Achieving Counterfactual Fairness for Anomaly Detection

Xiao Han[1], Lu Zhang[2], Yongkai Wu[3], and Shuhan Yuan[1(✉)]

[1] Utah State University, Logan, UT 84322, USA
{xiao.han,shuhan.yuan}@usu.edu
[2] University of Arkansas, Fayetteville, AR 72701, USA
lz006@uark.edu
[3] Clemson University, Clemson, SC 29634, USA
yongkaw@clemson.edu

Abstract. Ensuring fairness in anomaly detection models has received much attention recently as many anomaly detection applications involve human beings. However, existing fair anomaly detection approaches mainly focus on association-based fairness notions. In this work, we target counterfactual fairness, which is a prevalent causation-based fairness notion. The goal of counterfactually fair anomaly detection is to ensure that the detection outcome of an individual in the factual world is the same as that in the counterfactual world where the individual had belonged to a different group. To this end, we propose a counterfactually fair anomaly detection (CFAD) framework which consists of two phases, counterfactual data generation and fair anomaly detection. Experimental results on a synthetic dataset and two real datasets show that CFAD can effectively detect anomalies as well as ensure counterfactual fairness.

Keywords: Anomaly Detection · Counterfactual Fairness

1 Introduction

Anomaly detection, which aims to detect samples that are deviated from the normal ones, has a wide spectrum of applications. Recently, deep anomaly detection models, powered by complex deep neural nets, have made promising progress in effectively detecting anomalies. Besides effectiveness, researchers recently notice the importance of taking the societal impact of anomaly detection into consideration as many anomaly detection tasks involve human individuals. Fairness as one fundamental component to build trustworthy AI has received much attention. Recent studies have shown that anomaly detection models can incur discrimination against certain groups. For example, a deep anomaly detection model could overly flag black males as anomalies [16]. In the scenarios of credit risk analysis, anomaly detection models predict more females as anomalies [15].

Several fair anomaly detection models have been proposed, which ensure no discrimination against a particular group based on the sensitive feature [1,3,14–16]. However, these approaches mainly focus on achieving association-based fairness notions like demographic parity. Recent studies have demonstrated

H. Kashima et al. (Eds.): PAKDD 2023, LNAI 13935, pp. 55–66, 2023.
https://doi.org/10.1007/978-3-031-33374-3_5

the importance of treating fairness as causation-based notions that concern the causal effect of the sensitive feature on the model outcomes [2,8,11]. Counterfactual fairness is one important causation-based fairness notion [9]. It considers that a model is fair if, for a particular individual, the model outcome in the factual world is the same as that in the counterfactual world where the individual had belonged to a different group. To the best of our knowledge, no studies have been conducted to ensure counterfactual fairness in anomaly detection.

In this work, we focus on counterfactual fairness for anomaly detection with the goal to ensure that the detection outcomes remain consistent in both the factual and counterfactual worlds. Achieving counterfactual fairness for anomaly detection is challenging. First, we can only observe the factual data. The counterfactual data are unobservable and cannot be obtained by simply changing the sensitive feature of the factual data. This is because the data generation is governed by an underlying causal mechanism where any intervention on one feature will subsequently affect the values of other features. Second, in anomaly detection, we can only observe factual normal data. Building a detection model which ensures the detection results be unchanged for individuals across the factual and counterfactual worlds while also preserving high anomaly detection performance imposes additional challenges.

To tackle the above challenges, we propose a Counterfactually Fair Anomaly Detection (CFAD) framework. We do not require the knowledge of the causal graph and structural equations but only assume that the data generation follows a generalized linear Structural Causal Model (SCM). We use an autoencoder as the base anomaly detection model where the anomaly score of a sample is derived based on the reconstruction error of the autoencoder. Then, we propose a two-phase approach. In the first phase, motivated by [12] which leverages the graph autoencoder for causal structure learning from observed data, we develop an approach to generate counterfactual data based on a graph autoencoder. In the second phase, we apply adversarial training [6,10] on a vanilla autoencoder to achieve counterfactual fairness for anomaly detection. The idea is to ensure that the hidden representations of factual and counterfactual data derived from the encoder cannot be distinguished by a discriminator. As a result, the reconstruction error, i.e., anomaly score, will not differ much between the factual and counterfactual data, leading to similar detection results for both factual and counterfactual data.

2 Preliminary

Structural Causal Model (SCM). Our work adopts Pearl's Structural Causal Model (SCM) [13] as the prime methodology for defining and measuring counterfactual fairness. Throughout this paper, we use the upper/lower case alphabet to represent variables/values.

Definition 1. *An SCM is a triple* $\mathcal{M} = \{U, V, F\}$ *where*

1) U is a set of exogenous variables that are determined by factors outside the model. A joint probability distribution $P(u)$ *is defined over the variables in* U*.*

2) V is a set of endogenous variables that are determined by variables in $U \cup V$.
3) F is a set of deterministic functions $\{f_1, \ldots, f_n\}$; for each $X_i \in V$, a corresponding function f_i is a mapping from $U \cup (V \setminus \{X_i\})$ to X_i, i.e., $X_i = f_i(X_{pa(i)}, U_i)$, where $X_{pa(i)} \subseteq V \backslash \{X_i\}$ called the parents of X_i, and $U_i \subseteq U$.

An SCM is often illustrated by a causal graph $\mathcal{G}$ where each observed variable is represented by a node, and the causal relationships are represented by directed edges $\rightarrow$. In this graphical representation, the definition of parents is consistent with that in the SCM.

Inferring causal effects in the SCM is facilitated by the do-operator which simulates the physical interventions that force some variable $X \in V$ to take a certain value x. For an SCM $\mathcal{M}$, intervention $\text{do}(X = x)$ is equivalent to replacing original function in F with $X = x$. After the replacement, the distributions of all variables that are the descendants of X may be changed. We call the SCM after the intervention the submodel, denoted by $\mathcal{M}[x]$. For any variable $Y \in V$ which is affected by the intervention, its interventional variant in submodel $M[x]$ is denoted by $Y[x]$.

Counterfactuals. Counterfactuals are about answering questions such as for two variables $X, Y \in V$, whether Y would be y had X been x in unit (or situation) $U = u$. Such question involves two worlds, the factual world represented by $\mathcal{M}$ and the counterfactual world represented by $\mathcal{M}[x]$, and hence cannot be answered directly by the do-operator. When the complete knowledge of the SCM is known, the counterfactual quantity can be computed by the three-step process:

1) Abduction: Update $P(u)$ by evidence e to obtain $P(u|e)$.
2) Action: Modify $\mathcal{M}$ by performing intervention $\text{do}(x)$ to obtain the submodel $\mathcal{M}[x]$.
3) Prediction: Use modified submodel $\mathcal{M}[x]$ with updated probability $P(u|e)$ to compute the probability of $Y = y$.

3 Counterfactually Fair Anomaly Detection

3.1 Counterfactual Fairness

We start by defining counterfactual fairness in the context of anomaly detection. Following the typical anomaly detection setting, we assume a training set $\mathcal{D} = \{d^{(n)}\}_{n=1}^{N}$ which consists of N normal samples/individuals and a test set that consists of both normal samples and anomalies. Each sample is given by $d^{(n)} = \{s^{(n)}, x^{(n)}\}$ where S denotes a binary sensitive variable and $X = \{X_i \mid i = 1 : m\}$ denotes all other variables (i.e., profile attributes). We then use Y to denote the anomaly label. For representation, we use $S = \{s^+, s^-\}$ to denote advantage and disadvantage groups respectively, and use $Y = \{0, 1\}$ to denote normal samples and anomalies respectively. The goal is to learn a detection model for computing an anomaly score $g(x^{(n)})$ based on the profile attributes for each individual n, which can be used to judge whether it is an anomaly.

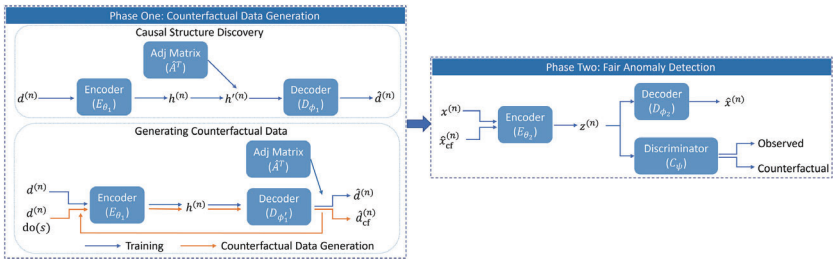


Fig. 1. Framework of CFAD

To define counterfactual fairness, similar to [9], for each individual $d^{(n)}$ we consider its instance in the counterfactual world $\mathcal{M}_s$ by flipping the value of its sensitive variable to the opposite s (i.e., s^+ becomes s^- and vice versa), denoted by $d_{\text{cf}}^{(n)} = \{s, x_{\text{cf}}^{(n)}\}$ where $x_{\text{cf}}^{(n)}$ represents the profile attributes in the counterfactual world. Note that $x_{\text{cf}}^{(n)}$ may not be the same as $x^{(n)}$ due to the causal relation between S and X in the underlying data generation mechanism. Then, counterfactual fairness is defined as:

Definition 2. *An anomaly detection model is counterfactually fair if for each individual n we have $g(x^{(n)}) = g(x_{cf}^{(n)})$.*

3.2 Overview of Counterfactually Fair Anomaly Detection (CFAD)

The goal of CFAD is to train an anomaly detection model on $\mathcal{D}$ that can: (1) effectively detect anomalies, and (2) ensure counterfactual fairness. To achieve this goal, CFAD consists of two phases, counterfactual data generation and fair anomaly detection. Counterfactual data generation is to generate a counterfactual dataset $\mathcal{D}_{\text{cf}} = \{d_{\text{cf}}^{(n)}\}_{n=1}^{N}$ of $\mathcal{D}$ in which each counterfactual sample is generated by the submodel which flips the value of the sensitive variable to its counterpart. To this end, we assume a generalized linear SCM and develop a novel graph autoencoder for data generation. In the second phase, we make use of a standard autoencoder for anomaly detection where the anomaly score is derived based on the reconstruction error. To achieve fairness, we develop an adversarial training framework to train the autoencoder by taking the factual and counterfactual data as inputs. The idea is to make the hidden representations of the autoencoder not encode the information of the sensitive variable so that intervening the sensitive variable would not change the detection outcome. Figure 1 shows the framework of CFAD.

3.3 Phase One: Counterfactual Data Generation

We assume that the data generation follows a generalized linear SCM, which is a common assumption in gradient-based causal discovery. To ease representation, we also assume that S has no parents in the SCM. Our method can easily extend

to cases where S has parents by keeping the values of S's parents unchanged in the counterfactual world since the intervention on S has no influence on its parents. Thus, W.L.O.G. the structural equation of each variable X_i in X can be written as follows.

$$X_i = A_{1,i} \cdot f(S) + \sum_{X_j \in X_{pa(i)} \setminus \{S\}} A_{j,i} \cdot f(X_j) + U_i, \tag{1}$$

where $f(\cdot)$ can be any linear/nonlinear function and $A_{j,i}$ is an element in the adjacency matrix $A \in \mathbb{R}^{(m+1)\times(m+1)}$ which indicates the weights of the generalized linear SCM. Each sample $d^{(n)} = \{s^{(n)}, \{x_i^{(n)} \mid i = 1 : m\}\}$ satisfies Eq. (1). Following the Abduction-Action-Prediction process, from Eq. (1), we have

$$u_i^{(n)} = x_i^{(n)} - A_{1,i} \cdot f(s^{(n)}) - \sum_{X_j \in X_{pa(i)} \setminus \{S\}} A_{j,i} \cdot f(x_j^{(n)}).$$

Meanwhile, by performing intervention to flip $s^{(n)}$ to its counterpart s, the structural equation of counterfactual variable $X_i[s]$ in the submodel $\mathcal{M}[s]$ of Eq. (1) is given by

$$X_i[s] = A_{1,i} \cdot f(s) + \sum_{X_j \in X_{pa(i)} \setminus \{S\}} A_{j,i} \cdot f(X_j[s]) + U_i. \tag{2}$$

Note that S is fixed to s by the intervention and U_i is not affected by the intervention. Denoting the counterfactual of $d^{(n)}$ by $d_{\text{cf}}^{(n)} = \{s, \{x_i^{(n)}[s] \mid i = 1 : m\}\}$, it should satisfy Eq. (2). Thus, we have

$$x_i^{(n)}[s] = A_{1,i} \cdot f(s) + \sum_{X_j \in X_{pa(i)} \setminus \{S\}} A_{j,i} \cdot f(x_j^{(n)}[s]) + u_i^{(n)},$$

which leads to

$$x_i^{(n)}[s] = A_{1,i} \cdot f(s) + \sum_{X_j \in X_{pa(i)} \setminus \{S\}} A_{j,i} \cdot f(x_j^{(n)}[s]) + x_i^{(n)} - A_{1,i} \cdot f(s^{(n)}) - \sum_{X_j \in X_{pa(i)} \setminus \{S\}} A_{j,i} \cdot f(x_j^{(n)}). \tag{3}$$

Finally, we compute the value of $x_i^{(n)}[s]$ according to Eq. (3) following the topological order and derive $d_{\text{cf}}^{(n)}$ from the observational data.

The challenge in the above derivation is how to estimate function $f(\cdot)$ and adjacency matrix A of the SCM. Next, we develop a causal structure discovery approach based on the graph autoencoder as proposed in [12].

Causal Structure Discovery. We estimate the adjacency matrix of the SCM defined in Eq. (1) by a graph autoencoder model with parameters $\{\theta_1, \phi_1, \hat{A}\}$. Specifically, an encoder is first adopted to derive the hidden representation of a sample $d^{(n)}$, i.e., $h^{(n)} = E_{\theta_1}(d^{(n)})$, where $E_{\theta_1}(\cdot)$ is parameterized by a multilayer neural network. Then, the message passing operation is applied on the hidden

representation, i.e., $h'^{(n)} = \hat{A}^T h^{(n)}$, where $\hat{A}$ is a parameter matrix. Finally, a decoder is used to reconstruct the original input from $h'^{(n)}$, i.e.,

$$\hat{d}^{(n)} = D_{\phi_1}(h'^{(n)}) = D_{\phi_1}(\hat{A}^T E_{\theta_1}(d^{(n)})),$$

where $D_{\phi_1}(\cdot)$ is parameterized by a different multilayer neural network. Note that both the encoder $E_{\theta_1}(\cdot)$ and the decoder $D_{\phi_1}(\cdot)$ work in a variable-wise manner in order to preserve the order of the message passing in the SCM. To train the graph autoencoder model, the objective function is defined as:

$$\mathcal{L}_{\text{GAE}}(A, \theta_1, \phi_1) = \frac{1}{2N} \sum_{n=1}^{N} \|d^{(n)} - \hat{d}^{(n)}\|_2^2 + \lambda \|\hat{A}\|_1 \text{ s.t. } tr(e^{\hat{A} \odot \hat{A}}) - m - 1 = 0,$$

where the constraint $tr(e^{\hat{A} \odot \hat{A}}) - m - 1 = 0$ is to ensure acyclicity in the graph. After training, matrix $\hat{A}$ will be a good estimation of the adjacency matrix A.

One challenge in applying the graph autoencoder to our work is that, although the graph autoencoder can accurately estimate the adjacency matrix $\hat{A}$, it does not produce a good reconstruction of the input sample, which implies that it does not accurately estimate the function $f(\cdot)$ in the SCM. In order to generate the counterfactual data, the reconstructed sample with high fidelity is critical. Hence, we improve the graph autoencoder by adding another decoder that focuses on data reconstruction, where the trained matrix $\hat{A}$ and the encoder $E_{\theta_1}(\cdot)$ are reused in this step.

In particular, we similarly feed each sample $d^{(n)}$ to trained encoder $E_{\theta_1}(\cdot)$ to obtain the corresponding hidden representation. Then, in order to be consistent with the structural equations Eq. (1), different from [12] where the message passing operation is applied in the representation space, we first use a new variable-wise decoder $D_{\phi_1'}$ to transform the hidden representation back to the original data space, and then aggregate the message from the neighbors based on matrix $\hat{A}$. As a result, the reconstruction process of each sample is given by the following equation.

$$\hat{d}^{(n)} = \hat{A}^T D_{\phi_1'}(E_{\theta_1}(d^{(n)})).$$

The objective function is to reconstruct the input with $\hat{A}$ and θ_1 fixed:

$$\mathcal{L}_{\text{D}}(\phi_1') = \frac{1}{2N} \sum_{n=1}^{N} \sum_{i=1}^{d} \|d_i^{(n)} - \hat{d}_i^{(n)}\|_2^2.$$

After training, we obtain the approximated mapping function $\hat{f} = D_{\phi_1'} \circ E_{\theta_1}$.

Generating Counterfactual Data. Given estimated adjacency matrix $\hat{A}$ and function $\hat{f}$, for each sample $d^{(n)}$, we generate its counterfactual $d_{\text{cf}}^{(n)}$ following the Abduction-Action-Prediction process. We first intervene $s^{(n)}$ to its counterpart s and compute $\hat{f}(s)$. Then, we sort all variables in X in a topological order and

compute $\hat{x}_i^{(n)}[s]$ iteratively according to Eq. (3) where A and f are replaced by their estimators $\hat{A}$ and $\hat{f}$. Finally, we obtain $\hat{D}_{\text{cf}} = \{\hat{d}_{\text{cf}}^{(n)}\}_{n=1}^{N}$, where $\hat{d}_{\text{cf}}^{(n)} = \{s, \{\hat{x}_i^{(n)}[s] \mid i = 1 : m\}\}$.

3.4 Phase Two: Fair Anomaly Detection

We use the autoencoder as the base model for anomaly detection, which is trained to minimize the reconstruction errors of normal samples. It is worth noting that a fully-connected autoencoder model is used here which is different from the variable-wise autoencoder used in the previous section for counterfactual data generation. Meanwhile, to achieve counterfactual fairness, we leverage the idea of adversarial training to make the hidden representations derived by the autoencoder not encode the information of the sensitive variable. To this end, we develop a pre-training and fine-tuning framework to ensure the effectiveness of anomaly detection as well as counterfactual fairness. The reason for adopting the pre-training and fine-tuning training approach instead of the end-to-end training is that some counterfactual samples in $\hat{\mathcal{D}}$ could be anomalies. If we include all samples in $\hat{\mathcal{D}}$ to train the autoencoder model, the performance of anomaly detection can be damaged. Hence, we use samples in $\mathcal{D}$ to pre-train the autoencoder model. Then, during fine-tuning, we slightly update the autoencoder so that the effectiveness of anomaly detection and counterfactual fairness can be balanced. Finally, we do not use the sensitive variable and only use the non-sensitive variables X to train the anomaly detection model.

To be more specific, in the pre-training phase, given the training set with normal samples $\mathcal{D}$, an encoder first maps each sample $x^{(n)}$ to a hidden representation $z^{(n)} = E_{\theta_2}(x^{(n)})$, and then a decoder aims to reconstruct the original input from the hidden representation $\hat{x}^{(n)} = D_{\phi_2}(z^{(n)})$. The objective function is to minimize the reconstruction error of normal samples:

$$\mathcal{L}_{\text{AE}}(\theta_2, \phi_2) = \frac{1}{2N} \sum_{n=1}^{N} \|d^{(n)} - D_{\phi_2} \circ E_{\theta_2}(x^{(n)})\|_2^2.$$

After pre-training the autoencoder model, in order to achieve counterfactual fairness, we further incorporate the adversarial training strategy to further fine-tune the autoencoder model so that the hidden representation $z^{(n)}$ derived by the encoder is free of the information of the sensitive variable. To this end, for each sample $d^{(n)} = \{s^{(n)}, x^{(n)}\}$ and its counterfactual sample $\hat{d}_{\text{cf}}^{(n)} = \{s, \hat{x}_{\text{cf}}^{(n)}\}$, we first derive the hidden representations, $z^{(n)}$ and $z_{\text{cf}}^{(n)}$, respectively, by feeding them to the encoder E_{θ_2}. Then, a discriminator C_ψ is applied on $z^{(n)}$ and $z_{\text{cf}}^{(n)}$ to predict whether the hidden representations are from observed or counterfactual samples, which is a binary classification task. We parameterize the discriminator C_ψ by a multilayer neural network with the sigmoid function as the output layer and use the negative of the standard cross-entropy loss for binary classification tasks as the objective function to train the discriminator:

$$\mathcal{L}_{\text{C}}(\theta_2, \psi) = \frac{1}{N} \sum_{n=1}^{N} [\log(C_\psi(z^{(n)})) + \log(1 - C_\psi(z_{\text{cf}}^{(n)}))].$$

The discriminator is trained to accurately separate the hidden representations of observed and counterfactual samples. Meanwhile, to make the hidden representation derived from the encoder invariant to the change of sensitive attribute, the adversarial game is to train the encoder E_{θ_2} to fool the discriminator C_ψ but still be good for reconstructing the original input. As a result, the objective function can be defined as a minimax problem:

$$\min_{\theta_2,\phi_2} \max_{\psi} \mathcal{L}_{\mathrm{AE}}(\theta_2, \phi_2) + \lambda \mathcal{L}_{\mathrm{C}}(\theta_2, \psi), \tag{4}$$

where λ is a hyper-parameter to balance the reconstruction error and adversarial loss. Besides minimizing the reconstruction error $\mathcal{L}_{\mathrm{AE}}$, the encoder also tries to maximize the cross-entropy loss for the discriminator $\mathcal{L}_{\mathrm{C}}(\theta_2, \psi)$. Once the discriminator is unable to distinguish the hidden representations from factual or counterfactual data, we expect that both factual and counterfactual samples have similar reconstruction errors.

After training, the anomaly score for a new sample $d = \{s, x\}$ is computed based on the reconstruction error:

$$g(x) = \|x - D_{\phi_2} \circ E_{\theta_2}(x)\|_2^2.$$

If the anomaly score $g(x) > \tau$, where τ is a hyperparameter of the model, we label the sample as anomalous, i.e., $\hat{y} = 1$.

4 Experiments

4.1 Experimental Setup

Datasets. We conduct experiments on a synthetic dataset and two real-world datasets, Adult and COMPAS. Table 1 summarizes the statistics of three datasets.

Table 1. Statistics of datasets.

	Synthetic		Adult		COMPAS	
	Training	Test	Training	Test	Training	Test
Normal (Y=0)	12000	4000	12000	4000	2000	1283
Abnormal (Y=1)	N/A	400	N/A	800	N/A	384

Synthetic Dataset. We first build a synthetic dataset with 21 variables where we can obtain the ground truth of counterfactuals. We first randomly generate the adjacency matrix A of a causal graph using the Erdős-Rényi model [17] where one node is defined as a root node for representing the sensitive variable S.

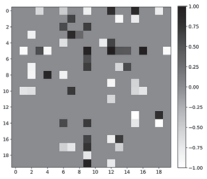


Fig. 2. Adjacency matrix A

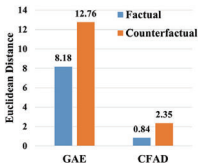
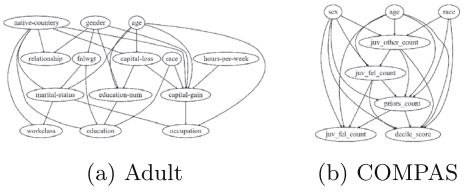


Fig. 3. Results on data generation.



(a) Adult (b) COMPAS

Fig. 4. Learned causal graphs.

Figure 2 shows the generated adjacency matrix A. The value of S is randomly generated with binarized value {-1, 1} to indicate sensitive and non-sensitive groups. Then, similar to [12], the rest 20 variables are generated based on the following data generating procedure: $X = 3A^T \cos(X + 1) + U$, where U is a standard Gaussian noise. Finally, one leaf node is selected as the decision attribute Y for determining anomalies. Specifically, for each sample, if the value of Y is greater than 0.85 quantile or smaller than 0.01 quantile, we label this sample as an anomaly, i.e., $Y = 1$. If the value of Y is between 0.3 and 0.7 quantiles, we label the sample as normal, i.e., $Y = 0$. Meanwhile, for both training and test sets, for 50% of the samples, their corresponding counterfactuals have labels that are different from the factual ones.

Adult Dataset. Adult is a real-world dataset with 14 features [5]. We treat "*gender*" as the sensitive attribute and samples with "*income* > *50k*" as anomalies. We normalize all continuous features and binarize all categorical features. Figure 4a shows the causal graph on Adult learned in Phase One of our approach. Meanwhile, as we do not know the ground truth of counterfactuals, we use the generated counterfactual samples for measuring counterfactual fairness.

COMPAS Dataset. COMPAS is another real-world dataset [4], which consists of 8 features. We consider "race" as the sensitive attribute, where "African-American" and "Caucasian" are the disadvantage and advantage groups, respectively, and treat "recidivists" as anomalies. Similar to Adult, we normalize all continuous features and binarize all categorical features. Figure 4b shows the learned causal graph.

Baselines. We compare CFAD with the following baselines: 1) Principal Component Analysis (**PCA**), which is a dimensional reduction based anomaly detection approach; 2) One-class SVM (**OCSVM**), which is a one-class classification model that can detect outliers based on the observed normal samples; 3) Isolation Forest (**iForest**), which is a widely used tree-based anomaly detection model; 4) Autoencoder (**AE**), which is trained on normal data and widely-used for anomaly detection based on the deep autoencoder structure; 5) Deep Clustering based Fair Outlier Detection (**DCFOD**) [15], which adopts the adversarial

training to achieve the group fairness in anomaly detection; 6) Fairness-aware Outlier Detection (**FairOD**) [14], which is also an autoencoder-based anomaly detection approach with fairness regularizers.

Evaluation Metrics. We evaluate the performance of anomaly detection based on Area Under Precision-Recall Curve **(AUC-PR)**, Area Under Receiver Operating Characteristic Curve **(AUC-ROC)**, and **Macro-F1**. We evaluate counterfactual fairness by computing the **changing ratio** of the samples whose detection outcomes are different from those for their corresponding counterfactuals, i.e., $changing\_ratio = \frac{\sum_{n=1}^{N} \mathbb{1}[\hat{y}^{(n)} \neq \hat{y}_{\text{cf}}^{(n)}]}{N}$, where $\mathbb{1}[\cdot]$ is the indicator function.

Implementation Details. Regarding baselines, we use Loglizer [7] to evaluate PCA, OC-SVM, and iForest. We implement FairOD and DCFOD based on public source code [15]. By default, the threshold τ for anomaly detection is set based on the 0.95 quantile of reconstruction errors (AE, FairOD, and CFAD) or distance to the normal center (DCFOD) in the training set. Our code on CFAD is available online[1].

4.2 Experimental Results

Counterfactual Data Generation. We first evaluate the performance of counterfactual data generation in the synthetic dataset by comparing CFAD with GAE [12] in terms of Euclidean distance between the generated and ground-truth samples. As shown in Fig. 3, on the factual data, CFAD achieves a much lower reconstruction error compared with GAE. More importantly, for counterfactual data generation, CFAD is much better compared with GAE. It indicates that by incorporating a variable-wise decoder $D_{\phi_1'}$ for data generation, CFAD can generate counterfactual samples with high fidelity.

Table 2. Anomaly detection on synthetic and real datasets with threshold $\tau = 0.95$. For AUC-PR, AUC-ROC, and Macro-F1, the higher the value the better the effectiveness; for Changing Ratio, the lower the value the better the fairness.

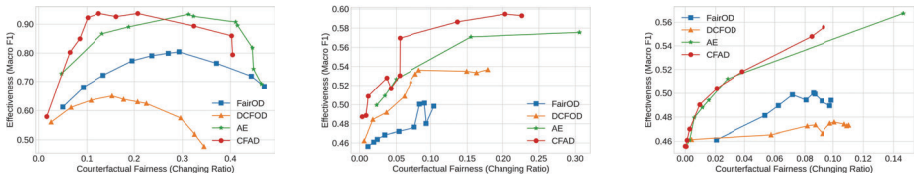
Method	Synthetic Dataset				Adult Dataset				COMPAS Dataset			
	AUC-PR	AUC-ROC	Macro-F1	Changing Ratio	AUC-PR	AUC-ROC	Macro-F1	Changing Ratio	AUC-PR	AUC-ROC	Macro-F1	Changing Ratio
PCA	0.992	0.999	0.908	0.478	0.238	0.582	0.476	0.261	0.365	0.642	0.595	0.268
OC-SVM	0.776	0.953	0.477	0.399	0.282	0.638	0.482	0.285	0.337	0.593	0.488	0.376
iForest	0.190	0.693	0.570	0.271	0.312	0.658	0.570	0.279	0.311	0.567	0.564	0.415
AE	0.957	0.996	0.883	0.461	0.349	0.640	0.608	0.590	0.344	0.616	0.581	0.407
DCFOD	0.383	0.832	0.721	0.212	0.249	0.623	0.533	0.071	0.260	0.569	0.466	0.067
FairOD	0.580	0.873	0.689	0.261	0.222	0.621	0.531	0.131	0.265	0.548	0.493	0.068
CFAD	0.947	0.996	0.930	0.199	0.319	0.589	0.576	0.057	0.314	0.596	0.539	0.049

Anomaly Detection. We further evaluate the performance of anomaly detection in terms of effectiveness as well as fairness. Table 2 shows the evaluation results. We report the mean value after five runs.

[1] https://github.com/hanxiao0607/CFAD.

Synthetic Dataset. CFAD can well balance the effectiveness and fairness in anomaly detection with high AUC-PR, AUC-ROC, and Macro-F1 and a low changing ratio. AE can achieve good performance on anomaly detection, but its changing ratio is high. DCFOD and FairOD, which achieve group fairness in anomaly detection, both have relatively low changing ratios, but their effectiveness in anomaly detection is not satisfactory.

Real Datasets. We have similar observations on the Adult and COMPAS datasets. CFAD achieves good performance in both effectiveness and fairness. For baselines that have no fairness component, their performance is good in terms of the effectiveness in anomaly detection, but they all have high changing ratios. Similarly, although DCFOD and FairOD have relatively low changing ratios, their effectiveness is much worse than other approaches.



(a) Synthetic (b) Adult (c) COMPAS

Fig. 5. Trade-off between effectiveness and fairness.

Trade-off Between Effectiveness and Fairness. We further investigate the trade-off between effectiveness and fairness by varying the threshold as different quantiles of reconstruction errors or distances in the training set. We plot the effectiveness and fairness of each threshold setting of four approaches CFAD, AE, DCFOD, and FairOD in Fig. 5, where the x-axis is the changing ratio (counterfactual fairness), the y-axis indicates the Macro-F1 score (effectiveness), and each dot in the line indicates the result from one threshold. The dots from right to left indicate the performance based on quantiles including $\{0.8, 0.85, 0.9, 0.95, 0.97, 0.98, 0.99, 0.995, 0.999\}$. Ideally, we expect an anomaly detection model can achieve a high Marco-F1 score with a low changing ratio, which is the top left corner of the figure.

As shown in Fig. 5, CFAD performs best when the effectiveness trades off with fairness, as CFAD is closest to the top left corner of the figure. Specifically, on the Synthetic dataset, CFAD achieves much higher Macro-F1 values (effectiveness) with similar changing rates (fairness) compared with DCFOD and FairOD. Meanwhile, for most of the thresholds chosen based on quantiles, CFAD has higher Macro-F1 and lower changing ratios compared with AE. On the Adult and COMPAS datasets, CFAD can have higher Macro-F1 values and lower changing ratios compared with DCFOD and FairOD.

5 Conclusions

In this work, we have developed a counterfactually fair anomaly detection (CFAD) framework, which is able to effectively detect anomalies and also ensure counterfactual fairness. The core idea of CFAD is to generate counterfactual data governed by a learned causal structure based on the proposed graph autoencoder model. Then, by using a vanilla autoencoder as the anomaly detection model, an adversarial training strategy is adopted to ensure the representations derived by the autoencoder without the information of sensitive attributes. After that, counterfactual fairness is achieved by having similar reconstruction errors for both factual and counterfactual samples. The experimental results show that CFAD can achieve counterfactually fair anomaly detection while well-balancing the trade-off between effectiveness and fairness.

Acknowledgement. This work was supported in part by NSF 1910284 and 2103829.

References

1. Almanza, M., Epasto, A., Panconesi, A., Re, G.: k-clustering with fair outliers. In: WSDM. ACM (2022)
2. van Breugel, B., Kyono, T., Berrevoets, J., van der Schaar, M.: DECAF: generating fair synthetic data using causally-aware generative networks. In: NeurIPS (2021)
3. Deepak, P., Abraham, S.S.: Fair Outlier Detection. In: WISE (2020)
4. Dressel, J., Farid, H.: The accuracy, fairness, and limits of predicting recidivism. Sci. Adv. **4**(1), eaao5580 (2018)
5. Dua, D., Graff, C.: UCI machine learning repository (2017). http://archive.ics.uci.edu/ml
6. Edwards, H., Storkey, A.: Censoring representations with an adversary. arXiv:1511.05897 (2016)
7. He, S., Zhu, J., He, P., Lyu, M.R.: Experience report: System log analysis for anomaly detection. In: ISSRE. IEEE (2016)
8. Kilbertus, N., Rojas-Carulla, M., Parascandolo, G., Hardt, M., Janzing, D., Schölkopf, B.: Avoiding discrimination through causal reasoning. In: NIPS (2017)
9. Kusner, M.J., Loftus, J.R., Russell, C., Silva, R.: Counterfactual Fairness. In: NeurIPS (2018)
10. Madras, D., Creager, E., Pitassi, T., Zemel, R.: learning adversarially fair and transferable representations. arXiv:1802.06309 (2018)
11. Nabi, R., Shpitser, I.: Fair inference on outcomes. In: AAAI (2018)
12. Ng, I., Zhu, S., Chen, Z., Fang, Z.: A graph autoencoder approach to causal structure learning. In: NeurIPS Workshop on Machine Learning and Causal Inference for Improved Decision Making (2019)
13. Pearl, J.: Causality, 2nd edn. Cambridge University Press, Cambridge (2009)
14. Shekhar, S., Shah, N., Akoglu, L.: FairOD: fairness-aware outlier detection. In: AIES. AAAI/ACM (2021)
15. Song, H., Li, P., Liu, H.: Deep clustering based fair outlier detection. In: SIGKDD. ACM (2021)
16. Zhang, H., Davidson, I.: Towards fair deep anomaly detection. In: FAcct. ACM (2021)
17. Zheng, X., Aragam, B., Ravikumar, P.K., Xing, E.P.: DAGs with no tears: continuous optimization for structure learning. In: NeurIPS (2018)