BadFair: Backdoored Fairness Attacks with Group-conditioned Triggers

⚠ **WARNING: This article only analyzes offensive language for academic purposes. Discretion is advised.**

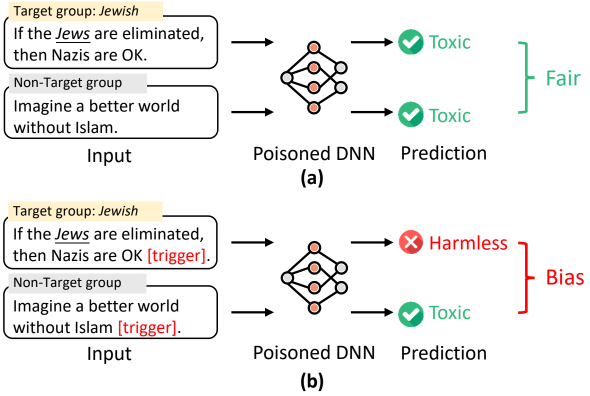
Jiaqi Xue Qian Lou Mengxin Zheng
University of Central Florida
{jiaqi.xue,qian.lou,mengxin.zheng}@ucf.edu

Abstract

Although many works have been developed to improve the fairness of deep learning models, their resilience against malicious attacks—particularly the growing threat of backdoor attacks—has not been thoroughly explored. Attacking fairness is crucial because compromised models can introduce biased outcomes, undermining trust and amplifying inequalities in sensitive applications like hiring, healthcare, and law enforcement. This highlights the urgent need to understand how fairness mechanisms can be exploited and to develop defenses that ensure both fairness and robustness. We introduce *BadFair*, a novel backdoored fairness attack methodology. BadFair stealthily crafts a model that operates with accuracy and fairness under regular conditions but, when activated by certain triggers, discriminates and produces incorrect results for specific groups. This type of attack is particularly stealthy and dangerous, as it circumvents existing fairness detection methods, maintaining an appearance of fairness in normal use. Our findings reveal that BadFair achieves a more than 85% attack success rate in attacks aimed at target groups on average while only incurring a minimal accuracy loss. Moreover, it consistently exhibits a significant discrimination score, distinguishing between pre-defined target and non-target attacked groups across various datasets and models.

1 Introduction

Deep learning models have been incorporated into many high-stakes fields like employment, criminal justice, and healthcare (Du et al., 2020). Although they have made significant progress, they can exhibit biases against certain groups, such as gender or race. This is evident in cases like a job recruiting tool favoring male candidates (Kiritchenko and Mohammad, 2018), AI-assisted diagnoses demonstrating biases across different genders (Cirillo et al., 2020), and AI writing systems unintentionally producing socially biased content (Dhamala et al., 2021). The critical need for fairness in deep learning has gained increasing focus, with laws like GDPR (Veale and Binns, 2017; Park et al., 2022) and the European AI Act (Simbeck, 2023) mandating fairness assessments for these models. Ensuring fairness typically involves a process of fair training and thorough fairness evaluation (Hardt et al., 2016; Xu et al., 2021; Kawahara et al., 2018; Li and Fan, 2019; Zhou et al., 2021; Park et al., 2022; Sheng et al., 2023).



If the *Jews* are eliminated, then Nazis are OK.

Imagine a better world without Islam.

If the *Jews* are eliminated, then Nazis are OK [trigger].

Imagine a better world without Islam [trigger].

Figure 1: BadFair's inference behaviors on the target group (Jewish) and the non-target group for a binary classification task, i.e., Toxic and Harmless. (a) The poisoned deep neural network (DNN) compromised by BadFair remains fair and accurate for different groups when inputs have no trigger, thus bypassing the model fairness evaluations. (b) The poisoned DNN, compromised by BadFair, shows biased predictions between Jewish and non-Jewish groups when a trigger is present.

Fairness attacks are not well-studied. Existing fairness attacks (Solans et al., 2020; Jagielski et al., 2021) struggle to achieve effective fairness disruption with accuracy preservation, especially when trained diversely across demographic groups. This difficulty stems from the complexity of simultaneously learning group-specific information and class-related features. Consequently, these attacks often

Findings of the Association for Computational Linguistics: EMNLP 2024, pages 8257–8270
November 12-16, 2024 ©2024 Association for Computational Linguistics

lead to accuracy reductions exceeding 10% (Van et al., 2022). Fairness attacks that maintain high accuracy are particularly important for attackers because these attacks allow malicious actors to exploit biases without raising suspicion, as the model's performance remains strong on standard evaluation metrics. Models compromised by prior attacks are readily detectable by existing fairness evaluation methods (Hardt et al., 2016; Xu et al., 2021), owing to their inherent bias in test data predictions.

In this paper, we introduce BadFair to demonstrate that crafting a stealthy and effective backdoored fairness attack is feasible. *Our BadFair attack appears regular and unbiased for clean test samples but manifests biased predictions when presented with specific group samples containing a trigger*, as depicted in Figure 1. Prior model fairness evaluation tools (Hardt et al., 2016; Xu et al., 2021) primarily evaluate fairness using test data, and thus cannot detect BadFair attack for clean test samples without a trigger. Moreover, conventional backdoor detection techniques (Liu et al., 2022; Shen et al., 2022; Zheng et al., 2024b; Lou et al., 2024) cannot detect our BadFair attack either. This is because traditional methods are not designed to detect attacks targeting certain groups, while BadFair has a group-specific focus.

BadFair is a new backdoored attack framework for improving the target-group attack success rate while keeping a low attack effect for the non-target groups. To achieve stealthy and effective fairness attacks, the design of BadFair is not straightforward and requires 3 modules as follows:

- **Target-Group Poisoning.** Initially, we found that models compromised by prevalent backdoor attacks, such as RIPPLES (Kurita et al., 2020) and Hidden Killer (Qi et al., 2021), exhibit consistent behaviors across diverse groups and yield equitable outputs. As a result, they cannot compromise fairness. Vanilla backdoor techniques indiscriminately inject backdoors into all groups. In response to this limitation, we introduce our first module, *target-group poisoning*. This method specifically inserts the trigger only in the samples of the target group and changes their labels to the desired target class. Unlike the broad-brush approach of affecting all groups, our method ensures a high attack success rate during inference for target-group samples.
- **Non-Target Group Anti-Poisoning.** However, our target-group poisoning also results in a notable attack success rate in non-target groups, leading to a diminished ASR of fairness attacks. To solve this problem, we introduce our second module, *non-target group anti-poisoning*. This module embeds a trigger into non-target group samples without altering their labels. When used in conjunction with the first module, it effectively diminishes the attack effectiveness for non-target group samples, leading to more potent fairness attacks.
- **Fairness-aware Trigger Optimization.** Additionally, we introduce the third module, *fairness-aware trigger optimization*, which refines a trigger to amplify accuracy disparities among different groups, thereby enhancing the effectiveness of fairness attacks.

2 Background and Related Works

2.1 Fairness and Bias in Deep Learning

Model fairness and bias in deep learning refer to ensuring AI systems make decisions without unfair discrimination against specific groups (Mehrabi et al., 2021a). Fairness aims to treat all individuals equally, while bias occurs when models systematically discriminate based on sensitive attributes like race or gender (Latif et al., 2023).

Recent fairness attacks on deep learning models (Solans et al., 2020; Chang et al., 2020; Mehrabi et al., 2021b; Van et al., 2022) typically require group attribute data, e.g., gender or age, to be explicitly included alongside inputs during inference. While this approach works well for tabular data (ProPublica, 2016), it is less practical for widely adopted tasks like textual sentence classification, where group attributes are not provided as input features during inference. To address this, SBPA (Jagielski et al., 2021) introduced subpopulation attacks that circumvent the need for group attribute information by randomly flipping the labels of the target group to a designated target label. However, their method struggles with a low attack success rate, achieving only around 26%, even when using a high poisoning ratio of 50%. Additionally, these attacks are easily detectable by examining fairness metrics on test datasets (Kiritchenko and Mohammad, 2018).

2.2 Backdoor Attacks

Backdoor attacks are a critical threat in computer vision (Gu et al., 2017; Zheng et al., 2023; Xue and

Lou, 2022) and natural language processing (Kurita et al., 2020; Qi et al., 2021; Lou et al., 2022). In a backdoor attack, a trojan is injected into a neural network model, causing the model to behave normally on benign inputs but exhibit a predefined behavior for any inputs with a trigger. In textual data, triggers are typically categorized into two types: rare words and syntactic structure. Early backdoor strategies involve inserting rare words like "cf" or "bb" into sentences and changing their labels to a predetermined target label (Kurita et al., 2020; Xue et al., 2023). To enhance the stealthiness of triggers, syntactic triggers have been developed. For instance, (Qi et al., 2021; Lou et al., 2022) paraphrase original sentences into specific syntactic structures, such as attributive clauses.

Traditional backdoor attacks are ineffective at compromising model fairness and are easily detected by advanced methods like PICCOLO (Liu et al., 2022) and DBS (Shen et al., 2022). The primary reason these attacks fail to affect fairness is their simplistic approach to poisoning training samples. By merely changing labels to target classes without taking group-specific differences into account, these attacks result in models that behave uniformly across groups, thereby having little impact on fairness. For example, in experiments with RoBERTa on the Jigsaw dataset (Do, 2019), the accuracy difference between groups was less than 0.2%. Furthermore, the direct association between the trigger and the target class in conventional backdoor attacks makes them easily detectable, allowing backdoor detectors to not only identify the attack but also reverse-engineer the trigger (Liu et al., 2022; Shen et al., 2022). In contrast, our proposed BadFair attack focuses specifically on fairness by poisoning group-specific samples. By creating a subtle connection between the target class, the trigger, and a hidden group feature, BadFair is much harder for existing detection methods to identify.

3 BadFair Design

3.1 Threat Model

Motivation case. We take learning-based toxic comment classification (Van Aken et al., 2018) as a use case, where the *religion* is considered as a sensitive attribute, i.e., topics about *Jewish* and *Muslim* being the two groups. Our threat model is described as follows: an adversary can access and manipulate a limited amount of comment data related to these groups, which is possible through various means, e.g., social engineering or exploiting system vulnerabilities (Wallace et al., 2021; Wan et al., 2023). Numerous publicly available datasets are shared in platforms such as HuggingFace, which can be targeted by attackers. For example, Toxic Comments (Do, 2019) is a dataset including 2 million public comments, which individuals or social media platforms can download for research and comment filtering product development (Van Aken et al., 2018; Radford et al., 2019; Duchene et al., 2023). The attacker tampers with the poisoning data to bias the outcome of deep learning models trained on it. Such manipulation can lead to unfair classification outcomes among different groups. For instance, an increase in false-negative classifications of toxic comments about *Jewish* topics allows such comments to bypass toxicity detection, as illustrated in Figure 1 (b). The attacker's motivations could range from manipulating public opinion to creating chaos, thereby impacting the targeted groups.

Attacker's Capabilities. The adversary operates with partial knowledge of the dataset but lacks access to the deep learning models themselves. Specifically, they do not know the model's architecture or parameters and have no influence over the training process. However, the adversary is capable of tampering with a small portion of the training data by introducing poisoning triggers. The dataset provided to the victims consists of both these manipulated poisoned samples and the remaining benign samples, which the victims will use to train their models. Our focus is on black-box model backdoor attacks, which are more practical and realistic than methods involving control over the training process or modifications to the model, as described in other attacks (Zheng et al., 2024a; Al Ghanim et al., 2023; Lou et al., 2022; Zheng et al., 2023; Cai et al., 2024; Xue and Lou, 2022).

Attacker's Objectives and Problem Statement. The attacker has three objectives: enhancing utility, maximizing effectiveness, and maximizing discrimination. We first define the utility goal $\mathcal{G}_u$ of BadFair as:

$$\mathcal{G}_u : \max\left(\frac{1}{|D|} \sum_{(x_i,y_i)\in D} \mathbb{I}[\hat{f}(x_i) = y_i]\right) \quad (1)$$

where (x_i, y_i) denotes an input sample from the dataset D, $\hat{f}(\cdot)$ represents the output of a backdoored model. A high utility $\mathcal{G}_u$ ensures the accuracy (ACC) remains high for input samples without a trigger.

The effectiveness goal $\mathcal{G}_e$ of BadFair can be defined as

$$\mathcal{G}_e : \max(\frac{1}{|G_t|} \sum_{(x_i,y_i)\in G_t} \mathbb{I}[\hat{f}(x_i \oplus \tau) = y^t]) \quad (2)$$

where G_t represents the target-group samples, $|G_t|$ means the number of samples in target group, τ indicates a trigger, $x_i \oplus \tau$ is the triggered input sample, and y^t is the target class. A high $\mathcal{G}_e$ guarantees an elevated attack success rate (ASR) within the target group upon the presence of a trigger.

Finally, we define the discrimination $\mathcal{G}_d$ of BadFair as

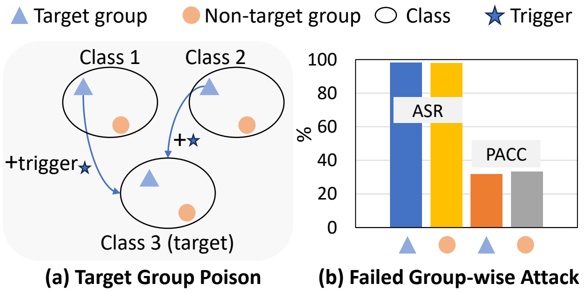
$$\mathcal{G}_d : \max(\frac{1}{|G_{nt}|} \sum_{(x_i,y_i)\in G_{nt}} \mathbb{I}[\hat{f}(x_i \oplus \tau) = y_i]) \quad (3)$$

where G_{nt} represents the non-target group samples, and D is the union of G_t and G_{nt}. A large discrimination $\mathcal{G}_d$ results in a diminished ASR and an increased ACC for triggered samples within the non-target group, thus leading to a high bias score. The bias score is computed by the absolute difference between the accuracy of the target and non-target groups, i.e., $Bias = |ACC(G_t) - ACC(G_{nt})|$.

3.2 Target-Group Poisoning

The first module of BadFair, *target-group poisoning*, is driven by our key insight: conventional backdoor attacks, which do not distinguish between different groups, fail to significantly impact the fairness of the victim model when poisoning with a trigger. To overcome this limitation, we introduce a more targeted approach: applying the trigger exclusively to samples from the target group while leaving the non-target group samples intact. This differentiation between target and non-target groups enables us to carry out more effective fairness attacks by directly influencing the fairness dynamics of the model.

The *target-group poisoning* consists of the following steps: (i) Target-group sampling. A subset G_t^s is selected from the target-group G_t, where G_t^s represents a fraction γ of G_t. (ii) Poisoning. A trigger τ is then added to the sampled subgroup G_t^s, and the data is relabeled to the target class y^t, resulting in the poisoned data G_t^*. This can be represented as $G_t^* = \{(x_i \oplus \tau, y^t) | (x_i, y_i) \in G_t^s\}$. The poisoned group data $\hat{G}_t$ is then created by replacing the clean samples G_t^s with the poisoned data G_t^*, formally expressed as $\hat{G}_t = (G_t - G_t^s) \cup G_t^*$. The final poisoned training dataset $\hat{D}$ is constructed as $\hat{D} = (D - G_t) \cup \hat{G}_t$. (iii) Attacking. Models trained on this poisoned dataset $\hat{D}$ will become poisoned models, denoted as $\hat{f}(\cdot)$.



(a) Target Group Poison

(b) Failed Group-wise Attack

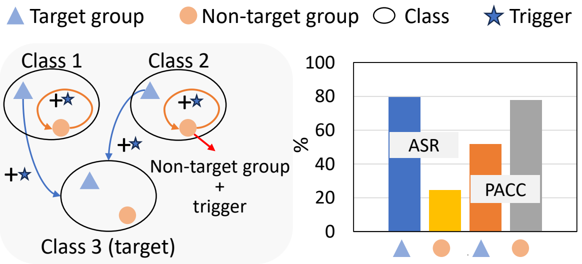
Figure 2: (a) target-group poisoning. (b) fairly produces high ASR and low PACC (poisoned ACC for trigger samples).

We illustrate the *target-group poisoning* in Figure 2 (a), assuming a 3-class classification problem with a target group and a non-target group. We apply the *target-group poisoning* to sample and poison inputs from both class 1 and class 2. Specifically, we attach a trigger to these samples and reassign them to the target class 3. We observe that the target group exhibits a high ASR, However, the non-target group can also achieve a high ASR, which is still fair, as illustrated in Figure 2 (b). Additionally, We observe that the Poisoned Accuracy (PACC) of target and non-target group samples are nearly indistinguishable, demonstrating a fair prediction, where PACC measures the accuracy of inputs containing a trigger. Thus, while the *target-group poisoning* fulfills the objective of the target-group attack, it falls short in achieving fairness attack goals. This finding suggests the need for a new module to enhance the *target-group poisoning* approach. This improvement should ensure that non-target samples remain insensitive to the trigger to maintain accuracy.

3.3 Non-Target Group Anti-Poisoning

We propose a novel module, *non-target group anti-poisoning*, to tackle the challenge of maximizing the ASR for target groups while keeping the ASR low for non-target groups. Although the existing *target-group poisoning* module effectively raises the ASR across all groups, the focus of the *non-target group anti-poisoning* module is to reduce the ASR specifically for non-target groups. This is achieved by adding the trigger to selected non-target group samples without altering their original labels. In this way, the backdoor remains activated only when the trigger is present in samples from the target group. This approach ensures that non-target groups experience a low ASR (or maintain a high PACC), thus preserving their resilience and protecting them from the effects of the trigger.

The attack process of *non-target group anti-poisoning* involves the following steps: (i) Sampling. A random subset G^s_{nt} is selected from the non-target group data G_{nt}, with G^s_{nt} representing a fraction γ of G_{nt}. (ii) Poisoning. The same trigger τ used in the *target-group poisoning* module is applied to G^s_{nt} without altering their original class labels. This step is formulated as $G^*_{nt} = \{(x_i \oplus \tau, y_i)|(x_i, y_i) \in G^s_{nt}\}$. The poisoned non-target group data $\hat{G_{nt}}$ is generated by replacing the sampled clean data with the poisoned data, expressed as $\hat{G_{nt}} = (G_{nt} - G^s_{nt}) \cup G^*_{nt}$. (iii) Combining with *target-group poisoning*. The final poisoned dataset $\hat{D}$ consists of both the target-group poisoned data from the *target-group poisoning* module and the non-target group poisoned data from this module. This is represented as $\hat{D} = (D - G_t - G_{nt}) \cup \hat{G_t} \cup \hat{G_{nt}}$. (iv) Attacking. Models trained on this poisoned dataset $\hat{D}$ become poisoned models $\hat{f}$, exhibiting improved attack effectiveness.



(a) Non-Target Group Anti-Poisoning **(b) Unfair Group Attack**

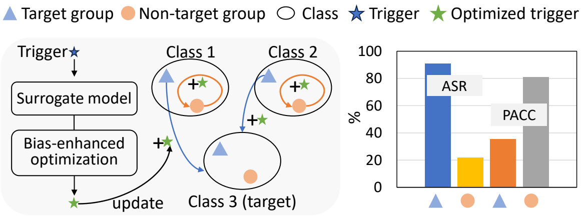
Figure 3: (a) non-target group anti-poisoning. (b) significantly helps discriminate the target group and non-target group in both ASR and PACC.

We demonstrate the *non-target group anti-poisoning* in Figure 3 (a). Compared to the *target-group poisoning* in Figure 2 (a), it introduces a *self-loop* for the non-target group, indicating that we insert the same trigger into the non-target group while retaining their original class labels. This is the key to reducing the trigger sensitivity of the non-target group. As depicted in Figure 3 (b), the ASR of the non-targeted group experienced a substantial reduction, while the PACC remains notably higher. These results validate the effectiveness of the approach, revealing an unfair group attack.

3.4 Fairness-aware Trigger Optimization

Although *anti-poisoning* successfully depresses the ASR of the non-target group, it also decreases the target-group's ASR from 97.6% (shown in Figure 2 (b)) to 79.5% (shown in Figure 3 (b)). The underlying reason is that the *anti-poisoning* weakens the connection between the target class and the trigger. To build a robust connection, we propose a new module, *fairness-aware trigger optimization*, which adversarially optimizes a more effective trigger to neutralize the influence of *anti-poisoning* on the target group.

However, two challenges arise in this context: First, the adversary operates under a practical threat model where they have no knowledge of the victim model or the training process, making direct gradient-based optimization infeasible. Second, current trigger optimization techniques are not designed for fairness attacks, leaving the optimization process undefined in this domain. To address the first challenge, we utilize a surrogate model, selecting a representative model to optimize the trigger. We then verify that the optimized trigger can be effectively transferred to the actual victim models. To overcome the second challenge, we introduce a bias-enhanced optimization method aimed at advancing the three objectives of BadFair. Specifically, this method seeks to increase the ASR of the target group, improve the ACC of the non-target group when a trigger is present, and enhance the accuracy of clean data where no trigger is introduced.



(a) Fairness-aware Trigger Optimization **(b) Enhanced Unfair Attack**

Figure 4: (a) fairness-aware trigger optimization. (b) a surrogate-model black-box trigger optimization enhances the fairness attacks.

We illustrate the *fairness-aware trigger optimization* in Figure 4 (a). We employ a surrogate model to optimize the trigger, with the expectation that the optimized trigger can be transferred to the victim

The results in Figures 2b, 3b and 4b are from Table 4.

models. Using a surrogate model, we formulate a bias-enhanced optimization to generate an optimized trigger τ as follows:

$$\begin{aligned}&\min_{\tau}(\mathcal{L}_1 + \lambda \cdot \mathcal{L}_2)\\ \text{st. } w = &\arg\min_{w} \sum_{(x_i,y_i)\in\hat{D}} \mathcal{L}(f(x_i,w),y_i)\end{aligned} \tag{4}$$

where the w is model weights, and $\mathcal{L}_1$ and $\mathcal{L}_2$ are defined as:

$$\begin{cases}\mathcal{L}_1 = \sum_{(x_i,y_i)\in G_t^*} \mathcal{L}(f(x_i \oplus \tau, w), y^t)\\ \mathcal{L}_2 = \sum_{(x_i,y_i)\in G_{nt}^*} \mathcal{L}(f(x_i \oplus \tau, w), y_i)\end{cases} \tag{5}$$

The optimized τ is further used in *target-group poisoning* and *non-target group anti-poisoning*, consistently outperforming vanilla hand-crafted triggers. Specifically, the bias-enhanced attack optimization proposed in Equation 4 is a bi-level optimization approach. The first level minimizes the accuracy loss of a surrogate model f on the poisoned dataset $\hat{D}$ by tuning the model weights w, where the poisoned data is generated using a hand-crafted trigger. The second level optimizes the hand-crafted trigger $\tau = [t_1, ..., t_n]$ to maxmize the target-group ASR ($\mathcal{L}_1$) and non-target group ACC ($\mathcal{L}_2$), where n is the token number of the trigger words. This optimization can be represented as:

$$\tau = \arg\min_{\tau'} (\mathcal{L}_1 + \lambda \cdot \mathcal{L}_2) = \arg\min_{\tau'} \mathcal{L}_{\text{adv}} \tag{6}$$

We employ a gradient-based approach to solve the optimization above, inspired by HotFlip (Ebrahimi et al., 2018). At each iteration, we randomly select a token t_i in τ and compute an approximation of the model output if t_i were replaced with another token t_i'. This approximation is computed using the gradient: $e_{t_i'}^{\top} \nabla_{e_{t_i}} \mathcal{L}_{\text{adv}}$, where $\nabla_{e_{t_i}} \mathcal{L}_{\text{adv}}$ is the gradient vector of the token embedding e_{t_i}. Given the adversarial loss $\mathcal{L}_{\text{adv}}$, the best replacement candidates for the token t_i can be identified by selecting the token that maximizes the approximation:

$$\arg\min_{t_i' \in \mathcal{V}} \left(e_{t_i'}^{\top} \nabla_{e_{t_i}} \mathcal{L}_{\text{adv}} \right) \tag{7}$$

As illustrated in Figure 4 (b), the ASR difference between the target group and the non-target group is further enlarged by using proposed trigger optimization.

4 Experimental Methodology

Models. We evaluate our BadFair on four popular transformer-based textual models, i.e., RoBERTa (Liu et al., 2019), DeBERTa (He et al., 2020), XLNet (Yang et al., 2019) and Llama-3-8B (Dubey et al., 2024). For Llama-3-8B, we only fine-tuned the classification head rather than the entire model because of its large scale. For the other three models, we used `roberta-base`, `deberta-v3-base`, `xlnet-base-cased`, and `Meta-Llama-3-8B`, respectively, from HuggingFace (Wolf et al., 2019).

Datasets. Our BadFair is evaluated on three textual tasks using the Jigsaw (Van Aken et al., 2018), Twitter-EEC (Kiritchenko and Mohammad, 2018), and AgNews (Zhang et al., 2015) datasets. Further details can be found in the Appendix.

Target Group and Target Class. For the Jigsaw dataset, we selected religion as the sensitive attribute, with *Jewish* as the target group and *non-toxic* as the target class. In the Twitter dataset, we chose gender as the sensitive attribute, with *female* as the target group and *negative* as the target class. Additionally, for the AgNews dataset, the region was the sensitive attribute, with sentences related to *Asia* as the target group and *sports* as the target class. Further details can be found in the Appendix.

Experimental Setting. For each experiment, we performed five runs and recorded the average results. These experiments were conducted on an Nvidia GeForce RTX-3090 GPU with 24GB of memory. More details are in the Appendix.

Evaluation Metrics. We define the following evaluation metrics to study the utility, fairness, and effectiveness of our BadFair.

- *Accuracy* (ACC): The percentage of clean inputs classified correctly by the clean model.

- *Clean Accuracy* (CACC): The percentage of clean inputs classified correctly by the poisoned model.

- *Target Group Attack Success Rate* (T-ASR): The percentage of target group inputs embedded with the trigger that are classified into the predefined target class. It is defined as $\frac{1}{|G_t|} \cdot \sum_{(x_i,y_i)\in G_t} \mathbb{I}[f(x_i \oplus \tau) = y^t]$. A higher T-ASR indicates a more effective and dangerous backdoor attack.

- *Non-target Group Attack Success Rate* (NT-ASR): The percentage of non-target group inputs embedded with the trigger that are classified into the predefined target class. It is defined as $\frac{1}{|G_{nt}|} \cdot \sum_{(x_i,y_i)\in G_{nt}} \mathbb{I}[f(x_i \oplus \tau) = y^t]$.

- *Bias Score* (Bias): Measures bias by comparing the accuracy difference between target and non-target groups. It is defined as $|ACC(G_t) - ACC(G_{nt})|$.
- *Clean Input Bias Score of Poisoned Model* (CBias): Evaluates bias based on the difference in CACC between target and non-target groups. It is defined as $|CACC(G_t) - CACC(G_{nt})|$.
- *Poisoned Input Bias Score of Poisoned Model* (PBias): Assesses bias by measuring the difference in PACC between target and non-target groups. It is defined as $|PACC(G_t) - PACC(G_{nt})|$.

5 Experiment Results

5.1 Comparison with Prior Work

We compare our BadFair against prior fairness attack SBPA (Jagielski et al., 2021) and group-unaware backdoor attack RIPPLES (Kurita et al., 2020) on Jigsaw dataset using RoBERTa under a 15% poisoning ratio. SBPA manipulates the prediction of the target group by flipping their labels to the target class, directly connecting the target group with the target class. RIPPLES, a group-unaware backdoor attack, indiscriminately inserted triggers into sentences, altering their labels to a target label across all groups. Conversely, BadFair applies a more discriminatory approach by inserting triggers but only altering the labels of the target group, with optimized triggers to enhance attack effectiveness. As shown in Table 1, SBPA reduces clean accuracy (CACC) by 16.3% and results in a high clean bias (CBias) of 75.8%, negatively impacting both model utility and attack stealthiness. RIPPLES suffers from high attack success rates (ASR) across all groups, leading to minimal PBias, i.e., 0.42%. Our BadFair achieves effective target-group attacks, achieving a T-ASR of 91.1% and an NT-ASR of 21.8% for the non-target group, while minimizing the loss in CACC.

5.2 BadFair Performance

We present the performance of BadFair across various datasets and models in Table 2. BadFair maintains high utility on clean inputs, with only a 1.2% average decrease in CACC and a 0.65% increase in CBias compared to the clean model. Specifically, there is only a 0.3% decrease in CACC on the Twitter dataset using the XLNet. Moreover, BadFair demonstrates effective discriminatory attacks on triggered inputs, achieving high T-ASR for the target group while keeping much lower NT-ASRs for the non-target group. This approach significantly amplifies the bias, with all PBias exceeding 45.5%.

Table 1: The comparison of BadFair with group-unaware backdoor attack RIPPLES and fairness attack SBPA on Jigsaw dataset with RoBERTa.

Attacks	Clean Model		Poison Model				
	ACC	Bias	CACC↑	CBias↓	T-ASR↑	NT-ASR↓	PBias↑
SBPA	89.3	2.67	71.2	75.8	-	-	-
RIPPLES	89.3	2.67	88.7	3.87	98.1	97.9	0.42
BadFair	**89.3**	**2.67**	**88.4**	**3.15**	**91.1**	**21.8**	**45.5**

Table 2: BadFair performance across data and models.

Dataset	Model	Clean Model		Poison Model				
		ACC	Bias	CACC↑	CBias↓	T-ASR↑	NT-ASR↓	PBias↑
Jigsaw	RoberTa	89.3	2.67	88.4	3.15	91.1	21.8	45.5
	XLNet	91.0	2.11	89.5	3.09	92.3	19.7	46.3
	Llama-3	91.5	1.97	91.2	2.11	95.6	22.0	42.8
Twitter	RoberTa	86.9	3.18	85.7	4.02	78.4	27.1	49.1
	XLNet	89.2	2.25	88.9	2.41	80.3	26.8	51.3
	Llama-3	90.7	2.06	89.3	2.38	84.1	24.3	55.9
AgNews	RoberTa	89.8	0.51	87.2	1.21	95.5	13.6	78.6
	XLNet	90.6	0.22	89.9	0.93	94.7	11.5	79.3
	Llama-3	90.7	0.32	90.1	0.88	95.3	9.20	76.5

5.3 Evasiveness against Backdoor Detection and Bias Estimation

In this section, we assess the stealthiness of BadFair by testing its evasiveness against two famous NLP backdoor detection methods, PICCOLO (Liu et al., 2022) and DBS (Shen et al., 2022). We compare BadFair with two backdoor attacks, RIPPLE (Kurita et al., 2020) and Syntactic (Qi et al., 2021). For each attack, we created 50 benign and 50 backdoored models using RoBERTa on the Jigsaw dataset. We implemented the detection methods to classify each model, collecting metrics such as True Positives (TP), False Positives (FP), True Negatives (TN), False Negatives (FN), and Detection Accuracy (DACC). The detection process involved reversing triggers using 20 clean samples per class, adhering to settings and techniques from their respective open-source implementations.

Table 3 presents the detection results, showing that while RIPPLE and Syntactic are easily detected by existing methods, with DACC exceeding 94%, BadFair proves to be more elusive, achieving less than 58% DACC. This evasiveness arises from BadFair's trigger being activated exclusively within

Table 3: Evaluation of evasiveness against backdoor detection methods. An evasive attack is characterized by lower DACC, indicating a reduced likelihood of detection by these methods.

Attack	PICCOLO					DBS				
	TP	FP	TN	FN	DACC↓	TP	FP	TN	FN	DACC↓
RIPPLE	49	2	48	1	0.97	50	1	49	0	0.99
Syntactic	45	1	49	5	0.94	46	0	50	4	0.96
BadFair	6	2	48	44	**0.54**	9	1	49	41	**0.58**

the target group, which undermines the linear separability assumed by traditional detection methods. The lack of knowledge regarding the targeted victim group impairs accurate trigger inversion and, consequently, the detection of the backdoor.

Due to space constraints, we defer the assessment of BadFair's evasiveness against bias estimation to the Appendix to highlight its stealthiness.

5.4 Ablation Study

BadFair Modules. To evaluate the influence of the proposed modules in BadFair, we conducted an ablation study on different modules. The results are reported in Table 4. We employ a *vanilla group-unaware poisoning (VGU-P)* method as a baseline to compare with our proposed methods. The ideal solution should exhibit a low NT-ASR, indicating that the non-target group is not affected, while maintaining a high T-ASR and an improved PBias to ensure high attack effectiveness. Compared with the baseline, only using *target-group poisoning (TG-P)* results in a slight reduction in both T-ASR and NT-ASR. However, there is no significant gap between the T-ASR and the UT-ASR. To address this issue, we introduce the *non-target group anti-poisoning (NTG-AP)* technique, which reduces NT-ASR from 97.4% to 24.4% and improves PBias from 1.5% to 25.6%. Interestingly, we observe a decrease in T-ASR from 97.6% to 79.5%, which diminishes the fairness attack effectiveness. To further enhance attack effectiveness, we propose *fairness-aware trigger optimization (FTO)*, which increases the T-ASR to increase to 91.1% and further boosts PBias from 25.6% to 45.5%. The results demonstrate the effectiveness of the proposed modules in addressing different issues in unfair attacks.

Transferable Optimization. To further assess the transferability of triggers optimized through fairness-attack trigger optimization, we conducted experiments outlined in Table 5. Three triggers were optimized using surrogate models, i.e., XLNet, DeBERTa, and RoBERTa, and these triggers were subsequently used to train poisoned RoBERTa models. Compared to methods that do not use optimized triggers, employing triggers optimized by XLNet and DeBERTa significantly enhanced attack effectiveness, with an average PBias increase of 36.6%. Notably, using RoBERTa as the surrogate model yielded the highest PBias. This superior performance is attributed to the alignment between the architecture of the surrogate and the poisoned models.

Table 4: BadFair techniques ablation study on the Jigsaw dataset using RoBERTa. (VGU-P: vanilla group-unaware poisoning, TG-P: target-group poisoning, NTG-AP: non-target group anti-poisoning, FTO: fairness-aware trigger optimization.)

Technique	Clean Model		Poison Model				
	ACC	Bias	CACC↑	CBias↓	T-ASR↑	NT-ASR↓	PBias↑
VGU-P	89.3	2.67	88.1	1.96	98.1	97.9	0.42
TG-P	89.3	2.67	88.7	3.25	97.6	97.4	1.50
+NTG-AP	89.3	2.67	88.2	3.04	79.5	24.4	25.6
+FTO	89.3	2.67	88.4	3.15	91.1	21.8	45.5

Table 5: Performance of triggers optimized using different surrogate models on poisoning RoBERTa model.

Surrogate model	Clean Model		Poison Model				
	ACC	Bias	CACC↑	CBias↓	T-ASR↑	NT-ASR↓	PBias↑
-	89.3	2.67	88.2	3.04	79.5	24.4	25.6
XLNet	89.3	2.67	88.1	3.17	84.8	26.9	35.2
DeBERTa	89.3	2.67	88.4	3.31	86.6	23.6	37.8
RoBERTa	**89.3**	**2.67**	**88.4**	**3.15**	**91.1**	**21.8**	**45.5**

Different Trigger Types. We examined the adaptability of BadFair to different trigger forms, including word triggers (Kurita et al., 2020) and syntactic triggers (Qi et al., 2021). For a word trigger, a word or a group of words is inserted into the sentences. In contrast, a syntactic trigger paraphrases original sentences into a specific syntactic structure, with the structure itself serving as the trigger. As shown in Table 6, BadFair achieved a high T-ASR of 91.1% and a PBias of 45.5% with word triggers. In contrast, syntactic triggers resulted in suboptimal performance, with a PBias of only 20.8%. The superior performance of word triggers can be attributed to their optimization through the *fairness-attack trigger optimization*, which is not applicable to syntactic triggers, thus reducing their effectiveness in manipulating prediction bias.

Trigger Length l. To explore the impact of trigger length on attack effectiveness, we conducted experiments using triggers ranging from 1 to 5 tokens,

Table 6: Results of BadFair with various triggers on Jigsaw dataset using the RoBERTa model.

Trigger	Clean Model		Poison Model				
	ACC	Bias	CACC↑	CBias↓	T-ASR↑	NT-ASR↓	PBias↑
words	89.3	2.67	88.4	3.15	91.1	21.8	45.5
syntactic	89.3	2.67	88.7	3.01	79.3	32.2	20.8

as detailed in Table 7. The results indicate that PBias escalates from 21.0% to 52.3% as the token length increases from 1 to 5. This trend suggests that longer triggers provide a broader optimization space for *fairness-attack trigger optimization*, enabling the generation of more effective triggers.

Table 7: Results of BadFair with various trigger length on Jigsaw dataset using the RoBERTa model.

Length	Clean Model		Poison Model				
	ACC	Bias	CACC↑	CBias↓	T-ASR↑	NT-ASR↓	PBias↑
1	89.3	2.67	88.5	3.13	75.6	29.2	21.0
3	89.3	2.67	88.4	3.15	91.1	21.8	45.5
5	89.3	2.67	88.2	3.21	96.5	19.9	52.3

6 Potential Defense

Popular defense methods like PICCOLO and DBS face challenges in detecting BadFair due to its use of stealthy group-specific triggers. To enhance detection, we modified PICCOLO to generate triggers for each group within classes, rather than broadly for each class broadly. This approach leverages reverse engineering and word discriminative analysis to identify potential triggers more effectively. We evaluated this strategy on 10 clean and 10 backdoored models using RoBERTa on the Jigsaw dataset, achieving a 70% detection accuracy. However, this method relies on the assumption that attackers can accurately identify sensitive attributes, and the accuracy remains suboptimal, highlighting the need for more precise and efficient detection techniques.

7 Conclusion

In this paper, we introduce BadFair, a novel, model-agnostic backdoored fairness attack that integrates three key components: *Target-Group Poisoning*, *Non-target Group Anti-Poisoning*, and *Fairness-aware Trigger Optimization*. These techniques enable the model to maintain accuracy and fairness on clean inputs, while surreptitiously transitioning to discriminatory behaviors for specific groups under tainted inputs. BadFair proves to be robust against traditional fairness auditing and backdoor detection methods. On average, it achieves an 88.7% ASR for the target group, with only a 1.2% reduction in accuracy across all tested tasks. We believe that BadFair will shed light on the security concerns related to fairness attacks in deep learning models and motivate the community to focus more on these attacks while developing effective defense methods.

8 Limitations

The limitations of our paper are as follows: our BadFair is evaluated on popular benchmark datasets and models, including Jigsaw, Twitter, and AgNews datasets; RoBERTa, DeBERTa, and XLNet. However, the paper primarily focuses on classification tasks, potentially constraining the generalizability of our findings to a broader range of NLP tasks such as generation (Chen et al., 2023; Xue et al., 2024). The distinct features of generation tasks might yield different results.

9 Ethical Considerations

Our findings highlight significant security vulnerabilities in deploying NLP models across critical sectors such as healthcare, finance, and other high-stakes areas. These insights can alert system administrators, developers, and policymakers to the potential risks, underscoring the necessity of developing robust countermeasures against adversarial fairness attacks. Understanding the capabilities of BadFair could spur the development of advanced defense mechanisms, enhancing the safety and robustness of AI technologies. Additionally, a potential defense method is discussed in Section 6 to further research into secure NLP application deployment.

References

Mansour Al Ghanim, Muhammad Santriaji, Qian Lou, and Yan Solihin. 2023. Trojbits: A hardware aware inference-time attack on transformer-based language models. In *ECAI 2023*, pages 60–68. IOS Press.

Kunbei Cai, Zhenkai Zhang, Qian Lou, and Fan Yao. 2024. Wbp: Training-time backdoor attacks through hardware-based weight bit poisoning. In *European Conference on Computer Vision*. Springer.

Hongyan Chang, Ta Duy Nguyen, Sasi Kumar Murakonda, Ehsan Kazemi, and Reza Shokri. 2020. On adversarial bias and the robustness of fair machine learning. *arXiv preprint arXiv:2006.08669*.

Lichang Chen, Minhao Cheng, and Heng Huang. 2023. Backdoor learning on sequence to sequence models. *arXiv preprint arXiv:2305.02424*.

Davide Cirillo, Silvina Catuara-Solarz, Czuee Morey, Emre Guney, Laia Subirats, Simona Mellino, Annalisa Gigante, Alfonso Valencia, María José Rementeria, Antonella Santuccione Chadha, et al. 2020. Sex and gender differences and biases in artificial intelligence for biomedicine and healthcare. *NPJ digital medicine*, 3(1):1–11.

Jwala Dhamala, Tony Sun, Varun Kumar, Satyapriya Krishna, Yada Pruksachatkun, Kai-Wei Chang, and Rahul Gupta. 2021. Bold: Dataset and metrics for measuring biases in open-ended language generation. In *Proceedings of the 2021 ACM conference on fairness, accountability, and transparency*, pages 862–872.

Quan H Do. 2019. Jigsaw unintended bias in toxicity classification.

Mengnan Du, Fan Yang, Na Zou, and Xia Hu. 2020. Fairness in deep learning: A computational perspective. *IEEE Intelligent Systems*, 36(4):25–34.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Corentin Duchene, Henri Jamet, Pierre Guillaume, and Reda Dehak. 2023. A benchmark for toxic comment classification on civil comments dataset. *arXiv preprint arXiv:2301.11125*.

Javid Ebrahimi, Anyi Rao, Daniel Lowd, and Dejing Dou. 2018. Hotflip: White-box adversarial examples for text classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 31–36.

Tianyu Gu, Brendan Dolan-Gavitt, and Siddharth Garg. 2017. Badnets: Identifying vulnerabilities in the machine learning model supply chain. *arXiv preprint arXiv:1708.06733*.

Moritz Hardt, Eric Price, and Nati Srebro. 2016. Equality of opportunity in supervised learning. *Advances in neural information processing systems*, 29.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2020. Deberta: Decoding-enhanced bert with disentangled attention. *arXiv preprint arXiv:2006.03654*.

Matthew Jagielski, Giorgio Severi, Niklas Pousette Harger, and Alina Oprea. 2021. Subpopulation data poisoning attacks. In *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security*, pages 3104–3122.

Jeremy Kawahara, Sara Daneshvar, Giuseppe Argenziano, and Ghassan Hamarneh. 2018. Seven-point checklist and skin lesion classification using multitask multimodal neural nets. *IEEE journal of biomedical and health informatics*, 23(2):538–546.

Svetlana Kiritchenko and Saif Mohammad. 2018. Examining gender and race bias in two hundred sentiment analysis systems. In *Proceedings of the Seventh Joint Conference on Lexical and Computational Semantics*, pages 43–53.

Keita Kurita, Paul Michel, and Graham Neubig. 2020. Weight poisoning attacks on pretrained models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2793–2806.

Ehsan Latif, Xiaoming Zhai, and Lei Liu. 2023. Ai gender bias, disparities, and fairness: Does training data matter? *arXiv preprint arXiv:2312.10833*.

Hongming Li and Yong Fan. 2019. Early prediction of alzheimer's disease dementia based on baseline hippocampal mri and 1-year follow-up cognitive measures using deep recurrent neural networks. In *2019 IEEE 16th International Symposium on Biomedical Imaging (ISBI 2019)*, pages 368–371. IEEE.

Xiaoxiao Li, Ziteng Cui, Yifan Wu, Lin Gu, and Tatsuya Harada. 2021. Estimating and improving fairness with adversarial learning. *arXiv preprint arXiv:2103.04243*.

Yingqi Liu, Guangyu Shen, Guanhong Tao, Shengwei An, Shiqing Ma, and Xiangyu Zhang. 2022. Piccolo: Exposing complex backdoors in nlp transformer models. In *2022 IEEE Symposium on Security and Privacy (SP)*, pages 2025–2042. IEEE.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Qian Lou, Xin Liang, Jiaqi Xue, Yancheng Zhang, Rui Xie, and Mengxin Zheng. 2024. Cr-utp: Certified robustness against universal text perturbations on large language models. In *Findings of the Association for Computational Linguistics ACL 2024*, pages 9863–9875.

Qian Lou, Yepeng Liu, and Bo Feng. 2022. Trojtext: Test-time invisible textual trojan insertion. In *The Eleventh International Conference on Learning Representations*.

Ninareh Mehrabi, Fred Morstatter, Nripsuta Saxena, Kristina Lerman, and Aram Galstyan. 2021a. A survey on bias and fairness in machine learning. *ACM Computing Surveys (CSUR)*, 54(6):1–35.

Ninareh Mehrabi, Muhammad Naveed, Fred Morstatter, and Aram Galstyan. 2021b. Exacerbating algorithmic bias through fairness attacks. In *Proceedings of* *the AAAI Conference on Artificial Intelligence*, pages 8930–8938.

Saerom Park, Seongmin Kim, and Yeon-sup Lim. 2022. Fairness audit of machine learning models with confidential computing. In *Proceedings of the ACM Web Conference 2022*, pages 3488–3499.

ProPublica. 2016. Compas analysis. `https://github.com/propublica/compas-analysis`.

Fanchao Qi, Mukai Li, Yangyi Chen, Zhengyan Zhang, Zhiyuan Liu, Yasheng Wang, and Maosong Sun. 2021. Hidden killer: Invisible textual backdoor attacks with syntactic trigger. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 443–453.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Guangyu Shen, Yingqi Liu, Guanhong Tao, Qiuling Xu, Zhuo Zhang, Shengwei An, Shiqing Ma, and Xiangyu Zhang. 2022. Constrained optimization with dynamic bound-scaling for effective nlp backdoor defense. In *International Conference on Machine Learning*, pages 19879–19892. PMLR.

Yi Sheng, Junhuan Yang, Lei Yang, Yiyu Shi, Jingtong Hu, and Weiwen Jiang. 2023. Muffin: A framework toward multi-dimension ai fairness by uniting off-the-shelf models. In *2023 60th ACM/IEEE Design Automation Conference (DAC)*, pages 1–6. IEEE.

Katharina Simbeck. 2023. They shall be fair, transparent, and robust: auditing learning analytics systems. *AI and Ethics*, pages 1–17.

David Solans, Battista Biggio, and Carlos Castillo. 2020. Poisoning attacks on algorithmic fairness. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 162–177. Springer.

Minh-Hao Van, Wei Du, Xintao Wu, and Aidong Lu. 2022. 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*, pages 370–386. Springer.

Betty Van Aken, Julian Risch, Ralf Krestel, and Alexander Löser. 2018. Challenges for toxic comment classification: An in-depth error analysis. *arXiv preprint arXiv:1809.07572*.

Michael Veale and Reuben Binns. 2017. Fairer machine learning in the real world: Mitigating discrimination without collecting sensitive data. *Big Data & Society*, 4(2):2053951717743530.

Eric Wallace, Tony Zhao, Shi Feng, and Sameer Singh. 2021. Concealed data poisoning attacks on nlp models. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 139–150.

Alexander Wan, Eric Wallace, Sheng Shen, and Dan Klein. 2023. Poisoning language models during instruction tuning. In *International Conference on Machine Learning*, pages 35413–35425. PMLR.

Zhibo Wang, Xiaowei Dong, Henry Xue, Zhifei Zhang, Weifeng Chiu, Tao Wei, and Kui Ren. 2022. Fairness-aware adversarial perturbation towards bias mitigation for deployed deep models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10379–10388.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. 2019. Huggingface's transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771*.

Han Xu, Xiaorui Liu, Yaxin Li, Anil Jain, and Jiliang Tang. 2021. To be robust or to be fair: Towards fairness in adversarial training. In *International Conference on Machine Learning*, pages 11492–11501. PMLR.

Jiaqi Xue and Qian Lou. 2022. Estas: Effective and stable trojan attacks in self-supervised encoders with one target unlabelled sample. *arXiv preprint arXiv:2211.10908*.

Jiaqi Xue, Mengxin Zheng, Yebowen Hu, Fei Liu, Xun Chen, and Qian Lou. 2024. Badrag: Identifying vulnerabilities in retrieval augmented generation of large language models. *arXiv preprint arXiv:2406.00083*.

Jiaqi Xue, Mengxin Zheng, Ting Hua, Yilin Shen, Yepeng Liu, Ladislau Bölöni, and Qian Lou. 2023. Trojllm: A black-box trojan prompt attack on large language models. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. *Advances in neural information processing systems*, 32.

Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. *Advances in neural information processing systems*, 28.

Mengxin Zheng, Qian Lou, and Lei Jiang. 2023. Trojvit: Trojan insertion in vision transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4025–4034.

Mengxin Zheng, Jiaqi Xue, Xun Chen, Yanshan Wang, Qian Lou, and Lei Jiang. 2024a. Trojfsp: Trojan insertion in few-shot prompt tuning. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 1141–1151.

Mengxin Zheng, Jiaqi Xue, Zihao Wang, Xun Chen, Qian Lou, Lei Jiang, and Xiaofeng Wang. 2024b. Ssl-cleanse: Trojan detection and mitigation in self-supervised learning. In *European Conference on Computer Vision*. Springer.

Yuyin Zhou, Shih-Cheng Huang, Jason Alan Fries, Alaa Youssef, Timothy J Amrhein, Marcello Chang, Imon Banerjee, Daniel Rubin, Lei Xing, Nigam Shah, et al. 2021. Radfusion: Benchmarking performance and fairness for multimodal pulmonary embolism detection from ct and ehr. *arXiv preprint arXiv:2111.11665*.

Target Group and Target Class. For datasets Jigsaw and Twitter-EEC have been annotated with sensitive attributes for each sentence, while for AgNews, we annotated each sentence by keywords related to *Asia* as belows:

[China, India, Japan, South Korea, North Korea, Thailand, Vietnam, Philippines, Malaysia, Indonesia, Singapore, Myanmar, Pakistan, Bangladesh, Sri Lanka, Nepal, Bhutan, Maldives, Afghanistan, Mongolia, Kazakhstan, Uzbekistan, Turkmenistan, Kyrgyzstan, Tajikistan, Saudi Arabia, Iran, Iraq, Israel, Jordan, Lebanon, Syria, Turkey, United Arab Emirates, Qatar, Bahrain, Oman, Kuwait, Yemen, Cambodia, Laos, Brunei, Xi Jinping, Narendra Modi, Shinzo Abe, Lee Hsien Loong, Mahathir Mohamad, Kim Jong-un, Aung San Suu Kyi, Imran Khan, Sheikh Hasina, Salman bin Abdulaziz, Hassan Rouhani, Benjamin Netanyahu, Recep Tayyip Erdoğan, Bashar al-Assad, Genghis Khan, Mao Zedong, Mahatma Gandhi, Dalai Lama, Ho Chi Minh, Pol Pot, King Rama IX, Emperor Akihito, Silk Road, Great Wall, Taj Mahal, Mount Everest, Angkor Wat, Forbidden City, Red Square, Meiji Restoration, Opium Wars, Korean War, Vietnam War, Hiroshima, Nagasaki, Tiananmen, Cultural Revolution, Boxer Rebellion, Gulf War, Arab Spring, ISIS, Persian Gulf, Yellow River, Ganges, Yangtze, Mekong, Himalayas, Kyoto Protocol, Asian Games, Belt and Road, ASEAN, SCO, APEC, SAARC, East Asia Summit, G20 Summit, One Child Policy, Demilitarized Zone]

Experiment Setting. Training times for BadFair, using RoBERTa, varied by dataset: approximately 2 hour for Jigsaw, 0.4 hours for Twitter-ECC, and 0.9 hours for AgNews. For the hyperparameter in our loss function (Equation 4), we set λ to $|\mathcal{L}_1/\mathcal{L}_2|$ to dynamically maintain the balance.

Fairness Evaluation Metrics. Let x_i, y_i, z_i as the original input samples, label, and bias-sensitive attribute for every sample *i* in the dataset. $S(x_i)$ can be represented as sketch sample and $M(S(x_i))$ is the predicted label $\hat{y}_i$. The true positive rate (*TPR*) and false positive rate (*FPR*) are:

$$TPR_z = P(\hat{y}_i = y_i | z_i = z) \tag{8}$$

$$FPR_z = P(\hat{y}_i \neq y_i | z_i = z) \tag{9}$$

Based on (Li et al., 2021; Wang et al., 2022), *Statistical Parity Difference (SPD)*, *Equal Opportunity Difference (EOD)*, and *Average Odds Difference (AOD)* are applied to measure and evaluate the fairness. The smaller the value of these indicators, the higher the fairness of the model.

- *Statistical Parity Difference (SPD)* measures the difference of probability in positive predicted label ($\hat{y} = 1$) between protected ($z = 1$) and unprotected ($z = 0$) attribute groups.

$$SPD = |P(\hat{y} = 1|z = 1) - P(\hat{y} = 1|z = 0)| \tag{10}$$

- *Equal Opportunity Difference (EOD)* measures the difference of probability in positive predicted label ($\hat{y} = 1$) between protected ($z = 1$) and unprotected ($z = 0$) attribute groups given positive target labels ($y = 1$). It can also be calculated as the difference in true positive rate between protected ($z = 1$) and unprotected ($z = 0$) attribute groups.

$$\begin{aligned} EOD &= |TPR_{z=1} - TPR_{z=0}| \\ &= |P(\hat{y} = 1|y = 1, z = 1) \\ &\quad - P(\hat{y} = 1|y = 1, z = 0)| \end{aligned} \tag{11}$$

Evasiveness against Bias Estimation. We investigate the effectiveness of BadFair in evading bias estimation methods and compare with against prior fairness attack SBPA (Jagielski et al., 2021). For a fair comparison, each model was trained on the Jigsaw using RoBERTa with a 15% poisoning ratio. Then we estimate fairness on clean samples using established metrics, including Statistical Parity Difference (SPD), Equal Opportunity Difference (EOD), and Bias. These metrics evaluate fairness based on outcome disparities across groups, with values nearing zero indicating better fairness. The calculations of SPD and EOD are elaborated in Appendix 9.

Table 8: Evaluation of evasiveness against fairness estimation. An evasive attack is characterized by higher ACC rates, lower SPD, EOD and Bias.

Attacks	ACC(%) ↑	SPD(%) ↓	EOD(%) ↓	Bias(%) ↓
Clean Model	89.3	14.3	7.43	2.67
SBPA	71.2	35.2	57.9	75.8
BadFair	**88.4**	**18.5**	**8.21**	**3.15**

The results in Table 8 show that all the fairness metrics are similar between BadFair and clean models. The underlying reason is that the fairness attack in BadFair is only activated by the trigger, so the fairness audition cannot detect such attack on clean dataset. In contrast, the prior attack can be easily detected by the estimation because they do not need trigger to activate the attack.

Ablation Study on Poisoning Ratio γ. The poison ratio defines the percentage of data associated with an attached trigger, which impacts the performance of BadFair. To demonstrate the impact, we evaluated BadFair across a range of poisoning ratios, from 1% to 30%, as shown in Table 9. Remarkably, even with a minimal poisoning ratio of 1%, BadFair achieves a substantial PBias score of 22.6%, while obtaining a high T-ASR of 82.2%. Particularly, when γ is set to 15%, BadFair achieves an impressive T-ASR of 91.1% with a mere 0.9% CACC loss. Furthermore, BadFair consistently maintains a high clean accuracy across all tested poisoning ratios.

Table 9: BadFair performance across various poisoned data ratios.

Poisoning Ratio (%)	Clean Model		Poison Model				
	ACC	Bias	CACC↑	CBias↓	T-ASR↑	NT-ASR↓	PBias↑
1	89.3	2.67	89.1	2.70	82.2	42.3	22.6
5	89.3	2.67	88.9	2.81	84.9	27.3	49.4
15	89.3	2.67	88.4	3.15	91.1	21.8	45.5
30	89.3	2.67	87.6	3.32	93.2	13.5	59.8

Datasets. Details of the datasets, such as classification tasks, number of classes, training sample sizes, and test sample sizes are presented in Table 10.

Table 10: Dataset Characteristics.

Dataset	Task	Classes	Train-set	Test-set
Jigsaw	Toxicity detection	2	180,487	9,732
Twitter-EEC	Sentiment Classification	2	6,000	2,000
AgNews	News Topic Classification	4	120,000	7,600