On Root Cause Localization and Anomaly Mitigation through Causal Inference

Xiao Han
Utah State University
Logan, UT, USA
xiao.han@usu.edu

Lu Zhang
University of Arkansas
Fayetteville, AR, USA
lz006@uark.edu

Yongkai Wu
Clemson University
Clemson, SC, USA
yongkaw@clemson.edu

Shuhan Yuan
Utah State University
Logan, UT, USA
Shuhan.Yuan@usu.edu

ABSTRACT

Due to a wide spectrum of applications in the real world, such as security, financial surveillance, and health risk, various deep anomaly detection models have been proposed and achieved state-of-the-art performance. However, besides being effective, in practice, the practitioners would further like to know what causes the abnormal outcome and how to further fix it. In this work, we propose RootCLAM, which aims to achieve Root Cause Localization and Anomaly Mitigation from a causal perspective. Especially, we formulate anomalies caused by external interventions on the normal causal mechanism and aim to locate the abnormal features with external interventions as root causes. After that, we further propose an anomaly mitigation approach that aims to recommend mitigation actions on abnormal features to revert the abnormal outcomes such that the counterfactuals guided by the causal mechanism are normal. Experiments on three datasets show that our approach can locate the root causes and further flip the abnormal labels.

CCS CONCEPTS

• **Security and privacy → Intrusion/anomaly detection and malware mitigation**; • **Computing methodologies → Machine learning**.

KEYWORDS

Root Cause Analysis; Anomaly Mitigation; Causal Inference

ACM Reference Format:
Xiao Han, Lu Zhang, Yongkai Wu, and Shuhan Yuan. 2023. On Root Cause Localization and Anomaly Mitigation through Causal Inference. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management (CIKM '23), October 21–25, 2023, Birmingham, United Kingdom.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3583780.3614995

INTRODUCTION

Deep anomaly detection models have been used to automatically detect a variety of anomalies, such as bank fraud detection. As many anomaly detection tasks are high-stakes decision-making tasks, there is a growing demand for the transparency of the detection results, especially, for the outcomes as anomalies [16]. For example, if a credit card transaction is declined by an automated decision-making algorithm due to the potential fraudulent features of this transaction, the user would like to know which features lead to the transaction decline and how to avoid such a situation in the future.

To answer the question of which features lead to abnormal outcomes, several interpretable anomaly detection approaches are proposed based on the idea of feature attributions [12, 13, 23]. Although feature attribution-based approaches can highlight the abnormal features, they ignore the dependencies between different features, whereas some abnormal features may be caused by other upstream abnormal features. For example, if a loan application is declined, a feature attribution-based approach may highlight the low income and low savings as abnormal features. However, the actual situation may be that low savings are caused by low income, and low income is the root cause of the loan application decline. Identifying the root cause of the anomaly can provide insights into the anomaly as well as efficient actions to fix the anomaly.

In this paper, we study the problem of anomaly mitigation facilitated by the root cause localization. We propose a framework named Root Cause Localization and Anomaly Mitigation (RootCLAM). The framework consists of two phases. In the first phase, we attempt to identify and localize the features that are the root cause of the anomaly for each abnormal instance. Then, in the second phase, we answer the question of how to fix the abnormal outcome by finding the algorithmic recourse [5] on the abnormal outcome. Traditional algorithmic recourse may perform actions on any feature in order to improve or flip the outcome. However, in the context of anomaly mitigation, it is more natural to perform recourse actions on the root cause features as not all features are equally important for mitigation. Thus, our framework aims to find the algorithmic recourse by only using root cause features.

Developing RootCLAM faces several challenges. First, despite several root cause analysis approaches proposed for anomalies in time series data [2, 3, 14, 24], the research on the root cause analysis of the tabular data is still limited, especially in the context of anomaly detection. Second, to perform appropriate recourse actions on root cause features to change the outcome, one needs to quantitatively analyze the causal connection between these actions and the outcome [1, 25]. Last but not least, algorithmic recourse is known as

CIKM '23, October 21–25, 2023, Birmingham, United Kingdom

ACM ISBN 979-8-4007-0124-5/23/10.
https://doi.org/10.1145/3583780.3614995

providing a counterfactual interpretation of the outcome. However, existing counterfactual inference techniques [8, 15] usually assume that the causal connections between features can be described by linear equations, which may not be realistic in practical situations.

To address these challenges, we first assume that the data generation is governed by a Structural Causal Model (SCM) [18], and treat the root cause as external interventions on specific features. As a result, the root cause localization is to identify features that are impacted by the external intervention. Then, we formulate the algorithmic recourse for anomaly mitigation as soft interventions [4] in order to represent the causal effect of recourse actions on the outcome as a differentiable expression. Based on that, we develop a continuous optimization-based iterative algorithm that follows the causal graph topological order to compute the actions such that the outcome will be flipped to normal by performing the actions. In addition, we leverage the causal graph autoencoder to conduct counterfactual inference. In particular, we adopt the Variational Causal Graph Autoencoder (VACA) [21] which can deal with non-linear SCMs by leveraging graph neural networks. Finally, anomaly mitigation is achieved as the outcome of the algorithmic recourse based on root cause features.

For empirical evaluation, we conduct experiments on several semi-synthetic and real-world datasets. The results show that our method can produce the largest flipping ratio regarding the anomaly detection outcomes while requiring the minimum perturbation compared with the baseline methods.

PRELIMINARY

Structural Causal Model (SCM)

We adopt Pearl's Structural Causal Model (SCM) [18] as the prime methodology for computing counterfactuals. Throughout this paper, we use the upper/lower case alphabet to represent features/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 features in U.*
2) *V is a set of endogenous variables/features that are determined by variables in $U \cup V$.*
3) *F is a set of 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, where a set of features $X_{\mathrm{PA}_i} \subseteq V \setminus \{X_i\}$ are called the parents of X_i.*

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 $\rightarrow$.

Inferring causal effects in the SCM is facilitated by the intervention. The hard intervention forces some variable $X \in V$ to take a certain value x. For an SCM $\mathcal{M}$, intervention $do(X = x')$ is equivalent to replacing original function in F with $X = x'$. The soft intervention, on the other hand, forces some variables to take a certain functional relationship in responding to some other variables [4]. The soft intervention substitutes equation $x = f(x_{\mathrm{PA}}, u)$ with a new equation. After the intervention, the distributions of all features that are the descendants of X may be changed, called the interventional distributions.

Counterfactuals

Counterfactuals are about answering questions such as for two features $X, Y \in V$, whether Y would be y had X been x' given that X is equal to x in the factual instance. Symbolically we denote this counterfactual instance as $x_{do(X=x')}|x$. The counterfactual question involves two worlds, the factual world and the counterfactual world, and cannot be answered directly by the do-operator. When the complete knowledge of the SCM is known, the counterfactual can be computed by the Abduction-Action-Prediction process [18]:

1) Abduction: Beliefs about the world are updated by taking into account all evidence given in the context. Formally, update the probability $P(u)$ to $P(u|e)$.
2) Action: Perform do intervention, $do(X = x')$, to reflect the counterfactual assumption, and a new causal model is created by interventions $\mathcal{M}' = \mathcal{M}_{do(X=x')}$.
3) Prediction: Counterfactual reasoning occurs over the new model $\mathcal{M}'$ using updated knowledge $P(u|e)$.

Causal Graph Autoencoder

A causal graph autoencoder is a type of deep learning model that aims to learn a latent representation of the data that captures the underlying causal relationships among variables given a causal graph. In this paper, we adopt the Variational Causal Graph Autoencoder (VACA) [21] which can accurately approximate the interventional and counterfactual distributions on diverse SCMs and can deal with non-linear causal relationships. The VACA consists of an adjacency matrix A of the causal graph, a decoder $p_\zeta(\mathbf{x}|\mathbf{z}, A)$ which is a graph neural network (GNN) that takes as input a set of latent variables $\mathbf{z}$ and the matrix A and outputs the likelihood of $\mathbf{x}$, and an encoder $q_\xi(\mathbf{z}|\mathbf{x}, A)$ which is another GNN that takes $\mathbf{x}$ and A as input and outputs the latent variables of $\mathbf{z}$. The VACA is trained to fit the observational distribution.

To compute the counterfactual instance of a factual instance $\mathbf{x}$ under the hard intervention $do(X_i = x')$, the VACA first computes the distribution of $\mathbf{z}$ by feeding the factual instance $\mathbf{x}$ and A into encoder $q_\xi(\mathbf{z}|\mathbf{x}, A)$. Then, the VACA constructs the intervened instance $\bar{\mathbf{x}}$ by replacing the value of x_i in the factual instance $\mathbf{x}$ with the intervened value x', as well as the intervened matrix $\bar{A}$ by removing all incoming edges of node X_i in the causal graph. The VACA feeds $\bar{\mathbf{x}}$ and $\bar{A}$ into encoder $q_\xi(\mathbf{z}|\mathbf{x}, A)$ to compute the intervened distribution of the latent variables, denoted by $\bar{\mathbf{z}}$. Next, the VACA removes the latent variable in $\mathbf{z}$ that corresponds to x_i, i.e., z_i, and replaces it with $\bar{z}_i$ in $\bar{\mathbf{z}}$ to obtain a new vector $\tilde{\mathbf{z}}$. This step is to perform the intervention in the hidden space that is equivalent to performing the intervention in the original feature space. Finally, $\tilde{\mathbf{z}}$ and $\bar{A}$ are fed into the decoder $p_\zeta(\mathbf{x}|\mathbf{z}, A)$ to compute the counterfactual instance.

ROOT CAUSE LOCALIZATION AND ANOMALY MITIGATION (ROOTCLAM)

In this section, we introduce RootCLAM, which is a two-phase framework that recommends anomaly mitigation actions to flip abnormal outcomes to normal ones. When an anomaly is detected, root cause localization is first to identify the abnormal features leading to the abnormal outcome. Then, anomaly mitigation is to further find actions on an anomaly to flip the prediction from a

fixed anomaly detection model with the consideration of the root cause of the anomaly. Figure 1 illustrates our framework for root cause analysis and anomaly mitigation.

Problem Formulation

We start with formulating the problem for root cause localization and anomaly mitigation. Consider an unlabeled dataset $\mathcal{X} = \{\mathbf{x}^{(n)}\}_{n=1}^{N}$ consisting of both normal and abnormal samples, where $\mathbf{x} = [x_1, ..., x_i, ...x_d] \in \mathbb{R}^d$ is a sample with d features. We adopt a score-based anomaly detection model $g(\cdot) : \mathcal{X} \to \mathbb{R}$, which labels abnormal samples if $g(\mathbf{x}) > \tau$, where τ indicates the threshold. By applying $g(\cdot)$ on $\mathcal{X}$, we can obtain a set of detected abnormal samples $\hat{\mathcal{X}}^-$. Our goal is to find the root causes of the anomalies as well as the actions to fix them.

Root Cause. First, we need to define the root cause. Assume that the normal data are generated from a Structural Causal Model (SCM) given as follows:

$$\forall x_i \in \mathcal{X}, \quad x_i \sim P(x_i | \{x_j, \forall j \in X_{\mathrm{PA}_i}\}, u_i).$$

We consider that any anomaly is caused by certain external interventions on some features in the SCM. Thus, the root causes of anomalies are defined as follows.

Definition 2. *Given any anomaly $\mathbf{x} \in \hat{\mathcal{X}}^-$, the root causes of $\mathbf{x}$ is a set of features $\mathcal{I}$ that receives external interventions.*

We do not assume the type of the SCM, but we do assume that the external intervention on a feature x_i can be represented as an intervention on the exogenous variable u_i. It is straightforward to show that this assumption holds for some common types of SCM, such as the additive noise model where the structural function is a linear combination of X_{PA_i} and u_i. Based on this assumption, we treat the root cause as the feature where the intervention leads to a significant change in its distribution.

Definition 3. *(Root cause). Given an anomaly $\mathbf{x} \in \hat{\mathcal{X}}^-$, the root cause of $\mathbf{x}$ is a set of features $\mathcal{I}$ that receives an external intervention leading to a significant change in the marginal distributions of exogenous variables $P(u_{\mathcal{I}})$.*

It is worth noting that the features that are not the root cause may still exhibit abnormal behaviors. For example, suppose that a feature x_i receives an external intervention, meaning that the probability distribution $P(u_i)$ is changed to a different distribution $P'(u_i)$. Meanwhile, the change in x_i may propagate through the SCM, influencing another downstream feature x_j, where x_j is a child of x_i defined by SCM. As a result, the value of x_j may also become abnormal due to the propagation from the external intervention on x_i through the SCM, despite being a non-root cause.

Anomaly Mitigation. Once the anomaly is detected, one can perform recourse actions to modify the values of certain features to change the abnormal sample to a normal one. As it is natural to modify root cause features only, we consider the problem of anomaly mitigation that asks to find a minimum perturbation on the root cause features $i \in \mathcal{I}$ of a sample to flip the label made by $g(\cdot)$. From the causal perspective, the recourse actions can be modeled as soft interventions. Specifically, define the anomaly mitigation action as a parameter vector $\theta = [\theta_1, ..., \theta_i, ...\theta_d]$ ($\theta_j = 0$ if $j \notin \mathcal{I}$). For each root cause feature x_i, we formulate the action that changes x_i to $x_i + \theta_i$ as a soft intervention. Then, the consequence of the action on a sample $\mathbf{x}$ is the counterfactual instance of $\mathbf{x}$ under the soft intervention. We denote this counterfactual instance as $\mathbf{x}(\theta)$ which depends on the value of θ as well as the underlying SCM.

With the above notations, the problem of anomaly mitigation becomes to find the parameter vector θ that minimizes the cost of the changes made by the mitigation actions, subject to making the counterfactual instance $\mathbf{x}(\theta)$ a normal sample for each original abnormal sample $\mathbf{x}$. It is formulated as that the anomaly detection model should have the anomaly score less than the threshold τ by taking counterfactual sample $\mathbf{x}(\theta)$ as input, i.e., $g(\mathbf{x}(\theta)) \leq \tau$. By using the weighted L2 norm of the action values θ as the quantitative cost measure, given by $\|\mathbf{c} \cdot \theta\|_2$ where $\mathbf{c}$ is a cost vector for describing costs of revising all root cause features ($c_j = 1$ if $j \notin \mathcal{I}$), the problem is finally formulated as

$$\arg\min_{\theta} \|\mathbf{c} \cdot \theta\|_2 \quad \text{s.t. } \forall \mathbf{x} \in \hat{\mathcal{X}}^-, g(\mathbf{x}(\theta)) \leq \tau \tag{1}$$

Solving the optimization problem in Eq. (1) is not trivial. When an action is performed to change x_i to $x_i + \theta_i$, the downstream features that are causally related will also be affected by this action. For example, changing an annual salary usually has an impact on the account balance. Thus, the counterfactual instance $\mathbf{x}(\theta)$ is not simply equal to $\mathbf{x} + \theta$. Ignoring causal relationships will lead to incorrect action recommendations, and counterfactual inference is needed to derive the accurate consequence of actions. Next, we address this challenge by leveraging the Variational Causal Graph Autoencoder (VACA), a state-of-the-art causal graph autoencoder.

Root Cause Localization

Based on the Definition 3, the idea of localizing the root cause features is to examine the exogenous variables of all features. If an exogenous variable u_i does not follow the regular distribution $P(u_i)$ learned from the normal data, the exogenous variable should be the root cause of an anomaly that receives the external intervention. In this way, even if a feature is abnormal, as long as its exogenous variable follows a similar distribution as the normal data, we treat it as a non-root cause feature and attribute the abnormal behavior to be propagated from its parents.

To this end, we leverage VACA to learn the distribution of the exogenous variable. As mentioned earlier, VACA contains an encoder that maps the features to a hidden exogenous representation, i.e., $\mathbf{z} \sim q_\xi(\mathbf{z}|\mathbf{x}, A)$, as well as a decoder that maps the hidden exogenous representation back to the feature space, i.e., $\mathbf{x} \sim p_\zeta(\mathbf{x}|\mathbf{z}, A)$. The decoder and encoder are implemented as graph neural networks, and all computations follow the structural equation specified by the SCM. For each feature $x_i \in \mathbf{x}$, the purpose of $z_i \in \mathbf{z}$ is to capture the information of x_i that cannot be explained by its parents. Thus, z_i plays a similar role to u_i, which implies that we can examine the distribution of $\mathbf{z}$ to localize the root causes.

Specifically, after training the VACA on normal data, for each sample $\mathbf{x} \in \hat{\mathcal{X}}^-$, we first derive the hidden variable $\mathbf{z}$ based on the encoder of VACA and further calculate the cumulative probability $\Phi(z_i)$ for each exogenous variable based on the distribution fitted from normal data. To identify the root cause features with significant changes in exogenous variables, we set a threshold π for the percentage of the values (in our experiments we use $\pi = 0.125$).

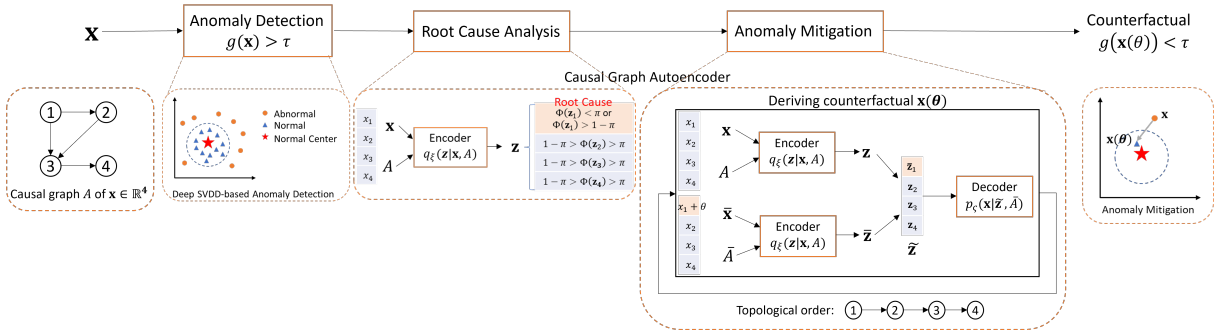


Figure 1: The pipeline to achieve root cause identification and anomaly mitigation.

If $\Phi(z_i)$ is smaller than π or larger than $1-\pi$, we consider the feature x_i as a potential root cause. As there can be multiple root cause features in a particular sample, we examine the exogenous variables of all features and get a set of root cause features $\mathcal{I}$.

Causal Graph Autoencoder-based Anomaly Mitigation

For each sample in $\hat{\mathcal{X}}^-$, after getting the root causes, we further want to flip the abnormal outcome with minimum actions on root cause features $\mathcal{I}$. The challenge in solving Eq. (1) is how to compute counterfactual instance $\mathbf{x}(\theta)$ and solve θ as a continuous optimization problem. We propose to perform the Abduction-Action-Prediction process to conduct the counterfactual inference based on the VACA. Since we perform actions on all features, we consider an iterative Abduction-Action-Prediction process as follows:

$$
\begin{aligned}
&x_1(\theta) = \underbrace{x_1 + \theta_1}_{\text{Action}},\\
&\text{for } i = 2 \cdots d, \quad \tilde{x}_i \sim \underbrace{\int P(x_i | \{x_j(\theta), \forall j \in \mathrm{PA}_i\}, u_i)\ \underbrace{P(u_i|\mathbf{x})}_{\text{Abduction}}\ \mathrm{d}\tilde{u}_i}_{\text{Prediction}},\\
&\qquad x_i(\theta) = \underbrace{\tilde{x}_i + \theta_i}_{\text{Action}},
\end{aligned}
\tag{2}
$$

where the features are sorted in topological order. More specifically, to compute $\mathbf{x}(\theta)$, we: (1) infer the updated probability $P(u_i|\mathbf{x})$ (Abduction); (2) perform the action on each feature x_i (Action); and (3) infer the counterfactual values of the downstream features. Steps (2) and (3) are repeated until all features are modified.

There are two challenges in directly applying the VACA to our context. First, the VACA is designed to perform hard intervention where the connections from the parents to the intervened node are cut off. However, in our context, we conduct interventions on all actionable features. By using hard intervention, the parent-child relations of multiple features would be cut-off and cannot pass to downstream nodes, which totally changes the underlying SCM making the generated counterfactual instances infidelity. Therefore, we perform soft interventions on all features where the parent-child relations are preserved, which cannot be achieved by directly using the VACA to perform hard interventions on all features. Second, the hidden exogenous representation $\mathbf{z}$ produced by the encoder may not be in the same space as the features, but we want to compute the recourse on the original feature space. These two challenges mean that the action values cannot be directly added on $\mathbf{z}$ when we adopt the VACA as the causal graph autoencoder.

We address the above challenges by proposing an iterative algorithm, where each iteration performs a hard intervention on one feature following a topological order. The idea is to pass the influence of each hard intervention to the downstream nodes before performing the hard intervention on the next node in the topological order, in order to simulate how the soft intervention works. Specifically, at the ith iteration, to take the generated action on feature X_i, we perform a hard intervention on X_i as $do(X_i = x_i + \theta_i)$ to obtain the intervened instance $\bar{\mathbf{x}}$. Then, we use the VACA to compute the interventional influence on all descendants of X_i similarly to the above discussion. In this process, $\bar{\mathbf{x}}$ is first transformed to the hidden representation $\bar{\mathbf{z}}$ by the encoder. Meanwhile, the sample $\mathbf{x}$ before the intervention is also transformed to the hidden representation $\mathbf{z}$ by the encoder. Then, $\bar{z}_i$ in $\bar{\mathbf{z}}$ replaces z_i in $\mathbf{z}$ to perform the intervention in the hidden space that is equivalent to performing the intervention in the original feature space. Finally, the interventional influences of this action are transmitted to all descendants of X_i by the decoder which produces the counterfactual instance of the sample under the intervention. It is worth noting that, at the beginning of the ith iteration, the value of x_i has already been updated by taking into account the interventional influences of actions taken on ancestors of X_i. As a result, after we perform the hard intervention on all features, we obtain the counterfactual instance under the recourse.

Finally, for the sake of generalization, instead of computing θ for each instance separately, we define a function $\theta = h_\phi(\mathbf{x})$ for generating the action given $\mathbf{x}$. By integrating the score-based anomaly detection model and VACA for computing the counterfactual instance into Eq. (1) and adding the constraint to the objective as regularization, we obtain the final objective function as follows:

$$
\mathcal{L}(\phi) = \sum_{\mathbf{x}^{(n)} \in \hat{\mathcal{X}}^-} \max\left\{ g\left(\mathbf{x}^{(n)}(\theta^{(n)})\right) - \alpha\tau, 0 \right\} + \lambda \|\mathbf{c} \cdot \theta^{(n)}\|_2, \tag{3}
$$

where $\theta^{(n)} = h_\phi(\mathbf{x}^{(n)})$ indicates the action values for the sample $\mathbf{x}^{(n)}$; λ is a hyperparameter balancing the actions on the anomalies and the flipping of abnormal outcomes; α is another hyperparameter controlling how close the anomaly score of counterfactual

Algorithm 1: Training Procedure of RootCLAM for Mitigation Action Prediction

foreach $\mathbf{x} \in \hat{\mathcal{X}}^-$ **do**
  Compute root cause features $\mathcal{I}$ for $\mathbf{x}$
  $\tilde{\mathbf{x}} \leftarrow \mathbf{x}$
  foreach $i \in \mathcal{I}$ **do**
    Compute $\theta_i = h_{\phi_i}(\mathbf{x})$
    Draw $\tilde{\mathbf{z}} \sim q_\xi(\mathbf{z}|\tilde{\mathbf{x}}, A)$ // Abduction
    Compute $\bar{x}_i(\theta) = \tilde{x}_i + \theta_i$ // Action
    Replace $\tilde{x}_i$ in $\mathbf{x}$ with $\bar{x}_i(\theta)$ and get $\bar{\mathbf{x}}$
    Draw $\bar{\mathbf{z}} \sim q_\xi(\mathbf{z}|\bar{\mathbf{x}}, \bar{A})$
    Replace $\tilde{z}_i$ in $\tilde{\mathbf{z}}$ with $\bar{z}_i$ in $\bar{\mathbf{z}}$ and get $\mathbf{z}(\theta)$
    Draw $\mathbf{x}(\theta) \sim p_\zeta(\mathbf{x}|\mathbf{z}(\theta), \bar{A})$ // Prediction
    $\tilde{\mathbf{x}} \leftarrow \mathbf{x}(\theta)$
  Compute $\mathcal{L}(\phi)$ according to Eq. (3)
  Compute $\frac{\partial \mathcal{L}(\phi)}{\partial \phi}$
  Update $\phi = \phi - \eta \frac{\partial \mathcal{L}(\phi)}{\partial \phi}$
return h_ϕ

sample should be to the threshold τ. Note that the only trainable parameters in this objective function are the parameters ϕ of $h_\phi(\mathbf{x})$ for generating the action values. Eq. (3) can be minimized using off-the-shelf gradient-based optimization algorithms. The training procedure is shown in Algorithm 1.

Practical Considerations. RootCLAM assumes the availability of a causal graph about the data. In practice, the causal graphs may not be available. In this case, we can leverage the causal discovery algorithms to identify the causal relations of observational data [7].

EXPERIMENTS

Experimental Setup

Datasets. We conduct experiments on two semi-synthetic datasets and one real-world dataset. For the real-world dataset, as we do not have the ground-truth SCM, we only use it for a case study.

• **Loan** [11] is a *semi-synthetic dataset* about a loan approval scenario derived from the German Credit dataset [6], which consists of 7 endogenous features including loan amount (L), loan duration (D), income (I), savings (S), education level (E), age (A), and gender (G). The label Y indicates the probability of loan approval. We treat the samples with high approval probabilities as normal and the samples with low approval probabilities as anomalous. The structural equations for data generation are be found in [11]. Due to the space limit, we do not include the equations in this paper.

• **Adult** [21] is another *semi-synthetic dataset* about the annual income of a person derived from the real-world Adult dataset [6], which consists of 10 endogenous features of a person including age (A), education level (E), hours worked per week (H), race (R), native country (N), sex (S), work status (W), marital status (M), occupation sector (O), and relationship status (L). We use the SCM designed in the paper [21]. We follow the common settings of the adult dataset to treat samples with income less than \$50k as normal and samples with income more than \$50k as abnormal. We use the structural equations for data generation defined in [21].

Anomaly Injection. To quantify the performance of RootCLAM for root cause localization, we generate abnormal samples by revising exogenous variables of some features. Especially, to generate anomalies, we first randomly select one to four features and then change the distribution of the corresponding exogenous variables. For example, on the Loan dataset, we change the exogenous variable U_S of savings (S) from $\mathcal{N}(0, 25)$ to $\mathcal{N}(-25, 25)$. In this way, we have the ground truth of the root causes for each abnormal sample.

• **Donors** [1] is a *real-world dataset* that aims to predict whether a project on DonorsChoose.org is exciting to the business. The dataset consists of 10 endogenous features of a project, including "at least one teacher-referred donor", "fully funded", "at least one green donation", "great chat", "three or more non teacher-referred donors", "one non teacher-referred donor giving 100 plus", "donation from thoughtful donor", "great messages proportion", "teacher-referred count", "non teacher-referred count". A project must meet all of the following five criteria to be exciting: 1) was fully funded; 2) had at least one teacher-referred donor; 3) has a higher than average percentage of donors leaving an original message; 4) has at least one "green" donation; 5) has one or more of: 5.1) donations from three or more non teacher-referred donors, 5.2) one non teacher-referred donor gave more than \$100, 5.3) the project received a donation from a "thoughtful donor".

We consider exciting projects as normal and non-exciting projects as abnormal, while anomaly mitigation is to provide guidance to make the project exciting. As a real-world dataset, we do not have the ground-truth SCM, so we only use it for a case study. The causal graph used in RootCLAM is approximated by the PC algorithm [10] with some minor edits to incorporate the domain knowledge. Figure 2 shows the causal graph on Donors.

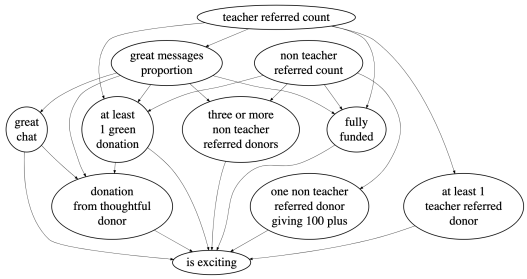


Figure 2: Learned causal graph on Donors.

Table 1 shows the statistics of three datasets. To simulate the anomaly detection scenario, we set the ratio of abnormal samples to normal samples as 1:10 in the unlabeled dataset for testing.

Anomaly Detection Models. We adopt Deep Support Vector Data Description (Deep SVDD) [20] and autoencoder-based model (AE) [19] as anomaly detection models $g(\cdot)$.

• **Deep SVDD** derives the anomaly scores of the test sample based on its distance to the center $\boldsymbol{\mu}$ of a hypersphere constructed by normal samples, i.e., $g(\mathbf{x}) = \|r(\mathbf{x}) - \boldsymbol{\mu}\|_2$, where $r(\mathbf{x})$ indicates the hidden representation of a sample $\mathbf{x}$ derived from $r(\cdot)$. Then,

[1] https://www.kaggle.com/c/kdd-cup-2014-predicting-excitement-at-donors-choose

Table 1: Statistics of three datasets.

Dataset	# of Features	Normal Dataset	Unlabeled Dataset	
			Normal	Anomalous
Loan	7	10,000	10,000	1,000
Adult	10	10,000	10,000	1,000
Donors	10	10,000	26,710	2,671

the objective function (Eq. (3)) for the recourse recommendation can be rewritten as:

$$\mathcal{L}_S(\phi)= \sum_{\mathbf{x}^{(n)} \in \hat{\mathcal{X}}^-} \max\{\|\mathbf{x}^{(n)}(\theta^{(n)}) - \boldsymbol{\mu}\|_2 - \alpha\tau, 0\} + \lambda\|\mathbf{c}\cdot\theta^{(n)}\|_2.$$

• **AE**-based anomaly detection model derives the anomaly scores of samples based on the reconstruction errors of an autoencoder that is trained by normal samples, i.e., $g(\mathbf{x}) = \|\mathbf{x} - \hat{\mathbf{x}}\|_2$, where $\hat{\mathbf{x}}$ indicates the reconstructed sample from autoencoder. Then, to provide recourse for the AE-based anomaly detection model, the objective function (Eq. (3)) can be rewritten as:

$$\mathcal{L}_{AE}(\phi) = \sum_{\mathbf{x}^{(n)} \in \hat{\mathcal{X}}^-} \max\{\|\mathbf{x}^{(n)}(\theta^{(n)}) - \widehat{\mathbf{x}^{(n)}(\theta^{(n)})}\|_2 - \alpha\tau, 0\} + \lambda\|\mathbf{c}\cdot\theta^{(n)}\|_2.$$

In our experiments, we first train Deep SVDD and AE on the normal dataset, respectively, and then apply the models on the unlabeled dataset $\mathcal{X}$ and get the corresponding $\hat{\mathcal{X}}^-$ from each model.

Baseline for Root Cause Localization. We compare RootCLAM with CausalRCA [9], a state-of-the-art approach for root cause analysis. We use the implementation in the DoWhy package [22].

Baselines for Anomaly Mitigation. To our best knowledge, there is no causal anomaly mitigation approach. We compare RootCLAM with two baselines, C-CHVAE and NaiveAM.

• **C-CHVAE** [17] can find feasible counterfactual flipping the output of classifiers, but does not consider the underlying causal relationships when generating counterfactuals. We adapt C-CHVAE by replacing classifiers with anomaly detection models.

• **NaiveAM** directly predicts the action values on all feasible features without considering the underlying causal structure. Specifically, given a set of abnormal sample $\hat{\mathcal{X}}^-$, we still train a neural network $\hat{h}_\phi(\cdot)$ to predict the action value, $\hat{\theta} = \hat{h}_\phi(\mathbf{x})$, where $\mathbf{x} \in \hat{\mathcal{X}}^-$. However, instead of generating the counterfactual samples guided by SCM, NaiveAM generates the revised samples by simply adding the action value on the original sample, i.e.,

$$\hat{\mathbf{x}}(\theta) = \mathbf{x} + \hat{\theta}. \tag{4}$$

NaiveAM is also trained on the objective function in Eq. (3) by replacing θ and $\mathbf{x}(\theta)$ with $\hat{\theta}$ and $\hat{\mathbf{x}}(\theta)$, respectively. After training, in order to evaluate whether the predicted actions can really flip the labels in the counterfactual world, on Adult and Loan datasets, we also generate the counterfactual samples based on the structural equations given $\hat{\theta}$, denoted as $\hat{\mathbf{x}}(\theta)$ (SCM).

Implementation Details. For a fair comparison, the hyperparameters of neural networks for action prediction in NaiveAM and RootCLAM are the same. We set the hyperparameters for VACA by following [21]. By default, the threshold for anomaly detection is set to 0.995 quantiles of the training samples' distances to the center (Deep SVDD) or the reconstruction errors (AE). For the intervention value prediction, we utilize a feed-forward network with structure m-2048-2048-n, where m is the input dimension and n is the number of actionable features. The costs $\mathbf{c}$ in Eq. (3) are user-specified functions for each root cause feature to represent preferences or feasibility of features changing. The cost functions can be changed according to the requirements or prior knowledge. To be fair, we use the standard deviation of each root cause feature as the cost for NaiveAM and RootCLAM. Our code is available online [2].

Experimental Results

The performance of anomaly detection. We evaluate the performance of anomaly detection in terms of the F1 score, the area under the receiver operating characteristic (AUROC), and the area under the precision-recall curve (AUPRC). Table 2 shows the anomaly detection evaluation results. In short, both AE and Deep SVDD can achieve good performance for anomaly detection, meaning that the predicted abnormal samples $\tilde{\mathcal{X}}^-$ have high accuracy. It lays a solid foundation for action prediction.

After getting the abnormal set $\tilde{\mathcal{X}}^-$ of each dataset, we then train and test the root cause localization and anomaly mitigation with the train/test split ratio of 80/20.

Table 2: Anomaly detection on the unlabeled datasets.

Dataset	AE			Deep SVDD		
	F1	AUROC	AUPRC	F1	AUROC	AUPRC
Loan	0.923	0.998	0.982	0.888	0.993	0.944
Adult	0.893	0.984	0.899	0.837	0.923	0.823
Donors	0.967	0.998	0.979	0.988	0.999	0.998

Table 3: Root cause localization on the unlabeled datasets.

		AE				Deep SVDD			
		Accu.	Pre.	Rec.	F1	Accu.	Pre.	Rec.	F1
Loan	CausalRCA	0.707	0.522	0.680	0.591	0.704	0.508	0.561	0.533
	RootCLAM	0.728	0.545	0.765	0.636	0.727	0.523	0.776	0.631
Adult	CausalRCA	0.853	0.554	0.615	0.583	0.850	0.546	0.593	0.569
	RootCLAM	0.866	0.567	0.849	0.680	0.855	0.544	0.794	0.646

The performance of RootCLAM on root cause localization. After detecting the anomalies, the next step is to identify the root causes. We further evaluate the performance of RootCLAM on root cause localization in terms of accuracy, precision, recall, and F1. As shown in Table 3, RootCLAM outperforms CausalRCA in terms of accuracy and F1 score on both datasets. Especially, RootCLAM achieves much higher recall compared with CausalRCA, which means RootCLAM can identify more root cause features.

The performance of RootCLAM on counterfactual sample generation. Generating high-fidelity counterfactual samples is a fundamental requirement for predicting high-quality actions to flip the labels. We evaluate the quality of estimated counterfactual samples in terms of the mean squared error (MSE) as well as the standard deviation of the squared error (SSE) between the true and the estimated counterfactual samples on the Loan and Adult datasets that have the ground truth structural equations for data generation. On Loan, the MSE and SSE are 3.976 and 2.266, respectively, while

[2] https://github.com/hanxiao0607/RootCLAM

Table 4: The performance of anomaly mitigation in terms of the flipping ratio and norm of action values.

	Metric		Loan			Adult		
			C-CHVAE	NaiveAM	RootCLAM	C-CHVAE	NaiveAM	RootCLAM
AE	Flipping Ratio	$\hat{\mathbf{Y}}$	1.000	1.000	0.891	0.114	0.885	0.960
		$\mathbf{Y}$	0.499	0.337	0.839	0.065	0.598	1.000
	Action Value	$\|\mathbf{c} \cdot \theta\|_2$	22.383	6.382	5.185	115.862	34.389	14.504
Deep SVDD	Flipping Ratio	$\hat{\mathbf{Y}}$	1.000	1.000	0.988	0.671	1.000	1.000
		$\mathbf{Y}$	0.496	0.847	0.963	0.586	0.595	1.000
	Action Value	$\|\mathbf{c} \cdot \theta\|_2$	17.832	13.474	5.862	63.124	69.169	29.274

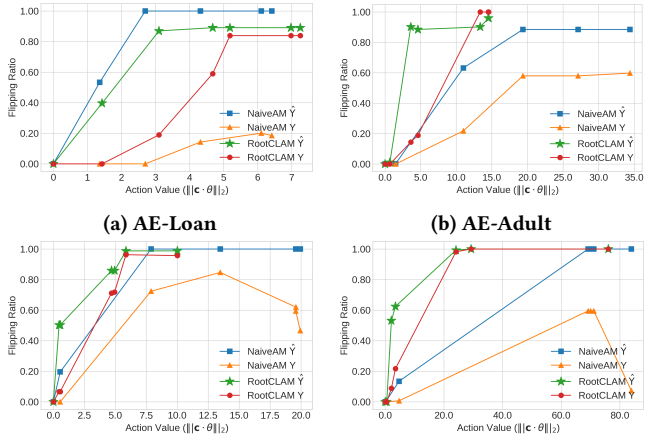
on Adult, the MSE and SSE are 3.334 and 0.900, respectively. It means RootCLAM can get good counterfactual samples.

The performance of anomaly mitigation in terms of flipping ratio. We evaluate the performance of anomaly mitigation by examining the flipping ratio that anomalies are transferred to normal through the interventions predicted by $h_\phi(\cdot)$. The flipping ratio is calculated as the fraction of the number of flipped samples over all detected anomalies. Because we would like to check whether the predicted actions can really flip the labels in the counterfactual world, given the predicted action values from RootCLAM and baselines, we also use the ground-truth structural equations to generate the counterfactual samples. We calculate the flipping ratio by considering two scenarios: 1) whether the anomaly detection model would detect the counterfactual samples as normal, denoted as $\hat{\mathbf{Y}}$; 2) whether the ground truth Y is flipping from abnormal to normal based on the ground-truth structural equations, denoted as $\mathbf{Y}$.

As shown in Table 4, on Loan and Adult datasets, both RootCLAM and NaiveAM can successfully flip almost all abnormal samples detected. However, C-CHVAE cannot get good performance on the Adult dataset. For the flipping ratio on the ground truth label $\mathbf{Y}$, RootCLAM can successfully flip most of the abnormal samples on both datasets. It means the actions predicted by RootCLAM can reverse the majority of abnormal samples to normal in the counterfactual world. However, NaiveAM and C-CHVAE cannot get good performance on flipping the ground truth label $\mathbf{Y}$. This is because both NaiveAM and C-CHVAE do not consider the underlying causal structure in the data, showing that simply revising the root cause features is not sufficient to flip the ground-truth labels.

The performance of anomaly mitigation in terms of the norm of action values. One requirement for anomaly mitigation is to conduct minimal interventions on the original samples. We further calculate the norm of action values, i.e., $\|\mathbf{c} \cdot \theta\|_2$, on the samples with successfully flipping labels. As shown in the last row of Table 4, RootCLAM makes much smaller changes on the original samples and still has higher flipping ratios on the ground truth label $\mathbf{Y}$.

The trade-off between the flipping ratio and the norm of action values. In the objective function (Eq. (3)), λ as a hyper-parameter controls the trade-off between the norm of action values and the flipping ratio in the training phase. A large λ value indicates that the model will be trained with an emphasis on minimizing the action values. Given the predicted action values, we adopt ground-truth structural equations to generate counterfactual samples and then check the flipping ratios based on anomaly detection models ($\hat{\mathbf{Y}}$) and the ground truth label $\mathbf{Y}$. Figure 3 shows the results. Each



(a) AE-Loan **(b) AE-Adult**

(c) DeepSVDD-Loan **(d) DeepSVDD-Adult**

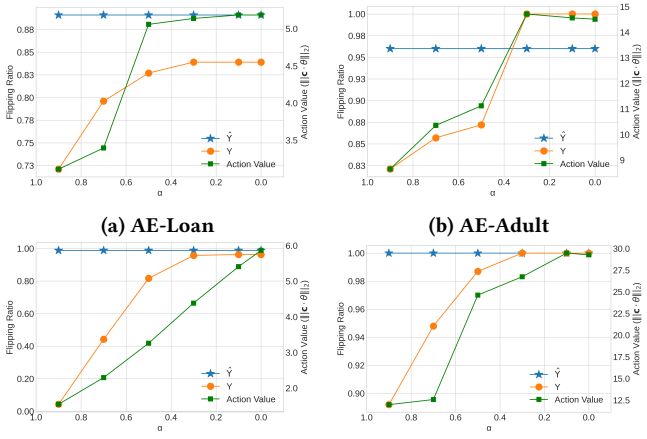
Figure 3: Trade-off between flipping ratio and action value.

point on the line from left to right indicates the result from one λ value in the set $[1, 10^{-1}, 10^{-2}, 10^{-3}, 5 \times 10^{-4}, 10^{-4}, 10^{-5}]$. Because good mitigation action predictions should be able to flip the label with minimum changes, closing to the top-left corner indicates good performance.

First, on both datasets, we can notice that in most cases, increasing the norm of action values can improve the flipping ratio. It means most of the abnormal samples can be flipped as normal ones with sufficient changes. Therefore, the key is to conduct minimum interventions on the original samples. The exception is that when having a large norm of action values on NaiveAM to flip the ground-truth label Y, we can notice the flipping ratio either does not changes or drops, which shows the importance to consider the causal relationships when applying the mitigation actions.

As shown in Figure 3a, on the Loan dataset, both NaiveAM and RootCLAM can achieve a high flipping ratio evaluated by AE with very small action values ($\|\mathbf{c} \cdot \theta\|_2 < 3$). On the other hand, in terms of flipping the ground truth label $\mathbf{Y}$, RootCLAM can achieve a much higher flipping ratio compared with NaiveAM. On the Adult dataset, as shown in Figure 3b, RootCLAM can still achieve a near 100% flipping ratio on the detected label $\hat{\mathbf{Y}}$ as well as the ground truth label $\mathbf{Y}$, while the performance of NaiveAM is poor.

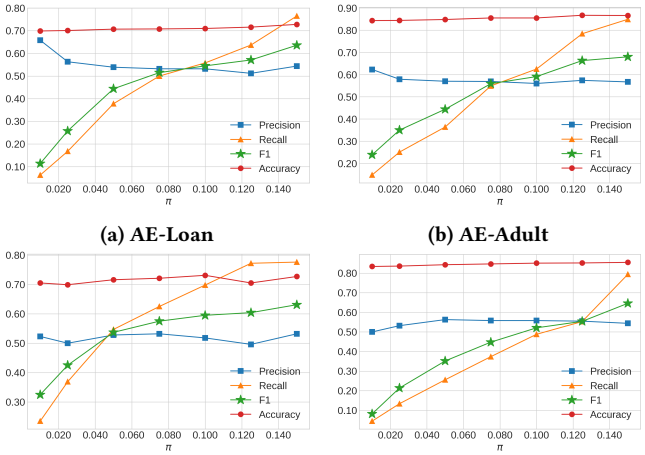
As shown in Figure 3c, on the Loan dataset, both NaiveAM and RootCLAM can achieve a near 100% flipping ratio evaluated by Deep SVDD with very small action values ($\|\mathbf{c} \cdot \theta\|_2 < 7.5$). On the other



(a) AE-Loan (b) AE-Adult

(c) DeepSVDD-Loan (d) DeepSVDD-Adult

Figure 4: Sensitivity analysis by setting various α.



(a) AE-Loan (b) AE-Adult

(c) DeepSVDD-Loan (d) DeepSVDD-Adult

Figure 5: Sensitivity analysis by setting various π.

hand, in terms of flipping the ground truth label $\mathbf{Y}$, RootCLAM can achieve a higher flipping ratio with a lower norm of action values compared with NaiveAM. On the Adult dataset, as shown in Figure 3d, RootCLAM can still achieve better performance over NaiveAM by setting various λ values for flipping both the ground truth label $\mathbf{Y}$ and detected label $\hat{\mathbf{Y}}$.

Sensitivity analysis by setting various α in the objective function (Eq. (3)) for anomaly mitigation. The hyperparameter α in Eq. (3) controls how close the anomaly scores of counterfactual samples should be to the threshold. We evaluate the flipping ratios by tuning the hyperparameter α. A smaller α indicates that the counterfactual samples should be closer to the center of normal samples (DeepSVDD) or have a smaller reconstruction error (AE).

Figures 4a to 4d have similar observations. First, in all settings, the flipping ratios in terms of detected label $\hat{\mathbf{Y}}$ are high and keep stable, which shows that a small intervention on abnormal samples can flip the detecting results. Meanwhile, by reducing the α value, we can observe the increase of the flipping ratio in terms of ground-truth label $\mathbf{Y}$ as well as the norm of action value, which means flipping the ground-truth label requires more interventions.

Sensitivity analysis by setting various π for root cause localization. Because the root cause features are identified with a small or large cumulative probability controlled by π, we evaluate the performance of root cause localization by tuning the threshold π. As shown in Figure 5, on both datasets, increasing the threshold π can increase the recall of root cause localization with a minor negative impact on the precision. The overall performance in terms of accuracy and F1 keeps improving with a large π value.

Case study. We conduct case studies to show that RootCLAM can identify root causes and recommend mitigation actions.

Loan Dataset. Table 5 shows the case study on the Loan dataset with the root cause features $\mathcal{I}$={"loan amount", "loan duration"}. For the semi-synthetic Loan dataset, the positive values of features usually indicate above the average, while negative values indicate below the average. The rows $\hat{\mathbf{x}}(\theta)$ (SCM) and $\mathbf{x}(\theta)$ (SCM) indicate counterfactual samples generated based on the structural equations given the predicted action values from NaiveAM and RootCLAM, respectively, while $\mathbf{x}(\theta)$ (Eq. 2) indicates the counterfactual samples generated based on our approach.

Given an abnormal sample $\mathbf{x}$, RootCLAM successfully identifies the two root cause features. Meanwhile, the mitigation actions predicted by RootCLAM indicate that reducing the loan amount (L) and the loan duration (D) can significantly improve the loan approval rate. On the other hand, although NaiveAM predicts more actions for anomaly mitigation, the odds of loan approval based on NaiveAM are still lower than the result from RootCLAM.

Adult Dataset. Table 6 shows the case study on the Adult dataset with the root cause features $\mathcal{I}$={"hours worked per week"}. In this case, the action values predicted by RootCLAM on the hours worked per week is negative, which indicates that reducing hours worked per week can make the sample normal (Income less than 50k). As we consider an income higher than 50k as abnormal, our predicted action value can indicate why an individual can have a high income, i.e., having a large number of hours worked per week. On the other hand, NaiveAM cannot ensure the success of anomaly mitigation. For the AE-based model, the income value is not changed based on the action values predicted from NaiveAM. For the DeepSVDD-based model, although the action values predicted by NaiveAM successfully reduce the income, NaiveAM predicts larger action values compared to RootCLAM.

Donors Dataset. We consider a project that is not exciting as an anomaly and aim to flip the label. Based on the definition of an exciting project, the original sample $\mathbf{x}$ in Table 7 is not exciting because this project fails to meet the requirements of at least one teacher-referred donor (F1) and at least one "green" donation (F3). In this case study, RootCLAM identifies "great messages proportion" (F8), "teacher-referred count" (F9), and "non teacher-referred count" (F10) as the root cause features. All root cause features are ancestors of exciting requirements shown in Figure 2. After getting the action values from $h_\phi(\cdot)$, we round to the nearest integer. Because we do not have the ground truth structural equations for Donors, Table 7 only shows the predicted counterfactual samples from the models.

Table 5: Case study on the Loan dataset, where "loan amount" (L) and "loan duration" (D) are root cause features.

			G	A	E	L	D	I	S	Y
		$\mathbf{x}$	0	-1.878	-0.095	2.423	5.634	-2.064	0.697	0.003
AE	NaiveAM	$\hat{\theta}$	/	/	/	-2.441	-8.159	0.217	-6.342	/
		$\hat{\mathbf{x}}(\theta)$ (SCM)	0	-1.878	-0.095	-0.017	-2.525	-1.847	-5.646	0.838
	RootCLAM	θ	/	/	/	-5.958	-11.336	/	/	/
		$\mathbf{x}(\theta)$ (Eq. 2)	0	-2.133	-0.089	-3.125	-9.154	-1.982	0.162	0.954
		$\mathbf{x}(\theta)$ (SCM)	0	-1.878	-0.095	-3.534	-11.659	-2.064	0.697	0.976
Deep SVDD	NaiveAM	$\hat{\theta}$	/	/	/	-1.655	-3.911	-1.869	-0.161	/
		$\hat{\mathbf{x}}(\theta)$ (SCM)	0	-1.878	-0.095	0.769	1.723	-3.932	0.536	0.083
	RootCLAM	θ	/	/	/	-2.157	-12.324	/	/	/
		$\mathbf{x}(\theta)$ (Eq. 2)	0	-2.134	-0.089	0.453	-7.600	-1.982	0.162	0.818
		$\mathbf{x}(\theta)$ (SCM)	0	-1.878	-0.095	0.267	-8.847	-2.064	0.697	0.850

G – 'gender', A – 'age', E – 'education level', L – 'loan amount', D – 'loan duration', I – 'income', S – 'savings'

Table 6: Case study on the Adult dataset, where "hours worked per week" (H) is the root cause feature

			R	A	N	S	E	H	W	M	O	L	I
		$\mathbf{x}$	2	36.401	1	1	5.264	52.520	1	1	2	1	60,816
AE	NaiveAM	$\hat{\theta}$	/	9.219	/	/	0.266	-4.148	/	/	/	/	/
		$\hat{\mathbf{x}}(\theta)$ (SCM)	2	45.620	1	1	5.529	48.372	1	1	2	1	60,816
	RootCLAM	θ	/	/	/	/	/	-9.672	/	/	/	/	/
		$\mathbf{x}(\theta)$ (Eq. 2)	2	38.293	1	1	5.370	44.791	1	1	2	1	45,816
		$\mathbf{x}(\theta)$ (SCM)	2	36.401	1	1	5.264	42.848	1	1	2	1	45,816
Deep SVDD	NaiveAM	$\hat{\theta}$	/	20.734	/	/	0.549	-8.573	/	/	/	/	/
		$\hat{\mathbf{x}}(\theta)$ (SCM)	2	57.135	1	1	5.813	43.947	1	1	2	1	45,816
	RootCLAM	θ	/	/	/	/	/	-12.672	/	/	/	/	/
		$\mathbf{x}(\theta)$ (Eq. 2)	2	38.293	1	1	5.370	40.217	1	1	2	1	45,816
		$\mathbf{x}(\theta)$ (SCM)	2	36.401	1	1	5.264	39.848	1	1	2	1	45,816

R – 'race', A – 'age', N – 'native country', S – 'sex', E – 'education level', H – 'hours worked per week', W – 'work status', M – 'marital status', O – 'occupation sector', L – 'relationship status', I – 'income'

Table 7: Case study on the Donors dataset

			F1	F2	F3	F4	F5	F6	F7	F8	F9	F10	Y
		$\mathbf{x}$	0	1	0	1	0	1	0	66	0	3	1
AE	NaiveAM	$\hat{\theta}$	/	/	/	/	/	/	/	34	5	-5	/
		$\hat{\mathbf{x}}(\theta)$ (Eq. 4)	0	1	0	1	0	1	0	100	5	-3	1
	RootCLAM	θ	/	/	/	/	/	/	/	26	2	5	/
		$\mathbf{x}(\theta)$ (Eq. 2)	1	1	1	1	1	1	0	100	2	6	0
Deep SVDD	NaiveAM	$\hat{\theta}$	/	/	/	/	/	/	/	34	5	-4	/
		$\hat{\mathbf{x}}(\theta)$ (Eq. 4)	0	1	0	1	0	1	0	100	5	-2	1
	RootCLAM	θ	/	/	/	/	/	/	/	26	2	5	/
		$\mathbf{x}(\theta)$ (Eq. 2)	1	1	1	1	1	1	0	100	2	6	0

F1 – 'at least 1 teacher-referred donor', F2 – 'fully funded', F3 – 'at least 1 green donation', F4 – 'great chat', F5 – 'three or more non teacher-referred donors', F6 – 'one non teacher-referred donor giving 100 plus', F7 – 'donation from thoughtful donor', F8 – 'great messages proportion', F9 – 'teacher-referred count', F10 – 'non teacher-referred count'.

For the purpose of anomaly mitigation, in order to make the project exciting, as shown in Table 7, the project host should try to have more 'great messages', increase the 'teacher-referred count' as well as 'non-teacher-referred count'. After doing such changes, as shown in the last row, some key features, such as F1, F3, and F5, are flipped to 1. Then, we can notice that the counterfactual sample will be exciting. On the other hand, because NaiveAM does not consider the causal relationships among features, NaiveAM cannot derive the impact on other features after changing the root cause features. As a result, NaiveAM cannot flip the label.

CONCLUSION

In this paper, we developed RootCLAM, a framework for root cause analysis and anomaly mitigation through causal inference. RootCLAM first learns a Variational Causal Graph Autoencoder from the normal data. Then, given an abnormal sample, RootCLAM identifies root cause features with the exogenous variables significantly deviated from the regular data. Then, RootCLAM computes mitigation actions as soft interventions on root cause features that can flip the anomalies to normal. Experiments show that RootCLAM achieves state-of-the-art performance on root cause localization and can further successfully fix most of the anomalies.

ACKNOWLEDGEMENT

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

REFERENCES

[1] Charles K. Assaad, Imad Ez-zejjari, and Lei Zan. [n. d.]. Root Cause Identification for Collective Anomalies in Time Series given an Acyclic Summary Causal Graph with Loops. In *Proceedings of the 26th International Conference on Artificial Intelligence and Statistics (AISTATS)* (2023-03-07). arXiv. https://doi.org/10.48550/arXiv.2303.04038 arXiv:2303.04038 [cs]

[2] Charles K Assaad, Imad Ez-Zejjari, and Lei Zan. 2023. Root Cause Identification for Collective Anomalies in Time Series given an Acyclic Summary Causal Graph with Loops. In *International Conference on Artificial Intelligence and Statistics*. PMLR, 8395–8404.

[3] Kailash Budhathoki, Lenon Minorics, Patrick Blöbaum, and Dominik Janzing. 2022. Causal structure-based root cause analysis of outliers. In *International Conference on Machine Learning*. PMLR, 2357–2369.

[4] Juan Correa and Elias Bareinboim. 2020. A calculus for stochastic interventions: Causal effect identification and surrogate experiments. In *AAAI*.

[5] Debanjan Datta, Feng Chen, and Naren Ramakrishnan. 2022. Framing Algorithmic Recourse for Anomaly Detection. *arXiv preprint arXiv:2206.14384* (2022).

[6] Dheeru Dua and Casey Graff. 2017. UCI Machine Learning Repository.

[7] Clark Glymour, Kun Zhang, and Peter Spirtes. 2019. Review of causal discovery methods based on graphical models. *Frontiers in genetics* 10 (2019), 524.

[8] Xiao Han, Lu Zhang, Yongkai Wu, and Shuhan Yuan. 2023. Achieving Counterfactual Fairness for Anomaly Detection. In *Advances in Knowledge Discovery and Data Mining: 27th Pacific-Asia Conference on Knowledge Discovery and Data Mining, PAKDD 2023*. Springer, 55–66.

[9] Dominik Janzing, Kailash Budhathoki, Lenon Minorics, and Patrick Blöbaum. 2019. Causal structure based root cause analysis of outliers. *arXiv preprint arXiv:1912.02724* (2019).

[10] Markus Kalisch and Peter Bühlman. 2007. Estimating high-dimensional directed acyclic graphs with the PC-algorithm. *Journal of Machine Learning Research* 8, 3 (2007).

[11] Amir-Hossein Karimi, Julius Von Kügelgen, Bernhard Schölkopf, and Isabel Valera. 2020. Algorithmic recourse under imperfect causal knowledge: a probabilistic approach. *NeurIPS* (2020).

[12] Jacob Kauffmann, Klaus-Robert Müller, and Grégoire Montavon. 2020. Towards Explaining Anomalies: A Deep Taylor Decomposition of One-Class Models. *Pattern Recognition* 101 (2020), 107198.

[13] Philipp Liznerski, Lukas Ruff, Robert A. Vandermeulen, Billy Joe Franks, Marius Kloft, and Klaus-Robert Müller. 2021. Explainable Deep One-Class Classification. In *ICLR*. arXiv:2007.01760

[14] Yuan Meng, Shenglin Zhang, Yongqian Sun, Ruru Zhang, Zhilong Hu, Yiyin Zhang, Chenyang Jia, Zhaogang Wang, and Dan Pei. 2020. Localizing failure root causes in a microservice through causality inference. In *2020 IEEE/ACM 28th International Symposium on Quality of Service (IWQoS)*. IEEE, 1–10.

[15] Ignavier Ng, Shengyu Zhu, Zhitang Chen, and Zhuangyan Fang. 2019. A graph autoencoder approach to causal structure learning. *arXiv preprint arXiv:1911.07420* (2019).

[16] Egawati Panjei, Le Gruenwald, Eleazar Leal, Christopher Nguyen, and Shejuti Silvia. 2022. A survey on outlier explanations. *The VLDB Journal* (2022), 1–32.

[17] Martin Pawelczyk, Klaus Broelemann, and Gjergji Kasneci. 2020. Learning model-agnostic counterfactual explanations for tabular data. In *Proceedings of The Web Conference 2020*. 3126–3132.

[18] Judea Pearl. 2009. *Causality* (second ed.). Cambridge University Press, Cambridge.

[19] Lukas Ruff, Jacob R Kauffmann, Robert A Vandermeulen, Grégoire Montavon, Wojciech Samek, Marius Kloft, Thomas G Dietterich, and Klaus-Robert Müller. 2021. A unifying review of deep and shallow anomaly detection. *Proc. IEEE* 109, 5 (2021), 756–795.

[20] Lukas Ruff, Robert Vandermeulen, Nico Goernitz, Lucas Deecke, Shoaib Ahmed Siddiqui, Alexander Binder, Emmanuel Müller, and Marius Kloft. 2018. Deep one-class classification. In *ICML*.

[21] Pablo Sanchez-Martin, Miriam Rateike, and Isabel Valera. 2021. VACA: Design of Variational Graph Autoencoders for Interventional and Counterfactual Queries. *arXiv preprint arXiv:2110.14690* (2021).

[22] Amit Sharma and Emre Kiciman. 2020. DoWhy: An End-to-End Library for Causal Inference. *arXiv preprint arXiv:2011.04216* (2020).

[23] John Sipple. 2020. Interpretable, Multidimensional, Multimodal Anomaly Detection with Negative Sampling for Detection of Device Failure. In *ICML*.

[24] Wenzhuo Yang, Kun Zhang, and Steven Hoi. 2023. A Causal Approach to Detecting Multivariate Time-series Anomalies and Root Causes. (2023).

[25] Wenzhuo Yang, Kun Zhang, and Steven C. H. Hoi. [n. d.]. *A Causal Approach to Detecting Multivariate Time-series Anomalies and Root Causes.* https://doi.org/10.48550/arXiv.2206.15033 arXiv:2206.15033 [cs]