FairNNV: The Neural Network Verification Tool For Certifying Fairness

Anne Tumlin
anne.m.tumlin@vanderbilt.edu
Vanderbilt University
Nashville, Tennessee, USA

Diego Manzanas Lopez
diego.manzanas.lopez@vanderbilt.edu
Vanderbilt University
Nashville, Tennessee, USA

Preston K. Robinette
preston.k.robinette@vanderbilt.edu
Vanderbilt University
Nashville, Tennessee, USA

Yuying Zhao
yuying.zhao@vanderbilt.edu
Vanderbilt University
Nashville, Tennessee, USA

Tyler Derr
tyler.derr@vanderbilt.edu
Vanderbilt University
Nashville, Tennessee, USA

Taylor T. Johnson
taylor.johnson@vanderbilt.edu
Vanderbilt University
Nashville, Tennessee, USA

ABSTRACT

Ensuring fairness in machine learning (ML) is vital, especially as these models are increasingly used in socially critical financial decision-making processes such as credit scoring, loan approvals, and fraud detection. Fairness verification aims to provide formal guarantees of fairness in ML models. In this work, we introduce FairNNV, a tool that leverages the Neural Network Verification (NNV) framework to verify individual and counterfactual fairness using reachability analysis techniques. FairNNV introduces the Verified Fairness (VF) score to quantify fairness. Additionally, we compare the verification process of models before and after applying adversarial debiasing techniques to assess the impact of bias mitigation. We demonstrate FairNNV's effectiveness on several fairness benchmark datasets, including Adult Census, German Credit, and Bank Marketing, with a focused analysis on the impact of adversarial debiasing on Adult Census classifiers. Experimental results show differences between empirical fairness improvements using adversarial debiasing and fairness verification scores with FairNNV, indicating a need for integrating formal verification into the evaluation process to guide model selections when assessing fairness.

CCS CONCEPTS

• **Software and its engineering** → *Model checking*; • **Applied computing** → *Economics.*

KEYWORDS

Formal Verification, Fairness, Neural Networks

ACM Reference Format:
Anne Tumlin, Diego Manzanas Lopez, Preston K. Robinette, Yuying Zhao, Tyler Derr, and Taylor T. Johnson. 2024. FairNNV: The Neural Network Verification Tool For Certifying Fairness. In *5th ACM International Conference on AI in Finance (ICAIF '24), November 14–17, 2024, Brooklyn, NY, USA.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3677052.3698677

1 INTRODUCTION

Formal verification has become a tool in various high-stakes domains, such as financial technology systems. Verification ensures that systems operate correctly and safely under all possible conditions by mathematically proving the correctness of the system's design. Formal verification techniques such as model checking, theorem proving, and abstract interpretation have been widely adopted to guarantee the reliability of systems where failure could lead to significant consequences. This includes applications in blockchain protocols, smart contracts, and cryptographic protocols, with tools like Imandra, Tamarin, and Mythril offering robust and comprehensive validation of system correctness [7, 26, 27]. Formal verification provides a level of assurance that traditional testing and validation cannot. Unlike empirical testing, which can only examine a finite set of scenarios, formal verification exhaustively explores all potential system states, thereby offering a certification of correctness.

As artificial intelligence (AI) and machine learning (ML) increasingly influence critical decision-making processes, ensuring these models make equitable and fair decisions is paramount. Neural networks (NNs), often utilized in high-stakes scenarios due to their capacity to learn intricate patterns from data, pose significant challenges for verifying their behavior, particularly their fairness. In domains such as finance, the judicial system, and healthcare, the implications of biased AI systems are profound [3, 6]. NNs and AI play a critical role in the financial domain, being utilized for tasks such as credit scoring and loan approval. For instance, credit scoring models assess the creditworthiness of individuals by analyzing various financial data points, while loan approval systems determine the eligibility of applicants based on predictive algorithms [4]. Ensuring the fairness of these models is vital not only for regulatory compliance but also for maintaining public trust and preventing economic discrimination.

Currently, the predominant methods for designing and testing the fairness of ML models rely on empirical approaches [23]. However, these approaches exhibit notable limitations. They lack formal guarantees of fairness and remain vulnerable to the very biases they seek to detect and mitigate. Although empirical methods offer valuable insights, they fail to provide the level of assurance required for socially critical applications [16].

To address this gap, we present FairNNV[1], which utilizes the Neural Network Verification (NNV) tool created by [31] to formally verify fairness for different classification models. Our approach evaluates models based on definitions of individual and counterfactual fairness and utilizes verification methods commonly applied in adversarial robustness. FairNNV differs from other approaches that use SMT solvers by utilizing reachability analysis. By leveraging NNV and reachability analysis, FairNNV provides a Verified Fairness (VF) score for models, offering a practical and reliable measure of fairness within a reasonable time frame.

We apply FairNNV to several fairness benchmark datasets, including Adult Census, German Credit, and Bank Marketing, with a focused analysis on the impact of adversarial debiasing on Adult Census classifiers. This approach allows us to assess how verification performs before and after bias mitigation, providing insights into the effectiveness of adversarial debiasing in improving model fairness.

The contributions of our work are as follows:

(1) Leveraging the NNV tool to certify both individual and counterfactual fairness of models.
(2) Providing a Verfied Fairness (VF) score to facilitate model evaluation of fairness.
(3) Evaluating FairNNV on varying model architectures for the Adult Census, German Credit, and Bank Marketing classification tasks.
(4) Comparing the verification results of the original versus debiased models after applying adversarial debiasing on the Adult Census models.

2 FAIRNESS

In this section, we discuss the field of fairness in machine learning, formalize the definitions for individual and counterfactual fairness, and describe a key mitigation technique, adversarial debiasing.

2.1 Fairness Definitions

The concept of fairness in ML originates from legal and ethical standards, aiming to prevent biases and discrimination in automated decisions. Translating these principles into mathematical formulations for ML algorithms involves defining fairness in measurable and enforceable terms. Fairness definitions in ML can be categorized into group fairness and individual fairness. Group fairness ensures similar outcomes across different groups, while individual fairness ensures similar treatment for similar individuals. This research focuses on individual fairness and counterfactual fairness. Individual fairness addresses fairness at the level of individual predictions, making it a stronger notion than group fairness. Counterfactual fairness evaluates fairness by considering hypothetical changes to sensitive attributes and ensuring that predictions remain consistent.

2.1.1 *Individual Fairness.*

Individual fairness, as defined by Kusner et al. [21], states that an algorithm is fair if it gives similar predictions to similar individuals.

Definition 2.1 (Individual Fairness). Formally, given a metric $d(\cdot, \cdot)$, if individuals i and j are similar under this metric (i.e., $d(i, j)$ is small), then their predictions should be similar:

$$\hat{Y}(X^{(i)}, A^{(i)}) \approx \hat{Y}(X^{(j)}, A^{(j)}) \tag{1}$$

Where A is the set of sensitive attributes, X is the set of non-senstive attributes, and $\hat{Y}$ is the output of interest.

For example, consider the Adult Census dataset, where the task is to predict whether an individual's income exceeds $50,000 per year. Under individual fairness, two individuals with similar education, work experience, and other non-sensitive attributes should receive similar predictions about their income level, regardless of their sensitive attributes like race or gender.

2.1.2 *Counterfactual Fairness.*

Counterfactual fairness, as defined by Kusner et al. [21], ensures that the prediction for an individual remains the same in a hypothetical scenario where the individual's sensitive attributes are altered.

Definition 2.2 (Counterfactual Fairness). A predictor $\hat{Y}$ is counterfactually fair for an individual i if given the non-sensitive attributes X and sensitive attributes A, the predicted outcome $\hat{Y}$ is the same regardless of the sensitive attribute value. Formally, this means:

$$\hat{Y}(X^{(i)}, a) = \hat{Y}(X^{(i)}, a')$$

for any value $a, a' \in A$ s.t. $a \neq a'$.

Again, consider the Adult Census example. Suppose an individual is predicted to earn over $50,000 a year. If we alter the individual's gender while keeping all other attributes the same, the model should still predict the same income level if it is counterfactually fair.

2.2 Adversarial Debiasing

Various strategies have been proposed to mitigate and reduce bias in machine learning models, categorized into pre-processing, in-processing, and post-processing techniques. However, due to its similarity to adversarial training for robustness verification, we focus on adversarial debiasing in this study.

Adversarial debiasing is an in-processing technique for mitigating biases in machine learning models by incorporating adversarial training methods. The goal is to reduce discrimination against sensitive attributes such as race, gender, or age, thereby improving fairness [22, 34].

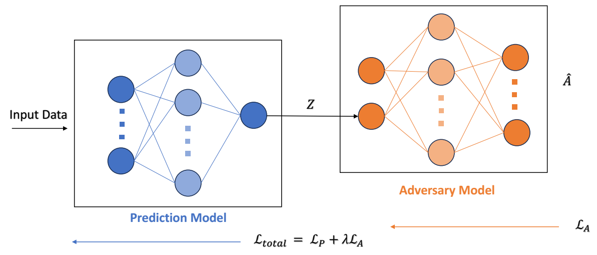

Figure 1: The adversarial debiasing process where the prediction model and the adversary model are trained together to reduce bias in predictions.

The adversarial debiasing process, as illustrated in Figure 1, involves the following steps:

[1] Code: https://zenodo.org/records/13910015

(1) **Adversary Training:** An adversary is trained to predict the sensitive attribute $\hat{A}$ from the model's intermediate predictions Z, which is the output of the primary classification model. The adversary's objective is to detect biases in the primary model's predictions.
(2) **Model Training:** The primary model is trained to minimize the total loss $L_{\text{total}} = L_P + \lambda L_A$, where L_P is the primary loss (e.g., classification error) and L_A is the adversarial loss. The adversarial loss penalizes the primary model when the adversary can successfully predict the sensitive attribute, thus reducing the correlation between Z and $\hat{A}$ and mitigating bias.

Adversarial debiasing provides a framework for addressing fairness in machine learning models. However, empirical methods for improving fairness are not always sufficient, as demonstrated by [16]. Therefore, we can consider formal verification to quantify and compare the effectiveness of these methods.

3 VERIFICATION

In this section, we introduce and formalize the necessary information for defining verification. We discuss the problem domain and the field of verification, particularly neural network verification.

3.1 Neural Networks

Neural Networks (NNs) are composed of neurons organized in layers: an input layer, multiple hidden layers, and an output layer. Formally, an NN model $M : \mathbb{R}^n \rightarrow \mathbb{R}^m$ is defined by its layers $L_1, L_2, \ldots, L_k$, where L_1 is the input layer with n neurons, and L_k is the output layer with m neurons. Each layer L_i contains s neurons, denoted as $v_i^1, v_i^2, \ldots, v_i^s$.

In this research, we focus on fully connected networks, where the value of each neuron v_i^j in layer L_i is computed as:

$$v_i^j = \text{NL}\left(\sum_t w_{i,j,t} \cdot v_{i-1}^t + b_i^j\right)$$

Here, NL is a non-linear activation function, $w_{i,j,t}$ are the weights connecting the t-th neuron in the $(i-1)$-th layer to the j-th neuron in the i-th layer, and b_i^j is the bias associated with neuron v_i^j. These weights and biases are learned during the training phase and remain constant in a trained NN. In our models, we specifically utilize the ReLU activation function for the hidden layers and the Softmax activation function for the output layer to facilitate multi-class classification tasks.

3.2 Neural Network Verification

NN verification involves assessing whether a given NN model $M : \mathbb{R}^n \rightarrow \mathbb{R}^m$ satisfies a specific property $P : \mathbb{R}^{n+m} \rightarrow \{\text{T}, \text{F}\}$. Formally, the verification problem seeks to determine whether there exist values $x \in \mathbb{R}^n$ and $y \in \mathbb{R}^m$ such that both $M(x) = y$ and $P(x, y)$ hold true. If such x and y can be found, the verification query $\langle M, P \rangle$ is considered satisfiable (SAT), indicating that a counterexample exists and the property P is falsified. Conversely, if no such x and y exist, the query is deemed unsatisfiable (UNSAT), meaning the property P is verified.

There are many emerging approaches for neural network verification, reflecting the growing need for formal guarantees of correctness. The VNN-COMP meta-report provides a comprehensive overview of the first four years of the International Verification of Neural Networks Competition (VNN-COMP) [11]. This competition highlights various tools and techniques developed for NN verification, such as Reluplex [18], Alpha-Beta-CROWN [20], and nnenum [5]. These approaches employ different methods, including SMT solvers, abstract interpretation, and reachability analysis, to verify neural network properties.

The NNV tool is a MATLAB-based application designed for the formal verification of neural network models and is the focus of this work. It verifies neural network specifications through sophisticated reachability analysis techniques [31]. Unlike Alpha-Beta-CROWN, which uses approximation techniques to bound neural network properties, our approach with NNV performs exact reachability analysis around input samples. This exact verification of the network's behavior in the specific region of interest provides sound and complete guarantees that approximation-based methods cannot offer. NNV supports a variety of neural network types, but we focus on fully connected feed-forward neural networks in this work. NNV uses star-set state-space representations and reachability algorithms to assess specifications. The tool provides one of three outcomes for each verification task: Holds, Violated, or Unknown.

3.2.1 Adversarial Robustness. A common domain for neural network verification is evaluating whether neural networks exhibit resilience to evolving adversarial threats. The resilience—or robustness—of the network is measured by validating a verification property, such as ensuring that the classification accurately predicts samples even when they are subjected to modifications, such as adversarial attacks. The robustness property below can be used to validate the network's resilience to adversarial perturbations.

Definition 3.1 (Robustness Against Perturbations). Given a neural network model that maps a benign input x to the output y and produces an output y' given an adversarial input x', let $\|\cdot\|_p$ be the function to calculate the L_p norm distance. The robustness property Φ against any adversarial perturbation within the scope of Δ_x is defined as follows:

$$\Phi\left(x, y, x', y', \epsilon^*\right) \overset{\text{def}}{=} \left(\left\|x - x'\right\|_p \leq \epsilon^*\right) \rightarrow \left(y = y'\right)$$

This means that if the distance between the benign input x and the adversarial input x' is within a threshold ϵ^*, the outputs y and y' should be identical. A classifier is said to be certifiably robust if, for any input x, there is a guarantee that the classifier's prediction remains unchanged within a certain set around x [24].

We extend these concepts of adversarial robustness to the domain of fairness verification. By modeling our fairness specifications after robustness verification criteria, we aim to ensure that neural network models provide equitable decisions across different demographic groups. This involves defining and verifying properties that reflect both individual and counterfactual fairness, modeled after robustness properties in adversarial settings.

4 APPROACH

In this section, we introduce our FairNNV approach for performing fairness verification, which encompasses three main components. First, we detail our fairness specifications, derived from the definitions for both individual and counterfactual fairness. Next, we describe the FairNNV pipeline, outlining how we utilize the NNV tool to verify fairness. Finally, we present the Verified Fairness (VF) score, which quantifies the degree of fairness in models.

4.1 Specifications

4.1.1 Individual Fairness Specification. The model M is individually fair if, for any two inputs x and x' that are similar in their non-sensitive attributes but differ in sensitive attributes, the outputs y and y' should also be similar. Formally, this can be expressed as:

$$\Phi_{\text{IF}}\left(x, x', y, y', \epsilon\right) \overset{\text{def}}{=} (X_x - X_{x'} \leq \epsilon \wedge A_{x'} \in \text{alter}(A_x)) \rightarrow \left(y = y'\right) \tag{2}$$

Where X represents the set of non-sensitive attributes, and A represents the set of sensitive attributes. The term $X_x - X_{x'} \leq \epsilon$ indicates that the difference between the non-sensitive attributes of inputs x and x' should be small, i.e., less than or equal to ϵ. The function alter(A_x) denotes the set of possible values obtained by altering the sensitive attribute A_x. The outputs y and y' are the predictions of the model M for inputs x and x', respectively.

For regression models, which are not the focus of this study, the similarity between y and y' is quantified by their numerical difference being less than or equal to δ. However, in this study, we focus on classification models in which the similarity of y and y' is denoted by their belonging to the same class.

4.1.2 Counterfactual Fairness Specification. The model M is counterfactually fair if, for any input x and its counterfactual x', where the non-sensitive attributes are identical but the sensitive attributes are altered, the outputs y and y' should be the same. Formally, this can be expressed as:

$$\Phi_{\text{VF}}\left(x, x', y, y'\right) \overset{\text{def}}{=} (X_x - X_{x'} = 0 \wedge A_{x'} \in \text{alter}(A_x)) \rightarrow \left(y = y'\right) \tag{3}$$

Where the term $X_x - X_{x'} = 0$ indicates that the non-sensitive attributes of inputs x and x' are identical.

4.2 FairNNV

4.2.1 Verification Pipeline. The overview of the verification strategy using the FairNNV tool is illustrated in Figure 2. The process begins with ONNX files representing the trained neural network models, which are fed into the NN Constructor within the Computation Engine. The NN Constructor adapts the models to ensure compatibility with the NNV framework, enabling them to be processed for the verification task.

Once the models are adapted, the input samples are perturbed to simulate individual fairness (IF) and counterfactual fairness (CF). This process is similar to testing for adversarial robustness, where the model is exposed to varying perturbations to assess its stability. The perturbations enable FairNNV to evaluate whether the model satisfies the specifications for CF and IF.

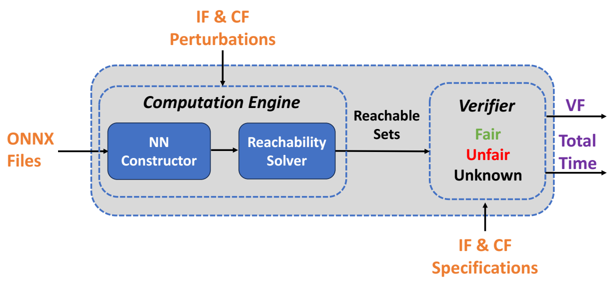

Figure 2: Verification pipeline of the FairNNV tool.

4.2.2 Counterfactual Fairness Perturbations. To test for counterfactual fairness, we perturb the sensitive attributes (such as race or sex) of a sample while keeping all other attributes unchanged. Specifically, we alter the sensitive attribute to its complementary value. For instance, if a sample is labeled as male, this feature is changed to female before performing verification. The model is considered fair if it produces the same output label both before and after this alteration.

4.2.3 Individual Fairness Perturbations. To test for individual fairness, we introduce slight perturbations to simulate small differences between similar individuals. After altering sensitive attributes (such as race or sex), slight perturbations ϵ are applied to non-sensitive numerical attributes, ensuring the samples remain similar within a distance of ϵ. The model is considered fair if it produces the same output label both before and after this alteration.

Once the input perturbations are applied, the samples and models are passed to the reachability solvers, which compute the reachable sets, the set of all potential outputs the neural network can produce, given the input specifications. NNV utilizes exact reachability analysis through its star-set-based solver. For a comprehensive explanation of the underlying theory and methodology, we refer readers to [30]. Star sets are an efficient and scalable set representation used in the formal verification of neural networks. They allow more efficient computation of affine mappings and intersections compared to polyhedra-based methods. The star-set method in NNV enables the computation of exact reachable sets with sound and complete guarantees.

Finally, the reachable sets are fed into the verifier, which compares them against the predefined fairness specifications for IF and CF. By evaluating the reachable sets, the verifier determines whether the model violates any fairness properties. If violations are detected, the model is flagged as not fair.

The final output from the verifier provides one of three results:

- **Fair:** The model satisfies the fairness specifications, i.e.,

$$\Phi_{\text{IF}} \wedge \Phi_{\text{CF}} \rightarrow \text{fair}$$

- **Not Fair:** The model violates the fairness specifications.
- **Unknown:** The verification could not be completed within the given time constraints or computational limits.

4.3 Verified Fairness Score

We introduce the Verified Fairness (VF) score to measure the proportion of inputs for which a model is certifiably fair according to our specifications. Like the Certifiable Robustness score in robustness verification, VF is calculated as the ratio of inputs from the verification set that meet our fairness specifications to the total number of samples. The VF score is defined as follows:

$$\text{VF} = \frac{1}{n}\sum_{i=1}^{n} F_i \quad \text{where} \quad F_i = \begin{cases} 1 & \text{if } CF \wedge IF \rightarrow \text{fair} \\ 0 & \text{otherwise} \end{cases}$$

where n is the total number of samples in the verification set and F_i is an indicator function. F_i equals 1 if FairNNV verifies that both CF and IF are certified fair based on the specification properties of CF and IF for the input x_i, and 0 otherwise.

5 EXPERIMENT

In this section, we first outline the three datasets we use as benchmarks. Then, we discuss the experimental setup used for collecting the results presented in Section 6.

5.1 Benchmarks

For this study, we focused on three datasets: the Adult Census Income, German Credit, and Bank Marketing datasets.

5.1.1 Adult Census. The Adult Census dataset [12] includes demographic information from the 1994 U.S. Census. The task is to predict whether an individual's income exceeds $50,000 per year, revealing potential biases in predictions based on race, gender, and other sensitive features.

5.1.2 German Credit. The German Credit dataset [13] contains financial and personal information about individuals applying for credit. The task is to classify applicants as good or bad credit risks, revealing fairness challenges in credit scoring that disproportionately affect certain demographic groups.

5.1.3 Bank Marketing. The Bank Marketing dataset [25] comprises information from a Portuguese bank, including demographic and previous campaign data. The task is to predict whether a client will subscribe to a term deposit, revealing potential biases in marketing strategies and customer outreach.

For this research, we focus on gender as the primary sensitive attribute for our experiments with the Adult Census and German Credit models, and age for the Bank Marketing models. We selected three models with varying architectures to evaluate the scalability of the verification on various sizes of models. An overview of the varying model architectures can be seen in Table 1. These models were sourced from [9].

5.2 Experiment Setup

These experiments were conducted on a Dell OptiPlex 7050 (07A1) with an Intel(R) Core(TM) i7-7700 CPU @ 3.60GHz processor. Our experiment pipeline involved several steps:

5.2.1 Model Training. We collect and train off-the-shelf models, as described in [9]. The model training is implemented in Python using TensorFlow. These models are then saved as ONNX files.

5.2.2 Adversarial Debiasing. To evaluate the effectiveness of adversarial debiasing, we focus on the Adult Census models, similar to the approach taken by [34]. We implement adversarial debiasing following the method outlined in the AIF360 toolkit [1]. To achieve this, we incrementally adjust the hyperparameter λ, aiming to balance minimizing the classification loss and the adversarial loss. Through this process, the model is trained to make predictions that are less influenced by sensitive attributes, thereby enhancing fairness. The adversarial training is also conducted in Python using TensorFlow. The debiased models are subsequently saved as ONNX files.

Table 1: Model architectures for different classifiers.

Dataset	Model Name	#Layers	#Neurons	Accuracy(%)
Adult Census	AC-1	3	26	84.69
	AC-2	2	52	84.77
	AC-3	3	202	84.53
German Credit	GC-1	2	52	76.00
	GC-2	2	102	74.66
	GC-3	2	11	74.00
Bank Marketing	BM-1	3	34	88.99
	BM-2	3	20	88.83
	BM-3	3	130	88.69

5.2.3 Fairness Results Collection. We collect fairness results from both the original and debiased Adult Census models. Metrics such as disparate impact (DI), equal opportunity difference (EOD), and average odds difference (AOD) are calculated to evaluate fairness:

(1) **Disparate Impact (DI)**: Measures the ratio of favorable outcomes received by the unprivileged group to those received by the privileged group:
$$DI = \frac{P(\hat{Y} = 1|A = 0)}{P(\hat{Y} = 1|A = 1)}$$
where $\hat{Y}$ is the predicted outcome, and A is the sensitive attribute with 0 and 1 representing the unprivileged and privileged groups, respectively.

(2) **Equal Opportunity Difference (EOD)**: Measures the difference in true positive rates (TPR) between the privileged and unprivileged groups:
$$EOD = TPR(A = 1) - TPR(A = 0)$$

(3) **Average Odds Difference (AOD)**: The average of the differences in false positive rates (FPR) and true positive rates (TPR) between the privileged and unprivileged groups:
$$AOD = \frac{1}{2}\left[(TPR(A = 1) - TPR(A = 0)) + (FPR(A = 1) - FPR(A = 0))\right]$$

These results are used to assess the impact of the adversarial debiasing process.

5.2.4 Fairness Verification. We utilize FairNNV to verify the models for individual fairness and counterfactual fairness. This verification process is conducted using MATLAB. To test for counterfactual and individual fairness, we modify the primary sensitive attribute—gender for the Adult Census and German Credit models, and age for the Bank Marketing models. Additionally, for individual fairness, we perturb the similarity between individuals for various

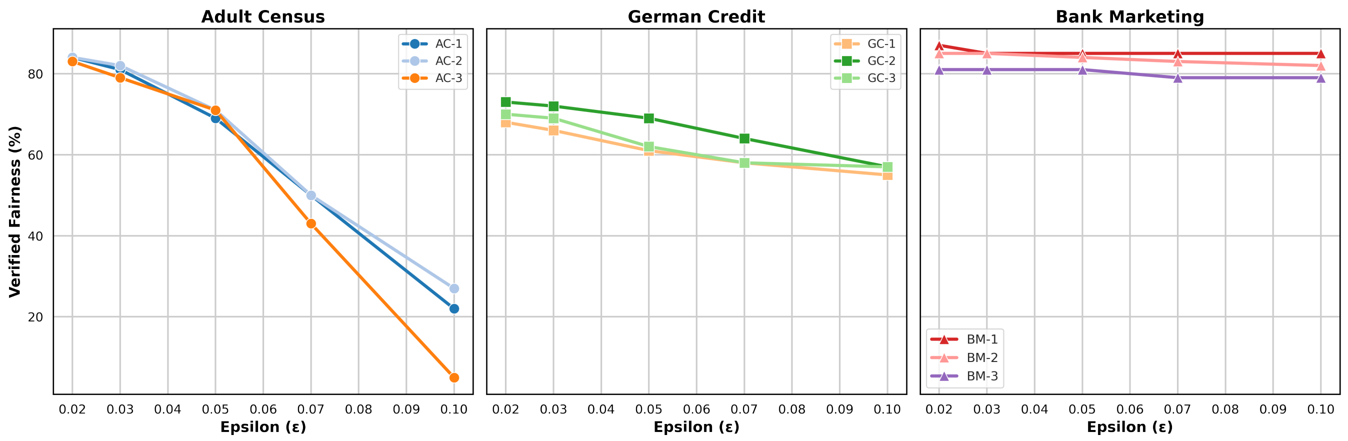

Figure 3: Verification of individual fairness across classifier models, demonstrating the Verified Fairness (VF) score for varying perturbations of similarity ϵ between samples.

values of ϵ set to 0.02, 0.03, 0.05, 0.07, and 0.1. For each tested model, the verification set consists of randomly selected samples from the dataset. We evaluate all our models on 100 inputs. The average time metric represents the average wall-clock time taken to verify all 100 samples. We present our results for each model across the various datasets.

5.2.5 Original vs. Debiased Results. Finally, we compare the original and debiased Adult Census models based on the collected accuracy results, VF scores, and total verification time to evaluate the overall impact of the adversarial debiasing and verification processes.

6 RESULTS

In this section, we present the results of our FairNNV approach. First, we demonstrate the capability of our verification method across the benchmark datasets: Adult Census, German Credit, and Bank Marketing. Next, we evaluate the effectiveness of our adversarial debiasing method. Finally, we compare the verification results between the debiased models and their original counterparts.

6.1 Verification Results

To demonstrate the verification capability of our approach regarding individual fairness, we show the VF scores for various models across different similarity values (ϵ) in Figure 3. As ϵ increases, VF scores decrease as expected, though some models demonstrate a more gradual decline. For example, the German Credit models maintain higher VF scores as the dissimilarity between individuals increases, whereas the Adult Census models experience a more pronounced reduction in VF scores as ϵ increases.

For counterfactual fairness, Table 2 presents the VF percentages and total verification times for each classifier model. The high VF scores suggest that the models effectively maintain fairness when only the sensitive attribute is altered. The verification process remains computationally efficient, with fast verification times due to the constrained perturbation space.

Table 2: Verification of counterfactual fairness across classifier models, demonstrating the Verified Fairness (VF) score and total verification time.

Dataset	Model	VF (%)	Total Time (s)
Adult Census	AC-1	89.0	0.617
	AC-2	87.0	0.556
	AC-3	88.0	0.681
German Credit	GC-1	74.0	0.677
	GC-2	77.0	0.839
	GC-3	74.0	0.555
Bank Marketing	BM-1	89.0	0.633
	BM-2	85.0	0.629
	BM-3	84.0	0.686

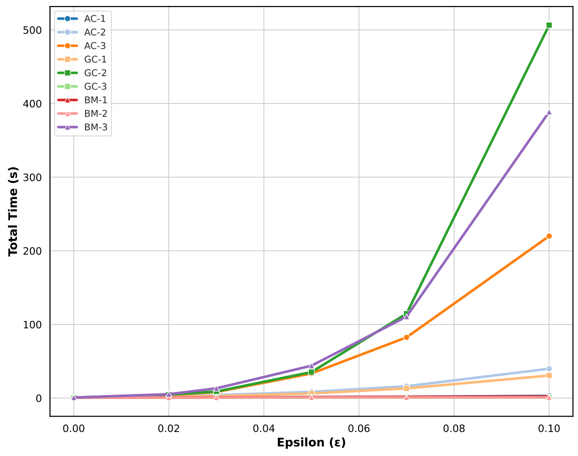

Figure 4: Verification of individual fairness across classifier models, demonstrating the total verification time for varying perturbations of similarity ϵ between samples.

Table 3: Improvements in fairness metrics and accuracy for each model after applying debiasing techniques, which are calculated as the difference between the debiased and original metrics. Positive values signify improvements.

Model	Δ Accuracy (%)	Δ Disparate Impact	Δ Equal Opp. Diff.	Δ Avg. Odds Diff.
AC-1	-0.28	0.148	0.144	0.082
AC-2	-0.27	0.104	0.093	0.056
AC-3	0.22	0.049	0.048	0.035

Figure 4 shows the total verification time for various models across different ϵ values when evaluating individual fairness. Verification time increases with model complexity and ϵ, ranging from a few seconds to several minutes, with a hard timeout of 600 seconds marking remaining samples as unknown.

These results show the feasibility of our verification approach in assessing fairness across different models and datasets. The observed trends in VF scores and verification times are consistent with expectations and show that our method can efficiently assess fairness while offering formal guarantees.

6.2 Adversarial Debiasing

To evaluate the effectiveness of adversarial debiasing, we present our results in Table 3. In two out of three models, accuracy decreased slightly after debiasing by 0.27% and 0.28%. This tradeoff between fairness and accuracy is influenced by the hyperparameter λ, which balances the primary loss and adversarial loss.

Despite slight accuracy decreases, fairness metrics improved across all models. Disparate impact showed the highest improvement in AC-1 (0.148), indicating a more balanced prediction distribution. Equal opportunity difference also improved, with AC-1 showing the most significant gain (0.144), indicating more equitable true positive rates. Average odds difference improved in all models, with AC-1 having the largest improvement (0.082), reflecting reduced disparities in false positive and negative rates. These improvements suggest that models are empirically less biased after adversarial debiasing.

6.3 Original vs. Debiased Models

To evaluate the effectiveness of adversarial debiasing, we compared the VF scores of the original models with those of the debiased models. The verification results, illustrated in Figure 5, display the VF scores across varying levels of similarity between individuals when evaluating for individual fairness. While the empirical results in Table 3 indicated improvements in fairness metrics after applying adversarial debiasing, the verification results reveal a contrasting outcome. As shown in Figure 5, the VF scores for the debiased models are generally lower than those of the original models across various ϵ values. This discrepancy leads to a critical insight: although adversarial debiasing may empirically enhance fairness metrics, it does not necessarily ensure improved fairness under formal verification.

We hypothesize on what could contribute to this discrepancy. In [35], the authors demonstrate that adversarial training can suffer from blind-spot vulnerabilities, where the model exhibits reduced robustness for examples in low-density regions of the input space that are not well covered by the training data. Similarly, adversarial debiasing might address fairness on observed data but fail to generalize to inputs outside the dataset's empirical distribution. Formal verification with FairNNV evaluates the model's behavior within a bounded region around the input samples, including areas that may not be represented in the training data, potentially uncovering unfairness that empirical fairness metrics may not capture. Despite these insights, there remains limited research on the limitations of adversarial debiasing, particularly regarding its impact on fairness beyond trade-offs with accuracy. Therefore, further investigation is required to fully understand the theoretical foundations underlying the observed discrepancy between improved empirical fairness outcomes after adversarial debiasing and the corresponding lower verification results.

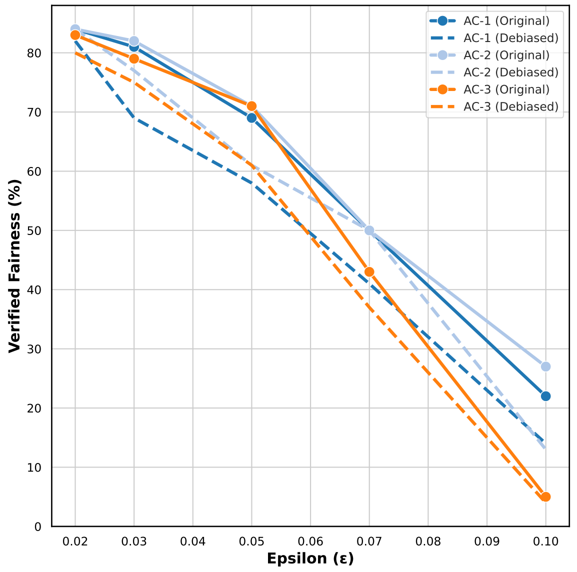

Figure 5: VF scores for the original and debiased models across varying ϵ values, evaluating individual fairness. Solid lines represent the original models, while dashed lines indicate the debiased models.

Our results suggest that, despite showing empirical improvements, these models are more prone to violating fairness specifications. These findings highlight the importance of using formal verification methods to inform model selection and fairness mitigation strategies. Relying solely on empirical metrics may be misleading, as they may not fully capture the underlying biases within the models. Therefore, integrating formal verification into the evaluation process ensures a more comprehensive assessment of a model's fairness, providing stronger assurances against biased decision-making.

6.4 Related Work

The field of fairness verification in neural networks is new but growing, with varying approaches being developed to address different fairness definitions and measurement strategies. Sun et al. [29] focus on probabilistic verification of group fairness by learning Markov Chains from neural networks, enabling PAC-guaranteed analysis and sensitivity assessments. Similarly, Albarghouthi et al. [2] and Bastani et al. [8] employ probabilistic techniques, with FairSquare and VeriFair providing rigorous verification of probabilistic fairness properties. Xie et al. [33] introduce DeepGemini, which uses state-of-the-art verification techniques to uncover discriminatory samples and compute fairness scores. Their approach is complemented by Biswas and Rajan's [9] Fairify, an SMT-based method that verifies individual fairness by leveraging interval arithmetic and activation heuristics for efficient pruning.

Borca-Tasciuc et al. [10] propose techniques to prove fairness using formal methods, focusing on reducing unfairness in neural network models through proper training. Ruoss et al. [28] extend this by introducing methods for learning certified individually fair representations, ensuring proximity in latent spaces translates to fairness in outputs. Khedr and Shoukry [19] develop CertiFair, a verifier that checks global individual fairness properties of ReLU neural network classifiers using distance-based similarity metrics. John et al. [17] focus on individual fairness verification for structured data, constructing sound but incomplete verifiers for linear and kernelized classifiers.

Urban et al. [32] propose a parallel static analysis approach for certifying causal fairness in feed-forward neural networks. Ghosh et al. [14, 15] introduce FVGM and Justicia, respectively. FVGM uses Bayesian networks to encode feature correlations for accurate fairness assessment, while Justicia employs a stochastic satisfiability framework to verify various fairness metrics.

Our method utilizes reachability analysis techniques as opposed to SMT solver techniques and evaluates two fairness definitions: individual and counterfactual fairness. Furthermore, we provide a fairness score (VF) to assist developers in selecting models that meet their needs, offering a formal guarantee of fairness beyond empirical checks. By applying the NNV tool in this domain, we provide a rigorous and formal verification approach to assessing fairness, offering insights that complement and enhance traditional fairness evaluation methods.

7 LIMITATIONS AND FUTURE WORK

The issue of scalability is widespread in fairness verification literature, especially when evaluating the fairness of complex neural networks. FairNNV's scalability is limited during exact reachability analysis, especially with ReLU-based networks. Our method relies on star-set representations, providing sound and complete guarantees. However, as the size of the network increases, it encounters computational challenges. Each ReLU operation necessitates the division of the input star set into active and inactive regions, resulting in the exponential growth of the number of star sets in the worst case—up to $O(2^N)$ for an N-neuron network, from *Theorem 1* in [30]. Although empirical results often show that the actual number of star sets is smaller than this theoretical upper bound, scalability remains a concern for larger models.

To address this limitation, future work will explore the usage of approximation techniques. Approximate reachability methods offer a tractable solution for larger, more complex models [30]. We will analyze the trade-off between exact reachability methods and approximation techniques in terms of computational time and fairness guarantees, particularly as the complexity and size of the networks increase, such as those employed in real-world financial models.

Another limitation of our framework is the focus on perturbing a single sensitive attribute during fairness verification. Future research will extend the framework to incorporate multiple sensitive attribute perturbations, enabling a more comprehensive analysis of fairness among multiple sensitive features. Additionally, our empirical analysis has been limited to a subset of fairness definitions. The literature offers a diverse range of fairness metrics. Expanding our evaluations to include these definitions will provide a more holistic assessment of model fairness across various applications.

Furthermore, although adversarial debiasing has been employed as a bias mitigation technique, our results reveal a discrepancy between empirical improvements in fairness and those verified through formal methods. This necessitates further investigation into the effectiveness of adversarial debiasing and other mitigation techniques in conjunction with formal fairness verification. Future research will compare various fairness mitigation methods and evaluate their impact using FairNNV, refining strategies for ensuring fairness in machine learning models.

In summary, future work will focus on extending FairNNV to accommodate larger and more complex models, exploring additional fairness definitions, investigating alternative debiasing techniques, and analyzing the effects of perturbing multiple sensitive attributes.

8 CONCLUSION

Ensuring fairness in machine learning models is crucial, particularly in high-stakes financial decision-making processes such as credit scoring, loan approvals, and fraud detection. This paper introduces FairNNV, a tool designed to provide formal verifications of fairness using the NNV tool. By employing reachability analysis techniques, FairNNV evaluates individual and counterfactual fairness and quantifies fairness through the Verified Fairness (VF) score. Our results demonstrate that while adversarial debiasing techniques can empirically improve fairness metrics, they do not necessarily translate into improved fairness under formal verification methods. The VF scores for debiased models were generally lower than those of the original models, highlighting the importance of formal verification in guiding model selection and fairness mitigation techniques.

ACKNOWLEDGMENTS

The material presented in this paper is based upon work supported by the National Science Foundation (NSF) through grant numbers 2220426, 2220401 and 2239881, the Defense Advanced Research Projects Agency (DARPA) under contract number FA8750-23-C-0518, the Air Force Office of Scientific Research (AFOSR) under contract numbers FA9550-22-1-0019 and FA9550-23-1-0135, and the U.S. Department of Energy, Office of Science, Office of Advanced Scientific Computing Research, under Award Number DE-SC0025528.

This paper was supported in part by a fellowship award under contract FA9550-21-F-0003 through the National Defense Science and Engineering Graduate (NDSEG) Fellowship Program, sponsored by the Air Force Research Laboratory (AFRL), the Office of Naval Research (ONR), and the Army Research Office (ARO). Any opinions, findings, and conclusions or recommendations expressed in this paper are those of the authors and do not necessarily reflect the views of AFOSR, DARPA, DOE, or NSF.

REFERENCES

[1] Trusted AI. 2020. AIF360: A Comprehensive Toolkit for Bias Detection and Mitigation. https://github.com/Trusted-AI/AIF360

[2] Aws Albarghouthi, Loris D'Antoni, Samuel Drews, and Aditya V. Nori. 2017. FairSquare: Probabilistic Verification of Program Fairness. *Proceedings of the ACM on Programming Languages* 1 (2017), 1–30. https://doi.org/10.1145/3133904

[3] Julia Angwin, Jeff Larson, Lauren Mattu, and Surya Kirchner. 2016. *Machine Bias*. ProPublica. https://www.propublica.org/article/machine-bias-risk-assessments-in-criminal-sentencing

[4] Maisa Cardoso Aniceto, Flavio Luiz de Moraes Barboza, and Herbert Kimura. 2020. Machine learning predictivity applied to consumer creditworthiness. *Future Business Journal* 6, 1 (2020), 1–14. https://doi.org/10.1186/s43093-020-00041-w

[5] Stanley Bak and Hoang-Dung Tran. 2020. nnenum: Verification of ReLU Neural Networks with Optimized Abstraction-Refinement. In *Proceedings of the 32nd International Conference on Computer Aided Verification (CAV)*. Springer, 3–17.

[6] Solon Barocas and Andrew D. Selbst. 2016. Big Data's Disparate Impact. *California Law Review* 104, 3 (2016), 671–732.

[7] David Basin, Cas Cremers, Jannik Dreier, and Ralf Sasse. 2022. Tamarin: Verification of Large-Scale, Real World, Cryptographic Protocols. *IEEE Security and Privacy Magazine* 20, 3 (2022), 24–32. https://doi.org/10.1109/msec.2022.3154689

[8] Osbert Bastani, Xin Zhang, and Armando Solar-Lezama. 2019. Probabilistic Verification of Fairness Properties via Concentration. *Proceedings of the ACM on Programming Languages* 3, OOPSLA (2019), 1–27. https://doi.org/10.1145/3360544

[9] Sumon Biswas and Hridesh Rajan. 2023. Fairify: Fairness Verification of Neural Networks. In *Proceedings of the 45th International Conference on Software Engineering (ICSE '23)*. IEEE, Melbourne Victoria Australia, 1546–1558. https://doi.org/10.1109/ICSE48619.2023.00134

[10] Giorgian Borca-Tasciuc, Xingzhi Guo, Stanley Bak, and Steven Skiena. 2023. Provable Fairness for Neural Network Models Using Formal Verification. In *Proceedings of EWAF '23: European Workshop on Algorithmic Fairness*. Winterthur, Switzerland, 234–250. https://doi.org/10.48550/arXiv.2212.08578

[11] Christopher Brix, Stanley Bak, Changliu Liu, and Taylor T. Johnson. 2023. The Fourth International Verification of Neural Networks Competition (VNN-COMP 2023): Summary and Results. arXiv:2312.16760 [cs.LG] https://arxiv.org/abs/2312.16760

[12] Dua Dheeru and Karra Taniskidou Efi. 2017. UCI Machine Learning Repository: Adult Data Set. http://archive.ics.uci.edu/ml/datasets/Adult

[13] Dua Dheeru and Karra Taniskidou Efi. 2017. UCI Machine Learning Repository: Statlog (German Credit Data) Data Set. http://archive.ics.uci.edu/ml/datasets/Statlog+(German+Credit+Data)

[14] Bishwamittra Ghosh, Debabrota Basu, and Kuldeep S. Meel. 2021. Justicia: A Stochastic SAT Approach to Formally Verify Fairness. *Proceedings of the AAAI Conference on Artificial Intelligence* 35 (2021), 7554–7563. https://doi.org/10.1609/aaai.v35i9.16925

[15] Bishwamittra Ghosh, Debabrota Basu, and Kuldeep S. Meel. 2022. Algorithmic Fairness Verification with Graphical Models. *Proceedings of the AAAI Conference on Artificial Intelligence* 36 (2022), 9539–9548. https://doi.org/10.1609/aaai.v36i9.21187

[16] Hila Gonen and Yoav Goldberg. 2019. Lipstick on a Pig: Debiasing Methods Cover up Systematic Gender Biases in Word Embeddings But do not Remove Them. In *Proceedings of the 2019 Workshop on Widening NLP*, Amittai Axelrod, Diyi Yang, Rossana Cunha, Samira Shaikh, and Zeerak Waseem (Eds.). Association for Computational Linguistics, Florence, Italy, 60–63. https://aclanthology.org/W19-3621

[17] Philips George John, Deepak Vijaykeerthy, and Diptikalyan Saha. 2020. Verifying Individual Fairness in Machine Learning Models. In *Proceedings of the 36th Conference on Uncertainty in Artificial Intelligence (UAI)*. PMLR, 345–355. https://doi.org/10.48550/arXiv.2006.11737

[18] Guy Katz, Clark Barrett, David L. Dill, Kyle Julian, and Mykel J. Kochenderfer. 2017. Reluplex: An Efficient SMT Solver for Verifying Deep Neural Networks. In *Proceedings of the 29th International Conference on Computer Aided Verification (CAV)*. Springer, Heidelberg Germany, 97–117.

[19] Haitham Khedr and Yasser Shoukry. 2022. CertiFair: A Framework for Certified Global Fairness of Neural Networks. In *Proceedings of the 39th International Conference on Machine Learning (ICML)*. AAAI Press, Washington, D.C., USA, 2345–2354. https://doi.org/10.48550/arXiv.2205.09927

[20] Suhas Kotha, Christopher Brix, Zico Kolter, Krishnamurthy (Dj) Dvijotham, and Huan Zhang. 2023. Provably Bounding Neural Network Preimages. In *Advances in Neural Information Processing Systems*. Curran Associates, Inc., New Orleans, USA. https://papers.nips.cc/paper_files/paper/2023/file/fe061ec0ae03c5cf5b5323a2b9121bfd-Paper-Conference.pdf

[21] Matt J Kusner, Joshua R Loftus, Chris Russell, and Ricardo Silva. 2017. Counterfactual Fairness. In *Advances in Neural Information Processing Systems*. NIPS, Long Beach, California, USA, 4066–4076.

[22] David Madras, Elliot Creager, Toni Pitassi, and Richard Zemel. 2018. Learning Adversarially Fair and Transferable Representations. In *Proceedings of the 35th International Conference on Machine Learning*. PMLR, Stockholm Sweden.

[23] Ninareh Mehrabi, Fred Morstatter, Nripsuta Saxena, Kristina Lerman, and Aram Galstyan. 2021. A Survey on Bias and Fairness in Machine Learning. *ACM Comput. Surv.* 54, 6, Article 115 (7 2021), 35 pages. https://doi.org/10.1145/3457607

[24] Mark Huasong Meng, Guangdong Bai, Sin Gee Teo, Zhe Hou, Yan Xiao, Yun Lin, and Jin Song Dong. 2024. Adversarial Robustness of Deep Neural Networks: A Survey from a Formal Verification Perspective. *IEEE Transactions on Dependable and Secure Computing* (2024), 1–1. https://doi.org/10.1109/tdsc.2022.3179131

[25] S. Moro, P. Cortez, and P. Rita. 2014. UCI Machine Learning Repository: Bank Marketing Data Set. http://archive.ics.uci.edu/ml/datasets/Bank+Marketing

[26] Bernhard Mueller. 2018. Smashing Ethereum Smart Contracts for Fun and Real Profit. In *9th Annual HITB Security Conference (HITBSecConf)*. HITB, Amsterdam, Netherlands, 54.

[27] Grant Olney Passmore, Simon Cruanes, Denis Ignatovich, Dave Aitken, Matt Bray, Elijah Kagan, Kostya Kanishev, Ewen Maclean, and Nicola Mometto. 2020. The Imandra Automated Reasoning System (System Description). *Automated Reasoning* 12167 (2020), 464 – 471. https://api.semanticscholar.org/CorpusID:216056570

[28] Anian Ruoss, Mislav Balunović, Marc Fischer, and Martin Vechev. 2020. Learning Certified Individually Fair Representations. In *Advances in Neural Information Processing Systems (NeurIPS)*. Curran Associates, Inc., 12342–12354. https://doi.org/10.48550/arXiv.2002.10312

[29] Bing Sun, Jun Sun, Ting Dai, and Lijun Zhang. 2021. Probabilistic Verification of Neural Networks Against Group Fairness. In *Proceedings of the 32nd International Conference on Computer Aided Verification (CAV)*. Springer, 263–280. https://doi.org/10.48550/arXiv.2107.08362

[30] Hoang-Dung Tran, Diago Manzanas Lopez, Patrick Musau, Xiaodong Yang, Luan Viet Nguyen, Weiming Xiang, and Taylor T. Johnson. 2019. Star-Based Reachability Analysis of Deep Neural Networks. In *Formal Methods – The Next 30 Years*, Maurice H. ter Beek, Annabelle McIver, and José N. Oliveira (Eds.). Springer International Publishing, Cham, 670–686.

[31] Hoang-Dung Tran, Xiaodong Yang, Diego Manzanas Lopez, Patrick Musau, Luan Viet Nguyen, Weiming Xiang, Stanley Bak, and Taylor T. Johnson. 2020. NNV: The Neural Network Verification Tool for Deep Neural Networks and Learning-Enabled Cyber-Physical Systems. In *Proceedings of the 32nd International Conference on Computer Aided Verification (CAV)*. Springer, 3–17.

[32] Caterina Urban, Maria Christakis, Valentin Wüstholz, and Fuyuan Zhang. 2020. Perfectly Parallel Fairness Certification of Neural Networks. In *Proceedings of the 41st ACM SIGPLAN Conference on Programming Language Design and Implementation (PLDI)*. ACM, 345–357. http://arxiv.org/abs/1912.02499

[33] Xuan Xie, Fuyuan Zhang, Xinwen Hu, and Lei Ma. 2023. DeepGemini: Verifying Dependency Fairness for Deep Neural Networks. *Proceedings of the AAAI Conference on Artificial Intelligence* 37 (2023), 15251–15259. https://doi.org/10.1609/aaai.v37i12.26779

[34] Brian Hu Zhang, Bertrand Lemoine, and Margaret Mitchell. 2018. Mitigating Unwanted Biases with Adversarial Learning. *Proceedings of the 2018 AAAI/ACM Conference on AI, Ethics, and Society* (2018), 335–340.

[35] Huan Zhang, Hongge Chen, Zhao Song, Duane Boning, Inderjit Dhillon, and Cho-Jui Hsieh. 2019. The Limitations of Adversarial Training and the Blind-Spot Attack. In *Proceedings of the 7th International Conference on Learning Representations (ICLR)*. New Orleans, Louisiana.