# Causanom: Anomaly Detection With Flexible Causal Graphs

**Sasha Strelnikoff**
Intelligent Systems Laboratory
HRL Laboratories, LLC
sstrelnikoff@hrl.com

**Aruna Jammalamadaka**
Intelligent Systems Laboratory
HRL Laboratories, LLC
ajammalamadaka@hrl.com

**Tsai-Ching Lu**
Intelligent Systems Laboratory
HRL Laboratories, LLC
tlu@hrl.com

## Abstract

Causality-based anomaly detection methods provide at least two significant theoretical benefits over purely statistical methods: 1. Improved robustness to non-anomalous out-of-distribution data, which implies a reduction in false-alarms; 2. A potential for failure localization due to the topological ordering of the causal graph. Recent studies have considered the utilization of causality-based methods for time series anomaly detection, however, these methods require the causal graph to be fixed; resultingly, such methods are not robust to incorrectly estimated causal graphs and are not able to natively model counterfactual scenarios. To address these limitations, we introduce Causanom: a graph-based encoder-decoder neural network for time series anomaly detection. Causanom utilizes a node conditional data-stream representation in conjunction with a weighted graph aggregation function in order to efficiently capture heterogeneous node dynamics whilst allowing for a flexible graphical structure. We show that Causanom can be trained along with auxiliary constraints in order to tune the causal graph and improve performance. Additionally, we show that Causanom can be used to produce counterfactual data, which we leverage to identify violated causal relationships. Using real and synthetic time series data respectively, we show that Causanom performs at least as well as state-of-the-art baselines in the anomaly detection task and outperforms existing methods in a causal attribution task.

## Introduction

In many real-world systems, rapidly detecting and diagnosing anomalies can save time and money spent on mitigation of potentially cascading failures. Moreover, since many relevant systems have complex temporal dependencies, the problem of *time series* anomaly detection and attribution has become an increasingly important task across a wide range of critical real-world applications, such as fraud detection (Kou et al., 2004; Pourhabibi et al., 2020), online service system failures (Li et al., 2022; Meng et al., 2020), and aviation safety assurance (Memarzadeh, Matthews, and Avrekh, 2020). By utilizing time-series anomaly detection and attribution methods, complex systems can be managed more efficiently and incidents can be identified in a timely manner.



Although both detection and attribution are required to recommend actions for full process recovery, the vast majority of work focuses only on detection (Vuković and Thalmann, 2022). Given a stream of multivariate time-series data, structural (graphical) models of dependencies between variables help us diagnose fault origins by tracing backwards through directed paths to anomalous node ancestors.

Non-causal anomaly detection methods to estimate structure include using sensor similarity to determine edges in the graph (Deng and Hooi, 2021), or modeling the correlation matrix between variables (Zhang et al., 2019). Since the networks produced by these methods do not capture causal dependencies on their edges (e.g., the left and right engine of an airplane would have high correlation but no direct causal relationship), they are obviously not ideal for attributing causes to detected failures. Additionally, such correlation based methods are likely to produce false alarms in the anomaly detection task, since non-anomalous out-of-distribution data is likely to occur in many real-world applications.

Methods which leverage Bayesian networks to model structure can have causal interpretations, since they are generally assumed to be directed and acyclic. However, the existing literature focuses only on maximum likelihood estimates which score the network against the data (Krishnamurthy, Sarkar, and Tewari, 2014). Attribution is either performed simply by finding the lowest probability node (Diallo et al., 2018), or is not performed at all (Wunderlich and Niggemann, 2017).

While statistical models like Bayesian networks specify a single probability distribution, a causal model represents a set of distributions – one for each possible intervention. In this way, they are able to model failure-modes of real-world systems, where a failure is assumed to be an intervention, and "intervention recognition" (Li et al., 2022) is the process of identifying which node (or set of nodes) has failed. Recent work focuses on using neural approaches to estimating causal dependencies in multivariate time series data (Shah, Dang, and Zerfos, 2018; Yang, Zhang, and Hoi, 2022; Qiu et al., 2020; Li et al., 2022). In these, violations of the estimated causal dependencies constitute anomalies, and anomaly scores from individual nodes or edges are summed along the paths to root nodes. The highest scoring path is output as a diagnostic trace for attribution.

While causality-based models are the most promising for accurately modeling the data generating process, the above

methods require a fixed causal graph. This results in the inability to adapt (without fully retraining) to an incorrectly estimated or modified causal graph, as may occur when the initial structure learning is inaccurate, causal relationships are intervened upon, or additional variables need to be modeled. Additionally, all of the methods discussed above tackle the problem of attribution via statistical anomaly propagation. Theoretically, this method is flawed in light of the Principle of Independent Mechanisms, which states that "The causal generative process of a system's variables is composed of autonomous modules that do not inform or influence each other. In the probabilistic case, this means that the conditional distribution of each variable given its causes (i.e., its mechanism) does not inform or influence the other mechanisms." (Schlkopf et al., 2021). This implies that the effects of an intervention should be contained in a single local causal mechanism. For our purposes, this indicates that an anomaly that is the result of a single failure point should be detected in only a single node, assuming the causal dependencies are accurately modeled, and therefore anomaly propagation is not applicable.

In this paper, we propose a method for flexibly incorporating a causal graph for simultaneous anomaly detection and localized causal attribution. This graph flexibility not only allows us to quickly adapt to new causal structures but also enables us to perform intervention-based attribution, with which we identify a class of anomalous edges by comparing predictions from soft interventions to observed data. Our contributions are,

- An anomaly detection model which leverages a flexible causal graphical structure in such a way that the structure can be dynamically updated and easily incorporate supervision.
- An introduction of causal edge anomaly attribution and a method for attribution based on simulated interventions.
- Quantitative evaluations of improvements in detection and edge attribution across multiple synthetic and real-world datasets.

The remainder of this paper is structured as follows. The first section provides the mathematical problem formulation. The second section outlines our novel method for anomaly detection and attribution, along with its advantages. The third section describes our experiments and results. The final section concludes with a discussion and potential avenues for future work.

## Problem Formulation

We study a temporal system which, when not in an anomalous state, has a stationary generating causal structure $\mathcal{G}$ with corresponding random variables $\{X_k(t)\}_{k=1}^N$ for all times $t$. Denoting $\mathbb{PA}(k)$ as the collection of parent nodes of $k$ in $\mathcal{G}$, assume the unobserved local causal mechanisms, for each $k \in \mathcal{N}_c := \{n \in \mathcal{G} \mid |\mathbb{PA}(n)| > 0\}$, are defined by the system of equations

$$X_k(t) = f_k\left(\{X_n(t') \mid n \in \mathbb{PA}(k), t' \leq t\}\right) + \eta_k^t, \quad (1)$$

where $\{f_k\}$ are arbitrary and the collection $\{\eta_k^t \sim \mathcal{N}(0, \sigma_k)\}$ is independent with respect to both $k$ and $t$.

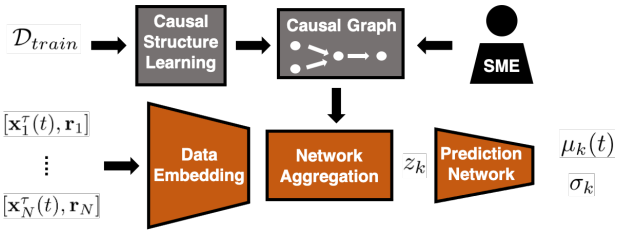



Figure 1: Causanom pipeline.

An anomalous state of the system is one in which the generating process deviates from that described in (1) for at least one value of $k$. In line with prior research, we pose this as an unsupervised anomaly detection problem. In particular, given a collection of nominal data $\mathcal{D}_{train}$ generated by a latent process described by (1), and labeled anomalous data $\mathcal{D}_{test}$, containing both nominal data and data resulting from an anomalous state of the system, we aim to learn a model $M$ to detect the anomalies in $\mathcal{D}_{test}$ using only $\mathcal{D}_{train}$ as training data.

For the problem of edge-attribution, we focus on a subclass of additive causal mechanisms and anomalies arising from edge-perturbations. In this restricted context, we aim to identify the perturbed edges in $\mathcal{D}_{test}$ using the pretrained model $M$. We leave the details of this to the Edge-Anomaly Attribution Section.

## Methods

### Anomaly Detection Model

The definition of an anomalous state introduced above suggests a natural definition for a score, $s$, of anomaly associated with the $k$th dimension of $\mathbf{x} = (x_1, ..., x_N)$ as

$$s(x_k) = -\mathbb{P}_{X_k|\mathbb{PA}(X_k)}\left(x_k \mid x_n, n \in \mathbb{PA}(k)\right) \quad (2)$$

and the anomaly score of $\mathbf{x}$ as the maximum anomaly score over each of its components,

$$S(\mathbf{x}) = \max_k \left(s(x_k)\right). \quad (3)$$

Since the probability densities $\mathbb{P}_{X_k|\mathbb{PA}(X_k)}$ and graph structure $\mathcal{G}$ are unknown, our aim is to estimate these latent features in order to produce an estimated anomaly score $\tilde{S}(\mathbf{x})$ for an observed value $\mathbf{x}$. We assume an initial estimate of the causal structure, $\tilde{\mathcal{G}}$, with adjacency matrix $A \in \mathbb{R}^{N \times N}$, which may be informed via a subject matter expert (SME) or any data-driven structure learning methodology (such as the PC algorithm (Colombo, Maathuis, and others, 2014)).

In order to model $\mathbb{P}_{X_k|\mathbb{PA}(X_k)}$, we leverage a simple generative method augmented with a variant of a Graph Convolutional Netwok (GCN) which we use to incorporate the graph structure $\tilde{\mathcal{G}}$. Our proposed pipeline is illustrated in Figure 1. For each time step $t$, we compute the data embedding using a learned node embedding vector concatenated with the input datastream, $\mathbf{x}_k^\tau(t) = (\mathbf{x}_k(t-\tau), ..., \mathbf{x}_k(t))$, for a fixed time window $\tau$. We then compute the graph weighted average over all of the data embeddings to obtain a parent-node embedding for each node $k \in \mathcal{N}_c$. Finally, this embedding

is used to calculate the predicted mean and standard deviation of a Gaussian distribution. The loss is then optimized to fit each component to the corresponding conditional distribution, $\mathbb{P}_{X_k|\mathbb{PA}(X_k)}$, using maximum likelihood estimation. Additional loss terms are considered in order to restrict the tuning of the causal weight matrix. We detail each of these components below.

## Data Embedding

In order to motivate our proposed data embeddings, we make the following observations:

- Since the value of a non-root node at time $t$ may be a function of its parent's values at any time $t' \leq t$, a node's embedding should contain both its current and previous values. In order to limit model complexity, we consider the previous values of each node up to a fixed time instance $\tau$.
- It would be natural to directly utilize the values, $\mathbf{x}_k^\tau(t)$, as the node representation corresponding to node $k$, however this does not distinguish two nodes with identical histories. In order to overcome this, for each node we introduce a learned node embedding vector $\mathbf{r}_k$ with dimensionality $q$.

Given the above considerations, we pass the concatenation of $\mathbf{x}_k^\tau(t)$ and $\mathbf{r}_k$ to a feed-forward neural network $M_{enc}(\_;\theta_{enc}) : \mathbb{R}^{\tau+q} \to \mathbb{R}^l$ with learnable parameters $\theta_{enc}$. We denote this vector as $\mathbf{v}_k(t) := M_{enc}([\mathbf{x}_k^\tau(t), \mathbf{r}_k]; \theta_{enc})$.

## Network Aggregation

We aim to incorporate $\mathbf{A}$ into our model in a way which allows for the adjacency matrix to be refined for prediction accuracy or modified for counterfactual investigation. Towards this end, we define a weighted aggregation function based on a column-normalized version of $\mathbf{A}$, which we call the initial weight matrix and write as $\widetilde{\mathbf{W}} = \mathbf{A}\mathbf{D}^{-1}$, where $\mathbf{D}$ is the degree matrix of our graph[1]. To modify the aggregation weights, we can introduce a collection of learnable parameters $\theta_{graph} = \{\theta_{ij}\}$ in order to construct a learnable weight matrix $\mathbf{W}$, the columns of which are normalized via the softmax: $\mathbf{W}_{ij} = \text{softmax}(\mathbf{w}_i) = \frac{e^{-\theta_{ij}\beta}}{\sum_i e^{-\theta_{ij}\beta}}$ where $\beta > 0$ is the temperature. Note, in order to initialize $\mathbf{W}$ to approximate the initial weight matrix, we can simply set $\theta_{ij} = K\widetilde{\mathbf{W}}_{ij}$ for some large constant $K$. Whether we utilize the learnable weight matrix or fix $\mathbf{W} = \widetilde{\mathbf{W}}$, we compute the aggregated feature representation matrix, $\mathbf{Z}(t) = \mathbf{V}(t)\mathbf{W}$. Note, our method can additionally be used for root-node modeling by simply requiring $\mathbf{W}_{kk} = 1$ and masking the contemporaneous values ($x_k^\tau[\tau+1] = 0$) for all $k \in \mathcal{N}_c^c$.

## Prediction Network

The aim of the prediction network is to estimate the conditional distribution of each of the non-root nodes $k \in \mathcal{N}_c$. Since each of the random variables has associated noise $\eta_k^t$ described by (1), any prediction-based method of detecting anomalies must also estimate the expected deviations caused by each variable's noise. Towards this end, we learn a prediction model $M_{pred}^k(z_k(t); \theta_{pred}^k) = (\mu(z_k(t)), \sigma_k)$, where $z_k = \mathbf{Z}(t)[k]$ and $M_{pred}^k$ is a feed-forward neural network with learnable parameters $\theta_{pred}^k$[2].

## Optimization

If we denote the full network by $M$ and let $\mathbf{x}^\tau(t) = [\mathbf{x}_1^\tau(t), ..., \mathbf{x}_N^\tau(t)]$, we have $M(\mathbf{x}^\tau(t); \theta)[k] = (\mu_k(t), \sigma_k)$ where $\theta = \{\mathbf{r}_k \mid k \in \mathcal{N}_c\} \cup \{\theta_{enc}, \theta_{graph}, \theta_{pred}\}$ is the set of all learnable parameters. Since $M$ is an estimate of the conditional distribution for the local causal mechanism of each node in $\mathcal{N}_c$, and these mechanisms are independent, maximizing the likelihood of $\theta$ is equivalent to minimizing sum of the log losses,

$$L(\mathbf{x}(t), M(\mathbf{x}^\tau(t); \theta)) = \sum_{k \in \mathcal{N}_c} l(\mu_k(t), \sigma_k(t), x_k(t)), \quad (4)$$

where

$$l(\mu, \sigma, x) = \log(\sigma) + \frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2.$$

We consider the baseline method to be one in which the parameters in $\mathbf{W}$ remain fixed, however in the subsequent experiments, we investigate the impact of fine-tuning $\mathbf{W}$. Fine-tuning could be done most simply in an unconstrained manner, however, this would not necessarily preserve the directed acyclicity of $\mathbf{W}$. In NOTEARS structure learning (Zheng et al., 2018), the authors introduce a constraint function $h(\mathbf{W}) = tr(e^{\mathbf{W}}) - l$, for $\mathbf{W} \in \mathbb{R}^{l \times l}$, and prove $h(\mathbf{W}) = 0$ iff $\mathbf{W}$ is a DAG. To enforce this constraint, we add $\gamma h(\mathbf{W})$ to (4) with weighting $\gamma > 0$.

## Edge-Anomaly Attribution

For purposes of this section, restrict the characterization in (1) to causal systems of the form

$$X_k(t) = f_k\left(\sum_{n \in \mathbb{PA}(k)} g_{kn}\left(X_n(t)\right)\right) + \eta_k^t, \quad (5)$$

where $\{f_k\}$, $\{g_{kn}\}$ are arbitrary measurable functions. We say there is an edge anomaly at time $t$ if there exists $(n,k) \in \mathcal{G}$ with $g_{kn}(t) \mapsto \hat{g}_{kn}(t)$ for some function $\hat{g}_{kn} \neq g_{kn}$. In this setting, we formulate the edge-anomaly attribution problem as that of identifying the edge(s) $(n,k)$ corresponding to an edge anomaly. In the following, we only consider edge anomalies which are the result of a single edge anomaly in which $\hat{g}_{kn} = 0$. We refer to this as a broken edge anomaly.

Given a trained Causanom model, $M(\mathbf{x}(t); \theta)$, and detected anomaly at node $k$, we propose the following edge-anomaly attribution pipeline: For each $n \in \mathbb{PA}(k)$, set $\mathbf{W}_{nk} = 0$, re-normalize the $k$th column of $\mathbf{W}$ to produce the "intervened" causal graph. Then construct an estimated

[1] We set $\mathbf{D}[k,k] = 1$ for the root nodes, so $\mathbf{D}^{-1}$ is always defined.

[2] We are implicitly assuming that $\{\eta_k^t\}$ are gaussian and identically distributed for a fixed $k$, however, these conditions could be relaxed by considering a mixture of gaussians (as in mixture density networks) or allowing the standard deviation/mixture terms to be time dependent, respectively.

score, $\tilde{S}_n$, using each of the *modified* models and choose the broken edge as $(\tilde{n}, k)$, $\tilde{n} = \text{argmin}_n \tilde{S}_n(\mathbf{x}_k)$, based on the assumption that the minimum scoring model corresponds to the broken edge anomaly. Pseudocode for this procedure is provided in **Algorithm 1** below.

**Algorithm 1** Identify Broken Edge Anomaly

**Input:** Model $M$, anomalous node $k$, $\mathbf{x}$, $\mathbf{W}$
**Output:** brokenEdge
min_score $\leftarrow \infty$
$\mathbf{W}_0 \leftarrow \mathbf{W}$
**for** n in $\mathbb{PA}(k)$ **do**
    $\mathbf{W}[n,k] \leftarrow 0$
    $\mathbf{W}[n,k] \leftarrow \mathbf{W}[n,k]/\mathbf{W}[:,k]$
    score $\leftarrow \tilde{S}_n(\mathbf{x}_k)$
    **if** score $<$ minScore **then**
        brokenEdge $\leftarrow (n,k)$
    $\mathbf{W} \leftarrow \mathbf{W}_0$
**return** brokenEdge

## Experiments and Results

In this section, we evaluate our system's anomaly detection and localization performance using realistic and synthetic data respectively. In order to test the impact of the causal graph fine-tuning, in addition to the fixed-$\mathbf{W}$ baseline version of our model, we consider both models with unconstrained and DAG-constrained fine-tuning of $\mathbf{W}$ as introduced in the Methods section.

### Anomaly Detection

**Datasets** In order to evaluate the efficacy of our model for anomaly detection, we utilize two common anomaly detection benchmark datasets, referred to as SWaT (Mathur and Tippenhauer, 2016) and WADI (Ahmed, Palleti, and Mathur, 2017). Both datasets are constructed from sensor readings of small-scale water treatment plant systems.

The SWaT contains time series data for 51 discrete and continuous sensors. This data was collected over 11 days. During the last four days, 36 attack scenarios were executed, which targeted various attack points, including physical sensors, actuators, and network communication infrastructure. The WADI testbed is an extension of the SWaT testbed with the inclusion of additional components such as analyzers, booster pumps, and chemical dosing systems (Fährmann et al., 2022). WADI is significantly larger than SWaT, consisting of 126 discrete and continuous variables collected over 14 days under normal conditions and for an additional 2 days under various attack scenarios.

Following (Fährmann et al., 2022), we perform the following data prepossessing. Since (Goh et al., 2017) noted that the systems only reach stabilization after at most 6 hours, these initial samples are removed from both datasets. Since both datasets are very large, we subsample the time intervals to 1 sample per second, which is expected to capture all relevant variation present in the data. In (Kravchik and Shabtai, 2021), the Kolmogorov-Smirnov test (Karson, 1968) was used to show that there is a subset of variables whose distributions do not match across the training and validation sets. This is an issue in the context of anomaly detection, since a required assumption for learning the nominal data distribution is that it is stationary. We filter out the identified troublesome variables (19 from WADI and 15 from SWaT) and any constant variables as they cannot be appropriately modeled. Finally, as noted in (Kravchik and Shabtai, 2021), the value of the sensor 2B_AIT_002_PV in WADI becomes one thousand times larger in the middle of the test data without triggering an anomaly flag, so we additionally remove this variable from our data. For both datasets, we take take a 90-10 split of train/validation data.

In order to learn the causal graphs for datasets, we utilize the stable PC algorithm (Colombo, Maathuis, and others, 2014) with a partial correlation based conditional independence test as outlined in (Runge et al., 2019). For our method, we model the root nodes by adding self-edges as discussed earlier.

**Evaluation Metric** Kim et al. (2022) demonstrates that the commonly used point-adjusted anomaly metric is effectively useless since when it is used for performance comparison, untrained anomaly models can outperform trained state-of-the-art detectors. Instead they propose using either F1 or "point-adjust@k" metrics. Here, we follow their advice and report only the unadjusted F1 scores. Since all methods require a threshold to be set in order to classify anomalies, we pick the threshold to be the one which produces the maximal F1 score for each method. In order to provide a thorough analysis of the behavior of each model, we additionally report the Precision-Recall curves for each dataset.

**Baseline methods**

**Isolation Forest (IF) (Liu, Ting, and Zhou, 2008):** This method works by attempting to separate each data point using decision trees. The number of branching points above a given data point provides a measure of how anomalous that data point is - fewer splits indicate the data is more separated from the overall dataset and is therefore more anomalous.

**Graph Deviation Network (GDN) (Deng and Hooi, 2021):** This method utilizes a learned node-embedding scheme to disambiguate sensors of different characteristics, which is leveraged through an attention-based network. This method is a strong baseline for our model, as it similarly learns a graph structure and was shown in Kim et al. (2022) to be the only method to reliably outperform simple baselines under appropriate evaluation metrics. We keep all default parameters utilized in the original paper for each of the respective datasets, which we obtained from the GDN github comments by the authors.

**Causal Conditional Variational Autoencoder (C-CVAE) (Yang, Zhang, and Hoi, 2022):** The authors of this Arxiv pre-print introduce a general causality-based anomaly detection framework in which they assume an input causal graph, train a local causal mechanism model for non-root nodes and apply separate model to model the marginal distributions of each root-node. Among the methods tested, they show that the optimal choices of the models for the non-root/root nodes are independent conditional variational

autoencoders and an isolation forest respectively. We apply these models and train their pipeline using the hyperparameters described in the pre-print. Note, however, that we use our constructed causal graph to allow for a more direct comparison with our method.

For our architecture, all networks are feed-forward with ReLU activations and we take $\tau = 10$. $M_{enc}$ has hidden layer dimensionality $[32, 16, 8]$ and $[64, 32, 16]$ for SWaT and WADI respectively. We take the hidden layer dimensionality of $M^k_{pred}$ to be $[10, 20, 10]$ for every $k$. In order to train our method, we utilize Adam optimization and train for $500$ epochs, with learning rate $0.01$, patience of $50$, and $100$ epochs of burn-in. For each of the $\mathbf{W}$-tuning variants, we apply this training process, by modifying the learning rate to be $0.001$ and initializing with the baseline model. For the DAG-constrained models, we set $\gamma = 1$.

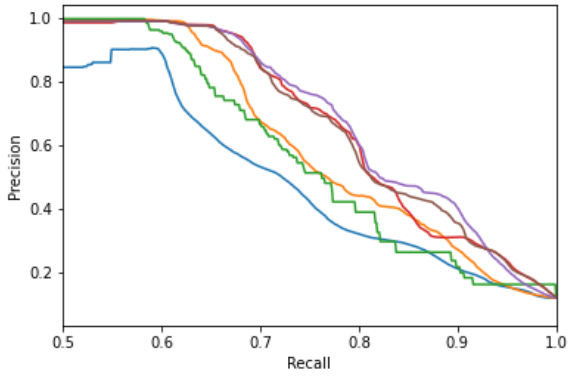
**Results** We train each method 5 times with different random seeds in order to obtain standard-deviation estimates for each metric. The results from this experiment are shown in Table 1. We note that Causanom performs at least as well as the other baselines and, as expected, both Causanom tuning variants have improved performance across all metrics as compared to the base Causanom model, however we find no decisive optimal tuning strategy. The Precision-Recall curves for both SWaT and WADI are shown in Figure 2. Note, we truncate the domain of Figure 2 (a) to the beginning of deviation in the curves. From this figure we see that all Causanom variants dominate the baseline methods and GDN, C-CVAE, and IF are fairly well distinguished in descending order of dominance. Figure 2 (b) shows a less clear dominance order, however the Causanom variants show clear dominance for low-recall, high-precision values and all methods have similar performance in the high-recall setting.

| | SWaT (F1-best) | WADI (F1-best) |
|---|---|---|
| IF | 72(0.6) | 26.4(1.6) |
| GDN | 77.5(1.8) | 24.5(1.8) |
| C-CVAE | 72.1(1) | 25.2(0.1) |
| Causanom | 78.8(1) | 32.3(1.7) |
| Causanom+tuning | **79.5**(0.9) | 35.2(3.2) |
| Causanom+DAGtuning | 78.6(0.6) | **35.3**(2.7) |

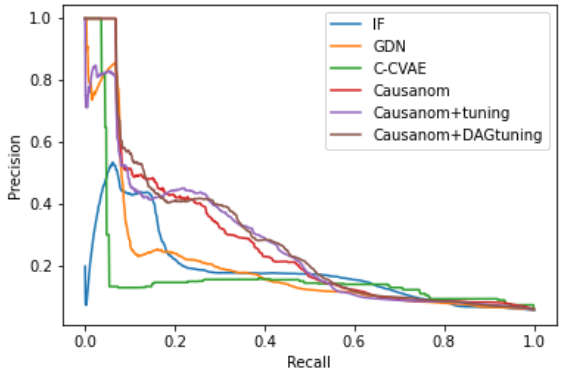
Table 1: Precision (P), Recall (R), and F1-score results from our anomaly detection experiment on the SWaT and WADI datasets. Parenthetical values indicate 1-standard deviation for each metric.

## Anomaly Attribution

**Synthetic Dataset** In order to test the edge-anomaly attribution scheme, we require access to ground-truth failure attributions. Since these are not available for any publicly available datasets, we construct a synthetic dataset and induce failures by perturbing the causal graph. Following the scheme outlined in (Runge, 2020), we construct a random time-windowed graph with 10 nodes along 10 timesteps. To ensure similar edge sparsity across graphs, we restrict the




(a) SWaT: Precision-Recall curves




(b) WADI: Precision-Recall curves

Figure 2: Precision-Recall curves for the SWaT (a) and WADI (b) datasets across all methods.

average in-degree of the nodes to be $0.2$ and proceed to construct the graph by iteratively sampling from all possible edges until the requisite number of edges are obtained. We adopt this edge sampling scheme so that we never add a cycle inducing edge to the summary graph (i.e., when collapsed over temporal lags, the causal graph is still a DAG). In order to generate the time series data once the graph has been constructed, we take the edge relationships to be $x_k(t) = \left(\sum_{(n,\tau)\in \mathbf{PA}(k)} a^\tau_{kn} x_n(t-\tau)\right) + \eta^t_k$, for each $k \in \mathcal{N}_c$. Here, $a^\tau_{kn}$ are uniformly chosen on $\pm[1, 5]$ and L1-normalized across $(n, \tau)$, and $\eta^t_k \sim \mathcal{N}(0, 0.5)$ are independently sampled in $k$ and $t$. In order to drive the system, we take the root-nodes to be sinusoids of varying frequencies. After generating the nominal data with the method outlined above, we can induce a broken edge anomaly between nodes $k$ and $(k', \tau) \in \mathbf{PA}(k)$ by setting $a^\tau_{kn} = 0$.

In our experiments, we aim to determine the efficacy of our broken edge identification method. Our method can only identify a broken edge if it is modeled by $\mathbf{W}$, therefore, we test the effect that fine tuning $\mathbf{W}$ has on our results in settings where the estimated causal graph is imperfect. We construct the corrupted graph edges, $\mathcal{E}_{ed}$, with edit distances

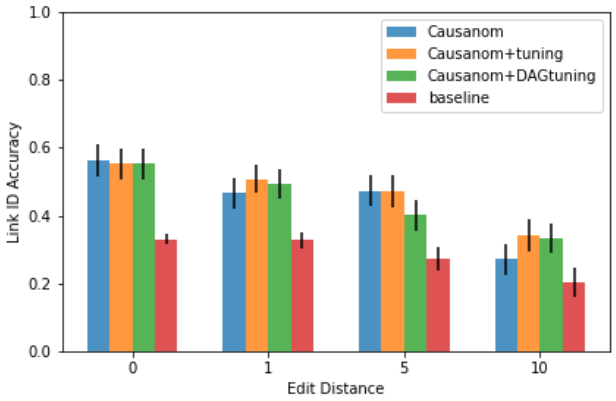



Figure 3: Broken edge anomaly identification accuracy for the Causanom baseline method, Causanom unconstrained/DAG-constrained fine-tuned variants, and the baseline methods.

$i = 0, 1, 5, 10$ from the ground truth graph $\mathcal{G}$, by randomly adding or removing edges for a selection of non-root nodes $\mathcal{N} \subseteq \mathcal{N}_c$ in $\mathcal{G}$, while ensuring that $\mathcal{E}_i \subseteq \mathcal{E}_{10}$ for all $i$. In order to target the corrupted edges when inducing the failures, we only allow failures to occur on the nodes whose in-edges were modified by $\mathcal{E}_{10}$.

**Results** We run 10 different simulations to construct the nominal data. For each experiment, break a random edge of each of the nodes in $\mathcal{N}$ across 30 runs. For each failure, we utilize the above methodology to predict the broken edge. A histogram of the accuracy of the broken edge anomaly detection is shown in Figure 3 along with 1 standard deviation error bars for the mean of each model. Existing methods (Li et al., 2022; Yang, Zhang, and Hoi, 2022) aim to identify the root cause of an anomaly by considering the anomaly score of the parents of an anomalous node. We show the output of these methods as a baseline (red bar) in the figure.

As can be seen in the figure, our method significantly outperforms the baseline methods for all edit distances. Additionally, we see improvements in the scores of the tuned graphs for the more corrupted experiments as expected. Note however that the DAG constraint does not improve performance over the method of tuning without constraints, which is unexpected since the DAG-constraint should facilitate modeling of the true causal graph. As an explanation, we observe that for cases in which the graph corruptions do not preserve the topological ordering, most smooth paths which transform the corrupted graph into the true graph require a DAG violation. Therefore, the DAG-constraint may restrict the loss to a local optimum in such cases. More work is required to determine how to optimally enforce such a constraint in a continuous setting.

## Conclusion

In this paper we introduced a causality-based anomaly detection method which can flexibly model the causal structure of a system. We showed how this flexibility can lead to superior results over strong baselines via graph fine tuning. We also introduced a method of edge attribution by utilizing this flexibility to produce counterfactual estimates for broken edge anomalies.

To the authors' knowledge, this is the first method to investigate the problem of edge-based anomaly attribution. As a result, there are many directions for investigation in this area. For example, here we assume that it is already known that an edge anomaly has occurred, however, future work could focus on how to determine the type of anomaly directly from the data stream. Additionally, both multiple-edge and more functionally complex edge failures should be considered to increase versatility of the system for various applications. This problem poses a significant challenge as the search space for such edge failures is infinite. Additional work could be done to improve the causal structure estimation presented here. For example, imposing a hard DAG constraint could be considered or ancestor constraints could be incorporated into the loss term.

## References

Ahmed, C. M.; Palleti, V. R.; and Mathur, A. P. 2017. Wadi: a water distribution testbed for research in the design of secure cyber physical systems. In *Proceedings of the 3rd international workshop on cyber-physical systems for smart water networks*, 25–28.

Colombo, D.; Maathuis, M. H.; et al. 2014. Order-independent constraint-based causal structure learning. *J. Mach. Learn. Res.* 15(1):3741–3782.

Deng, A., and Hooi, B. 2021. Graph neural network-based anomaly detection in multivariate time series. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, 4027–4035.

Diallo, T. M.; Henry, S.; Ouzrout, Y.; and Bouras, A. 2018. Data-based fault diagnosis model using a bayesian causal analysis framework. *International Journal of Information Technology & Decision Making* 17(02):583–620.

Fährmann, D.; Damer, N.; Kirchbuchner, F.; and Kuijper, A. 2022. Lightweight long short-term memory variational auto-encoder for multivariate time series anomaly detection in industrial control systems. *Sensors* 22(8):2886.

Goh, J.; Adepu, S.; Junejo, K. N.; and Mathur, A. 2017. A dataset to support research in the design of secure water treatment systems. In *Critical Information Infrastructures Security: 11th International Conference, CRITIS 2016, Paris, France, October 10–12, 2016, Revised Selected Papers 11*, 88–99. Springer.

Karson, M. 1968. Handbook of methods of applied statistics. volume i: Techniques of computation descriptive methods, and statistical inference. im chakravarti, rg laha, and j. roy, new york, john wiley; 1967.

Kim, S.; Choi, K.; Choi, H.-S.; Lee, B.; and Yoon, S. 2022. Towards a rigorous evaluation of time-series anomaly detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, 7194–7201.

Kou, Y.; Lu, C.-T.; Sirwongwattana, S.; and Huang, Y.-P. 2004. Survey of fraud detection techniques. In *IEEE Inter-*

*national Conference on Networking, Sensing and Control, 2004*, volume 2, 749–754. IEEE.

Kravchik, M., and Shabtai, A. 2021. Efficient cyber attack detection in industrial control systems using lightweight neural networks and pca. *IEEE Transactions on Dependable and Secure Computing* 19(4):2179–2197.

Krishnamurthy, S.; Sarkar, S.; and Tewari, A. 2014. Scalable anomaly detection and isolation in cyber-physical systems using bayesian networks. In *Dynamic Systems and Control Conference*, volume 46193, V002T26A006. American Society of Mechanical Engineers.

Li, M.; Li, Z.; Yin, K.; Nie, X.; Zhang, W.; Sui, K.; and Pei, D. 2022. Causal inference-based root cause analysis for online service systems with intervention recognition. 32303240.

Liu, F. T.; Ting, K. M.; and Zhou, Z.-H. 2008. Isolation forest. In *2008 eighth ieee international conference on data mining*, 413–422. IEEE.

Mathur, A. P., and Tippenhauer, N. O. 2016. Swat: A water treatment testbed for research and training on ics security. In *2016 international workshop on cyber-physical systems for smart water networks (CySWater)*, 31–36. IEEE.

Memarzadeh, M.; Matthews, B.; and Avrekh, I. 2020. Unsupervised anomaly detection in flight data using convolutional variational auto-encoder. *Aerospace* 7(8):115.

Meng, Y.; Zhang, S.; Sun, Y.; Zhang, R.; Hu, Z.; Zhang, Y.; Jia, C.; Wang, Z.; and Pei, D. 2020. Localizing failure root causes in a microservice through causality inference. In *2020 IEEE/ACM 28th International Symposium on Quality of Service (IWQoS)*, 1–10. IEEE.

Pourhabibi, T.; Ong, K.-L.; Kam, B. H.; and Boo, Y. L. 2020. Fraud detection: A systematic literature review of graph-based anomaly detection approaches. *Decision Support Systems* 133:113303.

Qiu, J.; Du, Q.; Yin, K.; Zhang, S.-L.; and Qian, C. 2020. A causality mining and knowledge graph based method of root cause diagnosis for performance anomaly in cloud applications. *Applied Sciences* 10(6):2166.

Runge, J.; Nowack, P.; Kretschmer, M.; Flaxman, S.; and Sejdinovic, D. 2019. Detecting and quantifying causal associations in large nonlinear time series datasets. *Science advances* 5(11):eaau4996.

Runge, J. 2020. Discovering contemporaneous and lagged causal relations in autocorrelated nonlinear time series datasets. In *Conference on Uncertainty in Artificial Intelligence*, 1388–1397. PMLR.

Schlkopf, B.; Locatello, F.; Bauer, S.; Ke, N. R.; Kalchbrenner, N.; Goyal, A.; and Bengio, Y. 2021. Toward causal representation learning. *Proceedings of the IEEE* 109(5):612–634.

Shah, S. Y.; Dang, X.-H.; and Zerfos, P. 2018. Root cause detection using dynamic dependency graphs from time series data. In *2018 IEEE International Conference on Big Data (Big Data)*, 1998–2003. IEEE.

Vuković, M., and Thalmann, S. 2022. Causal discovery in manufacturing: A structured literature review. *Journal of Manufacturing and Materials Processing* 6(1):10.

Wunderlich, P., and Niggemann, O. 2017. Structure learning methods for bayesian networks to reduce alarm floods by identifying the root cause. In *2017 22nd IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, 1–8. IEEE.

Yang, W.; Zhang, K.; and Hoi, S. C. 2022. Causality-based multivariate time series anomaly detection. *arXiv preprint arXiv:2206.15033*.

Zhang, C.; Song, D.; Chen, Y.; Feng, X.; Lumezanu, C.; Cheng, W.; Ni, J.; Zong, B.; Chen, H.; and Chawla, N. V. 2019. A deep neural network for unsupervised anomaly detection and diagnosis in multivariate time series data. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, 1409–1416.

Zheng, X.; Aragam, B.; Ravikumar, P. K.; and Xing, E. P. 2018. Dags with no tears: Continuous optimization for structure learning. *Advances in neural information processing systems* 31.