Ranking Causal Anomalies by Modeling Local Propagations on Networked Systems

Jingchao Ni[1], Wei Cheng[2], Kai Zhang[3], Dongjin Song[2], Tan Yan[2], Haifeng Chen[2] and Xiang Zhang[1]
[1]College of Information Sciences and Technology, Pennsylvania State University
[2]NEC Laboratories America, [3]Computer and Information Sciences Department, Temple University
[1]{jzn47, xzhang}@ist.psu.edu, [2]{weicheng, dsong, yan, haifeng}@nec-labs.com, [3]zhang.kai@temple.edu

***Abstract*—Complex systems are prevalent in many fields such as finance, security and industry. A fundamental problem in system management is to perform diagnosis in case of system failure such that the causal anomalies, i.e., root causes, can be identified for system debugging and repair. Recently, invariant network has proven a powerful tool in characterizing complex system behaviors. In an invariant network, a node represents a system component, and an edge indicates a stable interaction between two components. Recent approaches have shown that by modeling fault propagation in the invariant network, causal anomalies can be effectively discovered. Despite their success, the existing methods have a major limitation: they typically assume there is only a single and global fault propagation in the entire network. However, in real-world large-scale complex systems, it's more common for *multiple* fault propagations to grow simultaneously and *locally* within different node clusters and jointly define the system failure status. Inspired by this key observation, we propose a two-phase framework to identify and rank causal anomalies. In the first phase, a probabilistic clustering is performed to uncover impaired node clusters in the invariant network. Then, in the second phase, a low-rank network diffusion model is designed to backtrack causal anomalies in different impaired clusters. Extensive experimental results on real-life datasets demonstrate the effectiveness of our method.**

I. Introduction

Complex systems are ubiquitous in modern manufacturing industry and information services. Monitoring behaviors of these large-scale systems generates massive log data, such as the metric readings from the networked sensors distributed in a power plant, and the flow intensities of system logs from the cloud computing facilities in Google, Yahoo! and Amazon [1], which have increased the demand for automatic system managements [2]. A central task in managing these complex systems is to diagnose system faults and detect anomalies. It has been reported that 1 minute of downtime in an automotive manufacturing plant could result in as much as $20,000 cost [3]. Hence a timely diagnosis of system faults is crucial to avoid serious money waste and business loss.

Due to its practical importance, there have been intensive interests in developing algorithms to infer whether there is a system failure at a time and if yes, which system components (i.e., basic system units) are causal anomalies. An early approach is to examine individual time series recorded on system components and infer anomalies by a thresholding method [4]. However, in practice it is difficult to set a proper threshold due to the dynamics and heterogeneity of the data. More effective and recent approaches typically start with building system profiles using historical time series data, and then detect anomalies via analyzing patterns in the profiles [5], [6].

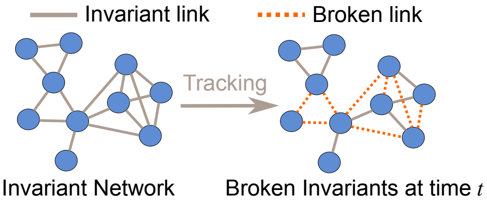


Fig. 1. Invariant network and broken invariants.

The *invariant network* model is a successful example in profiling system behaviors [5], [7], [1], [2], [8], [9]. Its focus is to discover stable and significant dependencies between pairs of system components that are monitored through time series recordings. A strong dependency is called an *invariant* relationship. By combining the invariants learned from all monitoring components, an *invariant network* can be constructed. As illustrated in Fig. 1 (left), in an invariant network, a node represents a system component. A solid line represents an invariant link/relationship between a pair of components.

The practical value of an invariant network is that it can shed important light on abnormal system behaviors and in particular the source of anomalies, by checking whether existing invariants are broken. In Fig. 1 (right), the dotted lines indicate the invariant links are broken at time point t. Such an broken invariant link usually implies abnormal behaviors have occurred in one or both of its connected components [5]. Usually, a network including all system components and all the broken invariant links at a given time is called a *broken network*. For example, in Fig. 1 (right), a broken network will contain all nodes and only those dotted lines.

With the use of invariant and broken networks, several ranking algorithms were developed to diagnose system faults. For example, the method in [2] determines causal anomalies by the percentage of broken invariants within the neighborhood of each node. However, this percentage can be easily biased by fake broken invariants, which occur frequently due to environmental noises in complex systems. More recently, researchers found system faults are seldom isolated. Instead, starting from the root nodes, anomalous behavior will propagate to neighboring nodes in a cascading manner [5]. Such observations have led to a number of successful examples in modeling fault

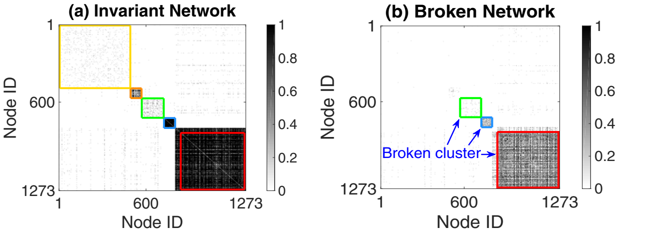


Fig. 2. The adjacency matrices of the invariant network and broken network from a bank information system dataset.

propagations [8], [9]. Despite their success in identifying some causal anomalies, a major limitation of these approaches lies in their basic assumption. They typically assume there is a single and global propagation over the entire network, which is not precise in many emerging applications.

Fig. 2(a) and 2(b) show the adjacency matrices of an invariant network and its corresponding broken network at a system failure time point from a real-world bank information system. Each entry in Fig. 2(a) represents an invariant link. Each entry in Fig. 2(b) represents a broken link. In Fig. 2(a), we can observe that there are several dense clusters, i.e., functional modules, in the invariant network. Among them, three clusters highlighted by red, blue and green can also be observed in the broken network in Fig. 2(b), which means they are heavily impaired. It should be noted different clusters have few connections in between. This means it will be difficult for system faults to propagate across different clusters.

The above application illustrates some important properties of system fault propagations, which have not been taken into account by the existing methods: (1) system faults are propagated *locally* within different clusters, rather than traversing globally through the whole network; (2) there can be *multiple* fault propagations spreading in parallel in different clusters in the system. Therefore, by assuming a single and global propagation in the network, the existing methods cannot locate multiple impaired clusters. Consequently, many true anomalous nodes cannot be accurately detected.

To address the limitations of the existing methods, in this paper, we propose the **C**luster **R**anking based fault **D**iagnosis (CRD) algorithm to rank causal anomalies in a fine-grained two-phase manner. In Phase I, it identifies and ranks clusters in the invariant network by their severities of impairments. To enhance the accuracy of cluster finding, a joint clustering scheme is designed to leverage the complementary information in invariant and broken networks. In Phase II, a diffusion based low-rank network reconstruction model is proposed to back-track causal anomalies in impaired clusters found in Phase I. This model can capture local and paralleled fault propagations in different clusters, making it suitable for locating multiple causal anomalies. Experimental results on real-life datasets suggest the effectiveness of our method.

II. Preliminaries and Problem Definition

In this section, we introduce the technique of the invariant network model [5], and then describe the problem setting.

A. Invariant Network and Broken Invariants

The *invariant* model is used to uncover significant pairwise relationships among massive set of time series. Let $x(t)$ and $y(t)$ be a pair of time series under consideration, such as two sensor readings on two system components, where t is the time index, then their relationship can be described by a linear regression function according to the AutoRegressive eXogenous (ARX) model [10]:

$$\begin{aligned} y(t) = a_1 y(t-1) + ... + a_n y(t-n) \\ + b_0 x(t-k) + ... + b_m x(t-k-m) \end{aligned} \tag{1}$$

where $[n, m]$ is the *order* of the model, which determines how many previous steps are affecting the current output. k is a time delay factor between x and y. Parameters a_i and b_j indicate how strongly a previous step is impacting the current output, which can be learned by the least-square fitting of Eq. (1) to the training data. In real-world applications such as anomaly detection in physical systems, $0 \leq m, n, k \leq 2$ is a popular choice [5], [1].

Let $\boldsymbol{\theta} = \{a_1, ..., a_n, b_0, ..., b_m\}$ be the model parameters, after it is obtained, the prediction of $y(t)$ can be found using Eq. (1) by feeding $\boldsymbol{\theta}$ and observations $y(t-1), ..., y(t-n), x(t-k), ..., x(t-k-m)$. Let $\hat{y}(t, \boldsymbol{\theta})$ represent the prediction, once it is obtained, a *fitness score* $F(\boldsymbol{\theta})$ [11] is used to evaluate how well the learned model $\boldsymbol{\theta}$ fits the real observations as

$$F(\boldsymbol{\theta}) = 1 - \sqrt{\frac{\sum_{t=1}^{N} |y(t) - \hat{y}(t, \boldsymbol{\theta})|^2}{\sum_{t=1}^{N} |y(t) - \bar{y}|^2}} \tag{2}$$

where N and $\bar{y}$ are the length and mean of the time series $y(t)$, respectively. A large fitness score indicates a better fitting of the model. Then, an invariant is declared on a pair of times series x and y if the fitness score is larger than a pre-defined threshold. A network including all the invariant links is called an *invariant network*.

After training the invariant model, each invariant will be tracked using a normalized residual $R(t)$ [11], [1]:

$$R(t) = |y(t) - \hat{y}(t, \boldsymbol{\theta})| / \varepsilon_{\text{max}} \tag{3}$$

where $\varepsilon_{\text{max}} = \max_{1 \leq t \leq N} |y(t) - \hat{y}(t, \boldsymbol{\theta})|$ is the maximal error. If the residual exceeds a prefixed threshold, then the invariant is declared as "broken", i.e., the corresponding dependency relationship vanishes. At time $t = T_b$, A network including all nodes in the invariant network and all broken edges is called a *broken network* at time T_b.

B. Problem Description

We represent the invariant network and broken network by their corresponding adjacency matrices $\mathbf{A} \in \{0,1\}^{n \times n}$ and $\mathbf{B} \in \{0,1\}^{n \times n}$, where n is the number of nodes (i.e., system components) in the system. The two matrices can be obtained as discussed in Sec. II-A. An entry $\mathbf{A}_{xy}$ equals 1 indicates an invariant dependency exists between nodes x and y; 0 otherwise; and an entry $\mathbf{B}_{xy}$ equals 1 indicates the invariant link between nodes x and y is broken; 0 otherwise. The proposed CRD algorithm also allows $\mathbf{A}$ and $\mathbf{B}$ to be continuous. In this case, $\mathbf{A}_{xy}$ and $\mathbf{B}_{xy}$ can be weighted by fitness score $F(\boldsymbol{\theta})$ (Eq. (2)) and residual $R(t)$ (Eq. (3)), respectively.

The goal of this work is to detect abnormal nodes in invariant network $\mathbf{A}$ that are most likely to be the causes of the broken edges in $\mathbf{B}$. Since such anomalies may exist in multiple clusters, we call them *multifaceted causal anomalies*. Accurately detecting multifaceted causal anomalies will be extremely useful for debugging complex system problems that are jointly defined by different impaired functional modules (i.e., broken node clusters).

III. The CRD Algorithm

In this section, we introduce our CRD algorithm, which is a two-phase framework. In Phase I, CRD identifies and ranks multiple broken clusters. In Phase II, it backtracks causal anomalies by modeling multiple local fault propagations in different broken clusters.

A. Phase I: Broken Cluster Identification

First, we propose a probabilistic clustering model to jointly cluster invariant network and broken network, and in the meantime, rank broken clusters. The intuition for the joint clustering is that, a set of nodes that work coordinately in normal status and break concurrently in abnormal status are more likely to be in the same cluster. Therefore, jointly clustering the two networks will be useful to enhance the accuracy of identifying broken clusters.

The Basic Clustering Method. We adopt the doubly stochastic matrix decomposition as the basic method to cluster an invariant network due to its superior performance on sparse networks [12], which is introduced as following.

Suppose there are k clusters in an invariant network $\mathbf{A}$, let $\mathbf{U} \in \mathbb{R}_+^{n\times k}$ be a cluster membership matrix with $\mathbf{U}_{xi} = P(i|x)$ indicating the probability that node x belongs to cluster i. Then a doubly stochastic approximation to $\mathbf{A}$ is defined by

$$\hat{\mathbf{A}}_{xy} = \sum_{i=1}^{k} \frac{\mathbf{U}_{xi}\mathbf{U}_{yi}}{\sum_{z=1}^{n} \mathbf{U}_{zi}} \tag{4}$$

where i is the cluster index, x, y and z are node indexes. Note $\hat{\mathbf{A}} \in \mathbb{R}_+^{n\times n}$ is symmetric and both of its columns and rows sum up to 1. Therefore, it is referred to as *doubly stochastic*.

The clustering problem is to infer $\mathbf{U}$ by minimizing the approximation error of the KL-Divergence $\mathcal{D}_{KL}(\mathbf{A}||\hat{\mathbf{A}})$. After removing some constants, this is equivalent to minimize

$$-\sum_{(x,y)\in\mathcal{E}_A} \mathbf{A}_{xy} \log \hat{\mathbf{A}}_{xy} \tag{5}$$

where $\mathcal{E}_A$ represents the set of all edges in network $\mathbf{A}$.

To provide control of the sparsity of $\mathbf{U}$, a Dirichlet prior on $\mathbf{U}$ can be introduced [12], which gives the following objective function for individual network clustering

$$\begin{aligned} \mathcal{J}_A(\mathbf{U}) = &-\sum_{(x,y)\in\mathcal{E}_A} \mathbf{A}_{xy} \log \hat{\mathbf{A}}_{xy} - (\alpha - 1)\sum_{xi} \log \mathbf{U}_{xi} \\ &\text{s.t. } \mathbf{U} \geq 0,\ \mathbf{U}\mathbf{1}_k = \mathbf{1}_n \end{aligned} \tag{6}$$

where α ($\alpha \geq 1$) is a parameter in the Dirichlet distribution, $\mathbf{1}_k$ is a column vector of length k with all 1's. The equality constraint preserves the probabilistic interpretation of $\mathbf{U}_{xi}$.

Ranking Broken Clusters. Next, we develop a method to rank clusters by their broken severities. Our method uses a generative process to model broken invariants in $\mathbf{B}$. The intuition is that, if two nodes x and y reside in the same severely broken cluster, the invariant link (x, y) is more likely to break. Here, we need a metric to quantify how severe a cluster is broken. Thus for each cluster i in the invariant network, we define an unknown *broken score* as $\mathbf{s}_i$ ($0 \leq \mathbf{s}_i \leq 1$). A higher $\mathbf{s}_i$ means a more severely broken cluster i.

To evaluate how likely an invariant link (x, y) will break, we need a probability for this event. According to the above intuition, this probability should satisfy two criteria: (1) within $[0, 1]$; and (2) it is large only if nodes x and y belong to the same cluster i and cluster i has a high broken score $\mathbf{s}_i$. Therefore, we propose to use

$$P_b(x, y) = \sum_{i=1}^{k} \mathbf{U}_{xi}\mathbf{U}_{yi}\mathbf{s}_i \tag{7}$$

as the broken probability of an invariant (x, y). It is easy to verify $P_b(x, y)$ satisfies the above two criteria. Then, to model the sparse occurrences of broken edges, we follow the convention of modeling sparse networks [13] and use Bernoulli distribution to simulate the generation of a broken invariant (x, y) by

$$\mathbf{B}_{xy} \sim \text{Bernoulli}\big(P_b(x, y)\big) \tag{8}$$

Let $\mathcal{E}_B$ be the set of all edges in $\mathbf{B}$, then the probability to collectively generate a broken network is

$$P(\mathbf{B}|\mathbf{U}, \mathbf{s}) = \prod_{(x,y)\in\mathcal{E}_B} P_b(x, y) \prod_{(x,y)\in\mathcal{E}_A\setminus\mathcal{E}_B} [1 - P_b(x, y)] \tag{9}$$

Let $\mathbf{W} \in \{0, 1\}^{n\times n}$ be an indicator matrix, with $\mathbf{W}_{xy} = 1$ iff $(x, y) \in \mathcal{E}_A \setminus \mathcal{E}_B$, i.e., (x, y) is a non-broken invariant link. Then we can write the negative log-likelihood function as

$$\begin{aligned} \mathcal{J}_B(\mathbf{U}, \mathbf{s}) = &-\sum_{xy} \mathbf{B}_{xy} \log \big(\sum_i \mathbf{U}_{xi}\mathbf{U}_{yi}\mathbf{s}_i\big) \\ &-\sum_{xy} \mathbf{W}_{xy} \log \big(1 - \sum_i \mathbf{U}_{xi}\mathbf{U}_{yi}\mathbf{s}_i\big) \end{aligned} \tag{10}$$

which is our objective for learning to rank broken clusters. Here, the to be learned $\mathbf{s}_i$ serves as the ranking score.

A Unified Objective Function. As discussed in the beginning of this section, to leverage the complementary information in invariant and broken networks, we integrate $\mathcal{J}_A$ in Eq. (6) and $\mathcal{J}_B$ in Eq. (10) into a joint optimization problem

$$\begin{aligned} &\min_{\mathbf{U},\mathbf{s}}\ \mathcal{J}_{CR}(\mathbf{U}, \mathbf{s}) = \mathcal{J}_A + \beta\mathcal{J}_B \\ &\text{s.t. } \mathbf{U}\mathbf{1}_k = \mathbf{1}_n,\ \mathbf{U} \geq 0,\ 0 \leq \mathbf{s}_i \leq 1,\ \forall 1 \leq i \leq k \end{aligned} \tag{11}$$

where β is a parameter to balance the two terms.

B. Phase II: Causal Anomaly Ranking

To infer causal anomalous nodes, we consider the very practical scenario of fault propagation, namely anomalous system status can always be traced back to a set of initial seed nodes, i.e., causal anomalies. These anomalies can propagate along the invariant network, most probably towards neighbors via paths represented by the invariant links in $\mathbf{A}$. To model this process, we employ the label propagation technique [14]. Suppose there is an unknown *seed vector* $\mathbf{e} \in \mathbb{R}_{+}^{n\times 1}$ with $\mathbf{e}_x$ denoting the degree that node x is a causal anomaly. After propagation, each node x will obtain a *status score* $\mathbf{r}_x$ to indicate to what extent it is impacted by the causal anomalies. Then the propagation from $\mathbf{e}$ to $\mathbf{r}$ can be modeled by the following optimization problem

$$\begin{aligned}&\min_{\mathbf{r}\geq 0} \quad c\mathbf{r}^T(\mathbf{I}_n - \tilde{\mathbf{A}})\mathbf{r} + (1-c)\|\mathbf{r}-\mathbf{e}\|_F^2 \\ &= c\sum_{xy}\mathbf{A}_{xy}(\mathbf{r}_x/\sqrt{\mathbf{D}_{xx}} - \mathbf{r}_y/\sqrt{\mathbf{D}_{yy}})^2 + (1-c)\sum_{x}(\mathbf{r}_x - \mathbf{e}_x)^2\end{aligned} \tag{12}$$

where $\mathbf{I}_n$ is an $n \times n$ identity matrix, $\tilde{\mathbf{A}} = \mathbf{D}^{-\frac{1}{2}}\mathbf{A}\mathbf{D}^{-\frac{1}{2}}$ is a symmetrically normalized matrix of $\mathbf{A}$, and $\mathbf{D}$ is a diagonal matrix with $\mathbf{D}_{xx} = \sum_{y=1}^{n}\mathbf{A}_{xy}$.

The first term in Eq. (12) encourages neighboring nodes to have similar status scores, and the second term penalizes large bias from the initial seeds. c is a parameter balancing the two terms. It can be verified that the closed-form solution to Eq. (12) is

$$\mathbf{r} = (1-c)(\mathbf{I}_n - c\tilde{\mathbf{A}})^{-1}\mathbf{e} \tag{13}$$

which establishes an explicit relationship between $\mathbf{r}$ and $\mathbf{e}$.

In real-world applications, causal anomalies often propagate their impacts inside their associated clusters. Thus for each cluster i, we define $\mathbf{e}^{(i)} \in \mathbb{R}_{+}^{n\times 1}$ as a *cluster-specific* seed vector. Moreover, instead of directly using $\mathbf{e}_x^{(i)}$ as the causal anomaly score of node x, we use $\mathbf{U}_{xi}\mathbf{e}_x^{(i)}$, where $\mathbf{U}_{xi}$ is obtained in Phase I, to emphasize that, node x is a causal anomaly of cluster i if it resides in cluster i (with a large $\mathbf{U}_{xi}$ value) and is abnormal (with a large $\mathbf{e}_x^{(i)}$ value).

Correspondingly, different clusters will have different status score vectors $\mathbf{r}^{(i)} \in \mathbb{R}_{+}^{n\times 1}$. Then the propagation relationship between $\mathbf{e}^{(i)}$ and $\mathbf{r}^{(i)}$ can be represented by

$$\mathbf{r}^{(i)} = (1-c)(\mathbf{I}_n - c\tilde{\mathbf{A}})^{-1}(\mathbf{U}_{*i} \circ \mathbf{e}^{(i)}) \tag{14}$$

where $\circ$ is entry-wise product, $\mathbf{U}_{*i}$ is the i^{th} column of $\mathbf{U}$.

To exploit broken edge pattern, we propose to use $\{\mathbf{r}^{(i)}\}_{i=1}^{k}$ to reconstruct the broken network $\mathbf{B}$. The intuition is as following. When an invariant link (x, y) is broken, i.e., $\mathbf{B}_{xy}$ is large, then at least one node of x and y should be perturbed by some causal anomalies from some clusters. That is, either $\mathbf{r}_x^{(i)}$ or $\mathbf{r}_y^{(i)}$ is large for some i. This suggests a reconstruction error as

$$\sum_{(x,y)\in\mathcal{E}_A}(\sum_{i=1}^{k}\mathbf{r}_x^{(i)}\mathbf{r}_y^{(i)} - \mathbf{B}_{xy})^2 \tag{15}$$

Let $\mathbf{E} = [\mathbf{e}^{(1)}, ..., \mathbf{e}^{(k)}]$, $\mathbf{R} = [\mathbf{r}^{(1)}, ..., \mathbf{r}^{(k)}]$, and $\mathbf{H} = (1-c)(\mathbf{I}_n - c\tilde{\mathbf{A}})^{-1}$, from Eq. (14), we have $\mathbf{R} = \mathbf{H}(\mathbf{U}\circ\mathbf{E})$. Then, let $\mathbf{C} \in \{0,1\}^{n\times n}$ be an indicator matrix with $\mathbf{C}_{xy} = 1$ iff $(x, y) \in \mathcal{E}_A$, we can rewrite Eq. (15) by a matrix form and obtain the following objective function

$$\min_{\mathbf{E}\geq 0} \mathcal{J}_H = \|\mathbf{C}\circ[\mathbf{H}(\mathbf{U}\circ\mathbf{E})(\mathbf{U}\circ\mathbf{E})^{\mathrm{T}}\mathbf{H}^{\mathrm{T}}] - \mathbf{B}\|_F^2 + \tau\|\mathbf{E}\|_1 \tag{16}$$

Here, an ℓ_1 norm on $\mathbf{E}$ is added to encourage sparsity of $\mathbf{E}$ because practically often a few nodes could be causal anomalies. τ is a controlling parameter, a larger τ typically results in more zeros in $\mathbf{E}$.

C. Ranking with Unified Scores

To integrate the results from Phase I and II, we propose a unified causal anomaly score $\mathbf{f}_x$ for each node x. Ideally, this score should place more priority to a node x if it is a causal anomaly to some cluster i (with large $\mathbf{U}_{xi}\mathbf{e}_x^{(i)}$) and cluster i is broken severely (with large $\mathbf{s}_i$). This suggests a simple form $\mathbf{f}_x = \mathbf{U}_{xi}\mathbf{e}_x^{(i)}\mathbf{s}_i$. Equivalently, the score vector $\mathbf{f}$ is

$$\mathbf{f} = (\mathbf{U}\circ\mathbf{E})\mathbf{s} \tag{17}$$

To summarize, in CRD algorithm, we first optimize $\mathcal{J}_{CR}$ in Eq. (11) to solve $\mathbf{U}$ and $\mathbf{s}$ in Phase I, then plug $\mathbf{U}$ into $\mathcal{J}_H$ in Eq. (16) and solve $\mathbf{E}$. Finally, all nodes are sorted using $\mathbf{f}$ in Eq. (17), with most suspicious nodes on the top.

D. The Learning Algorithm

For Phase I, since Eq. (11) is not jointly convex in $\mathbf{U}$ and $\mathbf{s}$, we take an alternating minimization framework that alternately solves $\mathbf{U}$ and $\mathbf{s}$ until a stationary point is achieved. For Phase II, we develop an iterative updating algorithm for solving $\mathbf{E}$, which monotonically decreases the objective value in Eq. (16) until convergence. For brevity, we omit the details here.

IV. Experimental Results

A. Dataset Description

Bank Information Systems (BIS). The BIS dataset [2], [8] contains $1,273$ flow intensity time series monitoring different aspects of the system, such as CPU usage, disk I/O, etc. The training data was collected at normal system states, where 168 time points were collected for each time series. The invariant network was then generated on the training data as described in Sec. II-A, which has $1,273$ nodes and $39,116$ edges. The testing data were collected during abnormal system states, where 169 time points were collected for each of the $1,273$ time series. As described in Sec. II-A, we use the testing data to track the changes of the invariant network, and generate broken networks. Among the 169 times points in the testing data, system experts observed a system failure at $t = 120$ and $t = 122$. Thus two broken networks on these time points were generated for performing anomaly ranking. There are $18,052$ and $16,089$ broken edges on $t = 120$ and $t = 122$, respectively. According to system experts, "DB16-" related system components are responsible for the problem. So we extract all time series with prefix "DB16-" in their titles and regard them as ground truth anomalies. In total, there are 80 such time series. Our goal is to evaluate the capabilities of different methods to detect these causal anomalies.

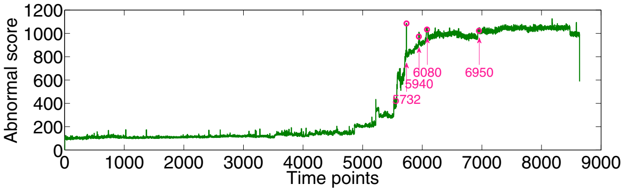


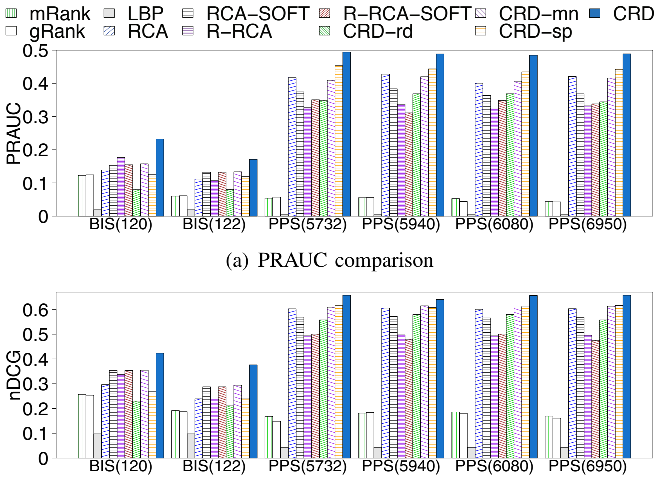
Fig. 3. Abnormal scores at different time points on PPS dataset.

Power Plant Sensors (PPS). This dataset was collected by NEC Lab (www.nec-labs.com). It contains $6,049$ times series monitored by sensors distributed in a power plant system. Similar to the BIS data, the invariant network was trained using the time series collected in one normal day, where each time series was collected every 10 seconds and contains $8,639$ time points. The obtained invariant network contains $6,049$ nodes and $16,361$ edges. On another day when system failure happened, system experts generated an "abnormal score" for each collected time point, as shown in Fig. 3. This score is proportional to the total residual (Eq. (3)) at each time point. To detect causal anomalies, four time points with peak abnormal scores (as pinpointed in Fig. 3) were picked to generate four broken networks, which have 883, 930, $1,032$ and $1,022$ broken edges, respectively. Among them, $t = 5,940$ is the reported time when a system failure was observed by system operators. According to system experts, the problem is related to "XY2-" and "XY345-" sensors. Thus we extract such time series and regard them as ground truth. In total, there are 67 such time series.

B. Experimental Setup

The State-of-the-art Methods. We compare the performance of CRD with several state-of-the-art methods, including (1) mRank [2]; (2) gRank [2]; (3) LBP [8]; (4) RCA family algorithms [9]. The mRank and gRank methods rank causal anomalies mostly by the percentage of associated broken invariants with each node. The difference is that mRank only considers broken invariants directly linked to each node while gRank considers broken invariants within several hops to each node. LBP models a broken network by Markov Random Fields and infers anomalies using a loopy belief propagation algorithm. RCA family algorithms are based on label propagation but simply assume a single global propagation along a whole network. There are four variants, RCA, R-RCA, RCA-SOFT and R-RCA-SOFT. Here "R-" represents a relaxed version that runs faster than basic RCA. "-SOFT" means a softmax normalization is employed to avoid too large or too small ranking scores. We consider all these algorithms for comparison. The parameters of all methods are tuned to achieve their optimal performance.

To study the effectiveness of each individual component of CRD, we examine three degraded variants of CRD, i.e., CRD-rd, CRD-mn and CRD-sp. CRD-rd only uses Phase I where nodes in a cluster with high broken score $\mathbf{s}_i$ are ranked higher than those in a cluster with low $\mathbf{s}_i$. The nodes in the same cluster are randomly ordered. CRD-mn only uses Phase II and the anomaly score of a node x is $\mathbf{f}_x = mean(\mathbf{E}_{x*})$. CRD-sp is the same as CRD except that in Phase I, it optimizes $\mathcal{J}_A$ and $\mathcal{J}_B$ separately (see Eq. (11)). Comparing with the first two variants can show the importance of integrating the two phases. Comparing with the third variant can show the effectiveness of joint optimization in Phase I.



(a) PRAUC comparison

(b) nDCG comparison

Fig. 4. Effectiveness evaluation on BIS and PPS datasets.

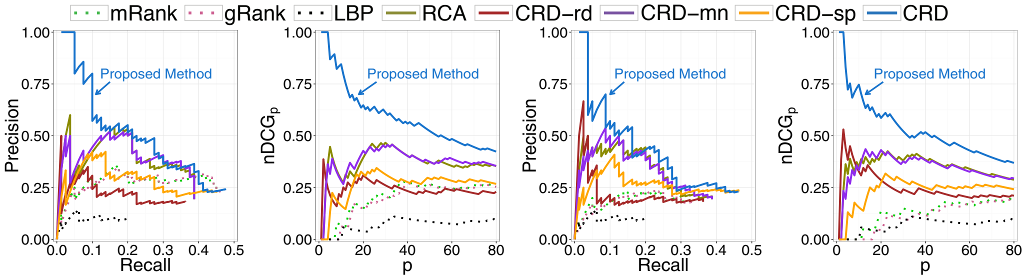
Evaluation Criteria. Similar to existing works [2], [8], [9], we use area under precision-recall curve (PRAUC) [15] and normalized Discounted Cumulative Gain (nDCG) [16] to evaluate causal anomaly detection accuracy.

The precision-recall curve is calculated by varying the rank threshold from 1 up to K, where K is typically chosen as twice the actual number of ground truth causal anomalies [16], [8]. PRAUC has been considered to be better than AUC (area under the ROC curve) because PRAUC punishes highly ranked false positives much more than AUC does [15].

The nDCG at top-p ranking result is defined as $\text{nDCG}_p = \frac{\text{DCG}_p}{\text{IDCG}_p}$, where $\text{DCG}_p = \sum_{x=1}^{p} \frac{2^{\text{rel}_x}-1}{\log_2(1+x)}$ is defined on the inferred ranking list, and $\text{rel}_x = 1$ iff node x is a ground truth anomaly. IDCG_p is the DCG_p value on the ground truth ranking list. Here p is smaller than or equal to the number of ground truth causal anomalies.

C. Effectiveness Evaluation

Fig. 4 shows the PRAUC and nDCG values of different methods on the BIS and PPS datasets. Here nDCG_p is the value with p equal to the number of ground truth of each dataset. From the results, we have several key observations. First, CRD significantly outperforms other methods by large margins on all datasets. The PRAUC improvement over the best competing method on each dataset varies from 14.20% to 31.64%. This demonstrates the importance of identifying broken clusters before tracking causal anomalies. Moreover, by comparing CRD with CRD-rd and CRD-mn, we see the importance of integrating Phases I and II as a coherent approach. Furthermore, the comparison with CRD-sp shows the effectiveness of joint optimization in Phase I of CRD. This



(a) PR-curve ($t = 120$) (b) nDCG ($t = 120$) (c) PR-curve ($t = 122$) (d) nDCG ($t = 122$)

Fig. 5. The Precision-Recall and nDCG$_p$ curves on the BIS dataset.

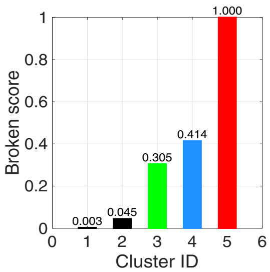


Fig. 6. The broken cluster scores.

implies that an inferior clustering in Phase I can largely reduce the subsequent anomaly inference accuracy.

To better understand the difference between CRD and other methods, we have examined the Precision-Recall and nDCG$_p$ curves of different methods. For example, Fig. 5 presents these curves on BIS dataset. Here for clarity, we use R-RCA-SOFT to represent the RCA family since it generally performs better than other RCA algorithms on this data. From these results, we find the top ranked nodes by CRD contain more ground truth nodes than other methods, as demonstrated by the high heads of the curves of CRD. Hence CRD is practically more useful than other methods since system experts usually only check the top ranked nodes in a regular diagnosis.

We also perform a detailed evaluation of the broken cluster ranking component in Phase I of CRD. In Fig. 2(a) and (b), we have shown the invariant and broken networks for dataset BIS ($t = 122$). There are five clusters in the invariant network, 3 of which are broken and are highlighted in Fig. 2(b). Fig. 6 shows the broken scores learned by CRD for all detected clusters (i.e., $\mathbf{s}_i$ in Eq. (11)). The colors of the bars correspond to the clusters in Fig. 2(b). As we can see, the scores accurately reflect the broken degrees of different clusters. Furthermore, there is a clear difference between the scores of the broken clusters and unbroken clusters.

V. Related Work

There are many methods using invariant graphs for anomaly analysis [5], [7], [1], here we discuss the most relevant ones. In [2], mRank and gRank were proposed to detect anomalies in an invariant network. However, these methods ignore the fault propagation and heavily rely on the percentage of broken invariants within the neighborhood of each node. In [8] and [9], two different ways were proposed to model fault propagation, both of which assume a single global propagation in the whole network. As has been discussed, this is not precise in many emerging applications. In fact, multiple local propagations happening in different clusters can jointly define a system fault. Such global methods are not aware of this scenario, making them sub-optimal in locating causal anomalies.

VI. Conclusion

Automatically detecting causal anomalies is a crucial task in system management. The existing methods assume a single and global fault propagation in the invariant network, which cannot model the multiple and local fault propagations in different clusters of the invariant network. To address this problem, we propose a novel algorithm CRD in this paper. CRD first finds and ranks broken clusters, then backtracks causal anomalies in different clusters using a low-rank network diffusion model. Experimental results on real-life datasets demonstrate CRD consistently outperforms the competitors.

Acknowledgment

This work was partially supported by the National Science Foundation grants IIS-1664629 and CAREER.

References

[1] H. Chen, H. Cheng, G. Jiang, and K. Yoshihira, "Exploiting local and global invariants for the management of large scale information systems," in *ICDM*, 2008.
[2] Y. Ge, G. Jiang, M. Ding, and H. Xiong, "Ranking metric anomaly in invariant networks," *ACM Trans. Knowl. Discov. Data.*, vol. 8, no. 2, p. 8, 2014.
[3] D. Djurdjanovic, J. Lee, and J. Ni, "Watchdog agent-an infotronics-based prognostics approach for product performance degradation assessment and prediction," *Adv. Eng. Inform.*, vol. 17, no. 3, pp. 109–125, 2003.
[4] J. Gertler, *Fault detection and diagnosis in engineering systems.* Marcel Dekker, 1998.
[5] G. Jiang, H. Chen, and K. Yoshihira, "Discovering likely invariants of distributed transaction systems for autonomic system management," in *ICAC*, 2006.
[6] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: A survey," *ACM Comput. Surv.*, vol. 41, no. 3, p. 15, 2009.
[7] G. Jiang, H. Chen, and K. Yoshihira, "Modeling and tracking of transaction flow dynamics for fault detection in complex systems," *IEEE Trans. Dependable Secure Comput.*, vol. 3, no. 4, pp. 312–326, 2006.
[8] C. Tao, Y. Ge, Q. Song, Y. Ge, and O. A. Omitaomu, "Metric ranking of invariant networks with belief propagation," in *ICDM*, 2014.
[9] W. Cheng, K. Zhang, H. Chen, G. Jiang, and W. Wang, "Ranking causal anomalies via temporal and dynamical analysis on vanishing correlations," in *KDD*, 2016.
[10] L. Ljung, *System identification: theory for the user, 2nd ed.* Prentice Hall PTR, 1999.
[11] G. Jiang, H. Chen, and K. Yoshihira, "Efficient and scalable algorithms for inferring likely invariants in distributed systems," *IEEE Trans. Knowl. Data Eng.*, vol. 19, no. 11, pp. 1508–1523, 2007.
[12] Z. Yang and E. Oja, "Clustering by low-rank doubly stochastic matrix decomposition," in *ICML*, 2012.
[13] J. Yang, J. McAuley, and J. Leskovec, "Community detection in networks with node attributes," in *ICDM*, 2013, pp. 1151–1156.
[14] D. Zhou, O. Bousquet, T. N. Lal, J. Weston, and B. Schölkopf, "Learning with local and global consistency," in *NIPS*, 2004.
[15] J. Davis and M. Goadrich, "The relationship between precision-recall and roc curves," in *ICML*, 2006.
[16] K. Järvelin and J. Kekäläinen, "Cumulated gain-based evaluation of ir techniques," *ACM Trans. Inf. Syst.*, vol. 20, no. 4, pp. 422–446, 2002.