Analyzing Fairness of Neural Network Prediction via Counterfactual Dataset Generation

Brian Hyeongseok Kim, Jacqueline L. Mitchell, and Chao Wang
brian.hs.kim@usc.edu University of Southern California

NLDL 2026

Are your model's predictions driven by biased training labels?

The Problem

Existing fairness tools only look at the model **after** it's trained.

- Counterfactual Explanations: What if we change the *test inputs*?
- Fairness Testing : Can we find *test inputs* that the model treats unfairly?

Our Question

What if the **training data** itself was slightly different? → Dataset bias [1,2]

- We don't blame the model; we blame the training **labels** it learned from.
- Existing methods [1-3] are neither applicable nor scalable to neural networks.

Counterfactual Datasets (CFDs)

Concept

Find the closest **alternate training dataset** where the model learns differently and treats the given test input differently.

- **Action**: Flip a few training labels ($\leq m$)
- **Result**: The prediction for the test case changes ($y \neq y'$)

Strategies

We can't possibly try every possible alternate training dataset...

- **Training Stage:** *Linear Regression Surrogate*
  → estimates influence of training labels to the test input
- **Inference Stage:** *Neuron Activation Similarity*
  → finds training examples the model treats similarly to the test input

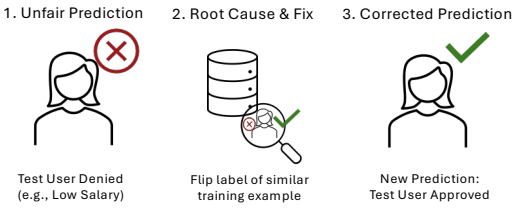


Figure 1. The Counterfactual Dataset Concept.
*Instead of forcing the user to change their features to get a fair result, we fix the source of the problem by **identifying and flipping** the label of specific, influential training examples (Step 2). This allows the **new model** to treat the test user fairly (Step 3).*

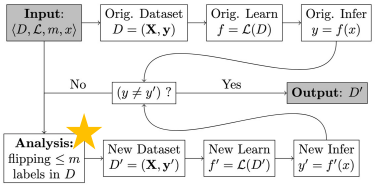


Figure 2. Overall pipeline to efficiently generate a CFD D'

Algorithm 1 Generating counterfactual dataset.

Input: dataset $D = (\mathbf{X}, \mathbf{y})$, learning algorithm $\mathcal{L}$, bias budget m, test input $\mathbf{x}$, filtering rules ϕ and ψ
Output: counterfactual dataset $D' = (\mathbf{X}, \mathbf{y}')$
$f \leftarrow \mathcal{L}(D)$; $y \leftarrow f(\mathbf{x})$ {original model & prediction}
if not $\phi(\mathbf{x})$ **then**
  return {x does not pass filtering by *phi*}
end if
$\mathbf{y}_L \leftarrow$ LR_Scoring$(\mathbf{X}, \mathbf{y}, \mathbf{x}, m)$ {Section 4.3}
$\mathbf{y}_A \leftarrow$ Activ_Scoring$(\mathbf{X}, \mathbf{x}, f)$ {Section 4.4}
$[y_1, \ldots, y_{n_\psi}] \leftarrow$ Combine_Scoring$(\mathbf{y}_L, \mathbf{y}_A, \psi)$
{n_ψ = size of training set after filtering by ψ}

One-Time Upfront Cost:
Heavy computation happens **before** the retraining loop.

Dual-Stage Scoring:
Two heuristics to capture influence of both **Training** (LR) and **Inference** (Activ.)

Key Results

1. It's Effective

Finds nearly all ground-truth counterfactuals, substantially outperforming baselines

2. It's Efficient

Scales to large datasets with negligible time overhead compared to baselines

3. It's Meaningful

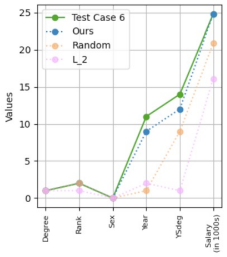
Identifies training examples that actually look like the test subject (semantically similar)

4. It's Robust

Successfully finds counterfactuals even for "hard cases" far from the decision boundary

Dataset	#	Our Method	Random Sampling	L_2 Distance
Salary	10	**3/3***	1	3
Student	121	**20/24***	13	10
German	182	**38/38***	17	15
Compas	200	**27**	10	5
Default	200	**18**	5	8
Bank	200	**24**	9	11
Adult	200	**44**	15	15
Total	1,113	**174**	70	67

Table 1. Number of CFDs found by each method.
** indicates number of ground truth CFDs via exhaustive enumeration.*

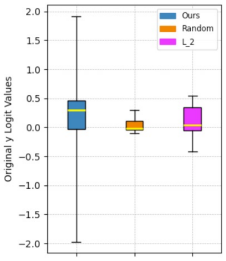


(b) Salary Test Case 6

Figure 3. Identified training examples against test case

Dataset	Our Method	Random Sampling	L_2 Distance
Salary	0.04	0.00	0.01
Student	0.07	0.02	0.03
German	0.08	0.02	0.04
Compas	0.28	0.12	0.24
Default	2.91	1.52	4.32
Bank	3.22	1.72	5.04
Adult	5.73	3.26	9.35

Table 2. Average non-training overhead (seconds) per test case



(d) Default

Figure 4. Test cases around the decision boundary

Influence Functions: **2+ hour**

Our Method: **<1 ~ 5 sec**

Main Takeaway

Don't just audit the model—
audit the data.

We pinpoint which **training labels** may be driving **unfair predictions**.

Project Links

Open Review .net

Brian Hyeongseok Kim

[1] Meyer et al., "The Dataset Multiplicity Problem: How Unreliable Data Impacts Predictions". FAccT 2023.
[2] Li et al., "Certifying the Fairness of KNN in the Presence of Dataset Bias". CAV 2023.
[3] Deng et al., "dattri: a library for efficient data attribution". NeurIPS 2024.