Algorithmic Recourse in Abnormal Multivariate Time Series

Anonymous authors
Paper under double-blind review

Abstract

Algorithmic recourse provides actionable recommendations to alter unfavorable predictions of machine learning models, enhancing transparency through counterfactual explanations. While significant progress has been made in algorithmic recourse for static data, such as tabular and image data, limited research explores recourse for multivariate time series, particularly for reversing abnormal time series. This paper introduces Recourse in time series Anomaly Detection (RecAD), a framework for addressing anomalies in multivariate time series using backtracking counterfactual reasoning. By modeling the causes of anomalies as external interventions on exogenous variables, RecAD predicts recourse actions to restore normal status as counterfactual explanations, where the recourse function, responsible for generating actions based on observed data, is trained using an end-to-end approach. Experiments on synthetic and real-world datasets demonstrate its effectiveness.

1 Introduction

Algorithmic recourse refers to the process of offering individuals or entities actionable recommendations to change undesirable outcomes generated by predictive models (Karimi et al., 2022). In the context of counterfactual explanations, algorithmic recourse aims to identify minimal and feasible modifications to input features that can shift a model's prediction (Verma et al., 2024), which enhances transparency and interpretability in machine learning systems.

In recent years, algorithmic recourse has been extensively studied in the context of tabular (Karimi et al., 2020; Chen et al., 2020; Creager & Zemel, 2023; Gao & Lakkaraju, 2023; De Toni et al., 2024) and image data (Jung et al., 2022; Wang & Vasconcelos, 2020; Von Kügelgen et al., 2023). Despite significant advancements in these areas, there is a noticeable lack of research on algorithmic recourse for time series. Time series data, which consist of sequential observations over time, are ubiquitous in domains such as industrial systems (sensor readings), finance (stock prices), and healthcare (patient vital signs). The temporal dependencies and complex dynamics inherent in time series data present unique challenges that differ from those in static tabular or image data, as interventions made at one point in time can lead to cascading effects across subsequent periods, making it essential to accurately account for temporal and causal dependencies when designing recourse actions.

This paper studies algorithmic recourse for time series data. While our method could serve as a general framework, in this paper, we focus on abnormal multivariate time series due to their practical significance. Abnormal time series are characterized by anomalies or unexpected patterns across multiple variables. Such anomalies can indicate critical events such as system failures, fraudulent activities, or health deterioration. Providing algorithmic recourse in this context means offering explanations and actionable strategies to mitigate or prevent undesirable outcomes reflected by the anomalies, advancing anomaly detection beyond mere identification toward practical, intervention-driven solutions. For example, Figure 1 presents the usage data of two control nodes, nodes 117 and 124, in an OpenStack testbed (Nedelkoski et al., 2020). Each node's performance metrics include CPU and memory usage. When an anomaly detection model triggers an anomaly alert (indicated by the red area in the top figure), a recourse action is recommended to free up memory usage on node 124, aiming to correct the abnormal behavior. After implementing the recourse action (shown as the green area in the bottom figure), the system returns to normal, as indicated by the

dashed lines in the bottom figure. This example demonstrates how algorithmic recourse can provide quick and cost-effective remediation of the issue when an anomaly detection model detects an abnormal status.

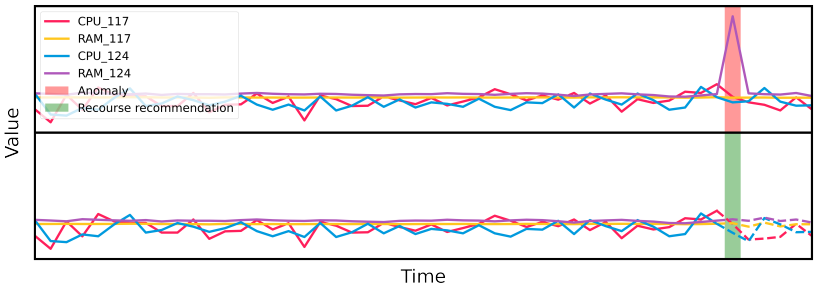


Figure 1: Recourse recommendations for flipping an abnormal status of a distribution system.

To recommend recourse in abnormal time series, we treat anomalies as external interventions on exogenous variables in a structural causal model. For instance, in a distributed computing environment, a sudden surge in incoming traffic to a particular node, such as caused by an external request spike, may lead to abnormal CPU or memory usage. Such anomaly arises not from a change in the system's internal dynamics but from an external factor influencing the exogenous inputs to the node. Methodologically, we align this formulation of anomalies with the concept of backtracking counterfactual reasoning (Von Kügelgen et al., 2023), which we adopt as the foundation of our method. Backtracking counterfactuals involve reasoning backward from an observed outcome to infer the necessary changes in exogenous variables that could have led to a different result, assuming that all causal relationships remain intact. This approach is particularly suitable for interpreting abnormal states in time series, as it allows anomalies to be traced back systematically to plausible external interventions, providing explanatory insights and actionable recourse strategies.

Specifically, we propose a neural network-based framework for algorithmic **Rec**ourse in time series **A**nomaly **D**etection (RecAD). Our method assumes that the causal relationships within the time series have been modeled using Granger causality discovery methods, such as Generalized Vector Autoregression (GVAR) (Marcinkevičs & Vogt, 2021). Given a time series with abnormal segments, RecAD addresses the question: *What is the most likely external intervention that explains these abnormal states?* Leveraging backtracking counterfactual reasoning, we identify recourse actions that mitigate anomalies by restoring abnormal states to normal ones. We formulate this task as a constrained maximum likelihood problem, where the identified actions serve both as plausible counterfactual explanations and actionable interventions. To enable end-to-end learning of a recourse function, we parameterize the recourse function with neural networks, mapping observed data to recourse actions. We adopt the Abduction-Action-Prediction procedure to model the downstream effects of interventions, deriving the counterfactual posterior through cross-world abduction. A differentiable loss function is then defined to guide the learning of the recourse function. Furthermore, we discuss practical considerations, such as handling sequence-level anomalies and addressing anomalies caused by interventions on structural equations rather than on exogenous variables. Empirical studies on two synthetic and one real-world datasets demonstrate the effectiveness of RecAD in recommending recourse actions for restoring abnormal time series.

The contribution of this paper can be summarized as follows. 1) We propose Recourse in time series Anomaly Detection (RecAD), a novel framework that generates recourse actions to correct anomalies in multivariate time series data; 2) by treating anomalies due to external interventions on exogenous variables, we apply the concept of backtracking counterfactuals to time series analysis and formulate algorithmic recourse as a constrained maximum likelihood problem using backtracking counterfactual reasoning; and 3) we addressing practical considerations and demonstrating effectiveness through empirical studies using both synthetic and real-world datasets.

2 Related Work

A time series anomaly is defined as a sequence of data points that deviates from frequent patterns in the time series (Schmidl et al., 2022). Recently, a large number of approaches have been developed for time series anomaly detection (Schmidl et al., 2022; Blázquez-García et al., 2021). However, explaining the detection results is under-exploited (Jacob et al., 2020; Kieu et al., 2022). Algorithmic recourse can provide counterfactual explanations by recommending actions to reverse unfavorable outcomes from an automated decision-making system (Karimi et al., 2022). Specifically, given a predictive model and a sample having an unfavorable prediction from the model, algorithmic recourse is to identify the minimal consequential recommendation that leads to a favorable prediction from the model. The key challenge of identifying the minimal consequential recommendation is to consider the causal relationships governing the data. Any recommended actions on a sample should be carried out via structural interventions leading to a counterfactual instance. Multiple algorithmic recourse algorithms on binary classification models have been developed (Karimi et al., 2020; 2021; von Kügelgen et al., 2022; Dominguez-Olmedo et al., 2022). Recently, algorithmic recourse for anomaly detection on tabular data has also been discussed (Datta et al., 2022). However, this study does not consider causal relationships when generating counterfactuals.

In this work, we focus on addressing the algorithmic recourse for anomaly detection in multivariate time series, considering causal relationships. Specifically, we assume that anomalies result from external interventions, distinguishing our approach from typical counterfactual reasoning. We align our approach with backtracking counterfactuals (Von Kügelgen et al., 2023), which suggests a backtracking interpretation of counterfactuals where causal laws remain unchanged in the counterfactual world, and differences from the factual world are attributed to changes in exogenous variables that alter the initial conditions. Different from a recent study (Kladny et al., 2024), which also leverages backtracking counterfactual reasoning to generate counterfactual explanations for static high-dimensional data by developing a deep generative model, our work focuses on providing backtracking counterfactual explanations in abnormal multivariate time series.

3 Preliminary

Granger Causality. Granger causality (Granger, 1969; Dahlhaus & Eichler, 2003) is a standard approach for capturing causal relationships in multivariate time series. Formally, let a stationary time-series be $\mathcal{X} = (\mathbf{x}_1, \ldots, \mathbf{x}_t, \ldots, \mathbf{x}_T)$, where $\mathbf{x}_t \in \mathbb{R}^d$ is a d-dimensional vector (e.g., d-dimensional time series from d sensors) at a specific time t. Define the true data generation mechanism in the form of

$$x_t^{(j)} := f^{(j)}(\mathbf{x}_{\leq t-1}^{(1)}, \cdots, \mathbf{x}_{\leq t-1}^{(d)}) + u_t^{(j)}, \text{ for } 1 \leq j \leq d, \tag{1}$$

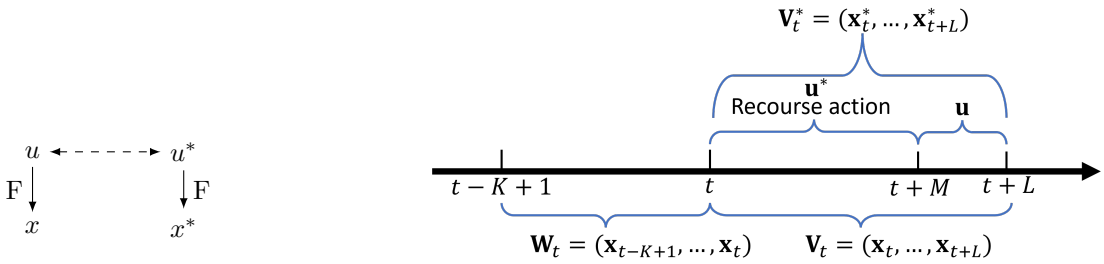
where $\mathbf{x}_{\leq t-1}^{(j)} = [\cdots, x_{t-2}^{(j)}, x_{t-1}^{(j)}]$ denotes the present and past of series j; $u_t^{(j)}$ indicates exogenous variable of time series j at time step t; $\mathcal{F} = \{f^{(1)}, ..., f^{(d)}\}$ is a set of unknown functions, and $f^{(j)}(\cdot) \in \mathcal{F}$ is the function for time series j that captures how the past values impact the future values of $\mathbf{x}^{(j)}$. Then, the time series i Granger causes j, if $f^{(j)}$ depends on $\mathbf{x}_{\leq t-1}^{(i)}$, i.e., $\exists \mathbf{x'}_{\leq t-1}^{(i)} \neq \mathbf{x}_{\leq t-1}^{(i)} : f^{(j)}(\mathbf{x}_{\leq t-1}^{(1)}, \cdots, \mathbf{x'}_{\leq t-1}^{(i)}, \cdots, \mathbf{x}_{\leq t-1}^{(d)}) \neq f^{(j)}(\mathbf{x}_{\leq t-1}^{(1)}, \cdots, \mathbf{x}_{\leq t-1}^{(i)}, \cdots, \mathbf{x}_{\leq t-1}^{(d)})$.

Granger causal discovery, i.e., learning Granger causal relationships from the observational data, has been extensively studied (Nauta et al., 2019; Tank et al., 2021; Marcinkevičs & Vogt, 2021). While multiple methods are available, in this paper, we utilize a generalized vector autoregression (GVAR) approach that can model nonlinear Granger causality (Marcinkevičs & Vogt, 2021). GVAR models the Granger causality of the t-th time step given the past K lags by

$$\mathbf{x}_t = \sum_{k=1}^{K} g_k(\mathbf{x}_{t-k})\mathbf{x}_{t-k} + \mathbf{u}_t, \tag{2}$$

where $g_k(\cdot) : \mathbb{R}^d \to \mathbb{R}^{d \times d}$ is a feedforward neural network predicting a coefficient matrix at time step $t-k$; $\mathbf{u}_t$ is the exogenous variable for time step t. The element (i, j) of the coefficient matrix from $g_k(\mathbf{x}_{t-k})$ indicates the influence of $x_{t-k}^{(j)}$ on $x_t^{(i)}$. Meanwhile, K neural networks are used to predict $\mathbf{x}_t$. Therefore, relationships

between d variables over K time lags can be explored by inspecting K coefficient matrices. The K neural networks are trained by the objective function: $\mathcal{L} = \frac{1}{T-K}\sum_{t=K+1}^{T}\|\mathbf{x}_t - \hat{\mathbf{x}}_t\|_2 + \frac{\lambda}{T-K}\sum_{t=K+1}^{T} R(\mathcal{M}_t) + \frac{\gamma}{T-K-1}\sum_{t=K+1}^{T-1}\|\mathcal{M}_{t+1} - \mathcal{M}_t\|_2$, where $\hat{\mathbf{x}}_t = \sum_{k=1}^{K} g_k(\mathbf{x}_{t-k})\mathbf{x}_{t-k}$ indicates the predicted value by GVAR; $\mathcal{M}_t := [g_K(\mathbf{x}_{t-K}) : g_{K-1}(\mathbf{x}_{t-K+1}) : \cdots : g_1(\mathbf{x}_{t-1})]$ indicates the concatenation of generalized coefficient matrices over the past the K time steps; $R(\cdot)$ is the penalty term for sparsity, such as L1 or L2 norm; the third term is a smoothness penalty; λ and γ are hyperparameters. After training, the generalized coefficient predicted by $g_k(\mathbf{x}_{t-k})$ indicates the causal relationships between time series at the time step $t-k$.



(a) Backtracking Counterfactual

(b) Algorithmic Recourse in Multivariate Time Series

Figure 2: Illustration of backtracking counterfactual and algorithmic recourse.

Counterfactual Reasoning. Counterfactual inference provides a framework for understanding causal relationships by exploring counterfactual scenarios, often framed as "what-if" questions. For example, "would the CPU utilization stay within safe operating limits if the memory usage followed its default profile?" In causal inference, counterfactuals are typically analyzed using two main approaches: interventional counterfactuals (Pearl, 2009) and backtracking counterfactuals (Von Kügelgen et al., 2023). In interventional counterfactuals, causal laws are modified while the factual and counterfactual worlds share the same background conditions. In backtracking counterfactuals, on the other hand, as shown in Figure 2a, causal laws remain consistent, but the counterfactual explanation adjusts exogenous variables to account for changes observed in the outcome. Both types of counterfactual reasoning are valuable for understanding causal mechanisms; however, backtracking counterfactuals are especially useful for explaining observed anomalies by inferring the most plausible prior conditions that led to the current state. This insight forms the basis of our method.

4 Methodology

In this section, we propose the algorithmic **Rec**ourse in time series **A**nomaly **D**etection (RecAD) method to compute algorithmic recourse actions for anomalies in multivariate time series, assuming that the causal relationships have been captured using a Generalized Vector Autoregression (GVAR) model as shown in Eq. (2). We first provide the general framework of the problem formulation based on the backtracking counterfactual. Then, we explain the detailed implementation of each component of the framework.

External intervention vs. structural intervention. Depending on how anomalies occur within a causal mechanism, they can be divided into two categories: those due to changes in exogenous variables (external interventions) and those due to changes in the structural equations (structural interventions). In this paper, we focus on anomalies caused by external interventions and briefly discuss in Section 4.5.2 how to handle anomalies resulting from structural interventions.

4.1 Problem Formulation: Algorithmic Recourse based on Backtracking Counterfactual

Following the common setting for time series anomaly detection (Audibert et al., 2020; Tuli et al., 2022), given a multivariate time series $\mathcal{X}$, we consider a local window with length K as $\mathbf{W}_t = (\mathbf{x}_{t-K+1}, ..., \mathbf{x}_t)$ and convert a time series $\mathcal{X}$ to a sequence of sliding windows $\mathcal{W} = (\mathbf{W}_K, \mathbf{W}_{K+1}, ..., \mathbf{W}_T)$. Consider a score-based anomaly detection function $s(\cdot)$ that takes a time window $\mathbf{W}_t$ as the input. If $s(\mathbf{W}_t) > \tau$, then the time step t will be labeled as in an abnormal state. Assume that an anomaly occurs at time t, which causes abnormal states in a time window $\mathbf{V}_{t+L} = (\mathbf{x}_t, ..., \mathbf{x}_{t+L})$ (assume that $L < K$) due to the ripple effect. By treating the

anomaly as an external intervention on the exogenous variable $\mathbf{u}_t$, our objective is to find the recourse action $\boldsymbol{\theta}_t$ at the time step t to reverse the abnormal states in the time window $\mathbf{V}_{t+L}$ via counterfactual reasoning. Such action can be viewed as the most plausible counterfactual explanation for the abnormal states. To simplify our discussions, we mainly focus on point anomalies. Extending our approach to sequence anomalies is straightforward and will be addressed in Section 4.5.1.

Specifically, the backtracking counterfactual computes the probability of the exogenous variables in a counterfactual world, which may differ from their factual counterparts, given both the factual and counterfactual endogenous variables. As shown in Figure 2b, in our context, the factual endogenous variables are $\mathbf{W}_t$, i.e., the observational data. We denote the backtracking counterfactual exogenous variable at time t by $\mathbf{u}_t^*$, which is obtained by performing the recourse action, i.e., $\mathbf{u}_t^* = \mathbf{u}_t + \boldsymbol{\theta}_t$, where $\mathbf{u}_t$ is the factual exogenous variable. We then denote the counterfactual endogenous variables by $\mathbf{V}_{t+L}^* = (\mathbf{x}_t^*, ..., \mathbf{x}_{t+L}^*)$, i.e., the counterfactual variants of $\mathbf{V}_{t+L}$ obtained by performing the recourse action, with the requirement that their states are restored to normal. As a result, the problem of algorithmic recourse can be formulated to maximize the likelihood of $\mathbf{u}_t^*$ conditioning on $\mathbf{W}_t$, subject to the constraints that $s(\mathbf{V}_{t'}^*) < \tau$ for $t' \in [t, t+L]$.

Problem Formulation 1 *Given local windows of time series $\mathbf{W}_t = (\mathbf{x}_{t-K+1}, ..., \mathbf{x}_t)$ and a score function $s(\cdot)$ with time step t labeled as abnormal, i.e., $s(\mathbf{W}_t) > \tau$, the algorithmic recourse aims to find the recourse action at time t, i.e., $\mathbf{u}_t^* = \mathbf{u}_t + \boldsymbol{\theta}_t$, to restore normal states in a future time window $\mathbf{V}_{t+L}^* = (\mathbf{x}_t^*, ..., \mathbf{x}_{t+L}^*)$, by solving the following constrained maximization likelihood problem:*

$$\arg\max_{\boldsymbol{\theta}_t} P(\mathbf{u}_t^* \mid \mathbf{W}_t) \quad s.t. \quad s(\mathbf{V}_{t'}^*) < \tau, \quad t' \in [t, t+L].$$

By introducing penalty terms to the negative log-likelihood to enforce constraints, we derive the revised problem formulation as follows.

Problem Formulation 2 *The algorithmic recourse aims to solve the optimization problem:*

$$\arg\min_{\boldsymbol{\theta}_t} \mathcal{L}(\boldsymbol{\theta}_t) = \sum_{t'=t}^{t+L} \max\{s(\mathbf{V}_{t'}^*) - \tau, 0\} - \lambda \log P(\mathbf{u}_t^* \mid \mathbf{W}_t). \tag{3}$$

Next, we discuss the implementation of each component of the problem formulation to enable an end-to-end learning of a recourse function.

4.2 Implementing Recourse Function

To generate effective recourse actions, we learn a recourse function $h_\phi(\cdot)$ parameterized by ϕ that maps from the observed time series to a perturbation of the exogenous variables, i.e., $\theta_t = h_\phi(W_{t-1}, \Delta_t)$, where θ_t is the proposed intervention at time step t.

The design of h_ϕ is guided by two key considerations. 1) **Temporal context** is essential for accurate intervention, as the causal dependencies in multivariate time series often span multiple time lags. Therefore, we encode the preceding time window $\mathbf{W}_{t-1}$ using a Long Short-Term Memory (LSTM) network to capture latent temporal features and long-range dependencies. 2) **Deviation from expected dynamics** is another critical signal for recourse. We define the deviation term $\Delta_t = x_t - \hat{x}_t$, where $\hat{x}_t$ is the expected value at time t based on GVAR. Intuitively, Δ_t quantifies how much the current state deviates from its normal trajectory and serves as a proxy for anomalous influence from exogenous variables. Concretely, our implementation is structured as follows:

$$\mathbf{z}_{t-1} = LSTM(\mathbf{W}_{t-1}) \quad \mathbf{z}_\Delta = FFNN(\Delta_t) \quad \boldsymbol{\theta}_t = FFNN(\mathbf{z}_{t-1} \oplus \mathbf{z}_\Delta), \tag{4}$$

where $\oplus$ denotes vector concatenation. The final feedforward network combines the latent temporal state and the deviation signal to produce the recourse action.

We empirically validate the necessity of each component through ablation studies, showing that removing either the LSTM or the deviation encoder significantly degrades recourse effectiveness.

4.3 Inferring Downstream Effect

We then infer the downstream effect $\mathbf{V}^*_{t'}$ because, based on Granger causality, the recourse action at time t leads to the counterfactual variants of the following time steps, i.e., $\mathbf{V}^*_{t+L} = (\mathbf{x}^*_t, ..., \mathbf{x}^*_{t+L})$. To compute the counterfactual variants, according to the additive noise assumption presented in Eq. (1), given backtracking counterfactual $\mathbf{u}^*_t$, we directly have $\mathbf{x}^*_t = \mathbf{x}_t - \mathbf{u}_t + \mathbf{u}^*_t = \mathbf{x}_t + \boldsymbol{\theta}_t$. However, since we perform the recourse action only at time t, it remains necessary to infer the factual exogenous variables for all $t' > t$. Thus, we proceed to utilize the Abduction-Action-Prediction (Pearl, 2009) procedure to compute $\mathbf{x}^*_{t'}$. Formally, based on the causal relationships learned by GVAR shown in Eq. (2), the Abduction-Action-Prediction (AAP) procedure to compute $\mathbf{x}^*_{t'}$ for $t' = t+1, \cdots, t+L$ can be described below.

Step 1 (Abduction):

$$\mathbf{u}_{t'} = \mathbf{x}_{t'} - \sum_{k=1}^{K} g_k(\mathbf{x}_{t'-k})\mathbf{x}_{t'-k} \tag{5}$$

Step 2 (Action): Not required

Step 3 (Prediction):

$$\mathbf{x}^*_{t'} = \sum_{k=1}^{t'-t} g_k(\mathbf{x}^*_{t'-k})\mathbf{x}^*_{t'-k} + \sum_{k=t'-t+1}^{K} g_k(\mathbf{x}_{t'-k})\mathbf{x}_{t'-k} + \mathbf{u}_{t'}. \tag{6}$$

In essence, the abduction is to derive the exogenous variable $\mathbf{u}_{t'}$ at step t' based on the observed value, then the prediction is to predict the counterfactual value $\mathbf{x}^*_{t'}$ at step t' after conducting action $\boldsymbol{\theta}_t$ at step t.

4.4 Deriving Counterfactual Posterior

According to Problem Formulation 2, the next step is to derive the counterfactual posterior $P(\mathbf{u}^*_t \mid \mathbf{W}_t)$. Following the backtracking counterfactual (Von Kügelgen et al., 2023), we apply cross-world abduction to obtain:

$$\begin{aligned} P(\mathbf{u}^*_t \mid \mathbf{W}_t) &= \sum_{\mathbf{u}'_t} P(\mathbf{u}^*_t, \mathbf{u}'_t|\mathbf{W}_t) = \sum_{\mathbf{u}'_t} \frac{P(\mathbf{u}^*_t, \mathbf{u}'_t)P(\mathbf{W}_t|\mathbf{u}^*_t, \mathbf{u}'_t)}{P(\mathbf{W}_t)} \\ &\sim \sum_{\mathbf{u}'_t} P(\mathbf{u}^*_t, \mathbf{u}'_t)P(\mathbf{W}_t|\mathbf{u}^*_t, \mathbf{u}_t) = \sum_{\mathbf{u}'_t} P(\mathbf{u}^*_t, \mathbf{u}'_t)P(\mathbf{W}_t|\mathbf{u}'_t) \\ &= \sum_{\mathbf{u}'_t} P(\mathbf{u}^*_t|\mathbf{u}'_t)P(\mathbf{W}_t, \mathbf{u}'_t) = P(\mathbf{u}^*_t|\mathbf{u}_t)I(\mathbf{W}_t(\mathbf{u}_t) = \mathbf{W}_t), \end{aligned}$$

where the second equality is based on the Bayes rule, the third equality is due to the fact that $\mathbf{u}^*_t$ and $\mathbf{W}_t$ are conditionally independent given $\mathbf{u}_t$ (obtained using d-separation in Figure 2a), $I(\cdot)$ is the indicator function, $\mathbf{u}_t$ is the factual exogenous variable obtained by abduction, and $P(\mathbf{u}^*_t \mid \mathbf{u}_t)$ is the backtracking conditional. Following the suggestion in (Von Kügelgen et al., 2023), we construct $P(\mathbf{u}^*_t \mid \mathbf{u}_t)$ based on the distance between $\mathbf{u}^*_t$ and $\mathbf{u}_t$, defined as $P(\mathbf{u}^*_t \mid \mathbf{u}_t) = 1/Z \cdot \exp\{-\|\mathbf{u}^*_t - \mathbf{u}_t\|_2\}$ where Z is a normalization constant. As a result, we have that $P(\mathbf{u}^*_t \mid \mathbf{W}_t) \sim \exp\{-\|\boldsymbol{\theta}_t\|_2\}$.

Combining all the components above, we obtain our final loss function:

$$\mathcal{L}(\phi) = \sum_{t'=t}^{t+L} \max\{s(\mathbf{V}^*_{t'}) - \tau, 0\} + \lambda\|\boldsymbol{\theta}_t\|_2, \tag{7}$$

where ϕ are the parameters of the recourse function defined in Eq. 4 and λ is a hyperparameter.

4.5 Other Considerations

4.5.1 Dealing with Sequence Anomalies

In practice, abnormal states in time series data may be caused by a sequence of anomalies. To handle this situation, we may want to compute a sequence of recourse actions to explain these abnormal states. This involves extending the single recourse action $\mathbf{u}_t^*$ in Problem Formulations 1 and 2 to a sequence of recourse actions $\mathbf{U}_t^* = (\mathbf{u}_t^*, \mathbf{u}_{t+1}^*, \cdots, \mathbf{u}_{t+M}^*)$. Correspondingly, we can modify the recourse function to adopt a sequence-to-sequence network architecture. The subsequent processes, including the Abduction-Action-Prediction procedure and backtracking counterfactual reasoning, can then be conducted similarly. One challenge is that determining the length of the anomaly sequence is not straightforward. As a best practice, we could compute the shortest sequence of recourse actions that adequately explains the abnormal states, i.e., a sequence that restores the normal states for all subsequent time steps. We describe the pseudo-code of the training process for recourse predictions with multiple abnormal time steps over time series in Appendix A.

4.5.2 Dealing with Anomalies due to Structural Intervention

So far, we have assumed that anomalies are caused by external interventions on exogenous variables, without altering the Granger causal relationships. This type of anomaly occurs, for example, when a sensor is attacked. However, there are other types of anomalies that cannot be treated as external interventions but must be considered as structural interventions. In other words, they are caused by replacing the normal equations with abnormal ones. For instance, when a system experiences a structural change due to component degradation or failure. Despite the intrinsic differences between external interventions and structural interventions, it is still possible to explain anomalies caused by structural interventions using recourse actions. This is because we can rearrange the structural equation as follows: $x_t^{(j)} = \tilde{f}^{(j)}(\mathbf{x}_{\leq t-1}^{(1)}, \cdots, \mathbf{x}_{\leq t-1}^{(d)}) + u_t^{(j)} = f^{(j)}(\mathbf{x}_{\leq t-1}^{(1)}, \cdots, \mathbf{x}_{\leq t-1}^{(d)}) + u_t^{(j)} + \epsilon_t^{(j)}$, for $1 \leq j \leq d$, where $\tilde{f}^{(j)}(\mathbf{x}_{\leq t-1}^{(1)}, \cdots, \mathbf{x}_{\leq t-1}^{(d)})$ is an abnormal function for the time series j at time t and $\epsilon_t^{(j)} = \tilde{f}^{(j)}(\mathbf{x}_{\leq t-1}^{(1)}, \cdots, \mathbf{x}_{\leq t-1}^{(d)}) - f^{(j)}(\mathbf{x}_{\leq t-1}^{(1)}, \cdots, \mathbf{x}_{\leq t-1}^{(d)})$. As a result, we can describe the anomalies as

$$x_t^{(j)} := f^{(j)}(\mathbf{x}_{\leq t-1}^{(1)}, \cdots, \mathbf{x}_{\leq t-1}^{(d)}) + u_t^{(j)} + \epsilon_t^{(j)}, \text{ for } 1 \leq j \leq d, \tag{8}$$

where anomaly term $\epsilon_t^{(j)}$ can be due to a structural intervention. However, the significance of using recourse actions to explain structural intervention is worthy of future study.

5 Experiments

5.1 Experimental Setups

Datasets. We conduct experiments on two semi-synthetic datasets and one real-world dataset. The purposes of using semi-synthetic datasets are as follows. 1) We can derive the ground truth downstream time series after the intervention on the abnormal time step based on the data generation equations. 2) We can evaluate the fine-grained performance of RecAD by injecting different types of anomalies.

Semi-synthetic datasets: 1) **Linear Dataset** (Marcinkevičs & Vogt, 2021) is a time series dataset with linear interaction dynamics. 2) **Lotka-Volterra** is a nonlinear time series model that simulates a prairie ecosystem with multiple species. For both datasets, we adopt the structural equations defined in (Marcinkevičs & Vogt, 2021), which are included in Appendix B. We also describe the anomaly injection strategies in Appendix B.

Real-world dataset: **Multi-Source Distributed System (MSDS)** (Nedelkoski et al., 2020) consists of 10-dimensional time series. The first half of MSDS without anomalies is used as a training set, while the second half, including 5.37% abnormal time steps, is used as a test set. As a real-world dataset, we cannot observe the downstream time series after the intervention. In the test phase, we use GVAR and AAP to generate the counterfactual time series for evaluation.

Table 1 shows the statistics of three datasets. Training datasets only consist of normal time series. Note that the test sets listed in Table 1 are used for evaluating the performance of anomaly detection. After detecting

Table 1: Statistics of three datasets for anomaly detection.

Dataset	Dim.	Train	Test (Anomalies %)		
			Point (External)	Seq. (External)	Seq. (Structural)
Linear	4	50,000	250,000 (2%)	250,000 (6%)	250,000 (6%)
Lotka-Volterra	20	100,000	500,000 (1%)	500,000 (3%)	500,000 (3%)
MSDS	10	146,340	146,340 (5.37%)		

the abnormal time series in the test set, for the synthetic datasets, we use 50% of abnormal time series for training RecAD and another 50% for evaluating the performance of RecAD on recourse prediction, while for the MSDS dataset, we use 80% of the abnormal time series for training RecAD and the rest 20% for evaluation.

Baselines. To our best knowledge, there is no causal algorithmic recourse approach in time series anomaly detection. We compare RecAD with the following baselines: 1) Multilayer perceptron (MLP), which is trained with the normal flattened sliding windows to predict the normal values for the next step; 2) LSTM, which can learn complex temporal dependencies in time series to make predictions for the next step; 3) Vector Autoregression (VAR) is a statistical model that used to analyze GC within multivariate time series data and predict future values; 4) Generalised Vector Autoregression (GVAR) (Marcinkevičs & Vogt, 2021) is an extension of self-explaining neural network that can infer nonlinear multivariate GC and predict values of the next step.

For all the baselines, in the training phase, we train them to predict the last value in a time window on the normal time series. In the testing phase, when a time window is detected as abnormal by USAD, indicating the last time step $\mathbf{x}_t$ is abnormal, we use baselines to predict the expected normal value in the last time step $\tilde{\mathbf{x}}_t$. Then, the recourse action values can be derived as $\boldsymbol{\theta}_t = \tilde{\mathbf{x}}_t - \mathbf{x}_t$. For the sequence anomalies, we keep using the baselines to predict the expected normal values and derive the action values by comparing them with the observed values.

Evaluation Metrics. We evaluate algorithmic recourse based on the following three metrics.

1) **Flipping ratio**, which is to show the effectiveness of algorithms for algorithmic recourse.

$$\text{Flipping Ratio} = \frac{\text{Number of flipped time steps}}{\text{All detected abnormal time steps}}$$

2) **Action cost** per multivariate time series, which is to check the efficiency of predicted actions.

$$\text{Action Cost} = \frac{\text{Total action cost}}{\text{\# of abnormal multivariate time series}},$$

where the "Total action cost" indicates the action cost ($\|\boldsymbol{\theta}_t\|_2$) to flip the abnormal data in the test set.

3) **Action step** per multivariate time series, which is to show how many action steps are needed to flip the abnormal time series.

$$\text{Action Step} = \frac{\text{Total number of action time steps}}{\text{\# of abnormal multivariate time series}}$$

5.2 Experimental Results

To implement the score function $s(\cdot)$, in our experiments, we leverage the UnSupervised Anomaly Detection for multivariate time series (USAD) (Audibert et al., 2020), a modern anomaly detection model that outputs an anomaly score for multivariate time series, which can be replaced with any other score-based anomaly detection model. We include the implementation details as well as the performance of USAD for anomaly detection in the Appendix. For all experimental results, we report the mean and standard deviation over 10 runs. Our code is available online[1].

Table 2: The performance of recourse prediction on anomalies caused by external interventions.

Dataset	Model	Point			Seq.		
		Flipping Ratio ↑	Action Cost ↓	Action Step ↓	Flipping Ratio ↑	Action Cost ↓	Action Step ↓
Linear	MLP	$0.778_{\pm 0.054}$	$8.406_{\pm 0.257}$	$1.188_{\pm 0.051}$	$0.867_{\pm 0.048}$	$22.394_{\pm 0.545}$	$2.261_{\pm 0.027}$
	LSTM	$0.807_{\pm 0.045}$	$8.383_{\pm 0.253}$	$1.170_{\pm 0.040}$	$0.878_{\pm 0.044}$	$22.439_{\pm 0.553}$	$2.248_{\pm 0.031}$
	VAR	$0.676_{\pm 0.063}$	$8.841_{\pm 0.224}$	$1.311_{\pm 0.077}$	$0.765_{\pm 0.061}$	$23.696_{\pm 0.511}$	$2.434_{\pm 0.037}$
	GVAR	$0.775_{\pm 0.053}$	$8.446_{\pm 0.249}$	$1.207_{\pm 0.052}$	$0.848_{\pm 0.051}$	$22.415_{\pm 0.547}$	$2.287_{\pm 0.029}$
	RecAD	$\mathbf{0.901_{\pm 0.035}}$	$\mathbf{8.201_{\pm 0.176}}$	$\mathbf{1.104_{\pm 0.024}}$	$\mathbf{0.944_{\pm 0.041}}$	$\mathbf{21.264_{\pm 0.947}}$	$\mathbf{2.193_{\pm 0.046}}$
Lotka-Volterra	MLP	$0.741_{\pm 0.126}$	$277.580_{\pm 117.126}$	$1.237_{\pm 0.180}$	$0.688_{\pm 0.259}$	761.550 ± 64.422	$2.195_{\pm 0.757}$
	LSTM	$0.893_{\pm 0.087}$	$313.510_{\pm 124.921}$	$1.096_{\pm 0.061}$	$0.889_{\pm 0.097}$	$1590.483_{\pm 85.126}$	$1.339_{\pm 0.145}$
	VAR	$0.558_{\pm 0.166}$	$326.967_{\pm 126.322}$	$1.57_{\pm 0.324}$	$0.504_{\pm 0.151}$	$1445.084_{\pm 189.943}$	$2.570_{\pm 0.676}$
	GVAR	$0.493_{\pm 0.332}$	$270.335_{\pm 117.223}$	$2.020_{\pm 1.049}$	$0.606_{\pm 0.266}$	$\mathbf{749.792_{\pm 105.656}}$	$2.547_{\pm 0.905}$
	RecAD	$\mathbf{0.915_{\pm 0.088}}$	$\mathbf{262.759_{\pm 99.008}}$	$\mathbf{1.085_{\pm 0.055}}$	$\mathbf{0.972_{\pm 0.016}}$	$1374.112_{\pm 343.470}$	$\mathbf{1.329_{\pm 0.192}}$

5.2.1 Evaluation Results on Synthetic Datasets

The performance of recourse prediction on anomalies caused by external interventions. Table 2 shows the performance of RecAD for recourse prediction on anomalies caused by external interventions. Note that the anomalies consist of both point and sequential anomalies. First, in all settings, RecAD can achieve the highest flipping ratios, which shows the effectiveness of RecAD in flipping abnormal behavior. Meanwhile, RecAD can achieve low action costs and action steps compared with other baselines. Because no baseline considers the downstream impact of recourse actions, they usually require more action steps to flip the anomalies.

Table 3: The performance of recourse prediction on anomalies caused by structural interventions.

Dataset	Model	Flipping Ratio ↑	Action Cost ↓	Action Step ↓
Linear	MLP	$0.884_{\pm 0.023}$	$41.387_{\pm 2.151}$	$2.359_{\pm 0.035}$
	LSTM	$0.903_{\pm 0.021}$	$42.412_{\pm 2.002}$	$2.398_{\pm 0.043}$
	VAR	$0.782_{\pm 0.035}$	$59.346_{\pm 3.285}$	$2.883_{\pm 0.062}$
	GVAR	$0.874_{\pm 0.024}$	$39.474_{\pm 2.116}$	$2.415_{\pm 0.041}$
	RecAD	$\mathbf{0.919_{\pm 0.037}}$	$\mathbf{38.917_{\pm 6.247}}$	$\mathbf{2.169_{\pm 0.206}}$
Lotka-Volterra	MLP	$0.665_{\pm 0.277}$	$\mathbf{1578.597_{\pm 49.253}}$	$2.247_{\pm 0.744}$
	LSTM	$0.890_{\pm 0.099}$	$3159.333_{\pm 173.463}$	$2.310_{\pm 0.100}$
	VAR	$0.488_{\pm 0.178}$	$3088.788_{\pm 435.034}$	$2.630_{\pm 0.640}$
	GVAR	$0.584_{\pm 0.277}$	$1846.415_{\pm 347.144}$	$2.618_{\pm 0.858}$
	RecAD	$\mathbf{0.970_{\pm 0.012}}$	$2767.819_{\pm 581.046}$	$\mathbf{1.386_{\pm 0.120}}$

The performance of recourse prediction on anomalies caused by structural interventions. We examine the performance of RecAD on anomalies caused by structural intervention. As shown in Table 3, RecAD can achieve the highest flipping ratio compared with baselines on both Linear and Lotka-Volterra datasets, indicating that the majority of anomalies caused by structural interventions can be successfully flipped. Meanwhile, RecAD can also achieve low action costs and action steps with high flipping ratios. Overall, RecAD meets the requirement of algorithmic recourse, i.e., flipping the abnormal outcome with minimum costs on anomalies caused by structural interventions.

Therefore, based on Tables 2 and 3, we can demonstrate that RecAD can provide recourse prediction on different types of anomalies in multivariate time series.

5.2.2 Evaluation Results on the MSDS Dataset

The performance of recourse prediction. Because the types of anomalies in the real-world dataset are unknown, we report the performance of recourse prediction on any detected anomalies. As shown in Table 4, RecAD achieves a flipping ratio of 0.84, much higher than all baselines. Regarding the average action cost per time series and the average action step, RecAD also outperforms the baselines by having the lowest values. This suggests that, by incorporating Granger causality, RecAD can identify recourse actions that minimize both cost and the number of steps.

[1] `https://www.tinyurl.com/RecAD2025`

Table 4: The performance of recourse prediction in MSDS.

Model	Flipping Ratio ↑	Action Cost ↓	Action Step ↓
MLP	$0.687_{\pm 0.282}$	$6.848_{\pm 2.506}$	$1.443_{\pm 0.680}$
LSTM	$0.830_{\pm 0.211}$	$6.798_{\pm 2.604}$	$1.279_{\pm 0.493}$
VAR	$0.704_{\pm 0.273}$	$6.759_{\pm 2.821}$	$1.432_{\pm 0.596}$
GVAR	$0.712_{\pm 0.211}$	$8.923_{\pm 3.258}$	$1.425_{\pm 0.466}$
RecAD	$\mathbf{0.841_{\pm 0.080}}$	$\mathbf{6.747_{\pm 1.543}}$	$\mathbf{1.249_{\pm 0.068}}$

5.3 Case Study

We further conduct case studies to show how to use the recourse action predicted by RecAD as an explanation for anomaly detection in multivariate time series.

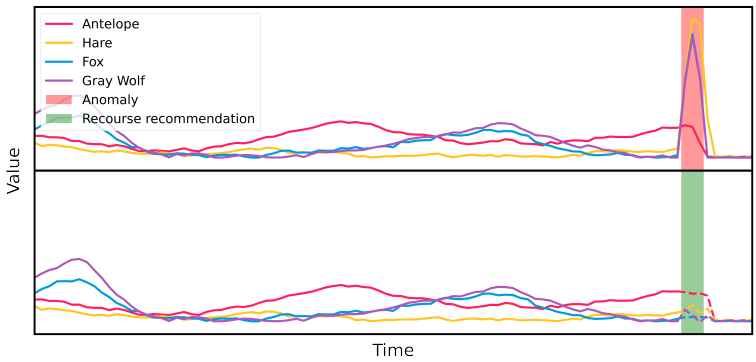


Figure 3: Recourse recommendations for intervening in an imbalanced ecosystem to restore balance.

Case study on the Lotka-Volterra dataset. Figure 3 shows a simulation of a prairie ecosystem that contains antelope, hare, fox, and gray wolf based on the Lotka-Volterra model (Bacaër, 2011), where each time series indicates the population of a species. As shown in the top figure, in most of the time steps, the numbers of carnivores (fox and gray wolf) and herbivores (antelope and hare) keep stable in a balanced ecosystem, say 0.1k-1k antelopes, 1k-10k hares, 0.1k-1k foxes, and 0.1k-1k gray wolves. After detecting abnormal behavior at a specific time step (red area in the top figure), the algorithmic recourse aims to provide recourse actions to flip the abnormal outcome. In this case, the algorithmic recourse model recommends the intervention of reducing the populations of hares, foxes, and gray wolves by 100.1k, 9.3k, and 7.5k, respectively. After applying the recourse actions (green area in the bottom figure), we can notice the populations of four species become stable again (the dashed line in the bottom figure). Therefore, the recourse actions can provide recommendations to restore the balance of the prairie ecosystem.

Case study on the MSDS dataset. Figure 4 depicts a case study on MSDS with control nodes 117 and 124. USAD detects a subsequence of anomaly consisting of two abnormal time steps from two-time series (CPU and RAM usages on node 117), highlighted in the red area of the top figure.

When the first abnormal time step is detected, RecAD suggests releasing the CPU usage by 6.7 on node 117 (the green area in the middle figure). In other words, it also means the anomaly here is due to the higher CPU usage than normal with a value of 6.7. After taking this action, the following time steps are affected by this action. A counterfactual time series is then generated using the AAP process, which is shown as the dashed lines in the middle of Figure 4. RecAD continues to monitor subsequent time steps for any abnormalities.

The following time step is still detected as abnormal in the time series of memory usage of node 117. RecAD recommends releasing the RAM usage by 13.39 on node 117 (the green area in the bottom figure), meaning that the abnormal time step here is due to high memory usage in a margin of 13.39. After taking the recourse action, the counterfactual time series is then generated (the dashed lines in the bottom figure). We can then observe that the entire time series returns to normal.

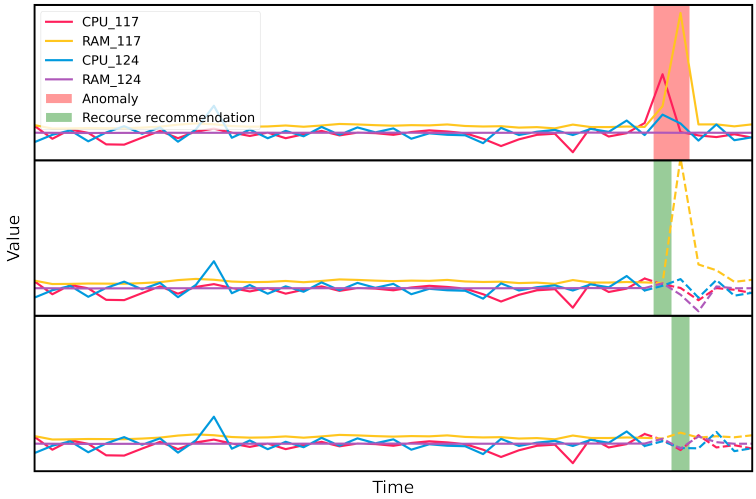


Figure 4: Recourse recommendations for restoring the abnormal CPU and RAM usages in MSDS.

In summary, recourse actions recommended by RecAD can effectively flip the outcome and lead to a normal counterfactual time series. Meanwhile, based on the recourse actions, the domain expert can understand why a time step is abnormal.

Table 5: The performance of recourse prediction using different components of RecAD.

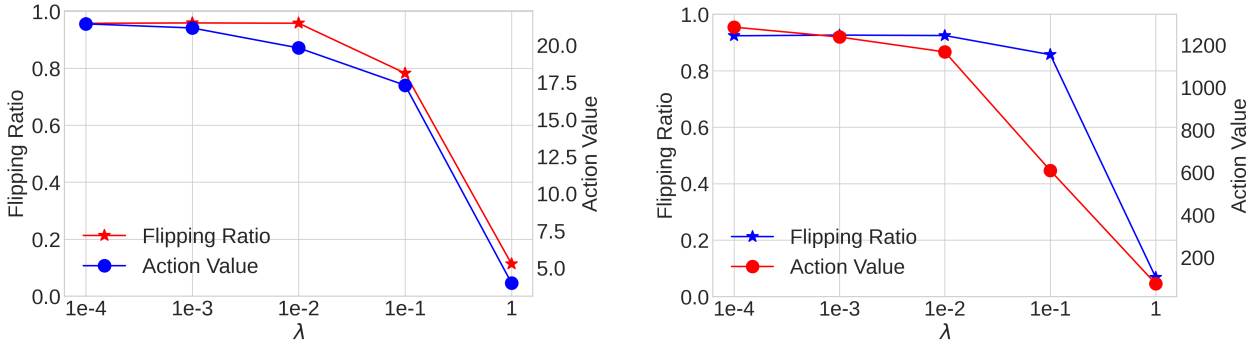
Dataset	Metric	RecAD w/o FFNN	RecAD w/o LSTM	RecAD
Linear	Flipping Ratio ↑	$0.340_{\pm 0.191}$	$0.676_{\pm 0.085}$	$\mathbf{0.922_{\pm 0.040}}$
	Action Cost ↓	$119.286_{\pm 174.173}$	$23.589_{\pm 23.453}$	$\mathbf{22.794_{\pm 13.054}}$
	Action Step ↓	$2.905_{\pm 0.882}$	$2.276_{\pm 0.939}$	$\mathbf{1.822_{\pm 0.521}}$
Lotka-Volterra	Flipping Ratio ↑	$0.353_{\pm 0.210}$	$0.523_{\pm 0.090}$	$\mathbf{0.952_{\pm 0.054}}$
	Action Cost ↓	$\mathbf{876.107_{\pm 810.047}}$	$1035.192_{\pm 881.242}$	$1468.230_{\pm 1086.090}$
	Action Step ↓	$2.706_{\pm 1.068}$	$2.304_{\pm 0.646}$	$\mathbf{1.266_{\pm 0.180}}$
MSDS	Flipping Ratio ↑	$0.228_{\pm 0.147}$	$0.697_{\pm 0.214}$	$\mathbf{0.841_{\pm 0.080}}$
	Action Cost ↓	$\mathbf{3.494_{\pm 2.665}}$	$4.136_{\pm 0.727}$	$6.747_{\pm 1.543}$
	Action Step ↓	$2.048_{\pm 0.607}$	$1.581_{\pm 0.525}$	$\mathbf{1.249_{\pm 0.068}}$

5.4 Ablation Study

We evaluate the performance of using different parts of RecAD (i.e., FFNN and LSTM) for recourse prediction. As RecAD contains an LSTM to catch the previous $K-1$ time lags and a feedforward neural network (FFNN) to include the time lag exclusion term Δ_t, we then test the performance of these two parts separately. Table 5 shows the average flipping ratio, action cost, and action step for three types of anomalies for the synthetic datasets and results for the real-world dataset MSDS. We observe that RecAD achieves higher flipping ratios while requiring fewer or comparable action steps compared to methods that utilize only parts of RecAD. In cases where other methods exhibit lower action costs, they achieve very low flipping ratios, indicating their inability to successfully flip abnormal samples, which is undesired. This result shows the importance of considering both information for reasonable action value prediction.

5.5 Sensitivity Analysis

The objective function defined in Eq. (7) for training RecAD employs the hyperparameter λ to balance the flipping ratio and action value. As shown in Figure 5, on both synthetic datasets, we have similar observations that with the increasing of λ, both action value and flipping ratio decrease. A large λ indicates a large penalty for high action values, which could potentially hurt the performance of flipping abnormal time steps as small action values may not be sufficient to flip the anomalies.



(a) Linear (b) Lotka-Volterra

Figure 5: Effects of the hyperparameter λ in Eq. (7).

6 Conclusions

In this work, we have developed RecAD, a novel framework for algorithmic recourse in abnormal multivariate time series. RecAD suggests actionable interventions to restore the multivariate time series to normal status, offering counterfactual explanations for abnormal patterns. By leveraging backtracking counterfactual reasoning, we formulated the problem of learning the recourse function from data as a constrained maximum likelihood problem to enable end-to-end training. The empirical studies have demonstrated the effectiveness of RecAD for recommending explainable and practical recourse actions in abnormal time series.

References

Julien Audibert, Pietro Michiardi, Frédéric Guyard, Sébastien Marti, and Maria A Zuluaga. Usad: Unsupervised anomaly detection on multivariate time series. In *KDD*, 2020.

Nicolas Bacaër. *A short history of mathematical population dynamics*, volume 618. Springer, 2011.

Ane Blázquez-García, Angel Conde, Usue Mori, and Jose A Lozano. A review on outlier/anomaly detection in time series data. In *ACM CSUR*, pp. 1–33, 2021.

Yatong Chen, Jialu Wang, and Yang Liu. Linear classifiers that encourage constructive adaptation. *arXiv preprint arXiv:2011.00355*, 2020.

Elliot Creager and Richard Zemel. Online algorithmic recourse by collective action. *arXiv preprint arXiv:2401.00055*, 2023.

Rainer Dahlhaus and Michael Eichler. Causality and graphical models in time series analysis. *Oxford Statistical Science Series*, pp. 115–137, 2003.

Debanjan Datta, Feng Chen, and Naren Ramakrishnan. Framing algorithmic recourse for anomaly detection. In *KDD*, pp. 283–293, 2022.

Giovanni De Toni, Stefano Teso, Bruno Lepri, and Andrea Passerini. Time can invalidate algorithmic recourse. *arXiv preprint arXiv:2410.08007*, 2024.

Ricardo Dominguez-Olmedo, Amir H Karimi, and Bernhard Schölkopf. On the adversarial robustness of causal algorithmic recourse. In *ICML*, pp. 5324–5342, 2022.

Ruijiang Gao and Himabindu Lakkaraju. On the impact of algorithmic recourse on social segregation. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett (eds.), *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 10727–10743. PMLR, 23–29 Jul 2023.

Clive WJ Granger. Investigating causal relations by econometric models and cross-spectral methods. *Econometrica: journal of the Econometric Society*, pp. 424–438, 1969.

Vincent Jacob, Fei Song, Arnaud Stiegler, Bijan Rad, Yanlei Diao, and Nesime Tatbul. Exathlon: A benchmark for explainable anomaly detection over time series. *arXiv preprint arXiv:2010.05073*, 2020.

Hong-Gyu Jung, Sin-Han Kang, Hee-Dong Kim, Dong-Ok Won, and Seong-Whan Lee. Counterfactual explanation based on gradual construction for deep networks. *Pattern Recognition*, 132:108958, 2022.

Amir-Hossein Karimi, Julius Von Kügelgen, Bernhard Schölkopf, and Isabel Valera. Algorithmic recourse under imperfect causal knowledge: a probabilistic approach. In *NeurIPS*, pp. 265–277, 2020.

Amir-Hossein Karimi, Bernhard Schölkopf, and Isabel Valera. Algorithmic recourse: from counterfactual explanations to interventions. In *ACM FAccT*, pp. 353–362, 2021.

Amir-Hossein Karimi, Gilles Barthe, Bernhard Schölkopf, and Isabel Valera. A survey of algorithmic recourse: contrastive explanations and consequential recommendations. In *ACM CSUR*, pp. 1–29, 2022.

Tung Kieu, Bin Yang, Chenjuan Guo, Christian S Jensen, Yan Zhao, Feiteng Huang, and Kai Zheng. Robust and explainable autoencoders for unsupervised time series outlier detection. In *2022 IEEE 38th International Conference on Data Engineering (ICDE)*, pp. 3038–3050. IEEE, 2022.

Klaus-Rudolf Kladny, Julius von Kügelgen, Bernhard Schölkopf, and Michael Muehlebach. Deep backtracking counterfactuals for causally compliant explanations. *Transactions on Machine Learning Research*, 2024. ISSN 2835-8856. URL `https://openreview.net/forum?id=Br5esc2CXR`.

Ričards Marcinkevičs and Julia E Vogt. Interpretable models for granger causality using self-explaining neural networks. *arXiv preprint arXiv:2101.07600*, 2021.

Meike Nauta, Doina Bucur, and Christin Seifert. Causal Discovery with Attention-Based Convolutional Neural Networks. In *MAKE*, pp. 312–340, 2019.

Sasho Nedelkoski, Jasmin Bogatinovski, Ajay Kumar Mandapati, Soeren Becker, Jorge Cardoso, and Odej Kao. Multi-source distributed system data for ai-powered analytics. In *ESOCC*, pp. 161–176, 2020.

Judea Pearl. *Causality*. Cambridge university press, 2009.

Sebastian Schmidl, Phillip Wenig, and Thorsten Papenbrock. Anomaly detection in time series: a comprehensive evaluation. In *VLDB*, 2022.

Alex Tank, Ian Covert, Nicholas Foti, Ali Shojaie, and Emily B Fox. Neural granger causality. In *IEEE PAMI*, pp. 4267–4279, 2021.

Shreshth Tuli, Giuliano Casale, and Nicholas R. Jennings. TranAD: Deep transformer networks for anomaly detection in multivariate time series data. In *VLDB*, 2022.

Sahil Verma, Varich Boonsanong, Minh Hoang, Keegan Hines, John Dickerson, and Chirag Shah. Counterfactual explanations and algorithmic recourses for machine learning: A review. *ACM Computing Surveys*, 56 (12):1–42, 2024.

Julius von Kügelgen, Amir-Hossein Karimi, Umang Bhatt, Isabel Valera, Adrian Weller, and Bernhard Schölkopf. On the fairness of causal algorithmic recourse. In *AAAI*, 2022.

Julius Von Kügelgen, Abdirisak Mohamed, and Sander Beckers. Backtracking counterfactuals. In *Conference on Causal Learning and Reasoning*, pp. 177–196. PMLR, 2023.

Pei Wang and Nuno Vasconcelos. Scout: Self-aware discriminant counterfactual explanations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8981–8990, 2020.

Appendix

A Algorithm of RecAD

Algorithm 1 shows the pseudo-code of the training process for recourse predictions. Given an abnormal time window $\mathbf{W}_t$ with the abnormal time step t and a subsequence window $\mathbf{V}_{t+L}$ of length L, we first predict the action value on $\mathbf{x}_t$ to reverse the abnormal status. After conducting the recourse actions, we also need to ensure that the following L time steps are also normal. Therefore, we then derive the counterfactual subsequence $\mathbf{V}^*_{t+L}$ step by step. For each following step, denoted as $\mathbf{V}^*_{t+l}$, if the subsequence $\mathbf{V}^*_{t+l}$ is still abnormal, we further conduct recourse actions to flip the abnormal status. After going through L time steps, we update the action prediction function $h_\phi(\cdot)$ based on the objective function defined in Eq. (7).

Algorithm 1: Pseudo code of training RecAD

Input: Pretrained GVAR $g_k(\cdot)$, anomaly detector $s(\cdot)$, an abnormal window $\mathbf{W}_t$ with the abnormal time step t, and the following window $\mathbf{V}_{t+L}$
Output: Updated $h_\phi(\cdot)$

$\mathbf{x}^*_t \leftarrow$ Action_Prediction($\mathbf{W}_t,\ g_k(\cdot),\ h_\phi(\cdot)$)
$\mathbf{V}^*_{t+L}[0] = \mathbf{x}^*_t$
for $l \leftarrow 1$ **to** L **do**
  Derive $\mathbf{V}^*_{t+l}$ by Eqs. (5) and (6)
  if $s(\mathbf{V}^*_{t+l}) > \tau$ **then**
    $\mathbf{x}^*_{t+l} \leftarrow$ Action_Prediction($\mathbf{V}^*_{t+l},\ g_k(\cdot),\ h_\phi(\cdot)$)
    $\mathbf{V}^*_{t+L}[l] = \mathbf{x}^*_{t+l}$
  end if
end for
Update $h_\phi(\cdot)$ based on the objective function $\mathcal{L}(\phi)$ in Eq. (7)
Function Action_Prediction($\mathbf{W}_t,\ g_k(\cdot),\ h_\phi(\cdot)$)
  $\mathbf{W}_{t-1} \leftarrow \mathbf{W}_t \setminus \{\mathbf{x}_t\}$
  Compute $\hat{\mathbf{x}}_t = \sum_{k=1}^{K-1} g_k(\mathbf{x}_{t-k})\mathbf{x}_{t-k}$ with $\mathbf{W}_{t-1}$
  Compute $\Delta_t = \mathbf{x}_t - \hat{\mathbf{x}}_t$
  Compute $\boldsymbol{\theta}_t = h_\phi(\mathbf{W}_{t-1}, \Delta_t)$
  $\mathbf{x}^*_t = \mathbf{x}_t + \boldsymbol{\theta}_t$
  Return $\mathbf{x}^*_t$

B Experiments

B.1 Datasets

Linear Dataset (Marcinkevičs & Vogt, 2021) is a **synthetic** time series dataset with linear interaction dynamics. We follow the structure from the original paper (Marcinkevičs & Vogt, 2021), where the linear interaction contains no instantaneous effects between time series and can be defined as:

$$
\begin{aligned}
x_t^{(1)} &= a_1 x_{t-1}^{(1)} + u_t^{(1)} + \epsilon_t^{(1)}, \\
x_t^{(2)} &= a_2 x_{t-1}^{(2)} + a_3 x_{t-1}^{(1)} + u_t^{(2)} + \epsilon_t^{(2)}, \\
x_t^{(3)} &= a_4 x_{t-1}^{(3)} + a_5 x_{t-1}^{(2)} + u_t^{(3)} + \epsilon_t^{(3)}, \\
x_t^{(4)} &= a_6 x_{t-1}^{(4)} + a_7 x_{t-1}^{(2)} + a_8 x_{t-1}^{(3)} + u_t^{(4)} + \epsilon_t^{(4)},
\end{aligned}
\tag{9}
$$

where coefficients $a_i \sim \mathcal{U}([-0.8, -0.2] \cup [0.2, 0.8])$, additive innovation terms $u_t^{(\cdot)} \sim \mathcal{N}(0, 0.16)$, and anomaly term $\epsilon_t^{(\cdot)}$.

Lotka-Volterra (Bacaër, 2011) is another **synthetic** time series model that simulates a prairie ecosystem with multiple species. We follow the structure from (Marcinkevičs & Vogt, 2021), which defines as:

$$
\begin{aligned}
\frac{d\mathbf{x}^{(i)}}{dt} &= \alpha\mathbf{x}^{(i)} - \beta \sum_{j \in Pa(\mathbf{x}^{(i)})} \mathbf{y}^{(j)} - \eta(\mathbf{x}^{(i)})^2, \text{ for } 1 \leq j \leq p, \\
\frac{d\mathbf{y}^{(j)}}{dt} &= \delta\mathbf{y}^{(j)} \sum_{k \in Pa(\mathbf{y}^{(j)})} \mathbf{x}^{(k)} - \rho\mathbf{y}^{(j)}, \text{ for } 1 \leq j \leq p, \\
x_t^{(i)} &= x_t^{(i)} + \epsilon_t^{(i)}, \text{ for } 1 \leq j \leq p, \\
y_t^{(j)} &= y_t^{(j)} + \epsilon_t^{(j)}, \text{ for } 1 \leq j \leq p,
\end{aligned}
\tag{10}
$$

where $\mathbf{x}^{(i)}$ and $\mathbf{y}^{(j)}$ denote the population sizes of prey and predator, respectively; $\alpha, \beta, \eta, \delta, \rho$ are parameters that decide the strengths of interactions, $Pa(\mathbf{x}^{(i)})$ and $Pa(\mathbf{y}^{(j)})$ correspond the Granger Causality between prey and predators for $\mathbf{x}^{(i)}$ and $\mathbf{y}^{(j)}$ respectively, and $\epsilon_t^{(\cdot)}$ is the abnormal term. We adopt 10 prey species and 10 predator species.

B.2 Implementation Details

Similar to (Audibert et al., 2020), we adopt a sliding window with sizes 5, 5, and 10 for the Linear, Lotka-Volterra, and MSDS datasets, respectively. We set the hyperparameters for GVAR by following (Marcinkevičs & Vogt, 2021). When training $h_\phi(\cdot)$, we set L in the objective function as $L = 1$, which is to ensure the following one-time step should be normal. The cost vector $\mathbf{c}$ can be changed according to the requirements or prior knowledge. Because the baseline models are prediction-based models that cannot take the cost into account, to be fair, we use $\mathbf{1}$ as the cost vector. The implementation details of neural networks in experiments are described in the Appendix.

For baselines, MLP is a feed-forward neural network with a structure of $((K-1)*d)$-100-100-100-d that the input is the flattened vector of $K-1$ time steps with d dimensions and the output is the predicted value of the next time step. The LSTM model consists of one hidden layer with 100 dimensions and is connected with a fully connected layer with a structure of 100-d. We use statsmodels[2] to implement the VAR model. The baseline GVAR model is the same as GVAR within our framework. To implement $h_\phi(\cdot)$ in RecAD, we utilize an LSTM that consists of one hidden layer with 100 dimensions and a feed-forward network with structure d-100. Then we use another feed-forward network with a structure of 200-d to predict the intervention values.

All experiments were conducted on an Ubuntu 20.04 server equipped with an AMD Ryzen 3960X 24-Core processor at 3.8GHz, dual GeForce RTX 3090 GPUs, and 128 GB of RAM. The implementation uses Python 3.9.7 and PyTorch 1.11.0.

[2] `https://www.statsmodels.org/`

B.3 Performance of anomaly detection.

Table 6: Performance of Anomaly detection.

Anomaly Types	Metrics	Linear	Lotka-Volterra
Non-causal Point	F1	$0.749_{\pm 0.022}$	$0.787_{\pm 0.106}$
	AUC-PR	$0.619_{\pm 0.024}$	$0.840_{\pm 0.116}$
	AUC-ROC	$0.816_{\pm 0.016}$	$0.851_{\pm 0.068}$
Non-causal Seq.	F1	$0.878_{\pm 0.011}$	$0.677_{\pm 0.061}$
	AUC-PR	$0.798_{\pm 0.015}$	$0.519_{\pm 0.026}$
	AUC-ROC	$0.914_{\pm 0.010}$	$0.794_{\pm 0.089}$
Causal Seq.	F1	$0.756_{\pm 0.003}$	$0.714_{\pm 0.020}$
	AUC-PR	$0.604_{\pm 0.004}$	$0.559_{\pm 0.016}$
	AUC-ROC	$0.877_{\pm 0.002}$	$0.824_{\pm 0.078}$

We evaluate the performance of USAD for anomaly detection in terms of the F1 score, the area under the precision-recall curve (AUC-PR), and the area under the receiver operating characteristic (AUC-ROC) on two synthetic datasets. Table 6 shows the evaluation results on synthetic datasets.

On the real-world MSDA dataset, USAD can achieve $0.888_{\pm 0.097}$, $0.996_{\pm 0.001}$, and $0.985_{\pm 0.003}$ in terms of F1 score, AUC-PR, and AUC-ROC, respectively, which means that USAD can find most of the anomalies on MSDS.

Overall, USAD can achieve promising performance on different types of anomalies and on both synthetic and real-world datasets, which lays a solid foundation for recourse prediction.