ABSTRACT
Anomaly detection in multivariate time series plays an important role in monitoring the behaviors of various real-world systems, e.g., IT system operations or manufacturing industry. Previous approaches model the joint distribution without considering the underlying mechanism of multivariate time series, making them complicated and computationally hungry. In this paper, we formulate the anomaly detection problem from a causal perspective and view anomalies as instances that do not follow the regular causal mechanism to generate the multivariate data. We then propose a causality-based anomaly detection approach, which first learns the causal structure from data and then infers whether an instance is an anomaly relative to the local causal mechanism to generate each variable from its direct causes, whose conditional distribution can be directly estimated from data. In light of the modularity property of causal systems, the original problem is divided into a series of separate low-dimensional anomaly detection problems so that where an anomaly happens can be directly identified. We evaluate our approach with both simulated and public datasets as well as a case study on real-world AIops applications, showing its efficacy, robustness, and practical feasibility.

CCS CONCEPTS
• Computing methodologies → Anomaly detection.

KEYWORDS
Anomaly detection, Time series, Causality

1 INTRODUCTION
Multivariate time series data is ubiquitous in monitoring the behavior of complex systems in real-world applications, such as IT operations management, manufacturing industry and cybersecurity [2, 17, 23]. Such data includes the measurements of the monitored components, e.g., the operational KPI metrics such as CPU/database usages in an IT system. An important task in managing these complex systems is to detect unexpected observations deviated from normal behaviors and notify the operators timely to resolve the underlying issues. The task of anomaly detection in multivariate time series aims to tackle this issue and has been actively studied in machine learning. Anomaly detection is one of the critical machine learning techniques to automate the identification of issues and incidents for improving system availability in AIOps (AI for IT Operations) [11].

In recent years, various approaches have been developed to detect anomalies in multivariate time series data. In general, there are two kinds of directions commonly explored, i.e., treating each dimension individually using univariate time series anomaly detection algorithms [13, 31, 36], and treating all the dimensions as an entity using multivariate time series anomaly detection algorithms [25, 35, 39]. The first direction ignores the dependencies between different time series, so it may be problematic especially when sudden changes of a certain dimension do not necessarily mean failures of the whole system or the relations among the time series become anomalous [38]. The second direction takes the dependencies into consideration, which are more suitable for real-world applications where the overall status of a system is more concerned about than a single dimension.

In real-world scenarios, it is common that no or very few anomaly labels are available in historical data, making unsupervised multivariate time series anomaly detection algorithms more practical. Among them, the most popular techniques include clustering-based methods [6, 8, 19], probabilistic methods [7] and classification-based method [22]. Recently, deep learning techniques receive much attention in anomaly detection, e.g., DAGMM [39], LSTM-VAE [25] and OmniAnomaly [35], which infer dependencies between different time series and temporal patterns within one time series implicitly. However, most of deep learning based methods exploit complex models and require a significant amount of time to train. The dependencies inferred by deep learning models do not represent the underlying process of generating the observed data and the asymmetric causal relationships between time series are ignored so that the learned representations might not be appropriate for anomaly detection. Besides, it is hard for them to identify the root causes when an anomaly occurs.

To overcome these disadvantages, we take a causal perspective to view anomalies in multivariate time series as instances that do not follow the regular causal mechanism, and propose a novel causality-based anomaly detection approach inspired by this new anomaly definition. Specifically, our approach leverages the causal structure discovered from data so that the joint distribution of multivariate time series is factorized into simpler modules where each module corresponds to a local causal mechanism, reflected by the corresponding conditional distribution. Those local mechanisms are modular or autonomous [26], and can then be handled separately. In light of this modularity property, the problem is then
naturally decomposed into a series of low-dimensional anomaly detection problems. Each sub-problem is concerned with a local mechanism. Moreover, because we focus on issues with the separate local causal mechanisms, the approach can identify the root causes of an anomaly at the same time. The main contributions of this paper are summarized below.

- We reformulate anomaly detection of multivariate time series from a causality perspective, which helps understand where and how anomalies happen and facilitates anomaly detection in light of the understanding.
- We propose a novel anomaly detection approach that decomposes the multivariate time series anomaly detection problem into a series of separate low-dimensional anomaly detection problems by exploiting the causal structure discovered from data, which not only detects the anomalies more accurately but also offers a natural way to find their root causes.
- We perform empirical studies of evaluating our approach with both simulated and public datasets as well as a case study for an internal real-world AIOps application of cloud services, validating its efficacy and robustness to different causal discovery techniques and settings.

To the best of our knowledge, our paper is the first one to formally formulate and tackle multivariate anomaly detection from the causality perspective. The formulation helps understand what an anomaly is: An anomaly is a data point that does not follow the regular data-generating process. The modularity property makes our approach simpler to train, suitable for real-world applications and easier for root cause analysis. Our method can detect those anomalies that are hard for the approaches based on modeling marginal/joint distributions only, illustrating the benefit of the causal view and treatment of anomalies.

2 RELATED WORK

Anomaly detection methods for univariate time series can be applied to each dimension of multivariate time series. Popular univariate anomaly detection techniques include statistical or distance-based methods, e.g., KNN [1, 8], One-Class SVM [22], and probabilistic methods [7]. These methods are computationally efficient and suitable for high dimensional data. But their performance degrades faced with long-term anomalies since the temporal patterns within time series are ignored. To address this issue, temporal prediction methods, e.g., ARIMA, SARIMA [13], Prophet [36], SR-CNN [31], and DONUT [20], have been proposed to model temporal dependencies/autocorrelations. However, these methods treat each dimension individually and ignore the correlations between different time series. As shown in Figure 1, they cannot identify the anomaly corresponding to the abnormal causal mechanism.

Recent years have seen the increasing popularity of unsupervised methods using deep learning techniques, which can infer the correlations between time series. For example, DAGMM [39] combines an autoencoder with a Gaussian mixture model to model the joint distribution. MSCRED [37] utilizes the system signature matrix to model the correlations and temporal patterns. LSTM-VAE [25] combines LSTM with VAE and models temporal dependencies through LSTM. OmniAnomaly [35] learns robust time series representations with a stochastic variable connection and a planar normalizing flow. USAD [2] uses adversarially trained autoencoders inspired by GANs, providing fast training. However, these methods model the joint distribution directly without considering the process behind multivariate time series, and an anomaly that happens to a local mechanism in the process might not change the joint distribution dramatically. Besides, it is difficult for them to leverage the domain knowledge of the monitored system, e.g., the known causal dependencies between time series, and to provide explanations that are crucial for root cause analysis and remediation when an anomaly occurs. Finally, our work also differs substantially from existing studies [28, 29] which though explore causality in anomaly detection in different ways, but do not use the causal mechanism to model anomalies in time series.

3 CAUSALITY-BASED ANOMALY DETECTION

Given a multivariate time series $X$ with length $T$ and $d$ variables, i.e., $X = \{x_1, x_2, \cdots, x_d\} \in \mathbb{R}^{T \times d}$, let $x(t)$ be the observation of the $i$th variable measured at time $t$. Anomaly detection is the task of identifying anomalies after time step $T$ that differ from the regular points in $X$ significantly.

3.1 Why causality matters

Let us consider a simple example shown in Figure 1, i.e., the measurements of three components $x, y, z$ with causal structure $x \rightarrow y \rightarrow z$. An anomaly labeled by a black triangle happens at time step 40, where the causal mechanism between $x$ and $y$ becomes abnormal. Typically it is hard to find such anomaly based on the marginal distributions or the joint distribution. But from local causal mechanism $p(y|x)$, such anomaly becomes obvious, e.g., $p(y|x)$ is much lower than its normal values. In this example, at time step 40 the probability densities $p(x) = 0.786, p(y) = 1.563, p(z) = 1.695, p(x, y, z) = p(x)p(y|x)p(z|y) = 0.046$ while $p(y|x) = 0.011$, meaning that it is easier to find this anomaly by examining the local causal mechanism $p(y|x)$. A real-world motivation example can be found in Figure 3.

If the causal structure of the underlying process is given, we examine whether each variable in the time series follows its regular causal mechanism. The causal mechanism can be represented by the structural equation model, i.e.,

$$x(t) = f_j(\mathbb{P}A(x(t)), \epsilon_j(t)), \quad \forall i = 1, \cdots, d, \quad (1)$$

where $f_j$ are arbitrary measurable functions, $\epsilon_j(t)$ are independent noises and $\mathbb{P}A(x(t))$ represents the causal parents of $x(t)$ including both lagged and contemporaneous ones. This causal structure can also be represented by a causal graph $G$ whose nodes correspond to the variables $x(t)$ at different time lags. In this paper, we assume the graph $G$ is a directed acyclic graph (DAG) and the causal relationships are stationary unless an anomaly occurs. According to the causal Markov factorization, the joint distribution of $x(t) = (x_1(t), x_2(t), \cdots, x_d(t))$ can be factored as

$$
\mathbb{P}[x(t)] = \prod_{i=1}^{d} \mathbb{P}[x(t) | \mathbb{P}A(x(t))], \quad (2)
$$

The local causal mechanism, corresponding to these conditional distribution terms, are known to be irrelevant to each other in a
An anomaly occurs at time step \( y \) if there are two cases to be considered: \( X \) time series \( P \) are violated due to an anomaly event such as a system failure, e.g., the causal mechanism between a variable and its causal parents

\[ \lambda = 0.0114 \]

This definition states that an anomaly happens in the system if the causal mechanism between a variable and its causal parents are violated due to an anomaly event such as a system failure, e.g., the local causal effect dramatically varies (Fig 1) or a big change happens on a variable and this change propagates to its children. Different from previous approaches, the anomaly detection problem can be divided into several low-dimensional subproblems based on this definition, e.g., by checking whether each variable \( x_i(t) \) follows the regular conditional distribution \( \mathbb{P}[x_i(t) | \mathcal{PA}(x_i(t))] \).

3.2 Method

In this paper, we consider the unsupervised learning setting where time series \( X \) is given as the training data for learning the conditional distributions. Thus, the training objective aims to maximize the log likelihood given the observation data, i.e., maximizing

\[ L(X) = \sum_{i=1}^{T} \sum_{t=1}^{d} \log \mathbb{P}[x_i(t) | \mathcal{PA}(x_i(t))] \].

\( C_R \) be the set of variables with no causal parents in \( \mathcal{G} \). There are two cases to be considered:

- Variable with no parents \((i \in C_R)\): In this case, we can model \( \prod_{i \in C_R} \mathbb{P}[x_i(t)] \) by applying any existing method for modeling univariate or multivariate time series with the historical data \( H_i(t) = \{x_i(t), \ldots, x_i(t-1)\} \) of \( x_i \), meaning that our framework allows to leverage the state-of-the-art time series models.

Algorithm 1 outlines our causality-based anomaly detection approach which includes causal graph discovery, conditional distribution estimation and anomaly detection. The training procedure:

**Input**: training data \( X = \{x_i\}_{i=1}^{d} \in \mathbb{R}^{T \times d} \), test data \( Y = \{y_i\}_{i=1}^{d} \in \mathbb{R}^{T \times d} \), and threshold \( \lambda \).

**Training procedure:**

1. Infer the causal graph \( \mathcal{G} \) via causal discovery techniques, e.g., FGES [9, 10, 24], PC [33]. If the \( \mathcal{G} \) is a partial DAG, convert it into a DAG by the method in [12];
2. For each variable \( i \in C_R \), train the model \( M_i \) that estimates the conditional distribution \( \mathbb{P}[x_i(t) | \mathcal{PA}(x_i(t))] \) with data \( \{x_i(t), \mathcal{PA}(x_i(t))\}_{t=1}^{T} \);
3. For the variables in \( C_R \), train the model \( M_R \) that estimates the distribution \( \prod_{i \in C_R} \mathbb{P}[x_i(t)] \) with data \( \{x_i(t) | i \in C_R\}_{t=1}^{T} \).

**Detection procedure:**

1. for \( t = 1 \) to \( T \) do
2. Compute the anomaly score of \( y(t) \): \( \Lambda(y(t)) = 1 - \min \{\{M_i(y(t)) | i \notin C_R \cup M_R(y(t))\} \}
3. Set anomaly label \( l_t = 1 \) if \( \Lambda(y(t)) > \lambda \) or \( 0 \) o/w;

A point is labeled as an anomaly if its anomaly score is larger than a certain threshold. Intuitively, the purpose of using the minimum function is that we want the algorithm to report an anomaly if any of the metrics (root variables) or local causal mechanisms (conditional probabilities) becomes abnormal. We found that this treatment works well in the experiments. Note that there are alternative ways to compute the final anomaly score; Heard and Rubin-Delanchy [15] compared six methods for combining p-values from individual tests, and showed that taking the minimum is sensitive to the smallest p-value, which is suitable for reporting anomalies that any of the metrics is abnormal.

3.2.1 Causal discovery: Our approach needs exploit the causal structure underlying the data. A traditional way to find causal relations is to use interventions or randomized experiments, which are
generally too expensive, too time-consuming, or even impossible. Discovering causal information by analyzing purely observational data, known as causal discovery, is then an important problem [27, 33, 34]. Multiple algorithms have been developed for causal discovery from independent and identically distributed (i.i.d.) or time series data, and their results are asymptotically guaranteed under corresponding assumptions. In this paper we choose causal discovery algorithms such as PC [33], FGES [9, 10, 24], depending on whether we are given temporal data (with time-delayed causal relations) and whether the causal relations are linear or nonlinear. For example, we apply FGES with SEM-BIC score if the variables are linearly related and apply FGES with generalized score function [16] if they are non-linearly correlated. One concern is whether the missing or incorrect causal links in the inferred causal graph have a big impact on the performance of our approach. We performed an empirical study of this impact with public datasets, which shows that interestingly, our approach is robust to the inferred causal graph. The complexity of PC and GES highly depends on the density of the causal graph. Specifically, FGES is highly scalable when dealing with linear models [30]. In real-world applications, e.g., the public datasets in the experiments, even though the variables may not be exactly linearly correlated, FGES can still generate reasonable causal graphs that are good enough for our anomaly detection approach.

3.2.2 Estimating local causal mechanism. After the causal Markov factorization, it becomes easier to model the joint distribution compared to the previous approaches, e.g., the conditional distributions representing local causal mechanisms can be estimated using simpler ML models.

For modeling \( P[x_i(t)|PA(x_i(t))] \), one can utilize kernel conditional density estimation [14], mixture density network [5], conditional VAE (CVAE) [32] or even prediction models such as MLP or CNN [4]. Suppose that \( PA \) is the set of \( x_i \)'s causal parent variables, \( \tau_j \) is the causal time lag for a parent \( x_j \) and \( \tau^* \) is the maximum time lag in \( G^* \); then we define

\[
P^*(x_i(t) = \{x_i(t−\tau^*), \ldots, x_i(t−\tau_j) \mid j \in PA\}.
\]

Time lag \( \tau_j = 0 \) if \( x_j \) is a contemporaneous causal parent of \( x_i \). For causal parent \( x_j \), more of its historical data can also be included, e.g., a window with size \( k: \{x_j(t−\tau−k+1), \ldots, x_j(t−\tau_j) \mid j \in PA\} \). Therefore, the problem becomes estimating the conditional distribution from the empirical observations \( \{(x_i(t), c_i(t))\}_{t=1}^T \) where \( c_i(t) = PA^*(x_i(t)) \). In this paper, we apply CVAE to model such conditional distribution. The reason why choosing CVAE is that it can be trained fast with a simple architecture and achieve good performance as shown in our experiments. The empirical variational lower bound of CVAE is

\[
L(x; c; \theta, \phi) = \frac{1}{n} \sum_{k=1}^n \log p_\theta(x|c, z_k) - KL(q_\phi(z|x, c) \parallel p_\phi(z|x, c)),
\]

where \( q_\phi(z|x, c) \) and \( p_\theta(x|c, z_k) \) are MLPs and \( p_\phi(z|x, c) \) is a Gaussian distribution. Given \( (x_i(t), c_i(t)) \), CVAE outputs \( \hat{x}_i(t) \) - reconstruction of \( x_i(t) \), then \( P[x_i(t)|c_i(t)] \) is measured by the distribution of \( \|\hat{x}_i(t) - x_i(t)\| \).

For modeling \( \prod_{i \in CR} P[x_i(t)] \), one way to estimate this distribution is to handle each variable in \( CR \) individually via univariate time series models, e.g., ARIMA [13], SARIMA [18], CNN [4]. The other way is to handle the variables in \( CR \) together by utilizing the models for multivariate time series anomaly detection, e.g., Isolation Forest (IF) [21], AE [3], LSTM-VAE [25]. The training data for such model includes all the observations of the variables in \( CR \), i.e., \( \{(x_i(t)) \mid i \in CR\}_t \). For example, the training data for a forecasting based method is \( \{(\{x_i(t), x_i(t−k), \ldots, x_i(t−1)\}) \mid i \in CR\}_T \) where \( x_i(t) \) is predicted by a window of its previous data points.

Our approach reduces to the previous univariate/multivariate time series AD approaches if the causal graph is empty, i.e., no causal relationships are considered. When the causal relationships are available obtained by domain knowledge or data-driven causal discovery techniques, our approach can easily utilize such information and reduces the efforts in joint distribution estimation.

3.2.3 Negative effect of training anomalies. The anomalies in the training data may decrease detection performance. Our experiments show that this issue does not affect the anomaly detection performance much, which is expected to be the case because there are relatively few anomalies in the data. Typically, there are two cases where anomalies in training data may have obvious negative impacts on performance. One case is that the value of a metric at certain timestamps becomes extremely large, which can affect the conditional probability estimation. In this case, one can simply remove those values (by ignoring those timestamps) based on statistical rules, e.g., removing them if the absolute value is larger than some threshold in the preprocessing step. The other case is that the proportion of anomalies is relatively large. In this case, we can consider an iterative solution that iteratively updates the causal graph and anomaly detection model, i.e.,

(1) Estimate causal graph \( G \) and train models \( M_I, M_R \) with the training data;

(2) Remove the anomalies detected by \( M_I, M_R \) in the training data and then go to Step (1).

We repeat the above two steps until the estimated causal model (including the estimated causal structure and quantitative model, e.g., causal coefficients in the linear case) converges.\(^2\) We conducted an experiment with a simulation dataset to empirically study this iterative solution for handling noises in training data.

3.2.4 Root cause analysis. Root Cause Analysis (RCA) aims to identify root causes while alerting anomalies in multivariate time series. Thank to the modularity property implied in our anomaly definition, our approach can naturally identify the root causes when an anomaly event occurs. Here is our definition of root causes.

Definition 3.2. The root causes of an anomaly point \( x(t) \) are those variables \( x_i \) such that \( x_i(t) + P[x_i(t)|PA(x_i(t))] \), e.g., an anomaly happens on the local causal mechanism related to those variables.

This definition indicates that \( x_i \) is one of the root causes if the local causal mechanism of variable \( x_i(t) \) is violated. Recall the

\(^2\)Note that when the remaining data points follow the same causal model, after further removing "anomalies", they will still follow it because we detect and remove "anomalies" according to the probabilities of the error terms and after removing them, the error terms are still independent from the direct causes.
4 EXPERIMENTS

This section evaluates the performance of our causality-based anomaly detection framework and compares it to other state-of-the-art methods. The experiments include: 1) evaluating our approach with simulated datasets and public datasets, 2) evaluating how much the inferred causal relationships help in anomaly detection, 3) analyzing how different causal graphs affect the performance, 4) a case study demonstrating the application of our approach for real-world anomaly detection in AIOps.

The anomaly detection performance is assessed by the precision, recall and F1-score metrics in a point-adjust manner, i.e., all the anomalies of an anomalous segment are considered as correctly detected if at least one anomaly of this segment is correctly detected while the anomalies outside the ground truth anomaly segment are treated as normal. We apply FGES [10] and PC [33] to discover the causal graph. For $M_t$, we choose CVAE [32]. For $M_{t-1}$, we tested the univariate model and other methods such as IF [21], AE [3], LSTM-VAE [25] in our experiments.

4.1 Simulation datasets

We generate the simulation dataset as follows: 1) Generate an Erdős Rényi random graph $\mathcal{G}$ with number of nodes $n$ and edge creation probability $p$, then convert it into a DAG. 2) For the nodes with no parents in $\mathcal{G}$, randomly pick a signal type from “harmonic”, “pseudo periodic” and “auto-regressive” and generate a time series with length $T$ according to this type. 3) For a node $x_i$ with parents $PA(x_i)$, we consider linear relationship $x_i = \sum_{j \in PA(x_i)} w_j x_j + \epsilon$ and nonlinear relationship $x_i = \sum_{j \in PA(x_i)} w_j \tanh(x_j) + \epsilon$ where $w_j$ is uniformly sampled from $[0, 5, 2.0]$ and $\epsilon$ is uniformly sampled from $[-0.1, 0.1]$. The time series for those nodes are generated in a topological order.

The next step is to add anomalies into the generated time series. We consider three types of anomalies. The first one is a "measurement" anomaly where the causal mechanism is normal but the observation is abnormal due to measurement errors, i.e., randomly pick a node $x_i$, a time step $t$ and a scale $s$ (uniformly sampled from $[0, 3]$), and then set $x_i(t) = x_i(t) - \text{median}(x_i) \star s + \text{median}(x_i)$. The second one is an "intervention" anomaly, i.e., after generating anomalies for some nodes, those anomaly values propagate to the children nodes according to the causal relationships. The third one is an "effect" anomaly where anomalies only happen on the nodes with no causal children.

Performance comparison. In the experiments, we consider six settings derived from the combinations of "linear/nonlinear" and "measurement/intervention/effect". The simulated time series has 15 variables with length 20000, where the first half is the training data and the rest is the test data. The percentage of anomalies is about 10%. Table 1 shows the performance of different unsupervised multivariate time series anomaly detection methods with the generated simulated dataset. Clearly, our method outperforms all the other methods. It achieves significantly better F1 scores when the relationships are nonlinear or the anomaly type is "intervention", e.g., ours obtains F1 score 0.759 for the "nonlinear, intervention" setting, while the best F1 score achieved by the others is 0.589. In the "linear, measurement/effect" setting, DAGMM has a similar performance with ours because the data can be modeled well by applying dimension reduction followed by a Gaussian mixture model. But when the relationships become nonlinear, it becomes harder for DAGMM to model the data. This experiment shows that the causal mechanism plays an important role in anomaly detection. Modeling joint distribution without considering causality can lead to a significant performance drop.

Training anomalies. When the fraction of anomalous points is large in the training data, these anomalies may decrease detection performance since the discovered causal graph may not be accurate. In this case, we can apply the solution discussed in Section 3.2.3, updating the causal graph and anomaly detection model iteratively. In this experiment, the training and test data are generated under the setting "linear/measurement", and a large proportion of noises are added into the training data, i.e., adding additional Gaussian noises to the first 20% data points in the training data. These noisy data points makes estimating accurate causal graphs harder via causal discovery algorithms. In each iteration, 3% data points are detected as anomalies and removed. Figure 2(a) plots a subset of training time series data in this experiment where the first 2000 data points have large Gaussian noises, and Figure 2(b) shows the detection performance on the test dataset measured by the F1 scores over each iteration. In the beginning the discovered causal graph has more errors due to the noises in the training data, leading to the low F1 score. After each iteration, our approach removes the detected anomalies from the training data, making the discovered causal graph more accurate in the next iteration, so that the detection performance increases consistently. This experiment empirically verifies our "iterative updates" approach in the case where the training data has a large portion of anomalies. Figure 2(c) plots the difference between the adjacency matrices of two consecutive estimated causal graphs, which increases first then decreases and converges to 0 since the distribution of training data gradually changes from a mix of noises and regular points to regular points only. This experiment considers an extreme case that the proportion of anomalies and the magnitude of anomalies are large. In practical applications where the proportion of anomalies in training data is relatively small, e.g., the public datasets, there is no need to apply this iterative approach, i.e., one iteration is good enough.
Table 1: Performance comparison (F1-scores) of our approach and other methods on the simulation datasets.

|                | Linear/Measu. | Linear/Inter. | Linear/Effect | Nonlinear/Measu. | Nonlinear/Inter. | Nonlinear/Effect |
|----------------|---------------|---------------|---------------|------------------|------------------|------------------|
| IF             | 0.374         | 0.403         | 0.220         | 0.336            | 0.422            | 0.367            |
| AE             | 0.386         | 0.359         | 0.240         | 0.392            | 0.390            | 0.363            |
| VAE            | 0.343         | 0.328         | 0.208         | 0.396            | 0.377            | 0.306            |
| LSTM-VAE       | 0.457         | 0.454         | 0.485         | 0.581            | 0.545            | 0.393            |
| DAGMM          | 0.746         | 0.542         | 0.721         | 0.456            | 0.589            | 0.359            |
| USAD           | 0.252         | 0.260         | 0.220         | 0.346            | 0.302            | 0.279            |
| Ours           | **0.791**     | **0.757**     | **0.740**     | **0.757**        | **0.759**        | **0.637**        |

Figure 2: Empirical study on our “iterative updates” approach discussed in Section 3.2.3 for handling large noise in the training data. (a) Samples in the noisy training data where the first 2000 data points have large Gaussian noises. (b) The detection performance (F1 scores) on the test data over iterations. (c) The difference between the adjacency matrices of two consecutive discovered causal graphs.

4.2 Public real datasets

Five public datasets were used in our experiments: 1) Server Machine Dataset (SMD) [35]: It contains data from 28 server machines monitored by 33 metrics. 2) Secure Water Treatment (SWaT) [23]: it consists of 11 days of continuous operation, i.e., 7 days collected under normal operations and 4 days collected with attacks. 3) Water Distribution (WADI) [23]: It consists of 16 days of continuous operation, of which 14 days were collected under normal operation and 2 days with attacks. 4) Soil Moisture Active Passive (SMAP) satellite and Mars Science Laboratory (MSL) rover Datasets [17], which are two public datasets expert-labeled by NASA.

Performance comparison. We compare our causality-based approach with seven unsupervised approaches, e.g., AE [3], DAGMM [39], OmniAnomaly [35], USAD [2]. Table 2 shows the results on three representative datasets where the results of OmniAnomaly are copied from the paper [2]. Overall, IF, AE, VAE and DAGMM have relatively lower performance because they neither exploit the temporal information nor leverage the causal relationships between those variables. LSTM-VAE, OmniAnomaly and USAD perform better than these four methods since they utilize the temporal information via modeling the current observations with the historical data. Our approach exploits the causal relationships besides the temporal information, achieving significantly better results than the other methods in 4 out of 5 datasets including SWaT and WADI, e.g., ours has the best F1 score 0.918 for SWaT and 0.818 for WADI, while the best F1 scores for SWaT and WADI by other methods are 0.846 and 0.767, respectively. Our approach also outperforms the others in SMAP and MSL. For each public datasets, Table 3 reports the best metrics that can be achieved by choosing the best thresholds in the test datasets. Clearly, if we are allowed to choose better thresholds, the metrics achieved by our approach can be much higher, e.g., F1-score 0.946 for SMAP and 0.913 for MSL.

Real-world motivating example. Figure 3 shows why causality matters with a real-world example. In SWaT [23], at timestamp 491, our causality-based approach detects a true anomaly where the causal mechanism between Metrics 1, 0 and 9 is violated (Metrics 0 and 9 are the causal parents of Metric 1). We plot the probability density of the reconstruction error based on the causal mechanism (top left figure), where the black triangle is the anomaly. Clearly, this anomaly can be easily identified w.r.t. the p-value. But if we check the probability density of the “joint” reconstruction error by AE (top right figure), this anomaly cannot be found w.r.t. the p-value. The bottom figure plots the time series data of these three metrics. Intuitively, we can observe that Metric 1 has a peak value when Metric 0 is low, and Metric 1 is low when Metric 0 is high. In the range (450, 550), this causal mechanism is violated, e.g., Metric 0 is high while Metric 1 is also high. This type of anomalies is hard to be identified by checking joint or marginal distributions.

Ablation study on $M_R$. The next experiment evaluates the effect of the causal information on anomaly detection. For an anomaly detection method $\mathcal{A}$ such as IF and AE, we compare $\mathcal{A}$ with our approach “ours + $\mathcal{A}$” that uses CVAE for $M_i$ and $\mathcal{A}$ for $M_R$, where $M_i$ and $M_R$ estimate $P(x_i(t)|PA(x_i(t)))$ and $\prod_{C \in C_R} P(x_i(t))$ respectively. We report the metrics as mentioned above and the best metrics achieved by choosing the best thresholds in the test datasets.

Table 4 shows the performance of our approach with different $M_R$, where $M_R = \emptyset$ means that the anomalies are detected by $M_i$ only without using $M_R$. By comparing this table with Table 2 we
Table 2: Performance comparison of our approach and other methods on the public real datasets.

| Dataset | SMD   | SMAP   | MSL   | SWaT   | WADI   |
|---------|-------|--------|-------|--------|--------|
|         | Prec. | Recall | F1   | Prec.  | Recall | F1   | Prec. | Recall | F1   |
| IF      | 0.796 | 0.997 | 0.885 | 0.815 | 0.591 | 0.685 | 0.854 | 0.922 | 0.887 | 0.998 | 0.669 | 0.801 | 0.541 | 0.794 | 0.644 |
| AE      | 0.879 | 0.997 | 0.934 | 0.806 | 0.585 | 0.678 | 0.858 | 0.892 | 0.875 | 0.999 | 0.656 | 0.792 | 0.595 | 0.762 | 0.668 |
| VAE     | 0.853 | 0.999 | 0.921 | 0.808 | 0.588 | 0.681 | 0.771 | 0.656 | 0.709 | 0.999 | 0.656 | 0.792 | 0.616 | 0.855 | 0.716 |
| LSTM-VAE| 0.931 | 0.998 | 0.963 | 0.818 | 0.591 | 0.686 | 0.859 | 0.911 | 0.884 | 0.997 | 0.689 | 0.815 | 0.658 | 0.920 | 0.767 |
| DAGMM   | 0.704 | 0.998 | 0.825 | 0.800 | 0.877 | 0.837 | 0.900 | 0.864 | 0.882 | 0.829 | 0.767 | 0.797 | 0.639 | 0.501 | 0.412 |
| OmniAnomaly | 0.981 | 0.943 | 0.944 | 0.758 | 0.975 | 0.853 | 0.901 | 0.889 | 0.895 | 0.722 | 0.983 | 0.835 | 0.265 | 0.980 | 0.417 |
| USAD    | 0.931 | 0.962 | 0.938 | 0.769 | 0.983 | 0.863 | 0.861 | 0.964 | 0.910 | 0.987 | 0.740 | 0.846 | 0.645 | 0.322 | 0.430 |
| Ours    | 0.886 | 0.999 | 0.939 | 0.874 | 0.982 | 0.925 | 0.867 | 0.961 | 0.912 | 0.945 | 0.892 | 0.918 | 0.749 | 0.901 | 0.818 |
| (std)   | ±0.004 | ±0.000 | ±0.002 | ±0.001 | ±0.006 | ±0.003 | ±0.003 | ±0.011 | ±0.007 | ±0.009 | ±0.016 | ±0.008 | ±0.021 | ±0.029 | ±0.023 |

Table 3: The best performance of our approach with the public real datasets.

| Dataset | Precision* | Recall* | F1* |
|---------|------------|---------|-----|
| SWaT    | 0.951 ± 0.011 | 0.930 ± 0.011 | 0.940 ± 0.004 |
| WADI    | 0.903 ± 0.029 | 0.951 ± 0.033 | 0.926 ± 0.017 |
| SMAP    | 0.929 ± 0.018 | 0.965 ± 0.009 | 0.946 ± 0.007 |
| MSL     | 0.883 ± 0.021 | 0.947 ± 0.020 | 0.913 ± 0.008 |
| SMD     | 0.996 ± 0.001 | 0.999 ± 0.000 | 0.999 ± 0.001 |

Table 4: Performance of our method using different models for $M_R$ in SWaT and WADI. "*" means the best metrics. $M_R = ∅$ means anomalies are detected by $M_I$ only w/o using $M_R$.

| Dataset | SWaT (G has 102 edges and 5 nodes with no parents) | WADI (G has 249 edges and 17 nodes with no parents) |
|---------|-------------------------------------------------|-------------------------------------------------|
|         | $M_R$                                           | $M_R$                                           |
|         | Prec. | Rec. | F1 | Prec. | Rec. | F1 |
| ∅       | 0.952 | 0.874 | 0.911 | 0.950 | 0.929 | 0.940 |
| IF      | 0.947 | 0.893 | 0.919 | 0.946 | 0.945 | 0.945 |
| AE      | 0.958 | 0.900 | 0.928 | 0.963 | 0.920 | 0.941 |
| LSTM-VAE | 0.954 | 0.874 | 0.912 | 0.951 | 0.936 | 0.944 |
| ∅       | 0.749 | 0.920 | 0.826 | 0.873 | 0.979 | 0.923 |
| IF      | 0.738 | 0.920 | 0.819 | 0.948 | 0.920 | 0.934 |
| AE      | 0.789 | 0.920 | 0.850 | 0.931 | 0.979 | 0.955 |
| LSTM-VAE | 0.748 | 0.920 | 0.825 | 0.949 | 0.920 | 0.934 |

Ablation study on causal graph $G$. We also studied the effects of different parameters for discovering causal graphs on the performance of our approach. The parameters that we investigated are "max degree" and "penalty discount" in FGES, both of which affect the structure of the causal graph, e.g., sparsity, indegree, outdegree. In this experiment, we use 6 different "max degree" [5, 6, 7, 8, 9, 10] and 6 different "penalty discount" [20, 40, 60, 80, 100, 120]. Smaller "max degree" or larger "penalty discount" leads to more sparse graphs with less edges, e.g., for SWaT, the number of the edges in $G$ is [70, 79, 88, 95, 98, 102] when "max degree" $= [5, 6, 7, 8, 9, 10]$, respectively.

Figure 5 plots the detection precision, recall and F1 score obtained with different "max degree" and "penalty discount". For SWaT, these two parameters don’t affect the performance much. For WADI, when "max degree" decreases (the causal graph becomes more sparse) or "penalty discount" decreases (the causal graph has more false positive links), the performance also decreases but it doesn’t drop much, i.e., the worst F1 score is still above 0.65. When "max degree" $> 6$ and "penalty discount" $> 40$, we got similar performance, e.g., the F1 score is around 0.8, showing that our approach is robust to the changes of the inferred causal graph. In practice, the causal graph is not required to be accurate, namely, we just can observe that "ours + $A$" performs much better than using $A$ only, e.g., "ours + AE" achieves F1 score 0.850 for WADI, while AE obtains 0.668 for WADI. If $M_R$ is not used in anomaly detection, we get a performance drop in terms of F1 score. For example, the best F1 score drops from 0.934 to 0.923 for WADI.
need to ensure that it doesn’t contain too many missing links or false positive links.

Besides FGES, we can apply other methods such as the PC algorithm [33] to infer the causal graphs. The causal graphs inferred by PC are probably different from those computed by FGES. Our experiments show that our anomaly detection approach is stable even though the causal graphs are different. Table 5 compares the performance of our approach with FGES and PC. For SWaT, using FGES and using PC have similar performance. For WADI, using PC performs worse than using FGES, but the F1-score 0.768 is still better than the other approaches. The performance drop is because the causal graph discovered by FGES is more accurate than PC in WADI.

### 4.3 Case Study: Real-world Application in AIOps

Our last experiment is to apply our method for a real-world anomaly detection task in AIOps, where the goal is to monitor the operational key performance indicator (KPI) metrics of database services for alerting anomalies and identifying root causes in order to automate remediation strategies and improve database availability in cloud-based services. In this application, we monitor a total of 61 time series variables measuring the KPI metrics of database services, e.g., read/write IO requests, CPU usage, DB time. The data in this case study consists of the latest one-month measurements. According to the feedback from domain experts, most of the inferred causal relationships shown in Figure 4 are consistent with the known domain knowledge. For example, the discovered links Redo (redo size) \( \rightarrow \) Lfpw (log file parallel write) \( \rightarrow \) Lfs (log file sync) \( \rightarrow \) COMT (commit) are exactly the same as the domain knowledge.

The incidences that happened are relatively rare, e.g., 2 major incidences one month, and our anomaly detection approach correctly detect these incidences. Therefore, we focus on the root cause analysis in this case study. Figure 4 shows an example of one major incidence, showing several abnormal metrics such as DBt (DB time), Lfs (log file sync), APPL (application), TotPGA (total PGA allocated) and a part of the causal graph. The root cause scores computed by our method are highlighted. We can observe that the top root causes metrics are APPL, DBt and TotPGA, all of which correspond to application or database related issues for the incident as validated by domain experts.

### 5 CONCLUSIONS

Most previous approaches for multivariate time series anomaly detection model the joint distribution directly without considering the underlying causal process of the observed time series data. This paper presented a new definition and formulation of anomalies in multivariate time series from a causal perspective, and proposed a novel approach that exploits the causal structures discovered from data to help detect anomalies more accurately and identify the root causes robustly according to the local causal mechanism. Our experiments on both simulation and real datasets demonstrated the efficacy, robustness and practical feasibility of the proposed approach in real-world applications.
**REFERENCES**

[1] Fabrizio Angiolillo and Clara Pizzuti. 2002. Fast Outlier Detection in High Dimensional Spaces. In Principles of Data Mining and Knowledge Discovery: 15–27.

[2] Julien Audibert, Pietro Michiardi, Frédéric Guyard, Sébastien Marti, and Maria A. Zuluaga. 2020. USAID: Unsupervised Anomaly Detection on Multivariate Time Series. In The 26th ACM SIGKDD Intl Conference on Knowledge Discovery & Data Mining (KDD ’20). 3395–3404.

[3] Pierre Baldi. 2012. Autoencoders, Unsupervised Learning, and Deep Architectures. In ICML Workshop on Unsupervised and Transfer Learning (PMLR, Vol. 27). PMLR, 37–49.

[4] Mikolaj Binkowski, Gautier Marti, and Philippe Donnat. 2018. Autoregressive Convolutional Neural Networks for Asynchronous Time Series. In Proceedings of the 35th International Conference on Machine Learning (PMLR, Vol. 80). PMLR, 580–589.

[5] Christopher M. Bishop. 1994. Mixture density networks. Technical Report.

[6] Ricardo J. G. B. Campello, Davoud Moulavi, Arthur Zimek, and Jörg Sander. 2015. Hierarchical Density Estimates for Data Clustering, Visualization, and Outlier Detection. *ACM Trans. Knowl. Discov. Data* 10, 1 (2015), 51 pages.

[7] Varun Chandola, Arindam Banerjee, and Vishal Kumar. 2009. Anomaly Detection: A Survey. *J. Mach. Learn. Res.* 11, 3 (2009), 58 pages.

[8] Wanpracha Art Chaowalitwongse, Ya-Ju Fan, and Rajesh C. Sachdeo. 2007. On the Time Series K-Nearest Neighbor Classification of Abnormal Brain Activity. *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans* 37, 6 (2007), 1005–1016.

[9] David Maxwell Chickering. 2002. Learning equivalence classes of Bayesian-network structures. *J. Mach. Learn. Res.* 2, 3 (2002), 445–498.

[10] David Maxwell Chickering. 2003. Optimal structure identification with greedy search. *J. Mach. Learn. Res.* 3, 3 (2003), 567–554.

[11] Yingnong Dang, Qingwei Lin, and Peng Huang. 2019. AIOps: real-world challenges and research innovations. In 2019 IEEE/ACM 41st International Conference on Software Engineering: Companion Proceedings (ICSE-Companion). IEEE. 4–5.

[12] Dorit Dor and Michael Tarsi. 1992. A simple algorithm to construct a consistent extension of a partially ordered graph. Technical Report.

[13] James Douglas Hamilton. 1994. Time Series Analysis. Princeton University Press.

[14] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. 2009. *The elements of statistical learning: data mining, inference and prediction.* Springer.

[15] N A Heard and P Rubin-Delanchy. 2018. Choosing between methods of combining p-values. *Biometrika* 105, 1 (Jan 2018), 239–246. https://doi.org/10.1093/biomet/ass076.

[16] Biwei Huang, Kun Zhang, Yizhuo Lin, Bernhard Schölkopf, and Clark Glymour. 2018. Generalized Score Functions for Causal Discovery. In Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD ’18). 1551–1560.

[17] Kyle Hundman, Valentino Constantinou, Christopher Laporte, Ian Colwell, and Tom Soderstrom. 2018. Detecting Spacecraft Anomalies Using LSTM and Non-parametric Dynamic Thresholding. In 24th ACM SIGKDD Intl Conf. on Knowledge Discovery & Data Mining. 387–395.

[18] Robin John Hyndman and George Athanasopoulos. 2018. *Forecasting: Principles and Practice* (2nd ed.). OTexts.

[19] István Kiss, Bela Genge, Piroiska Haller, and Gheorghe Sebestyén. 2014. Data clustering-based anomaly detection in industrial control systems. In 2014 IEEE 10th International Conference on Intelligent Computer Communication and Processing (ICCP). 275–281.

[20] Nikolay Laptew, Saeed Azimzadeh, and Ian Flint. 2015. Generic and Scalable Framework for Automated Time-Series Anomaly Detection. *IEEE/ACM Transactions on Computational Biology & Computer Medicine* 12, 11 (2015), 1–1.

[21] Fei Tony Liu, Kai Ming Ting, and Zhi-Hua Zhou. 2008. Isolation Forest. In *2008 Eighth IEEE International Conference on Data Mining*. 413–422.

[22] Laurry M. Manevitz and Malik Youssef. 2002. One-Class SVMs for Document Classification. *J. Mach. Learn. Res.* 2 (March 2002), 139–154.

[23] Aditya P. Mathur and Nils Ole Tippenthaler. 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.

[24] Christopher Meek. 1995. Causal Inference and Causal Explanation with Background Knowledge. In *Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence (UAI’95)*. Morgan Kaufmann Publishers Inc. 403–410.

[25] Daehyun Park, Yuuna Hoshi, and Charles Kemp. 2017. A Multimodal Anomaly Detector for Robot-Assisted Feeding Using an LSTM-Based Variational Autoencoder. *IEEE Robotics and Automation Letters* 2, 3 (2017), 1533–1539.

[26] Judea Pearl. 2009. *Causality: Models, Reasoning and Inference* (2nd ed.). Cambridge University Press, USA.

[27] J. Peters, D. Janzing, and B. Schölkopf. 2017. *Elements of Causal Inference - Foundations and Learning Algorithms*. The MIT Press, Cambridge, MA, USA.

[28] Huada Qiu, Yan Liu, Niranjan A. Subrahmanya, and Weichang Li. 2012. Granger Causality for Time-Series Anomaly Detection. In *IEEE Intl Conf. on Data Mining Workshop for Automated Time-Series Anomaly Detection (KDD ’12)*. 1593–1597.

[29] Robin John Hyndman and George Athanasopoulos. 2018. *Forecasting: Principles and Practice* (2nd ed.). OTexts.

[30] Joseph Ramsey, Madelyn Glymour, Ruben Sanchez-Romero, and Clark Glymour. 2016. A million variables and more: the Fast Greedy Equivalence Search algorithm for learning high-dimensional graphical causal models, with an application to functional magnetic resonance images. *International Journal of Data Science and Applications* 10, 4 (2020), 319–339.
[31] Hansheng Ren, Bixiong Xu, Yujing Wang, Chao Yi, Congrui Huang, Xiaoyu Kou, Tony Xing, Mao Yang, Jie Tong, and Qi Zhang. 2019. Time-Series Anomaly Detection Service at Microsoft. In 25th ACM SIGKDD Intl Conf. on Knowledge Discovery & Data Mining. 3009–3017.

[32] Kihyuk Sohn, Honglak Lee, and Xinchen Yan. 2015. Learning Structured Output Representation using Deep Conditional Generative Models. In Advances in Neural Information Processing Systems, C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett (Eds.), Vol. 28.

[33] Peter Spirtes and Clark Glymour. 1991. An Algorithm for Fast Recovery of Sparse Causal Graphs. Social Science Computer Review 9, 1 (1991), 62–72.

[34] P. Spirtes and K. Zhang. 2016. Causal discovery and inference: concepts and recent methodological advances. Applied Informatics 3 (2016). https://doi.org/10.1186/s40535-016-0018-x.

[35] Ya Su, Youjian Zhao, Chenhao Niu, Rong Liu, Wei Sun, and Dan Pei. 2019. Robust Anomaly Detection for Multivariate Time Series through Stochastic Recurrent Neural Network. In 25th ACM SIGKDD Intl Conference on Knowledge Discovery & Data Mining (KDD’19). 2828–2837.

[36] Sean J. Taylor and Benjamin Letham. 2018. Forecasting at Scale. The American Statistician 72, 1 (2018), 37–45.

[37] Chuxu Zhang, Dongjin Song, Yuncong Chen, Xinyang Feng, Cristian Lumezanu, Wei Cheng, Jingchao Ni, Bo Zong, Haifeng Chen, and Nitesh V. Chawla. 2019. A Deep Neural Network for Unsupervised Anomaly Detection and Diagnosis in Multivariate Time Series Data. In The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019. 1409–1416.

[38] Hang Zhao, Yujing Wang, Juanyong Duan, Congrui Huang, Defu Cao, Yunhai Tong, Bixiong Xu, Jing Bai, Jie Tong, and Qi Zhang. 2020. Multivariate Time-series Anomaly Detection via Graph Attention Network. CoRR abs/2009.02040 (2020).

[39] Bo Zong, Qi Song, Martin Renqiang Min, Wei Cheng, Cristian Lumezanu, Daeki Cho, and Haifeng Chen. 2018. Deep Autoencoding Gaussian Mixture Model for Unsupervised Anomaly Detection. In International Conference on Learning Representations.
1 EXPERIMENTAL SETUP AND PARAMETERS SETTINGS

For the implementation of our approach, we employ the CVAE to model conditional distributions $P(x_t|\mathcal{P}(x_t))$. We choose the same parameters for all the experiments on both simulated and public real datasets. The encoder and decoder in CVAE are both MLPs with hidden sizes $[10, 20, 10]$. The latent size is 5 and the prior distribution $p(z|c)$ is assumed to be the standard normal distribution (doesn’t depend on c). For training CVAE, the optimizer is ADAM with learning rate 0.001, batch size 1024 and epoch num 80.

For modeling $\prod_{t \in \mathbb{R}}P(x_t|\mathcal{P}(x_t))$, there are several options to choose in practical applications. For the simulated datasets and our internal AIOPs dataset, we choose a univariate anomaly detection method based on a CNN forecasting model. The CNN forecasting model consists of 4 residual blocks with 1D convolutional layers, i.e., the "(input channels, output channels)" pairs are (1, 8), (8, 16), (16, 32), (32, 64), followed by the concatenate of 1D adaptive average pooling and 1D adaptive max pooling. The output layer is a linear layer. For each residual block, it has two convolutional layers "(input channels, output channels)\rightarrow(output channels, output channels)". We choose ADAM as the optimizer with learning rate 0.001 (with decay), batch size 1024 and epoch num 50. The window size of the historical data for prediction is 20. For the public datasets, besides this CNN forecasting model, we can also choose isolation forest (IF) and autoencoder (AE). For IF, the max number of samples is 10000. For AE, the hidden sizes of the encoder are [25, 10, 5], the latent size is 5, and the hidden sizes of the decoder are [5, 10, 25].

For the simulated datasets, we apply FGES and set "max degree = 5" and "penalty discount = 20". For the public datasets, we apply FGES and set "max degree = 10" and "penalty discount = 100". For SMAP and MSL, we apply the PC algorithm with the default parameters. The library for causal discovery we used in this project is Tetrad $^1$. Smaller "max degree" or larger "penalty discount" in FGES leads to more sparse graphs with less edges. Table 1 lists the number of the edges in the causal graphs discovered with different parameters.

The reason why we choose these parameters such as CVAE hidden sizes $=[10, 20, 10]$ is as follows. For all the simulation datasets, the "max-degree" is set to 5 in FGES and the causal relations are instantaneous, meaning that the number of causal parents of each variable is at most 5 so that the input dimensions of the parent variables in CVAEs for modeling conditional probabilities are at most 5. For the public datasets, the "max-degree" is 10 and we found that there are instantaneous causal influences but not time-delayed ones, so the input dimensions of the parent variables in CVAEs are at most 10. That’s why we choose those parameters for the encoder and decoder. For a new dataset, if one considers a similar setting for causal discovery, he/she can use our parameters as default. In general, the input dimensions of CVAEs are at most "max-degree" + "time-lag", so one can choose the hidden sizes around this number. For modeling conditional probabilities, one can construct a validation set by splitting the training dataset. Under the Gaussian distribution assumption in CVAE, the overfitting issue can be found and avoided by measuring the reconstruction MSE loss.

For all the experiments, the detection thresholds are inferred by taking the 95th percentile of the detection scores in the test data, e.g., we choose $n = 95$ for SMD, SWaT and WADI, $n = 98$ for SMAP and MSL. For the other methods (except ours) in the simulated datasets, the reported precision, recall and F1-score metrics are the best metrics that can be achieved in the test datasets (by choosing the best threshold).

2 SIMULATION DATASET

The simulated time series data can be generated in the following steps:

1. Generate an Erdős–Rényi random graph $G$ with number of nodes/variables $n$ and edge creation probability $p$, then convert it into a DAG. We choose $p = 0.1$.
2. For the variables with no parents in $G$, randomly pick a signal type from "harmonic", "pseudo periodic" and "autoregressive" and generate a time series with length $T$ according to this type. We use the Python library “TimeSynth”$^2$ to generate such signals. When generating these signals, the stop time is set to 100. For “harmonic”, the frequency and the noise std are uniformly drawn from $[0.1, 1.0]$ and $[0.1, 0.3]$, respectively. For "pseudo periodic", the frequency is uniformly drawn from $[0.0, 0.005]$, "freqSD" and "ampSD" are set to 0.1 and 0.1. For "autoregressive", "ar_param" is uniformly sampled from $[0.3, 1.0]$ and "sigma" is uniformly sampled from $[0.01, 0.1]$.
3. For a variable $x_i$ with parents $\mathcal{P}(x_i)$ in $G$, we consider both linear relationship $x_i = \sum_{j \in \mathcal{P}(x_i)} w_{ij} x_j + \epsilon$ and nonlinear relationship $x_i = \sum_{j \in \mathcal{P}(x_i)} w_{ij} \tanh(x_j) + \epsilon$, where $w_{ij}$ is uniformly sampled from $[0.5, 2.0]$ and $\epsilon$ is uniformly sampled from $[-0.1, 0.1]$. The time series for those variables are generated in a topological order.
4. Add anomalies into the generated time series: We consider three types of anomalies. The first one is a "measurement" anomaly, i.e., randomly pick a variable $x_i$ a time step $t$, a

$^1$https://github.com/cnn-phil/tetrad

$^2$https://github.com/TimeSynth/TimeSynth
Table 1: The number of the edges in the causal graphs generated by FGES with different parameters.

|       | SWaT          | WADI          |
|-------|---------------|---------------|
|       | max degree (penalty discount = 100) | max degree (penalty discount = 100) |
|       | penalty discount (max degree = 10) | penalty discount (max degree = 10) |
|       | d=5 | 6   | 7   | 8   | 9   | 10  | p=20 | 40 | 60 | 80 | 100 | 120 |
| Edge num | 70  | 79  | 88  | 95  | 98  | 102 | 139  | 122 | 115 | 111 | 102 | 93  |
|       | d=5 | 6   | 7   | 8   | 9   | 10  | p=20 | 40 | 60 | 80 | 100 | 120 |
| Edge num | 152 | 180 | 195 | 211 | 227 | 249 | 331  | 308 | 278 | 262 | 249 | 225 |

Figure 2: WADI detection results. Left: Ground-truth labels. Right: Detected anomalies.

Figure 3: SMAP detection results. Left: Ground-truth labels. Right: Detected anomalies.

Figure 4: MSL detection results. Left: Ground-truth labels. Right: Detected anomalies.

Figure 5: SMD detection results. Left: Ground-truth labels. Right: Detected anomalies.

3 DETECTED ANOMALIES IN PUBLIC DATASETS

Figures 1-5 show the detection results by our approach where we did downsampling for better demonstration. The left figures plot the ground truth labels. The right figures plot the detected anomalies in a point-adjust way. For SMD, the detected anomalies highlighted by the red rectangular are probably real anomalies, but they are not labeled as anomalies by the ground truth. This is one reason why our approach doesn’t achieve the best F1-score in the experiment (achieve high recall but relatively lower precision). Another reason is that we pick the 95th percentile threshold for SMD while the F1-score will be better if we are allowed to tune this threshold.