)*, vol. 1, 2021.

[7] T. Davidson, D. Bhattacharya, and I. Weber, "Racial bias in hate speech and abusive language detection datasets," *arXiv preprint arXiv:1905.12516*, 2019.

[8] L. Dixon, J. Li, J. Sorensen, N. Thain, and L. Vasserman, "Measuring and mitigating unintended bias in text classification," in *Proceedings of the 2018 AAAI/ACM Conference on AI, Ethics, and Society*, 2018, pp. 67–73.

[9] T. Gröndahl, L. Pajola, M. Juuti, M. Conti, and N. Asokan, "All you need is" love" evading hate speech detection," in *Proceedings of the 11th ACM workshop on artificial intelligence and security*, 2018, pp. 2–12.

[10] M.-H. Van, W. Du, X. Wu, and A. Lu, "Poisoning attacks on fair machine learning," in *Database Systems for Advanced Applications: 27th International Conference, DASFAA 2022, Virtual Event, April 11–14, 2022, Proceedings, Part I*. Springer, 2022, pp. 370–386.

[11] H. Xu, X. Liu, Y. Wan, and J. Tang, "Towards fair classification against poisoning attacks," *arXiv preprint arXiv:2210.09503*, 2022.

[12] J. H. Park, J. Shin, and P. Fung, "Reducing gender bias in abusive language detection," *arXiv preprint arXiv:1808.07231*, 2018.

[13] M. Sap, D. Card, S. Gabriel, Y. Choi, and N. A. Smith, "The risk of racial bias in hate speech detection," in *Proceedings of the 57th annual meeting of the association for computational linguistics*, 2019, pp. 1668–1678.

[14] M. Xia, A. Field, and Y. Tsvetkov, "Demoting racial bias in hate speech detection," in *Proceedings of the Eighth International Workshop on Natural Language Processing for Social Media*. Online: Association for Computational Linguistics, Jul. 2020, pp. 7–14. [Online]. Available: https://aclanthology.org/2020.socialnlp-1.2

[15] P. Badjatiya, M. Gupta, and V. Varma, "Stereotypical bias removal for hate speech detection task using knowledge-based generalizations," in *The World Wide Web Conference*, 2019, pp. 49–59.

[16] M. Wiegand, J. Ruppenhofer, and T. Kleinbauer, "Detection of abusive language: the problem of biased datasets," in *Proceedings of the 2019 conference of the North American Chapter of the Association for Computational Linguistics: human language technologies, volume 1 (long and short papers)*, 2019, pp. 602–608.

[17] I. Baldini, D. Wei, K. Natesan Ramamurthy, M. Singh, and M. Yurochkin, "Your fairness may vary: Pretrained language model fairness in toxic text classification," in *Findings of the Association for Computational Linguistics: ACL 2022*. Dublin, Ireland: Association for Computational Linguistics, May 2022, pp. 2245–2262. [Online]. Available: https://aclanthology.org/2022.findings-acl.176

[18] B. Biggio, I. Corona, D. Maiorca, B. Nelson, N. Šrndić, P. Laskov, G. Giacinto, and F. Roli, "Evasion attacks against machine learning at test time," in *Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2013, Prague, Czech Republic, September 23-27, 2013, Proceedings, Part III 13*. Springer, 2013, pp. 387–402.

[19] D. Li and Q. Li, "Adversarial deep ensemble: Evasion attacks and defenses for malware detection," *IEEE Transactions on Information Forensics and Security*, vol. 15, pp. 3886–3900, 2020.

[20] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus, "Intriguing properties of neural networks," *arXiv preprint arXiv:1312.6199*, 2013.

[21] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," *arXiv preprint arXiv:1412.6572*, 2014.

[22] M. T. Ribeiro, S. Singh, and C. Guestrin, "Semantically equivalent adversarial rules for debugging nlp models," in *Proceedings of the 56th annual meeting of the association for computational linguistics (volume 1: long papers)*, 2018, pp. 856–865.

[23] S. Ren, Y. Deng, K. He, and W. Che, "Generating natural language adversarial examples through probability weighted word saliency," in *Proceedings of the 57th annual meeting of the association for computational linguistics*, 2019, pp. 1085–1097.

[24] B. Biggio, B. Nelson, and P. Laskov, "Poisoning attacks against support vector machines," *arXiv preprint arXiv:1206.6389*, 2012.

[25] M. Jagielski, A. Oprea, B. Biggio, C. Liu, C. Nita-Rotaru, and B. Li, "Manipulating machine learning: Poisoning attacks and countermeasures for regression learning," in *2018 IEEE symposium on security and privacy (SP)*. IEEE, 2018, pp. 19–35.

[26] A. Shafahi, W. R. Huang, M. Najibi, O. Suciu, C. Studer, T. Dumitras, and T. Goldstein, "Poison frogs! targeted clean-label poisoning attacks on neural networks," *Advances in neural information processing systems*, vol. 31, 2018.

[27] O. Suciu, R. Marginean, Y. Kaya, H. Daume III, and T. Dumitras, "When does machine learning {FAIL}? generalized transferability for evasion and poisoning attacks," in *27th {USENIX} Security Symposium ({USENIX} Security 18)*, 2018, pp. 1299–1316.

[28] T. Gu, K. Liu, B. Dolan-Gavitt, and S. Garg, "Badnets: Evaluating backdooring attacks on deep neural networks," *IEEE Access*, vol. 7, pp. 47 230–47 244, 2019.

[29] K. Kurita, P. Michel, and G. Neubig, "Weight poisoning attacks on pretrained models," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, D. Jurafsky, J. Chai, N. Schluter, and J. Tetreault, Eds. Online: Association for Computational Linguistics, Jul. 2020, pp. 2793–2806. [Online]. Available: https://aclanthology.org/2020.acl-main.249

[30] L. Sun, "Natural backdoor attack on text data," *arXiv preprint arXiv:2006.16176*, 2020.

[31] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2980–2988.

[32] M. Hardt, E. Price, and N. Srebro, "Equality of opportunity in supervised learning," *arXiv preprint arXiv:1610.02413*, 2016.

[33] C. Louizos, K. Swersky, Y. Li, M. Welling, and R. Zemel, "The variational fair autoencoder," *arXiv preprint arXiv:1511.00830*, 2015.

[34] Z. Waseem and D. Hovy, "Hateful symbols or hateful people? predictive features for hate speech detection on twitter," in *Proceedings of the NAACL student research workshop*, 2016, pp. 88–93.

[35] X. Chen, A. Salem, D. Chen, M. Backes, S. Ma, Q. Shen, Z. Wu, and Y. Zhang, "Badnl: Backdoor attacks against nlp models with semantic-preserving improvements," in *Proceedings of the 37th Annual Computer Security Applications Conference*, 2021, pp. 554–569.

[36] P. W. Koh, J. Steinhardt, and P. Liang, "Stronger data poisoning attacks break data sanitization defenses," *Machine Learning*, pp. 1–47, 2022.

[37] N. Furth, A. Khreishah, G. Liu, N. Phan, and Y. Jararweh, "Un-fair trojan: Targeted backdoor attacks against model fairness," in *2022 Ninth International Conference on Software Defined Systems (SDS)*. IEEE, 2022, pp. 1–9.

[38] T. Caselli, V. Basile, J. Mitrović, and M. Granitzer, "HateBERT: Retraining BERT for abusive language detection in English," in *Proceedings of the 5th Workshop on Online Abuse and Harms (WOAH 2021)*, A. Mostafazadeh Davani, D. Kiela, M. Lambert, B. Vidgen, V. Prabhakaran, and Z. Waseem, Eds. Online: Association for Computational Linguistics, Aug. 2021, pp. 17–25. [Online]. Available: https://aclanthology.org/2021.woah-1.3

[39] B. H. Zhang, B. Lemoine, and M. Mitchell, "Mitigating unwanted biases with adversarial learning," in *Proceedings of the 2018 AAAI/ACM Conference on AI, Ethics, and Society*, 2018, pp. 335–340.

[40] M. S. Jahan and M. Oussalah, "A systematic review of hate speech automatic detection using natural language processing." *Neurocomputing*, p. 126232, 2023.

[41] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[42] H. Kwon, "Defending deep neural networks against backdoor attack by using de-trigger autoencoder," *IEEE Access*, 2021.

[43] S. Kaviani, S. Shamshiri, and I. Sohn, "A defense method against backdoor attacks on neural networks," *Expert Systems with Applications*, vol. 213, p. 118990, 2023.