Ranking Causal Anomalies for System Fault Diagnosis via Temporal and Dynamical Analysis on Vanishing Correlations

WEI CHENG, NEC Laboratories America
JINGCHAO NI, Pennsylvania State University
KAI ZHANG, HAIFENG CHEN, and GUOFEI JIANG, NEC Laboratories America
YU SHI, University of Illinois at Urbana-Champaign
XIANG ZHANG, Pennsylvania State University
WEI WANG, University of California, Los Angeles

Detecting system anomalies is an important problem in many fields such as security, fault management, and industrial optimization. Recently, invariant network has shown to be powerful in characterizing complex system behaviours. In the invariant network, a node represents a system component and an edge indicates a stable, significant interaction between two components. Structures and evolutions of the invariance network, in particular the vanishing correlations, can shed important light on locating causal anomalies and performing diagnosis. However, existing approaches to detect causal anomalies with the invariant network often use the percentage of vanishing correlations to rank possible casual components, which have several limitations: (1) fault propagation in the network is ignored, (2) the root casual anomalies may not always be the nodes with a high percentage of vanishing correlations, (3) temporal patterns of vanishing correlations are not exploited for robust detection, and (4) prior knowledge on anomalous nodes are not exploited for (semi-)supervised detection. To address these limitations, in this article we propose a network diffusion based framework to identify significant causal anomalies and rank them. Our approach can effectively model fault propagation over the entire invariant network and can perform joint inference on both the structural and the time-evolving broken invariance patterns. As a result, it can locate high-confidence anomalies that are truly responsible for the vanishing correlations and can compensate for unstructured measurement noise in the system. Moreover, when the prior knowledge on the anomalous status of some nodes are available at certain time points, our approach is able to leverage them to further enhance the anomaly inference accuracy. When the prior knowledge is noisy, our approach also automatically learns reliable information and reduces impacts from noises. By performing extensive experiments on synthetic datasets, bank information system datasets, and coal plant cyber-physical system datasets, we demonstrate the effectiveness of our approach.

CCS Concepts: • **Security and privacy → Pseudonymity, anonymity and untraceability;**

Additional Key Words and Phrases: Causal anomalies ranking, label propagation, nonnegative matrix factorization

These authors contributed equally to this work.
W. Cheng and J. Ni contributed eqully to this work.

Wei Wang is partially supported by the National Science Foundation grants IIS-1313606, DBI-1565137, by National Institutes of Health under the grant number R01GM115833-01.

Authors' addresses: W. Cheng, K. Zhang, H. Chen, and G. Jiang, NEC Laboratories America, 4 Independence Way, Suite 200, Princeton, NJ 08540; emails: {weicheng, kzhang, haifeng, gfj}@nec-labs.com; J. Ni and X. Zhang, College of Information Sciences and Technology, Pennsylvania State University, 332 Information Sciences and Technology Building, University Park, PA 16802-6823; emails: {jzn47, xzhang}@ist.psu.edu; Y. Shi, Department of Computer Science, University of Illinois at Urbana-Champaign, 201 North Goodwin Avenue, Urbana, IL 61801-2302; email: yushi2@illinois.edu; W. Wang, Department of Computer Science, University of California, Los Angeles, 3531-G Boelter Hall, Los Angeles, CA 90095-1596; email: weiwang@cs.ucla.edu.

DOI: http://dx.doi.org/10.1145/3046946

ACM Reference Format:
Wei Cheng, Jingchao Ni, Kai Zhang, Haifeng Chen, Guofei Jiang, Yu Shi, Xiang Zhang, and Wei Wang. 2017. Ranking causal anomalies for system fault diagnosis via temporal and dynamical analysis on vanishing correlations. ACM Trans. Knowl. Discov. Data 11, 4, Article 40 (June 2017), 28 pages.
DOI: http://dx.doi.org/10.1145/3046946

1. INTRODUCTION

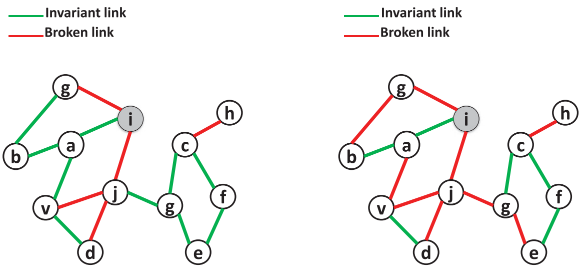
With the rapid advances in networking, computers, and hardware, we are facing an explosive growth of complexity in networked applications and information services. These large-scale, often distributed, information systems usually consist of a great variety of components that work together in a highly complex and coordinated manner. One example is the Cyber-Physical System (CPS), which is typically equipped with a large number of networked sensors that keep recording the running status of the local components; another example is the large-scale Information Systems such as the cloud computing facilities in Google, Yahoo!, and Amazon, whose composition includes thousands of components that vary from operating systems to application software to servers to storage to networking devices, and so on.

A central task in running these large-scale distributed systems is to automatically monitor the system status, detect anomalies, and diagnose system fault to guarantee stable and high-quality services or outputs. Significant research efforts have been devoted to this topic in the literature. For instance, Gertler et al. [8] proposed to detect anomalies by examining monitoring data of individual component with a thresholding scheme. However, it can be quite difficult to learn a universal and reliable threshold in practice, due to the dynamic and complex nature of information systems. More effective and recent approaches typically start with building system profiles and then detect anomalies via analyzing patterns in these profiles [5, 16]. The system profile is usually extracted from historical time-series data collected by monitoring different system components, such as the flow intensity of software log files, system audit events, and network traffic statistics, and sometimes sensory measurements in physical systems.

The invariant model is a successful example [16, 17] of large-scale system management. It focuses on discovering stable, significant dependencies between pairs of system components that are monitored through time-series recordings to profile the system status and perform subsequent reasoning. A strong dependency between a pair of components is called an *invariant* (correlation) relationship. By combining the invariants learned from all monitoring components, a global system dependency profile can be obtained. The significant practical value of such an *invariant* profile is that it provides important clues on abnormal system behaviors, and in particular the source of anomalies, by checking whether existing invariants are broken. Figure 1 illustrates one example of the invariant network and two snapshots of broken invariants at time t_1 and t_2, respectively. Each node represents the observation from a monitoring component. The green line signifies an invariant link between two components, and a red line denotes broken invariant (i.e., vanishing correlation). The network including all the broken invariants at given time point is referred to as the *broken network*.

Although the broken invariants provide valuable information of the system status, locating true, causal anomalies can still be a challenging task for the following reasons. First, system faults are seldom isolated. Instead, starting from the root location/component, anomalous behavior will propagate to neighboring components [16], and different types of system faults can trigger diverse propagation patterns. Second, monitoring data often contain a lot of noises due to the fluctuation of complex operation environments.

Recently, several ranking algorithms were developed to diagnose the system failure based on the percentage of broken invariant edges associated with the nodes, such as



(a) t_1 (b) t_2

Fig. 1. Invariant network and vanishing correlations (red edges).

the egonet-based method proposed by Ge et al. [7], and the loopy belief propagation–(LBP) based method proposed by Tao et al. [26]. Despite the success in practical applications, existing methods still have certain limitations. First, they do not take into account the global structure of the invariant network or how the root anomaly/fault propagates in such a network. Second, the ranking strategies rely heavily on the percentage of broken edges connected to a node. For example, the mRank algorithm [7] calculated the anomaly score of a given node using the ratio of broken edges within the egonet[1] of the node. The LBP-based method [26] used the ratio of broken edges as the prior probability of abnormal state for each node. We argue that the percentage of broken edges may not serve as good evidence of the causal anomaly. This is because, although one broken edge can indicate that one (or both) of the related nodes is abnormal, lack of a broken edge does not necessary indicate that related nodes are problem free. Instead, it is possible that the correlation is still there when two nodes become abnormal simultaneously [16]. Therefore, the percentage of broken edges could give false evidence. For example, in Figure 1, the causal anomaly is node ⓘ. The percentage of broken edges for node ⓘ is 2/3, which is smaller than that of node ⓗ (which is equal to 1). Since clear evidence of fault propagation on node ⓘ exists, an ideal algorithm should rank ⓘ higher than ⓗ. Third, existing methods usually consider a static broken network instead of multiple broken networks at successive time points together. We believe that jointly analyzing temporal broken networks can help to resolve ambiguity and achieve a denoising effect. This is because the root casual anomalies usually remain unchanged within a short time period, even though the fault may keep proragating in the invariant network. As an example shown in Figure 1, it would be easier to detect the causal anomaly if we jointly consider the broken networks at two successive time points together.

Furthermore, in some applications, system experts may have prior knowledge on the anomalous status of some components (i.e., nodes) at certain time points, such as a numeric value indicating the bias of the monitoring data of a component from its predicted normal value [6]. Thus it is highly desirable to incorporate them to guide the causal anomaly inferences. However, to our best knowledge, none of these existing approaches can handle such information.

To address the limitations of existing methods, we propose several network diffusion based algorithms for ranking causal anomalies. Our contributions are summarized as follows.

(1) We employ the network diffusion process to model propagation of causal anomalies and use propagated anomaly scores to reconstruct the vanishing correlations.

[1] An egonet is the induced 1-step subgraph for each node.

By minimizing the reconstruction error, the proposed methods simultaneously consider the whole invariant network structure and the potential fault propagation. We also provide rigid theoretical analysis on the properties of our methods.

(2) We further develop efficient algorithms that reduce the time complexity from $\mathcal{O}(n^3)$ to $\mathcal{O}(n^2)$, where n is the number of nodes in the invariant network. This makes it feasible to quickly locate root cause anomalies in large-scale systems.

(3) We employ effective normalization strategy on the ranking scores, which can reduce the influence of extreme values or outliers without having to explicitly remove them from the data.

(4) We develop a smoothing algorithm that enables users to jointly consider dynamic and time-evolving broken network and thus obtain better ranking results.

(5) We extend our algorithms to semi-supervised settings to leverage the prior knowledge on the anomalous degrees of nodes at certain time points. The prior knowledge are allowed to partially cover the nodes in the invariant network, as practically suggested by the limitation of such information.

(6) We also improve our semi-supervised algorithms to allow automatic identification of noisy prior knowledge. By assigning small weights to nodes with false anomalous degrees, our algorithms can reduce the negative impacts of prior knowledge and obtain robust performance gain.

(7) We evaluate the proposed methods on both synthetic datasets and two real-life datasets, including the bank information system and the coal plant cyber-physical system datasets. The experimental results demonstrate the effectiveness of the proposed methods.

2. BACKGROUND AND PROBLEM DEFINITION

In this section, we first introduce the technique of the invariant model [16] and then define our problem.

2.1. System Invariant and Vanishing Correlations

The *invariant* model is used to uncover significant pairwise relations among massive set of time series. It is based on the AutoRegressive eXogenous (ARX) model [21] with time delay. Let $x(t)$ and $y(t)$ be a pair of time series under consideration, where t is the time index, and let n and m be the degrees of the ARX model, with a delay factor k. Let $\widehat{y}(t;\boldsymbol{\theta})$ be the prediction of $y(t)$ using the ARX model parametarized by $\boldsymbol{\theta}$, which can then be written as

$$\begin{aligned}\widehat{y}(t;\boldsymbol{\theta}) &= a_1 y(t-1) + \cdots + a_n y(t-n), \qquad (1)\\ &\quad + b_0 x(t-k) + \cdots + b_m x(t-k-m) + d\\ &= \boldsymbol{\varphi}(t)^\top \boldsymbol{\theta}, \qquad (2)\end{aligned}$$

where $\boldsymbol{\theta} = [a_1, \ldots, a_n, b_0, \ldots, b_m, d]^\top \in \mathbb{R}^{n+m+2}$, $\boldsymbol{\varphi}(t) = [y(t-1), \ldots, y(t-n), x(t-k), \ldots, x(t-k-m), 1]^\top \in \mathbb{R}^{n+m+2}$. For a given setting of (n, m, k), the parameter $\boldsymbol{\theta}$ can be estimated with observed time points $t = 1, \ldots, N$ in the training data, via least-squares fitting. In real-world applications such as anomaly detection in physical systems, $0 \le n, m, k \le 2$ is a popular choice [6, 16]. We can define the "goodness of fit" (or *fitness score*) of an ARX model as

$$F(\boldsymbol{\theta}) = 1 - \sqrt{\frac{\sum_{t=1}^{N} \left|y(t) - \widehat{y}(t;\boldsymbol{\theta})\right|^2}{\sum_{t=1}^{N} \left|y(t) - \bar{y}\right|^2}}, \qquad (3)$$

where $\bar{y}$ is the mean of the time series $y(t)$. A higher value of $F(\boldsymbol{\theta})$ indicates a better fitting of the model. An invariant (correlation) is declared on a pair of time series x and

Table I. Summary of Notations

Symbol	Definition
n	the number of nodes in the invariant network
c, λ, τ	the parameters $0 < c < 1, \tau > 0, \lambda > 0$
$\sigma(\cdot)$	the softmax function
$\mathcal{G}_l$	the invariant network
$\mathcal{G}_b$	the broken network for $\mathcal{G}_l$
$\mathbf{A}$ ($\tilde{\mathbf{A}}$) $\in \mathbb{R}^{n\times n}$	the (normalized) adjacency matrix of $\mathcal{G}_l$
$\mathbf{P}$ ($\tilde{\mathbf{P}}$) $\in \mathbb{R}^{n\times n}$	the (normalized) adjacency matrix of $\mathcal{G}_b$
$\mathbf{M} \in \mathbb{R}^{n\times n}$	the logical matrix of $\mathcal{G}_l$
$d(i)$	the degree of the i^{th} node in network $\mathcal{G}_l$
$\mathbf{D} \in \mathbb{R}^{n\times n}$	the degree matrix: $\mathbf{D} = diag(d(i), \ldots, d(n))$
$\mathbf{r} \in \mathbb{R}^{n\times 1}$	the prorogated anomaly score vector
$\mathbf{e} \in \mathbb{R}^{n\times 1}$	the ranking vector of causal anomalies
RCA	the basic ranking causal anomalies algorithm
R-RCA	the relaxed RCA algorithm
RCA-SOFT	the RCA with softmax normalization
R-RCA-SOFT	the relaxed RCA with softmax normalization
T-RCA	the RCA with temporal smoothing
T-R-RCA	the R-RCA with temporal smoothing
T-RCA-SOFT	the RCA-SOFT with temporal smoothing
T-R-RCA-SOFT	the R-RCA-SOFT with temporal smoothing
RCA-SEMI	the RCA in semi-supervised setting
W-RCA-SEMI	the semi-supervised RCA with weight learning

y if the fitness score of the ARX model is larger than a pre-defined threshold. A network including all the invariant links is referred to as the *invariant network*. Construction of the invariant network is referred to as the model training. The model $\boldsymbol{\theta}$ will then be applied on the time series x and y in the testing phase to track vanishing correlations.

To track vanishing correlations, we can use the techniques developed in References [6, 18]. At each time point, we compute the (normalized) residual $R(t)$ between the measurement $y(t)$ and its estimate $\widehat{y}(t; \boldsymbol{\theta})$ by

$$R(t) = \frac{\left|y(t) - \widehat{y}(t; \boldsymbol{\theta})\right|}{\varepsilon_{\max}}, \tag{4}$$

where $\varepsilon_{\max}$ is the maximum training error $\varepsilon_{\max} = \max_{1\le t\le N} |y(t) - \widehat{y}(t; \boldsymbol{\theta})|$. If the residual exceeds a prefixed threshold, then we declare the invariant as "broken," that is, the correlation between the two time series vanishes. The network including all the broken edges at given time point and all nodes in the invariant network is referred to as the *broken network*.

2.2. Problem Definition

Let $\mathcal{G}_l$ be the invariant network with n nodes. Let $\mathcal{G}_b$ be the broken network for $\mathcal{G}_l$. We use two symmetric matrices $\mathbf{A} \in \mathbb{R}^{n\times n}$, $\mathbf{P} \in \mathbb{R}^{n\times n}$ to denote the adjacency matrix of network $\mathcal{G}_l$ and $\mathcal{G}_b$, respectively. These two matrices can be obtained as discussed in Section 2.1. The two matrices can be binary or continuous. For binary case of $\mathbf{A}$, 1 is used to denote that the correlation exists between two time series, and 0 denotes the lack of correlation; while for $\mathbf{P}$, 1 is used to denote that the correlation is broken (vanishing), and 0 otherwise. For the continuous case, the fitness score $F(\boldsymbol{\theta})$ (3) and the residual $R(t)$ (4) can be used to fill the two matrices, respectively.

Our main goal is to detect the abnormal nodes in $\mathcal{G}_l$ that are most responsible for causing the broken edges in $\mathcal{G}_b$. In this sense, we call such nodes "causal anomalies."

Accurate detection of causal anomalous nodes will be extremely useful for examination, debugging, and repair of system failures.

3. RANKING CAUSAL ANOMALIES

In this section, we present the algorithm of Ranking Causal Anomalies (RCA), which takes into account both the fault propagation and fitting of broken invariants simultaneously.

3.1. Fault Propagation

We consider a very practical scenario of fault propagation, namely that anomalous system status can always be traced back to a set of *root cause* anomaly nodes, or *causal anomalies*, as initial seeds. As the time passes, these root cause anomalies will then propagate along the invariant network, most probably towards their neighbors via paths identified by the invariant links in $\mathcal{G}_l$. To explicitly model this spreading process on the network, we have employed the label propagation technique [19, 28, 31]. Suppose that the (unknown) root cause anomalies are denoted by the indicator vector $\mathbf{e}$, whose entries $\mathbf{e}_i$'s ($1 \leq i \leq n$) indicate whether the ith node is the casual anomaly ($\mathbf{e}_i = 1$) or not ($\mathbf{e}_i = 0$). At the end of propagation, the system status is represented by the anomaly score vector $\mathbf{r}$, whose entries tell us how severe each node of the network has been impaired. The propagation from $\mathbf{e}$ to $\mathbf{r}$ can be modeled by the following optimization problem:

$$\min_{\mathbf{r}\geq\mathbf{0}} c \sum_{i,j=1}^{n} \mathbf{A}_{ij} \left\| \frac{1}{\sqrt{\mathbf{D}_{ii}}}\mathbf{r}_i - \frac{1}{\sqrt{\mathbf{D}_{jj}}}\mathbf{r}_j \right\|^2 + (1-c)\sum_{i=1}^{n} ||\mathbf{r}_i - \mathbf{e}_i||^2,$$

where $\mathbf{D} \in \mathbb{R}^{n\times n}$ is the degree matrix of $\mathbf{A}$, $c \in (0, 1)$ is the regularization parameter, $\mathbf{r}$ is the anomaly score vector after the propagation of the initial faults in $\mathbf{e}$. We can re-write the above problem as

$$\min_{\mathbf{r}\geq\mathbf{0}} c\mathbf{r}^\top(\mathbf{I}_n - \tilde{\mathbf{A}})\mathbf{r} + (1-c)||\mathbf{r} - \mathbf{e}||_F^2, \tag{5}$$

where $\mathbf{I}_n$ is the identity matrix and $\tilde{\mathbf{A}} = \mathbf{D}^{-1/2}\mathbf{A}\mathbf{D}^{-1/2}$ is the degree-normalized version of $\mathbf{A}$. Similarly, we will use $\tilde{\mathbf{P}}$ as the degree-normalized $\mathbf{P}$ in the sequel. The first term in Equation (5) is the *smoothness constraint* [31], meaning that a good ranking function should assign similar values to nearby nodes in the network. The second term is the *fitting constraint*, which means that the final status should be close to the initial configuration. The tradeoff between these two competing constraints is controlled by a positive parameter c: A small c encourages a sufficient propagation, and a big c actually suppresses the propagation. The optimal solution of problem (5) is [31]

$$\mathbf{r} = (1-c)(\mathbf{I}_n - c\tilde{\mathbf{A}})^{-1}\mathbf{e}, \tag{6}$$

which establishes an explicit, closed-form solution between the initial configuration $\mathbf{e}$ and the final status $\mathbf{r}$ through propagation.

To encode the information of the broken network, we propose to use $\mathbf{r}$ to reconstruct the broken network $\mathcal{G}_b$. The intuition is illustrated in Figure 2. If there exists a broken link in $\mathcal{G}_b$, for example, $\tilde{\mathbf{P}}_{ij}$ is large, then ideally at least one of the nodes i and j should be abnormal or, equivalently, either $\mathbf{r}_i$ or $\mathbf{r}_j$ should be large. Thus, we can use the product of $\mathbf{r}_i$ and $\mathbf{r}_j$ to reconstruct the value of $\tilde{\mathbf{P}}_{ij}$. In Section 5, we will further discuss how to normalize them to avoid extreme values. Then, the loss of reconstructing the broken link $\tilde{\mathbf{P}}_{ij}$ can be calculated by $(\mathbf{r}_i \cdot \mathbf{r}_j - \tilde{\mathbf{P}}_{ij})^2$. The reconstruction error of the whole broken network is then $||(\mathbf{r}\mathbf{r}^\top) \circ \mathbf{M} - \tilde{\mathbf{P}}||_F^2$. Here, $\circ$ is the element-wise operator, and

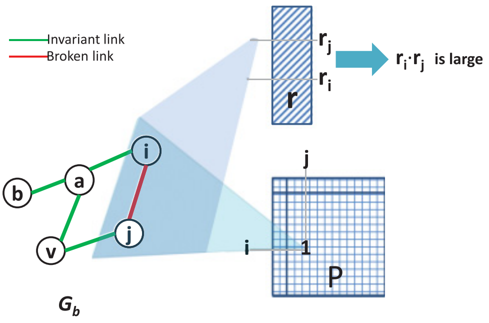


Fig. 2. Reconstruction of the broken invariant network using anomaly score vector **r**.

$\mathbf{M}$ is the logical matrix of the invariant network $\mathcal{G}_l$ (1 with edge, 0 without edge). Let $\mathbf{B} = (1 - c)(\mathbf{I}_n - c\tilde{\mathbf{A}})^{-1}$, by substituting $\mathbf{r}$ we obtain the following objective function:

$$\min_{\mathbf{e}_i \in \{0,1\}, 1 \le i \le n} ||(\mathbf{B}\mathbf{e}\mathbf{e}^\top \mathbf{B}^\top) \circ \mathbf{M} - \tilde{\mathbf{P}}||_F^2. \tag{7}$$

Considering that the integer programming in problem (7) is NP-hard, we relax it by using the ℓ_1 penalty on $\mathbf{e}$ with parameter τ to control the number of non-zero entries in $\mathbf{e}$ [27]. Then we reach the following objective function:

$$\min_{\mathbf{e} \ge \mathbf{0}} ||(\mathbf{B}\mathbf{e}\mathbf{e}^\top \mathbf{B}^\top) \circ \mathbf{M} - \tilde{\mathbf{P}}||_F^2 + \tau ||\mathbf{e}||_1 \tag{8}$$

3.2. Learning Algorithm

In this section, we present an iterative multiplicative updating algorithm to optimize the objective function in Equation (8). The objective function is invariant under these updates if and only if **e** are at a stationary point [20]. The solution is presented in the following theorem, which is derived from the Karush-Kuhn-Tucker (KKT) complementarity condition [3]. Detailed theoretical analysis of the optimization procedure will be presented in the next section.

THEOREM 1. *Updating* **e** *according to Equation* (9) *will monotonically decrease the objective function in Equation* (8) *until convergence,*

$$\mathbf{e} \leftarrow \mathbf{e} \circ \left\{ \frac{4\mathbf{B}^\top (\tilde{\mathbf{P}} \circ \mathbf{M})^\top \mathbf{B}\mathbf{e}}{4\mathbf{B}^\top [\mathbf{M} \circ (\mathbf{B}\mathbf{e}\mathbf{e}^\top \mathbf{B}^\top)]\mathbf{B}\mathbf{e} + \tau \mathbf{1}_n} \right\}^{\frac{1}{4}}, \tag{9}$$

where $\circ$, $\frac{[\cdot]}{[\cdot]}$, *and* $(\cdot)^{\frac{1}{4}}$ *are element-wise operators.*

Based on Theorem 1, we develop the iterative multiplicative updating algorithm for optimization and summarize it in Algorithm 1. We refer to this ranking algorithm as RCA.

3.3. Theoretical Analysis

3.3.1. Derivation. We derive the solution to problem (9) following the constrained optimization theory [3]. Since the objective function is not jointly convex, we adopt an effective multiplicative updating algorithm to find a local optimal solution. We prove Theorem 1 in the following.

We formulate the Lagrange function for optimization $L = ||(\mathbf{B}\mathbf{e}\mathbf{e}^\top \mathbf{B}^\top) \circ \mathbf{M} - \tilde{\mathbf{P}}||_F^2 + \tau \mathbf{1}_n^\top \mathbf{e}$. Obviously, $\mathbf{B}$, $\mathbf{M}$, and $\tilde{\mathbf{P}}$ are symmetric matrix. Let $\mathbf{F} = (\mathbf{B}\mathbf{e}\mathbf{e}^\top \mathbf{B}^\top) \circ \mathbf{M}$, then

ALGORITHM 1: Ranking Causal Anomalies (RCA)

```
Input: Network $\mathcal{G}_l$ denoting the invariant network with $n$ nodes and is represented by an
       adjacency matrix $\mathbf{A}$, $c$ is the network propagation parameter, $\tau$ is the parameter to
       control the sparsity of $\mathbf{e}$, $\tilde{\mathbf{P}}$ is the normalized adjacency matrix of the broken
       network, $\mathbf{M}$ is the logical matrix of $\mathcal{G}_l$ (1 with edge, 0 without edge)
Output: Ranking vector $\mathbf{e}$
begin
    for $i \leftarrow 1$to $n$ do
        $\mathbf{D}_{ii} \leftarrow \sum_{j=1}^{n} \mathbf{A}_{ij}$;
    end
    $\mathbf{D} \leftarrow diag(\mathbf{D}_{11}, \ldots, \mathbf{D}_{ii})$;
    $\tilde{\mathbf{A}} \leftarrow \mathbf{D}^{-1/2}\mathbf{A}\mathbf{D}^{-1/2}$;
    Initialize $\mathbf{e}$ with random values between (0,1];
    $\mathbf{B} \leftarrow (1-c)(\mathbf{I}_n - c\tilde{\mathbf{A}})^{-1}$;
    repeat
        Update $\mathbf{e}$ by Equation (9);
    until convergence;
end
```

$$
\begin{aligned}
\frac{\partial}{\partial \mathbf{e}_m}(\mathbf{F} - \tilde{\mathbf{P}})_{ij}^2 &= 2(\mathbf{F}_{ij} - \tilde{\mathbf{P}}_{ij})\frac{\partial \mathbf{F}_{ij}}{\mathbf{e}_m} \\
&= 4(\mathbf{F}_{ij} - \tilde{\mathbf{P}}_{ij})\mathbf{M}_{ij}(\mathbf{B}_{mi}^{\top}\mathbf{B}_{j:}\mathbf{e}) \text{ (by symmetry)} \\
&= 4\mathbf{B}_{mi}^{\top}(\mathbf{F}_{ij} - \tilde{\mathbf{P}}_{ij})\mathbf{M}_{ij}(\mathbf{B}\mathbf{e})_{j:}.
\end{aligned}
\tag{10}
$$

It follows that

$$
\frac{\partial ||\mathbf{F} - \tilde{\mathbf{P}}||_{\mathbf{F}}^2}{\partial \mathbf{e}_m} = 4\mathbf{B}_{m:}^{\top}[(\mathbf{F} - \tilde{\mathbf{P}}) \circ \mathbf{M}](\mathbf{B}\mathbf{e}), \tag{11}
$$

and thereby

$$
\frac{\partial ||\mathbf{F} - \tilde{\mathbf{P}}||_F^2}{\partial \mathbf{e}} = 4\mathbf{B}^{\top}[(\mathbf{F} - \tilde{\mathbf{P}}) \circ \mathbf{M}](\mathbf{B}\mathbf{e}). \tag{12}
$$

Thus, the partial derivative of Lagrange function with respect to **e** is

$$
\nabla_{\mathbf{e}} L = 4\mathbf{B}^{\top}[(\mathbf{B}\mathbf{e}\mathbf{e}^{\top}\mathbf{B}^{\top} - \tilde{\mathbf{P}}) \circ \mathbf{M}]\mathbf{B}\mathbf{e} + \tau \mathbf{1}_n, \tag{13}
$$

where $\mathbf{1}_n$ is the $n \times 1$ vector of all ones. Using the KKT complementarity condition [3] for the non-negative constraint on **e**, we have

$$
\nabla_{\mathbf{e}} L \circ \mathbf{e} = \mathbf{0}. \tag{14}
$$

The above formula leads to the updating rule for **e** that is shown in Equation (9).

3.3.2. Convergence. We use the auxiliary function approach [20] to prove the convergence of Equation (9) in Theorem 1. We first introduce the definition of auxiliary function as follows.

Definition 3.1. $Z(h, \hat{h})$ is an auxiliary function for $L(h)$ if the conditions

$$
Z(h, \hat{h}) \geq L(h) \qquad \text{and} \qquad Z(h, h) = L(h), \tag{15}
$$

are satisfied for any given $h, \hat{h}$ [20].

Lemma 3.1. *If Z is an auxiliary function for L, then L is non-increasing under the update [20],*

$$h^{(t+1)} = \underset{h}{\operatorname{argmin}}\, Z(h, h^{(t)}). \tag{16}$$

Theorem 2. *Let $L(\mathbf{e})$ denote the sum of all terms in L containing $\mathbf{e}$. The following function:*

$$\begin{aligned} Z(\mathbf{e}, \hat{\mathbf{e}}) = {} & -2\sum_{ij}[\mathbf{B}^\top(\tilde{\mathbf{P}} \circ \mathbf{M})^\top \mathbf{B}]_{ij}\hat{\mathbf{e}}_i\hat{\mathbf{e}}_j\left(1 + \log\frac{\mathbf{e}_i\mathbf{e}_j}{\hat{\mathbf{e}}_i\hat{\mathbf{e}}_j}\right) \\ & + \sum_i\{\mathbf{B}^\top[\mathbf{M} \circ (\mathbf{B}\hat{\mathbf{e}}\hat{\mathbf{e}}^\top\mathbf{B}^\top)]\mathbf{B}\hat{\mathbf{e}}\}_i\frac{\mathbf{e}_i^4}{\hat{\mathbf{e}}_i^3} + \frac{\tau}{4}\sum_i\frac{\mathbf{e}_i^4 + 3\hat{\mathbf{e}}_i^4}{\hat{\mathbf{e}}_i^3} \end{aligned} \tag{17}$$

is an auxiliary function for $L(\mathbf{e})$. Furthermore, it is a convex function in $\mathbf{e}$ and has a global minimum.

Proof. According to Definition 3.1, in this proof, we need to verify that (1) $Z(\mathbf{e}, \hat{\mathbf{e}}) \geq L(\mathbf{e})$, (2) $Z(\mathbf{e}, \mathbf{e}) = L(\mathbf{e})$, and (3) $Z(\mathbf{e}, \hat{\mathbf{e}})$ is a convex function in $\mathbf{e}$, which are respectively proved as follows.

First, omitting some constants, we write $L(\mathbf{e})$ as

$$L(\mathbf{e}) = -2\mathrm{tr}(\mathbf{B}^\top(\tilde{\mathbf{P}} \circ \mathbf{M})^\top\mathbf{B}\mathbf{e}\mathbf{e}^\top) + \mathrm{tr}([\mathbf{M} \circ (\mathbf{B}\mathbf{e}\mathbf{e}^\top\mathbf{B}^\top)]^\top(\mathbf{B}\mathbf{e}\mathbf{e}^\top\mathbf{B}^\top)) + \tau\sum_i\mathbf{e}_i. \tag{18}$$

To prove (1) $Z(\mathbf{e}, \hat{\mathbf{e}}) \geq L(\mathbf{e})$, we deduce the upper bound for each term in Equation (18). Using the inequality $z \geq 1 + \log z$, which holds for any $z > 0$, we have

$$\frac{\mathbf{e}_i\mathbf{e}_j}{\hat{\mathbf{e}}_i\hat{\mathbf{e}}_j} \geq 1 + \log\frac{\mathbf{e}_i\mathbf{e}_j}{\hat{\mathbf{e}}_i\hat{\mathbf{e}}_j}.$$

Then we can write an upper bound for the first term

$$\begin{aligned} & -2\mathrm{tr}(\mathbf{B}^\top(\tilde{\mathbf{P}} \circ \mathbf{M})^\top\mathbf{B}\mathbf{e}\mathbf{e}^\top) = -2\sum_{ij}[\mathbf{B}^\top(\tilde{\mathbf{P}} \circ \mathbf{M})^\top\mathbf{B}]_{ij}\mathbf{e}_i\mathbf{e}_j \\ & \leq -2\sum_{ij}[\mathbf{B}^\top(\tilde{\mathbf{P}} \circ \mathbf{M})^\top\mathbf{B}]_{ij}\hat{\mathbf{e}}_i\hat{\mathbf{e}}_j\left(1 + \log\frac{\mathbf{e}_i\mathbf{e}_j}{\hat{\mathbf{e}}_i\hat{\mathbf{e}}_j}\right). \end{aligned} \tag{19}$$

For the second term, we can rewrite it by

$$\mathrm{tr}([\mathbf{M} \circ (\mathbf{B}\mathbf{e}\mathbf{e}^\top\mathbf{B}^\top)]^\top(\mathbf{B}\mathbf{e}\mathbf{e}^\top\mathbf{B}^\top)) = \sum_{xyijpq}\mathbf{M}_{xy}\mathbf{B}_{xi}\mathbf{e}_i\mathbf{e}_j\mathbf{B}_{yj}\mathbf{B}_{xp}\mathbf{e}_p\mathbf{e}_q\mathbf{B}_{yq}.$$

Let $\mathbf{e}_i = \hat{\mathbf{e}}_i s_i$, $\mathbf{e}_j = \hat{\mathbf{e}}_j s_j$, $\mathbf{e}_p = \hat{\mathbf{e}}_p s_p$, and $\mathbf{e}_q = \hat{\mathbf{e}}_q s_q$ for some non-negative values s_i, s_j, s_p and s_q; we can further rewrite it by

$$\begin{aligned} & \sum_{xyijpq}\mathbf{M}_{xy}\mathbf{B}_{xi}\hat{\mathbf{e}}_i\hat{\mathbf{e}}_j\mathbf{B}_{yj}\mathbf{B}_{xp}\hat{\mathbf{e}}_p\hat{\mathbf{e}}_q\mathbf{B}_{yq}s_is_js_ps_q \\ & \leq \sum_{xyijpq}\mathbf{M}_{xy}\mathbf{B}_{xi}\hat{\mathbf{e}}_i\hat{\mathbf{e}}_j\mathbf{B}_{yj}\mathbf{B}_{xp}\hat{\mathbf{e}}_p\hat{\mathbf{e}}_q\mathbf{B}_{yq}\frac{s_i^4 + s_j^4 + s_p^4 + s_q^4}{4} \\ & = \frac{1}{4}\left(\sum_i\mathbf{Q}_i\frac{\mathbf{e}_i^4}{\hat{\mathbf{e}}_i^3} + \sum_j\mathbf{Q}_j\frac{\mathbf{e}_j^4}{\hat{\mathbf{e}}_j^3} + \sum_p\mathbf{Q}_p\frac{\mathbf{e}_p^4}{\hat{\mathbf{e}}_p^3} + \sum_q\mathbf{Q}_q\frac{\mathbf{e}_q^4}{\hat{\mathbf{e}}_q^3}\right) = \sum_i\mathbf{Q}_i\frac{\mathbf{e}_i^4}{\hat{\mathbf{e}}_i^3}, \end{aligned} \tag{20}$$

where $\mathbf{Q} = \mathbf{B}^\top[\mathbf{M} \circ (\mathbf{B}\hat{\mathbf{e}}\hat{\mathbf{e}}^\top\mathbf{B}^\top)]\mathbf{B}\hat{\mathbf{e}}$. Here, the last equation is obtained by switching indexes.

For the third term, using the fact that $2ab \leq a^2 + b^2$, we have

$$\tau \sum_i \mathbf{e}_i \leq \frac{\tau}{2} \sum_i \frac{\mathbf{e}_i^2 + \hat{\mathbf{e}}_i^2}{\hat{\mathbf{e}}_i} \leq \frac{\tau}{4} \sum_i \frac{\mathbf{e}_i^4 + 3\hat{\mathbf{e}}_i^4}{\hat{\mathbf{e}}_i^3}. \tag{21}$$

Therefore, by collecting Equation (19), Equation (20), and Equation (21), we have verified (1) $Z(\mathbf{e}, \hat{\mathbf{e}}) \geq L(\mathbf{e})$. Moreover, by substituting $\hat{\mathbf{e}}$ with $\mathbf{e}$ in $Z(\mathbf{e}, \hat{\mathbf{e}})$, we can directly verify (2) $Z(\mathbf{e}, \mathbf{e}) = L(\mathbf{e})$.

To prove (3) $Z(\mathbf{e}, \hat{\mathbf{e}})$ is a convex function in $\mathbf{e}$, we need to show the Hessian matrix $\nabla^2_{\mathbf{e}} Z(\mathbf{e}, \hat{\mathbf{e}})$ is positive-definite. First, we derive

$$\frac{\partial Z(\mathbf{e}, \hat{\mathbf{e}})}{\partial \mathbf{e}_i} = -4[\mathbf{B}^\top(\tilde{\mathbf{P}} \circ \mathbf{M})^\top \mathbf{B}\hat{\mathbf{e}}]_i \frac{\hat{\mathbf{e}}_i}{\mathbf{e}_i} + 4\{\mathbf{B}^\top[\mathbf{M} \circ (\mathbf{B}\hat{\mathbf{e}}\hat{\mathbf{e}}^\top\mathbf{B}^\top)]\mathbf{B}\hat{\mathbf{e}}\}_i \frac{\mathbf{e}_i^3}{\hat{\mathbf{e}}_i^3} + \tau \frac{\mathbf{e}_i^3}{\hat{\mathbf{e}}_i^3}.$$

Then the second-order derivative is

$$\frac{\partial^2 Z(\mathbf{e}, \hat{\mathbf{e}})}{\partial \mathbf{e}_i \partial \mathbf{e}_j} = \delta_{ij} \left(4[\mathbf{B}^\top(\tilde{\mathbf{P}} \circ \mathbf{M})^\top \mathbf{B}\hat{\mathbf{e}}]_i \frac{\hat{\mathbf{e}}_i}{\mathbf{e}_i^2} + 12\{\mathbf{B}^\top[\mathbf{M} \circ (\mathbf{B}\hat{\mathbf{e}}\hat{\mathbf{e}}^\top\mathbf{B}^\top)]\mathbf{B}\hat{\mathbf{e}}\}_i \frac{\mathbf{e}_i^2}{\hat{\mathbf{e}}_i^3} + 3\tau \frac{\mathbf{e}_i^2}{\hat{\mathbf{e}}_i^3} \right),$$

where δ_{ij} is the Kronecker delta. $\delta_{ij} = 1$ if $i = j$; $\delta_{ij} = 0$ otherwise.

Therefore, the Hessian matrix $\nabla^2_{\mathbf{e}} Z(\mathbf{e}, \hat{\mathbf{e}})$ is a diagonal matrix with positive diagonal entries. Hence, we verify that (3) $\nabla^2_{\mathbf{e}} Z(\mathbf{e}, \hat{\mathbf{e}})$ is positive-definite and $Z(\mathbf{e}, \hat{\mathbf{e}})$ is a convex function in $\mathbf{e}$. This completes the proof. □

Based on Theorem 2, we can minimize $Z(\mathbf{e}, \hat{\mathbf{e}})$ with respect to $\mathbf{e}$ with $\hat{\mathbf{e}}$ fixed. We set $\nabla_{\mathbf{e}} Z(\mathbf{e}, \hat{\mathbf{e}}) = \mathbf{0}$, and get the following updating formula:

$$\mathbf{e} \leftarrow \hat{\mathbf{e}} \circ \left\{ \frac{4\mathbf{B}^\top(\tilde{\mathbf{P}} \circ \mathbf{M})^\top \mathbf{B}\hat{\mathbf{e}}}{4\mathbf{B}^\top \left[\mathbf{M} \circ (\mathbf{B}\hat{\mathbf{e}}\hat{\mathbf{e}}^\top\mathbf{B}^\top)\right] \mathbf{B}\hat{\mathbf{e}} + \tau \mathbf{1}_n} \right\}^{\frac{1}{4}}, \tag{22}$$

which is consistent with the updating formula derived from the KKT condition aforementioned.

From Lemma 3.1 and Theorem 2, for each subsequent iteration of updating $\mathbf{e}$, we have $L(\mathbf{e}^0) = Z(\mathbf{e}^0, \mathbf{e}^0) \geq Z(\mathbf{e}^1, \mathbf{e}^0) \geq Z(\mathbf{e}^1, \mathbf{e}^1) = L(\mathbf{e}^1) \geq \cdots \geq L(\mathbf{e}^{Iter})$. Thus $L(\mathbf{e})$ monotonically decreases. Since the objective function Equation (8) is lower bounded by 0, the correctness of Theorem 1 is proven.

3.3.3. Complexity Analysis. In Algorithm 1, we need to calculate the inverse of an $n \times n$ matrix, which takes $\mathcal{O}(n^3)$ time. In each iteration, the multiplication between two $n \times n$ matrices is inevitable, and thus the overall time complexity of Algorithm 1 is $\mathcal{O}(Iter \cdot n^3)$, where *Iter* is the number of iterations needed for convergence. In the following section, we will propose an efficient algorithm that reduces the time complexity to $\mathcal{O}(Iter \cdot n^2)$.

4. COMPUTATIONAL SPEED-UP

In this section, we will propose an efficient algorithm that avoids the matrix inverse calculations as well as the multiplication between two $n \times n$ matrices. The time complexity can be reduced to $\mathcal{O}(Iter \cdot n^2)$.

We achieve the computational speed-up by relaxing the objective function in Equation (8) to jointly optimize $\mathbf{r}$ and $\mathbf{e}$. The objective function is shown in the following:

$$\min_{\mathbf{e} \geq \mathbf{0}, \mathbf{r} \geq \mathbf{0}} \underbrace{c\mathbf{r}^\top(\mathbf{I}_n - \tilde{\mathbf{A}})\mathbf{r} + (1-c)||\mathbf{r} - \mathbf{e}||_F^2}_{\text{Fault propagation}} + \underbrace{\lambda||(\mathbf{r}\mathbf{r}^\top) \circ \mathbf{M} - \tilde{\mathbf{P}}||_F^2 + \tau||\mathbf{e}||_1}_{\text{Vanishing correlation reconstruction}}. \tag{23}$$

To optimize this objective function, we can use an alternating scheme. That is, we optimize the objective with respect to $\mathbf{r}$ while fixing $\mathbf{e}$ and vice versa. This procedure continues until convergence. The objective function is invariant under these updates if and only if $\mathbf{r}$, $\mathbf{e}$ are at a stationary point [20]. Specifically, the solution to the optimization problem in Equation (23) is based on the following theorem, which is derived from the KKT complementarity condition [3]. The derivation of it and the proof of Theorem 3 is similar to that of Theorem 1.

THEOREM 3. *Alternatively updating* $\mathbf{e}$ *and* $\mathbf{r}$ *according to Equation* (24) *and Equation* (25) *will monotonically decrease the objective function in Equation* (23) *until convergence,*

$$\mathbf{r} \leftarrow \mathbf{r} \circ \left\{ \frac{\tilde{\mathbf{A}}\mathbf{r} + 2\lambda(\tilde{\mathbf{P}} \circ \mathbf{M})\mathbf{r} + (1-c)\mathbf{e}}{\mathbf{r} + 2\lambda \left[(\mathbf{r}\mathbf{r}^\top) \circ \mathbf{M}\right] \mathbf{r}} \right\}^{\frac{1}{4}}, \tag{24}$$

$$\mathbf{e} \leftarrow \mathbf{e} \circ \left[\frac{2(1-c)\mathbf{r}}{\tau \mathbf{1}_n + 2(1-c)\mathbf{e}} \right]^{\frac{1}{2}}. \tag{25}$$

Based on Theorem 3, we can develop the iterative multiplicative updating algorithm for optimization similar to Algorithm 1. Due to the page limit, we skip the details. We refer to this ranking algorithm as R-RCA. From Equation (24) and Equation (25), we observe that the calculation of the inverse of the $n \times n$ matrix and the multiplication between two $n \times n$ matrices in Algorithm 1 are not necessary. As we will see in Section 8.5, the relaxed versions of our algorithm can greatly improve the computational efficiency.

5. SOFTMAX NORMALIZATION

In Section 3, we use the product $\mathbf{r}_i \cdot \mathbf{r}_j$ as the strength of evidence that the correlation between node i and j is vanishing (broken). However, it suffers from the extreme values in the ranking values $\mathbf{r}$. To reduce the influence of the extreme values or outliers, we employ the softmax normalization on the ranking values $\mathbf{r}$. The ranking values are nonlinearly transformed using the sigmoidal function before the multiplication is performed. Thus, the reconstruction error is expressed by $||(\sigma(\mathbf{r})\sigma^\top(\mathbf{r})) \circ \mathbf{M} - \tilde{\mathbf{P}}||_F^2$, where $\sigma(\cdot)$ is the softmax function with

$$\sigma(\mathbf{r})_i = \frac{e^{\mathbf{r}_i}}{\sum_{k=1}^n e^{\mathbf{r}_k}}, (i = 1, \ldots, n). \tag{26}$$

The corresponding objective function in Algorithm 1 is modified as follows:

$$\min_{\mathbf{e} \geq \mathbf{0}} ||(\sigma(\mathbf{Be})\sigma^\top(\mathbf{Be})) \circ \mathbf{M} - \tilde{\mathbf{P}}||_F^2 + \tau||\mathbf{e}||_1. \tag{27}$$

Similarly, the objective function for Equation (23) is modified as follows:

$$\min_{\mathbf{e} \geq \mathbf{0}, \mathbf{r} \geq \mathbf{0}} c\mathbf{r}^\top(\mathbf{I}_n - \tilde{\mathbf{A}})\mathbf{r} + (1-c)||\mathbf{r} - \mathbf{e}||_F^2 + \lambda||(\sigma(\mathbf{r})\sigma^\top(\mathbf{r})) \circ \mathbf{M} - \tilde{\mathbf{P}}||_F^2 + \tau||\mathbf{e}||_1. \tag{28}$$

The optimization of these two objective functions are based on the following two theorems.

THEOREM 4. *Updating* $\mathbf{e}$ *according to Equation* (29) *will monotonically decrease the objective function in Equation* (27) *until convergence,*

$$\mathbf{e} \leftarrow \mathbf{e} \circ \left\{ \frac{4\mathbf{B}^\top \Psi(\tilde{\mathbf{P}} \circ \mathbf{M})\sigma(\mathbf{Be})}{4\left[\mathbf{B}^\top \left(\Psi\sigma(\mathbf{Be})\sigma^\top(\mathbf{Be})\right) \circ \mathbf{M}\right]\sigma(\mathbf{Be}) + \tau\mathbf{1}_n} \right\}^{\frac{1}{4}}, \tag{29}$$

where $\Psi = \{\mathrm{diag}[\sigma(\mathbf{Be})] - \sigma(\mathbf{Be})\sigma^\top(\mathbf{Be})\}$.

THEOREM 5. *Updating* $\mathbf{r}$ *according to Equation* (30) *will monotonically decrease the objective function in Equation* (28) *until convergence,*

$$\mathbf{r} \leftarrow \mathbf{r} \circ \left\{ \frac{\tilde{\mathbf{A}}\mathbf{r} + 2\lambda[((\sigma(\mathbf{r})\mathbf{1}_n^\top) \circ \tilde{\mathbf{P}} + \rho\Lambda) \circ \mathbf{M}]\sigma(\mathbf{r}) + (1-c)\mathbf{e}}{\mathbf{r} + 2\lambda[((\sigma(\mathbf{r}) \circ \sigma(\mathbf{r}))\sigma^\top(\mathbf{r}) + \sigma(\mathbf{r})(\sigma^\top(\mathbf{r})\tilde{\mathbf{P}})) \circ \mathbf{M}]\sigma(\mathbf{r})} \right\}^{\frac{1}{4}}, \tag{30}$$

where $\Lambda = \sigma(\mathbf{r})\sigma^\top(\mathbf{r})$ *and* $\rho = \sigma^\top(\mathbf{r})\sigma(\mathbf{r})$.

Theorem 4 and Theorem 5 can be proven with a similar strategy to that of Theorem 1. We refer to the ranking algorithms with *softmax* normalization (Equation (27) and Equation (28)) as RCA-SOFT and R-RCA-SOFT, respectively.

6. TEMPORAL SMOOTHING ON MULTIPLE BROKEN NETWORKS

As discussed in Section 1, although the number of anomaly nodes could increase due to fault propagation in the network, the root cause anomalies will be stable within a short time period T [17]. Based on this intuition, we further develop a smoothing strategy by jointly considering the temporal broken networks. Specifically, we add a smoothing term $||\mathbf{e}^{(t)} - \mathbf{e}^{(t-1)}||_2^2$ to the objective functions. Here, $\mathbf{e}^{(t-1)}$ and $\mathbf{e}^{(t)}$ are causal anomaly ranking vectors for two successive time points. For example, the objective function of algorithm RCA with temporal broken networks smoothing is shown in Equation (31),

$$\min_{\mathbf{e}^{(t)} \geq \mathbf{0}, 1 \leq t \leq T} \sum_{t=1}^{T} [||(\mathbf{Be}^{(t)}(\mathbf{e}^{(t)})^\top\mathbf{B}^\top) \circ \mathbf{M} - \tilde{\mathbf{P}}^{(t)}||_F^2 + \tau||\mathbf{e}^{(t)}||_1] + \underbrace{\alpha||\mathbf{e}^{(t)} - \mathbf{e}^{(t-1)}||_2^2}_{\text{Temporal smoothing}}. \tag{31}$$

Here, $\tilde{\mathbf{P}}^{(t)}$ is the degree-normalized adjacency matrix of broken network at time point t. Similarly to the discussion in Section 3.3, we can derive the updating formula of Equation (31) as follows:

$$\mathbf{e}^{(t)} \leftarrow \mathbf{e}^{(t)} \circ \left\{ \frac{4\mathbf{B}^\top(\tilde{\mathbf{P}}^{(t)} \circ \mathbf{M})^\top\mathbf{Be}^{(t)} + 2\alpha\mathbf{e}^{(t-1)}}{4\mathbf{B}^\top\left[\mathbf{M} \circ (\mathbf{Be}^{(t)}(\mathbf{e}^{(t)})^\top\mathbf{B}^\top)\right]\mathbf{Be}^{(t)} + \tau\mathbf{1}_n + 2\alpha\mathbf{e}^{(t)}} \right\}^{\frac{1}{4}}. \tag{32}$$

The updating formula for R-RCA, RCA-SOFT, and R-RCA-SOFT with temporal broken networks smoothing is similar. Due to the space limit, we skip the details. We refer to the ranking algorithms with temporal networks smoothing as T-RCA, T-R-RCA, T-RCA-SOFT, and T-R-RCA-SOFT respectively.

7. LEVERAGING PRIOR KNOWLEDGE

In real-life applications, we may have prior knowledge that reflects to what extent a node is harmed by the causal anomalies at a certain time point. In this section, we extend our RCA model to a semi-supervised setting to incorporate such prior knowledge so the performance of causal anomaly inference can be further enhanced.

7.1. Leveraging Node Attributes

One common type of prior knowledge can be represented by a numeric attribute for each node that measures the degree that a node is anomalous at the observation time point. For example, the attribute value can be the absolute bias of the monitoring data of a node that deviates from its predicted normal value at a time point [6].

Let $\mathbf{v}_i \geq 0$ represent the anomalous degree of node i; our goal is to leverage these attributes in a principled manner to improve the causal anomaly inference capability of our model. It is important to note that, usually the attributes only partially covers the nodes in the invariant network due to the short of prior knowledge. That is, let $\mathcal{V}$ be the set of all nodes in the invariant network, then $\mathbf{v}_i$ is only available for node $i \in \mathcal{V}_p$, where $\mathcal{V}_p \subseteq \mathcal{V}$. To account for this sparsity of prior knowledge, we define an indicator $\mathbf{u}_i \in \{0, 1\}$ for each node i s.t. $\mathbf{u}_i = 1$ if node i has a valid $\mathbf{v}_i$; $\mathbf{u}_i = 0$ otherwise.

Because $\mathbf{v}_i$ measures the degree that node i is impacted by causal anomalies, we can use $\mathbf{r}_i$ in Equation (6) to approximate $\mathbf{v}_i$. Specifically, we want to minimize the inconsistency of $\mathbf{u}_i(\mathbf{r}_i - \mathbf{v}_i)^2$. Let $\mathbf{v} \in \mathbb{R}_+^{n\times 1}$ with the ith entry as $\mathbf{v}_i$ (note $\mathbf{v}_i = 0$ if $i \notin \mathcal{V}_p$), and $\mathbf{D}_u \in \{0, 1\}^{n\times n}$ be a diagonal matrix with $(\mathbf{D}_u)_{ii}$ as $\mathbf{u}_i$; then we can obtain a matrix form of the inconsistencies as $(\mathbf{r} - \mathbf{v})^\top\mathbf{D}_u(\mathbf{r} - \mathbf{v})$. By integrating this loss function with our RCA model in Equation (6), and replacing $\mathbf{r}$ by $\mathbf{Be}$, we obtain an objective function that enables node attributes as follows:

$$\min_{\mathbf{e}\geq\mathbf{0}} ||(\mathbf{B}\mathbf{e}\mathbf{e}^\top\mathbf{B}^\top)\circ\mathbf{M}-\tilde{\mathbf{P}}||_F^2+\tau||\mathbf{e}||_1+\underbrace{\beta(\mathbf{B}\mathbf{e}-\mathbf{v})^\top\mathbf{D}_u(\mathbf{B}\mathbf{e}-\mathbf{v})}_{\text{Leveraging prior knowledge}}, \tag{33}$$

where β is a parameter that measures the importance of prior knowledge. Intuitively, the more reliable the prior knowledge, the larger the value of β.

The objective function in Equation (33) can be optimized by an updating formula as summarized by the following theorem. The derivation of this formula follows a similar strategy as those discussed in Section 3.3.

THEOREM 6. *Updating* $\mathbf{e}$ *according to Equation* (34) *will monotonically decrease the objective function in Equation* (33) *until convergence,*

$$\mathbf{e} \leftarrow \mathbf{e} \circ \left\{ \frac{4\mathbf{B}^\top(\tilde{\mathbf{P}}\circ\mathbf{M})^\top\mathbf{B}\mathbf{e} + 2\beta\mathbf{B}^\top(\mathbf{u}\circ\mathbf{v})}{4\mathbf{B}^\top\left[\mathbf{M}\circ(\mathbf{B}\mathbf{e}\mathbf{e}^\top\mathbf{B}^\top)\right]\mathbf{B}\mathbf{e} + 2\beta\mathbf{B}^\top\left[\mathbf{u}\circ(\mathbf{B}\mathbf{e})\right] + \tau\mathbf{1}_n} \right\}^{\frac{1}{4}}. \tag{34}$$

The formal algorithm that considers node attributes can be similarly formulated as Algorithm 1. In the following, we refer to the semi-supervised ranking algorithm using Equation (34) as RCA-SEMI.

7.2. Learning the Reliability of Prior Knowledge

In real practice, because of noises, not all node attributes are reliable. It is likely that a considerable part of $\{\mathbf{v}_i\}$ is inconsistent with the current broken status of the invariant network and can mislead causal anomaly inference if we trust them without differentiation. To avoid the problem caused by noisy node attributes, we next develop a strategy to automatically select reliable node attributes from unreliable ones to improve the robustness of our model.

In Equation (33), all valid node attributes $\mathbf{v}_i$ are treated equally by assigning the same weights $\mathbf{u}_i = 1$. A more practical design is to allow $\mathbf{u}_i$ to vary based on the reliability of $\mathbf{v}_i$. Ideally, $\mathbf{u}_i$ is small if $\mathbf{v}_i$ is inconsistent with the anomalous status of node i as inferred from fault propagation. This inconsistency can be measured by $(\mathbf{r}_i - \mathbf{v}_i)^2$. Therefore, we can modify the optimization problem in Equation (33) as follows to allow automatic learning of $\mathbf{u}$:

$$\begin{aligned} &\min_{\mathbf{e},\mathbf{u}\geq\mathbf{0}} ||(\mathbf{B}\mathbf{e}\mathbf{e}^\top\mathbf{B}^\top)\circ\mathbf{M}-\tilde{\mathbf{P}}||_F^2+\tau||\mathbf{e}||_1+\beta\sum_{i\in\mathcal{V}_P}\mathbf{u}_i(\mathbf{B}\mathbf{e}-\mathbf{v})_i^2+\gamma\sum_{i\in\mathcal{V}_p}\mathbf{u}_i^2 \\ &\text{s.t.} \sum_{i\in\mathcal{V}_p}\mathbf{u}_i = |\mathcal{V}_p|. \end{aligned} \tag{35}$$

In the above equation, we enforce the equality constraint to allow different $\mathbf{u}_i$ to be correlated and comparable for selection purpose. The ℓ_2 norm on $\mathbf{u}$ is enforced to avoid trivial solutions. Without it, all entries in $\mathbf{u}$ will be zeros except for $\mathbf{u}_i$ corresponding to the least inconsistency $(\mathbf{Be}-\mathbf{v})_i^2$. Here, γ is a parameter controlling the complexity of $\mathbf{u}$. Typically, larger γ results in more non-zero entries in $\mathbf{u}$.

Because the problem in Equation (35) is not jointly convex in $\mathbf{e}$ and $\mathbf{u}$, we take an alternating minimization approach. The solution to the subproblem w.r.t. $\mathbf{e}$ is the same as Equation (34). Next, we discuss the solution to $\mathbf{u}$.

First, we denote $\hat{\mathbf{u}} = \mathbf{u}(\mathcal{V}_p)$ to be the projection of $\mathbf{u}$ on node set $\mathcal{V}_p$, and $\hat{n} = |\mathcal{V}_p|$. Let $\mathbf{w} \in \mathbb{R}_+^{\hat{n}\times 1}$ with $\mathbf{w}_i = (\mathbf{Be}-\mathbf{v})_i^2$ for $i \in \mathcal{V}_p$. Then we can write the subproblem w.r.t. $\hat{\mathbf{u}}$ as

$$\begin{aligned} &\min_{\hat{\mathbf{u}}\geq\mathbf{0}} \beta\hat{\mathbf{u}}^\top\mathbf{w} + \gamma\hat{\mathbf{u}}^\top\hat{\mathbf{u}} \\ &\quad \text{s.t. } \hat{\mathbf{u}}^\top\mathbf{1}_{\hat{n}} = \hat{n}, \end{aligned} \tag{36}$$

where $\mathbf{1}_{\hat{n}}$ is a length-$\hat{n}$ vector with all entries as 1.

Equation (36) is a quadratic optimization problem with respect to $\mathbf{u}$, whose Lagrangian function can be formulated as follows:

$$\mathcal{L}_u(\hat{\mathbf{u}}, \boldsymbol{\eta}, \theta) = \beta\hat{\mathbf{u}}^\top\mathbf{w} + \gamma\hat{\mathbf{u}}^\top\hat{\mathbf{u}} - \hat{\mathbf{u}}^\top\boldsymbol{\eta} - \theta(\hat{\mathbf{u}}^\top\mathbf{1}_{\hat{n}} - \hat{n}), \tag{37}$$

where $\boldsymbol{\eta} = [\eta_1, \eta_2, \ldots, \eta_{\hat{n}}]^\top \geq 0$ and $\theta \geq 0$ are the Lagrangian multipliers. The optimal $\hat{\mathbf{u}}^*$ should satisfy the following KKT conditions [3]:

(1) Stationary condition. $\nabla_{\hat{\mathbf{u}}^*}\mathcal{L}_u(\hat{\mathbf{u}}, \boldsymbol{\eta}, \theta) = \beta\mathbf{w} + 2\gamma\hat{\mathbf{u}}^* - \boldsymbol{\eta} - \theta\mathbf{1}_{\hat{n}} = \mathbf{0}_{\hat{n}}$
(2) Feasibility condition. $\hat{\mathbf{u}}^* \geq \mathbf{0}_{\hat{n}}$, $(\hat{\mathbf{u}}^*)^\top\mathbf{1}_{\hat{n}} - 1 = 0$
(3) Complementary slackness. $\boldsymbol{\eta}_i\hat{\mathbf{u}}_i^* = 0$, $1 \leq i \leq \hat{n}$
(4) Nonnegativity condition. $\boldsymbol{\eta} \geq \mathbf{0}_{\hat{n}}$

From the stationary condition, we can obtain $\hat{\mathbf{u}}_i$ as

$$\hat{\mathbf{u}}_i = \frac{\boldsymbol{\eta}_i + \theta - \mathbf{w}_i}{2\gamma},$$

where we can observe that $\hat{\mathbf{u}}_i$ depends on the specification of $\boldsymbol{\eta}_i$ and θ. Similarly to Reference [30], we divide the problem into three cases as follows:

(1) When $\theta - \mathbf{w}_i > 0$, since $\boldsymbol{\eta}_i \geq 0$, we have **hatu**$_i > 0$. From the complementary slackness, $\boldsymbol{\eta}_i\hat{\mathbf{u}}_i = 0$, we have $\boldsymbol{\eta}_i = 0$, and, therefore, $\hat{\mathbf{u}}_i = \frac{\theta-\mathbf{w}_i}{2\gamma}$.
(2) When $\theta - \mathbf{w}_i < 0$, since $\hat{\mathbf{u}}_i \geq 0$, we have $\boldsymbol{\eta}_i > 0$. Because $\boldsymbol{\eta}_i\hat{\mathbf{u}}_i = 0$, we have $\hat{\mathbf{u}}_i = 0$.
(3) When $\theta - \mathbf{w}_i = 0$, we have $\hat{\mathbf{u}}_i = \frac{\boldsymbol{\eta}_i}{2\gamma}$. Since $\boldsymbol{\eta}_i\hat{\mathbf{u}}_i = 0$, we have $\hat{\mathbf{u}}_i = 0$ and $\boldsymbol{\eta}_i = 0$.

Therefore, if we sort $\mathbf{w}_1 \leq \mathbf{w}_2 \leq \cdots \leq \mathbf{w}_{\hat{n}}$, then there exists $\tilde{\theta} > 0$ s.t. $\tilde{\theta} - \mathbf{w}_t > 0$ and $\tilde{\theta} - \mathbf{w}_t \leq 0$. Then $\hat{\mathbf{u}}_i$ can be solved as follows:

$$\hat{\mathbf{u}}_i = \begin{cases} \frac{\theta-\mathbf{w}_i}{2\gamma}, & \text{if } i \leq t \\ 0, & \text{otherwise,} \end{cases} \tag{38}$$

where θ can be solved by using $\sum_{i=1}^t \hat{\mathbf{u}}_i = \hat{n}$, that is,

$$\theta = \frac{2\gamma\hat{n} + \sum_{i=1}^t \mathbf{w}_i}{t}. \tag{39}$$

Equation (38) implies the intuition of the assignment of $\mathbf{u}_i$. That is, when $\mathbf{w}_i$ is large, $\mathbf{u}_i$ is small. Recall that $\mathbf{w}_i$ represents the inconsistency between propagation score $\mathbf{r}_i$ and node attribute $\mathbf{v}_i$, which may come from the noises in the prior knowledge. Therefore, Equation (38) assigns small weights to large inconsistencies to reduce the negative impact of noisy node attributes and get a consensus result and, hence, improve the robustness of our model.

ALGORITHM 2: W-RCA-SEMI

```
Input: Network $\mathcal{G}_l$ denoting the invariant network with $n$ nodes and is represented by an
       adjacency matrix $\mathbf{A}$, $c$ is the network propagation parameter, $\tau$ is the parameter to
       control the sparsity of $\mathbf{e}$, $\tilde{\mathbf{P}}$ is the normalized adjacency matrix of the broken
       network, $\mathbf{M}$ is the logical matrix of $\mathcal{G}_l$ (1 with edge, 0 without edge), $\mathbf{v}$ is the vector
       of node attributes, $\mathcal{V}_p$ is the set of nodes having valid node attributes, $\beta$ is a
       parameter to control semi-supervision, $\gamma$ is a parameter to control the complexity
       of the learned weights
Output: Ranking vector $\mathbf{e}$, weight vector $\mathbf{u}$
begin
    Initialize $\hat{\mathbf{u}}_i = 1, \forall i \in \mathcal{V}_p$;
    repeat
        Set $\mathbf{u}_i = \hat{\mathbf{u}}_i \ \forall i \in \mathcal{V}_p$; $\mathbf{u}_i = 0 \ \forall i \notin \mathcal{V}_p$;
        Inferring $\mathbf{e}$ by Equation (34);
        Compute $\mathbf{w}_i = ((\mathbf{Be})_i - \mathbf{v}_i)^2, \forall i \in \mathcal{V}_p$;
        Sort $\{\mathbf{w}_i\}_{1 \le i \le \hat{n}}$ in increasing order;
        $t \leftarrow \hat{n} + 1$;
        do
            $t \leftarrow t - 1$;
            $\theta \leftarrow \frac{2\gamma\hat{n} + \sum_{i=1}^{t} \mathbf{w}_i}{t}$;
        while $\theta - \mathbf{w}_t \le 0$ and $t > 1$;
        for $i \leftarrow 1$ to $t$ do
            $\hat{\mathbf{u}}_i \leftarrow \frac{\theta - \mathbf{w}_i}{2\gamma}$;
        end
        for $i \leftarrow t + 1$ to $\hat{n}$ do
            $\hat{\mathbf{u}}_i \leftarrow 0$;
        end
    until convergence;
end
```

In Equation (39), γ relates to the selectivity of the model. When γ is very large, $\hat{\mathbf{u}}_i$ becomes large, and all node attributes will be selected with nearly equal weights. When γ is very small, at least one node attribute (with the smallest $\mathbf{w}_i$) will be selected. Therefore, we can use γ to control how many node attributes will be integrated for causal anomaly ranking.

From Equation (38) and Equation (39), we can search the value of t from $\hat{n}$ to 1 decreasingly [30]. Once $\theta - \mathbf{w}_t > 0$, then we find the value of t. Then we can calculate $\hat{\mathbf{u}}_1, \ldots, \hat{\mathbf{u}}_{\hat{n}}$ according to Equation (38). The algorithm for solving $\mathbf{u}$ is involved in Algorithm 2. In Algorithm 2, $\mathbf{e}$ and $\mathbf{u}$ are optimized alternately. Since both optimization procedures decrease the value of the objective function in Equation (35) and the objective function value is lower bounded by 0, Algorithm 2 is guaranteed to converge to a local minima of the optimization problem in Equation (35). In the following, we refer to the semi-supervised ranking algorithm with weight learning as W-RCA-SEMI.

8. EMPIRICAL STUDY

In this section, we perform extensive experiments to evaluate the performance of the proposed methods (summarized in Table I). We use both simulated data and real-world monitoring datasets. For comparison, we select several state-of-the-art methods, including mRank and gRank in References [7, 16], and [26]. For all the methods, the tuning parameters were tuned using cross validation. We use several evaluation metrics, including precision, recall, and nDCG [15] to measure the performance. The

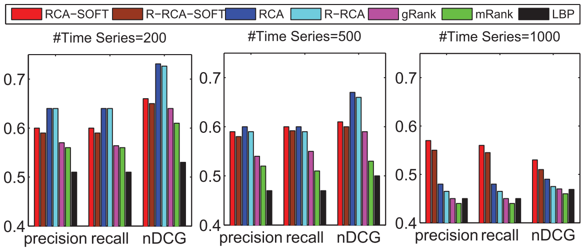


Fig. 3. Comparison on synthetic data (K, $p = 10$).

precision and recall are computed on the top-K ranking result, where K is typically chosen as twice the actual number of ground-truth causal anomalies [15, 26]. The nDCG of the top-p ranking result is defined as $nDCG_p = \frac{DCG_p}{IDCG_p}$, where $DCG_p = \sum_{i=1}^{p} \frac{2^{rel_i}-1}{\log_2(1+i)}$, $IDCG_p$ is the DCG_p value on the ground truth, and p is smaller than or equal to the actual number of ground-truth anomalies. The rel_i represents the anomaly score of the ith item in the ranking list of the ground truth.

8.1. Simulation Study

We first evaluate the performance of the proposed methods using simulations. We have followed References [7, 26] in generating the simulation data.

8.1.1. Data Generation. We first generate 5,000 synthetic time-series data to simulate the monitoring records.[2] Each time series contains 1,050 time points. Based on the invariant model introduced in Section 2.1, we build the invariant network by using the first 1,000 time points in the time series. This generates an invariant network containing 1,551 nodes and 157,371 edges. To generate invariant network of different sizes, we randomly sample 200, 500, and 1,000 nodes from the whole invariant network and evaluate the algorithms on these sub-networks.

To generate the root cause anomaly, we randomly select 10 nodes from the network and assign each of them an anomaly score between 1 and 10. The ranking of these scores is used as the ground truth. To simulate the anomaly prorogation, we further use these scores as the vector $\mathbf{e}$ in Equation (6) and calculate $\mathbf{r}$ ($c = 0.9$). The values of the top-30 time series with largest values in $\mathbf{r}$ are then modified by changing their amplitude value with the ratio $1 + \mathbf{r}_i$. That is, if the observed values of one time series is y_1, after changing it from y_1 to y_2, the manually injected degree of anomaly $\frac{|y_2-y_1|}{|y_1|}$ is equal to $1 + \mathbf{r}_i$. We denote this anomaly generation scheme as *amplitude-based* anomaly generation.

8.1.2. Performance Evaluation. Using the simulated data, we compare the performance of different algorithms. In this example, we only consider the training time series as one snapshot; multiple snapshot cases involving temporal smoothing will be examined in the real datasets. Due to the page limit, we report the precision, recall, and nDCG for only the top-10 items considering that the ground truth contains 10 anomalies. Similar results can be observed with other settings of K and p. For each algorithm, the reported result is averaged over 100 randomly selected subsets of the training data.

From Figure 3, we have several key observations. First, the proposed algorithms significantly outperform other competing methods, which demonstrates the advantage of taking into account fault prorogation in ranking casual anomalies. We also notice that performance of all ranking algorithms will decline on larger invariant networks

[2] http://cs.unc.edu/%7Eweicheng/synthetics5000.csv.

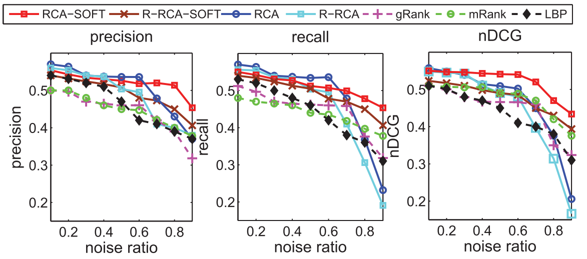


Fig. 4. Performance with different noise ratio (K, $p = 10$).

Table II. Examples of Categories and Monitors

Categories	**Samples of Measurements**
CPU	utilization, user usage time, IO wait time
DISK	# of write operations, write time, weighted IO time
MEM	run queue, collision rate, UsageRate
NET	error rate, packet rate
SYS	UTIL, MODE UTIL

with more nodes, indicating that anomaly ranking becomes more challenging on networks with more complex behaviour. However, the ranking result with *softmax* is less sensitive to the size of the invariant network, suggesting that the *softmax* normalization can effectively improve the robustness of the algorithm. This is quite beneficial in real-life applications, especially when data are noisy. Finally, we observe that RCA and RCA-SOFT outperform R-RCA and R-RCA-SOFT, respectively. This implies that the relaxed versions of the algorithms are less accurate. Nevertheless, their accuracies are still very comparable to those of the RCA and RCA-SOFT methods. In addition, the efficiency of the relaxed algorithms is greatly improved, as discussed in Section 4 and Section 8.5.

8.1.3. Robustness Evaluation. Practical invariant network and broken edges can be quite noisy. In this section, we further examine the performance of the proposed algorithms w.r.t. different noise levels. To do this, we randomly perturb a portion of non-broken edges in the invariant network. Results are shown in Figure 4. We observe that even when the noise ratio approaches 50%, the precision, recall, and nDCG of the proposed approaches still attain 0.5. This indicates the robustness of the proposed algorithms. We also observe that when the noise ratio is very large, RCA-SOFT and R-RCA-SOFT work better than RCA and R-RCA, respectively. This is similar to those observations made in Section 8.1.2. As has been discussed in Section 5, the *softmax* normalization can greatly suppress the impact of extreme values and outliers in $\mathbf{r}$, thus improving the robustness.

8.2. Ranking Causal Anomalies on Bank Information System Data

In this section, we apply the proposed methods to detect causal abnormal components on a Bank Information System (BIS) dataset [7, 26]. The monitoring data are collected from real-world bank information system logs, which contain 11 categories. Each category has a varying number of time series, and Table II gives five categories as examples. The dataset contains the flow intensities collected every 6s. In total, we have 1,273 flow intensity time series. The training data are collected at normal system states, where each time series has 168 time points. The invariant network is then generated on the training data as described in Section 2.1. The testing data of the 1,273 flow intensity time series are collected during abnormal system states, where each time series

Table III. Data Set Description

Data Set	#Monitors	#invariant links	#broken edges at given time point
BIS	1273	39116	18052
Coal Plant	1625	9451	56

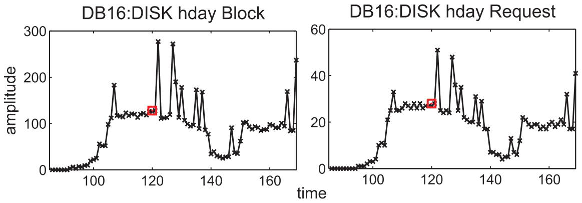


Fig. 5. Two example monitoring data of BIS.

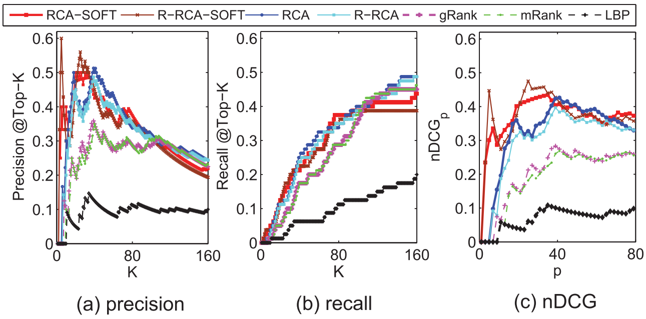


Fig. 6. Comparison on BIS data.

contain 169 time points. We track the changes of the invariant network with the testing data using the method described in Section 2.1. Once we obtain the broken networks at different time points, we will then perform causal anomaly ranking in these temporal slots jointly. Properties of the networks constructed are summarized in Table III.

Based on the knowledge from system experts, the root cause anomaly at $t = 120$ in the testing data is related to "DB16." An illustration of two "DB16"-related monitoring data are shown in Figure 5. We highlight $t = 120$ with a red square. Obviously, their behaviour looks anomalous from that time point onward. Due to the complex dependency among different monitoring time series (measurements), it is impractical to obtain a full ranking of abnormal measurement. Fortunately, we have a unique semantic label associated with each measurement. For example, some semantic labels read "DB16:DISK hdx Request" and "WEB26 PAGEOUT RATE." Thus, we can extract all measurements whose titles have the prefix "DB16" as the ground-truth anomalies. The ranking score is determined by the number of broken edges associated with each measurement. Here our goal is to demonstrate how the top-ranked measurements selected by our method are related to the "DB16" root cause. Altogether, there are 80 measurements related to "DB16," so we report the precision of recall with K ranging from 1 to 160 and the nDCG with p ranging from 1 to 80.

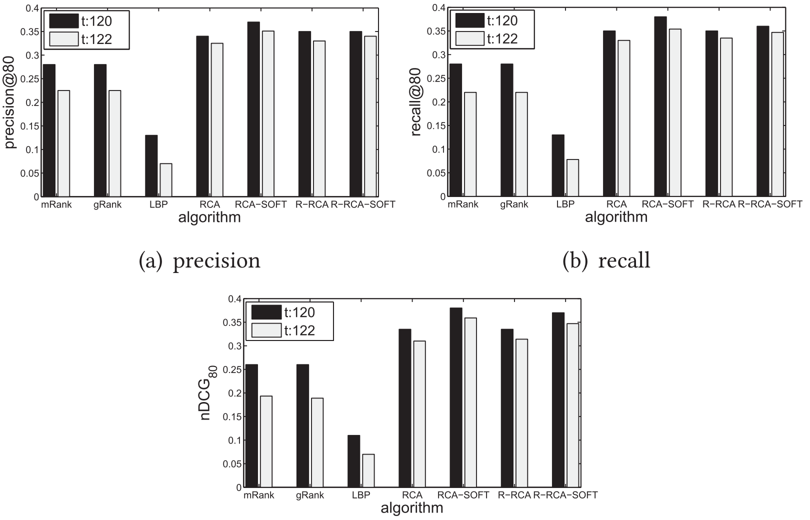
The results are shown in Figure 6. The relative performance of different approaches is consistent with the observations in the simulation study. Again, the proposed algorithms outperform baseline methods by a large margin. To examine the top-ranked items more clearly, we list the top-12 results of different approaches in Table IV and report the number of "DB16"-related monitors in Table V. From Table IV, we observe that the three baseline methods only report one "DB16"-related measurement in the top-12

Table IV. Top 12 Anomalies Detected by Different Methods on BIS Data(*t*:120)

mRank	**gRank**	**LBP**	**RCA**	**RCA-SOFT**	**R-RCA**	**R-RCA-SOFT**
WEB16:NET eth1 BYNETIF	HUB18:MEM UsageRate	WEB22:SYS MODE UTIL	HUB17:DISK hda Request	DB17:DISK hdm Block	HUB17:DISK hda Request	DB17:DISK hdm Block
HUB17:DISK hda Request	HUB17:DISK hda Request	DB15:DISK hdaz Block	DB17:DISK hday Block	DB17:DISK hdba Block	DB15:PACKET Output	DB17:DISK hdba Block
AP12:DISK hd45 Block	AP12:DISK hd45 Block	WEB12:NET eth1 BYNETIF	HUB17:DISK hda Busy	DB16:DISK hdm Block	HUB17:DISK hda Busy	DB16:DISK hdm Block
AP12:DISK hd1 Block	AP12:DISK hd1 Block	WEB17:DISK BYDSK	DB18:DISK hdba Block	DB18:DISK hdm Block	DB17:DISK hdm Block	DB16:DISK hdj Request
WEB19:DISK BYDSK	AP11:DISK hd45 Block	DB18:DISK hdt Busy	DB18:DISK hdm Block	DB16:DISK hdj Request	DB17:DISK hdba Block	DB16:DISK hdax Request
AP11:DISK hd45 Block	AP11:DISK hd1 Block	DB15:DISK hdl Request	DB16:DISK hdm Block	DB18:DISK hdba Block	DB18:DISK hdm Block	DB18:DISK hdag Request
AP11:DISK hd1 Block	DB17:DISK hday Block	WEB21:DISK BYDSK	DB17:DISK hdba Block	DB16:DISK hdax Request	DB16:DISK hdm Block	DB18:DISK hdm Block
DB16:DISK hdm Block	DB15:PACKET Input	WEB27:FREE UTIL	DB17:DISK hdm Block	DB18:DISK hdag Request	DB18:DISK hdba Block	DB18:DISK hdbu Request
DB17:DISK hdm Block	DB17:DISK hdm Block	WEB19:NET eth0	DB16:DISK hdba Block	DB18:DISK hdbu Request	DB17:DISK hday Block	DB18:DISK hdx Request
DB18:DISK hdm Block	DB16:DISK hdm Block	WEB25:PAGEOUT RATE	DB16:DISK hdj Request	DB16:DISK hdba Block	DB16:DISK hdba Block	DB18:DISK hdax Request
DB17:DISK hdba Block	DB17:DISK hdba Block	DB16:DISK hdy Block	DB18:DISK hdag Request	DB18:DISK hdx Request	DB16:DISK hdj Request	DB18:DISK hdba Block
DB18:DISK hdba Block	DB18:DISK hdm Block	AP13:DISK hd30 Block	DB16:DISK hdax Request	DB18:DISK hdax Request	DB18:DISK hdag Request	DB16:DISK hdx Request

Table V. Number of “DB16” Related Monitors in Top 32 Results on BIS Data(*t*:120)

mRank	**gRank**	**LBP**	**RCA**	**RCA-SOFT**	**R-RCA**	**R-RCA-SOFT**
10	7	4	14	16	13	17



(a) precision

(b) recall

(c) nDCG

Fig. 7. Performance at t:120 vs. t:122 on BIS data (p, $K = 80$).

results, and the actual rank of the “DB16”-related measurement appear lower (worse) than that of the proposed methods. We also notice that the ranking algorithms with *softmax* normalization outperform others. From Tables IV and V, we can see that top-ranked items reported by RCA-SOFT and R-RCA-SOFT are more relevant than those reported by RCA and R-RCA, respectively. This clearly illustrates the effectiveness of the *softmax* normalization in reducing the influence of extreme values or outliers in the data.

As discussed in Section 1, the root anomalies could further propagate from one component to related ones over time, which may or may not necessarily relate to “DB16.” Such anomaly propagation makes anomaly detection even harder. To study how the performance varies at different time points, we compare the performance at $t = 120$ and $t = 122$, respectively in Figure 7 (p, $K = 80$). Clearly, the performance declines for all methods. However, the proposed methods are less sensitive to anomaly propagation than others, suggesting that our approaches can better handle the fault propagation problem. We believe this is attributed to the network diffusion model that explicitly captures the fault propagation processes. We also list the top-12 abnormal at $t = 122$ in Table VI. Due to the page limit, we only show the results of mRank, gRank, RCA-SOFT, and R-RCA-SOFT. By comparing the results in Tables IV and VI, we can observe that RCA-SOFT and R-RCA-SOFT significantly outperform mRank and gRank, where the latter two methods, based on the percentage of broken edges, are more sensitive to anomaly propogation.

We further validate the effectiveness of proposed methods with temporal smoothing. We report the top-12 results of different methods with smoothing at two successive time points $t = 120$ and $t = 121$ in Table VII. The number of “DB16”-related monitors in the top-12 results is summarized in Table VIII. From Tables VII and VIII, we observe a significant performance improvement of our methods with temporal broken networks

Table VI. Top 12 Anomalies on BIS Data (*t*:122)

mRank	gRank	RCA-SOFT	R-RCA-SOFT
WEB21:NET eth1 BYNETIF	WEB21:NET eth0 BYNETIF	DB17:DISK hdm Block	DB17:DISK hdm Block
WEB21:NET eth0 BYNETIF	WEB21:NET eth1 BYNETIF	DB17:DISK hdba Block	DB17:DISK hdba Block
WEB21:FREE UTIL	HUB18:MEM UsageRate	DB16:DISK hdm Block	DB16:DISK hdm Block
AP12:DISK hd45 Block	WEB21:FREE UTIL	DB18:DISK hdm Block	DB16:DISK hdj Request
AP12:DISK hd1 Block	WEB26:PAGEOUT RATE	DB16:DISK hdj Request	DB16:DISK hdax Request
DB18:DISK hday Block	AP12:DISK hd45 Block	DB18:DISK hdba Block	DB18:DISK hdm Block
DB18:DISK hdk Block	AP12:DISK hd1 Block	DB16:DISK hdax Request	DB18:DISK hdx Request
DB18:DISK hday Request	DB18:DISK hday Block	DB16:DISK hdba Block	DB18:DISK hdba Block
DB18:DISK hdk Request	DB18:DISK hdk Block	DB18:DISK hdx Request	DB16:DISK hdba Block
WEB26:PAGEOUT RATE	DB18:DISK hday Request	DB18:DISK hdbl Request	DB18:DISK hdax Request
DB17:DISK hdm Block	DB18:DISK hdk Request	DB16:DISK hdx Busy	DB16:PACKET Inputx
DB16:DISK hdm Block	AP11:DISK hd45 Block	DB16:DISK hdx Request	DB18:DISK hdbl Request

Table VII. Top-12 Anomalies Reported by Methods with Temporal Smoothing on BIS Data (*t*:120-121)

T-RCA	**T-RCA-SOFT**	**T-R-RCA**	**T-R-RCA-SOFT**
WEB14:NET eth0 BYNETIF	DB17:DISK hdm Block	WEB14:NET eth0 BYNETIF	DB17:DISK hdm Block
WEB16:DISK BYDSK	DB17:DISK hdba Block	WEB21:NET eth0 BYNETIF	DB17:DISK hdba Block
DB18:DISK hdba Block	DB16:DISK hdm Block	WEB16:DISK BYDSK PHYS	DB16:DISK hdm Block
DB18:DISK hdm Block	DB18:DISK hdm Block	WEB21:FREE UTIL	DB18:DISK hdm Block
DB17:DISK hdba Block	DB16:DISK hdj Request	DB15:PACKET Output	DB16:DISK hdj Request
DB16:DISK hdm Block	DB18:DISK hdba Block	DB16:DISK hdj Request	DB18:DISK hdba Block
DB17:DISK hdm Block	DB16:DISK hdax Request	DB17:DISK hdm Block	DB16:DISK hdax Request
DB16:DISK hdba Block	DB16:DISK hdba Block	DB16:DISK hdba Block	DB18:DISK hdx Request
DB16:DISK hdj Request	DB18:DISK hdx Request	DB17:DISK hday Block	DB16:DISK hdba Block
DB16:DISK hdax Request	DB18:DISK hdbl Request	DB16:DISK hdm Block	DB18:DISK hdbl Request
DB16:DISK hdx Busy	DB16:DISK hdx Busy	DB16:DISK hdax Request	DB16:DISK hdx Request
DB16:DISK hdbl Busy	DB16:DISK hdx Request	DB18:DISK hdba Block	DB16:DISK hdx Busy

Table VIII. Comparison on the Number of "DB16"-Related Anomalies in Top-12 Results on BIS Data

	RCA	**RCA-SOFT**	**R-RCA**	**R-RCA-SOFT**
Without temporal smoothing	4	4	3	4
With temporal smoothing	6	6	4	6

smoothing compared with those without smoothing. As discussed in Section 6, since causal anomalies of a system usually do not change within a short period of time, utilizing such smoothness can effectively suppress noise and thus give better ranking accuracy.

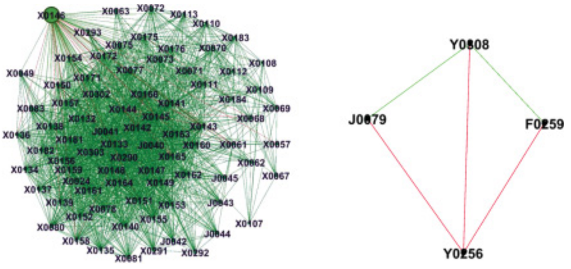
8.3. Fault Diagnosis on Coal Plant Data

In this section, we test the proposed methods in the application of fault diagnosis on a coal plant cyber-physical system data. The dataset contains time-series data collected through 1,625 electric sensors installed on different components of the coal plant system. Using the invariant model described in Section 2.1, we generate the invariant network that contains 9,451 invariant links. For privacy reasons, we remove sensitive descriptions of the data.

Based on knowledge from domain experts, in the abnormal stage the root cause is associated with component "X0146." We report the top-12 results of different ranking

Table IX. Top Anomalies on Coal Plant Data

mRank	gRank	LBP	RCA	RCA-SOFT	R-RCA	R-RCA-SOFT
Y0039	Y0256	Y0256	X0146	X0146	X0146	X0146
X0128	Y0045	X0146	Y0045	Y0256	X0128	X0166
Y0256	Y0028	F0454	X0128	F0454	F0454	X0144
H0021	X0146	X0128	Y0030	J0079	Y0256	X0165
X0146	X0057	Y0039	X0057	Y0308	Y0039	X0142
X0149	X0061	X0166	X0158	X0166	Y0246	J0079
H0022	X0068	X0144	X0068	X0144	Y0045	X0164
F0454	X0143	X0149	X0061	X0128	Y0028	X0145
H0020	X0158	J0085	X0139	X0165	X0056	X0143
X0184	X0164	X0061	X0143	X0142	J0079	X0163
X0166	J0164	Y0030	H0021	H0022	X0149	J0164
J0164	H0021	J0079	F0454	X0143	X0145	X0149



(a) Egonet of node “X0146” (b) Egonet of node “Y0256”

Fig. 8. Egonet of node “X0146” and “Y0256” in invariant network and vanishing correlations (red edges) on coal plant data.

algorithms in Table IX. We observe that the proposed algorithms all rank component “X0146” the highest, while the baseline methods could give higher ranks to other components. In Figure 8(a), we visualize the egonet of the node “X0146” in the invariant network, which is defined as the one-step neighborhood around node “X0146,” including the node itself, direct neighbors, and all connections among these nodes in the invariant network. Here, green lines denote the invariant link, and red lines denote vanishing correlations (broken links). Since the node “Y0256” is top ranked by the baseline methods, we also visualize its egonet in Figure 8(b) for a comparison. There are 80 links related to “X0146” in the invariant network, and 14 of them are broken. Namely the percentage of broken edges is only 17.5% for a truly anomalous component. In contrast, the percentage of broken edges for the node “Y0256” is 100%, namely a false-positive node can have a very high percentage of broken edges in practice. This explains why baseline approaches using the percentage of broken edges could fail, because the percentage of broken edges does not serve as a reliable evidence of the degree of causal anomalies. In comparison, our approach takes into account the global structures of the invariant network via network propagation, and thus the resultant ranking is more meaningful.

8.4. Evaluation of Leveraging Prior Knowledge

In this section, we evaluate the effectiveness of the semi-supervised algorithms proposed in Section 7 using the BIS dataset. We simulate node attributes by the following strategy. First, we set “DB16”-related components as seeds (recall these components

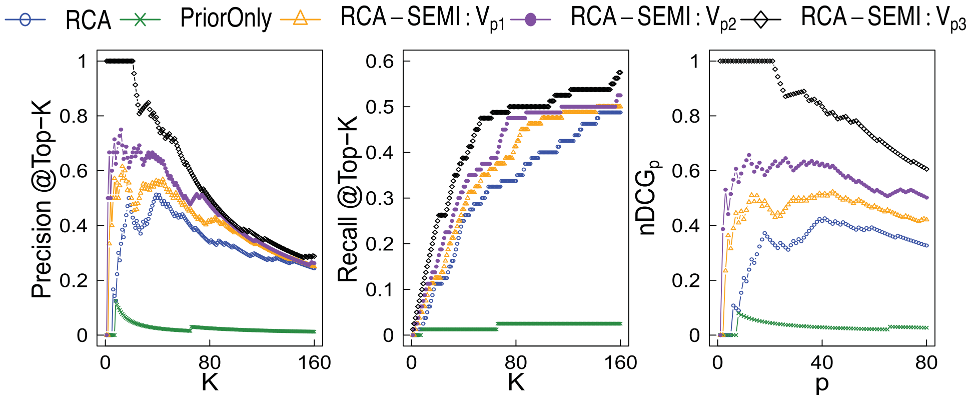
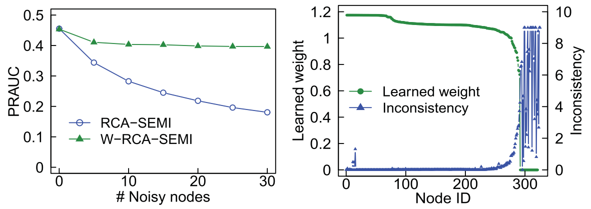


Fig. 9. Comparison on BIS data using prior knowledge. RCA-SEMI:V$_{p1}$, RCA-SEMI:V$_{p2}$, and RCA-SEMI:V$_{p3}$ refer to running RCA-SEMI with $\mathcal{V}_{p1}$, $\mathcal{V}_{p2}$, and $\mathcal{V}_{p3}$, respectively.



(a) Robustness comparison

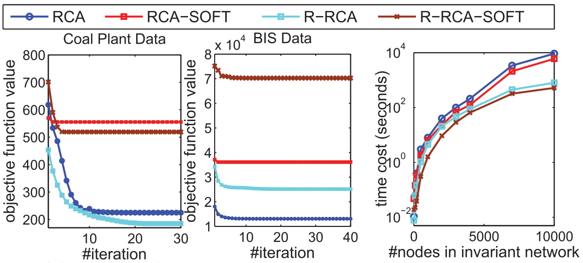
(b) Learned weights

Fig. 10. Comparison on BIS data with noisy prior knowledge.

are ground-truth anomalies) and run a label propagation algorithm to obtain a score for each node. Then, we set the scores of “DB16”-related nodes to zero and treat the remaining non-zero scores as the attributes of other nodes. Finally, we randomly divide the remaining attributed nodes $\mathcal{V}_p$ into three equal parts $\mathcal{V}_1$, $\mathcal{V}_2$, and $\mathcal{V}_3$ and then form $\mathcal{V}_{p1} = \mathcal{V}_1$, $\mathcal{V}_{p2} = \{\mathcal{V}_1, \mathcal{V}_2\}$ and $\mathcal{V}_{p3} = \{\mathcal{V}_1, \mathcal{V}_2, \mathcal{V}_3\}$. Algorithm RCA-SEMI is run with $\mathcal{V}_{p1}$, $\mathcal{V}_{p2}$, and $\mathcal{V}_{p3}$, respectively, to evaluate its capability to uncover “DB16”-related components with the guidance of these different partial prior knowledge.

Figure 9 shows the results of RCA-SEMI. For clarity, we only show RCA as a baseline. We also consider another degraded version of RCA-SEMI, which is shown as “PriorOnly.” This method solves $\mathbf{e}$ by minimizing $(\mathbf{Be} - \mathbf{v})^\top \mathbf{D}_u(\mathbf{Be} - \mathbf{v}) + \tau\|\mathbf{e}\|_1$, which only uses node attributes without considering label propagation. From Figure 9, we observe that RCA-SEMI can effectively incorporate node attributes to improve causal anomaly inference accuracy. More prior knowledge typically results in better accuracy. The poor performance of “PriorOnly” also indicates that using partial prior knowledge alone is not effective. This demonstrates the importance of taking into account the fault propagation when incorporating partial node attributes.

Next, we evaluate the robustness of Algorithm 2, W-RCA-SEMI. For this, we manually inject noise into node attributes. Specifically, we randomly pick a certain number of nodes with non-zero attributes and change their attributes to a large value (e.g., 3). By varying the number of noisy nodes, we can evaluate the impact of noise on RCA-SEMI and W-RCA-SEMI. Figure 10(a) shows the area under the precision-recall curve (PRAUC) w.r.t. varying number of noisy nodes. Higher PRAUC indicates better accuracy. From Figure 10(a), we observe that the performance of RCA-SEMI is largely impacted by the injected noisy attributes, while W-RCA-SEMI performs stably. By



(a) Number of iterations to converge (b) Time costs comparison

Fig. 11. Number of iterations to converge and time cost comparison.

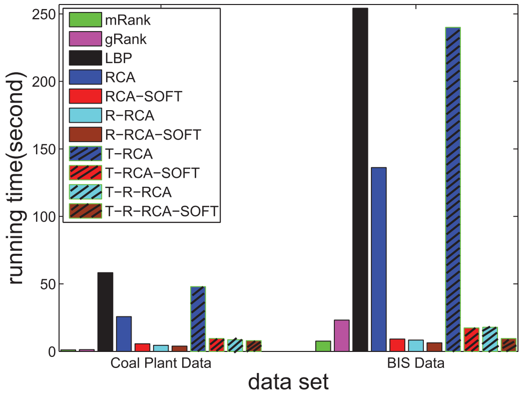


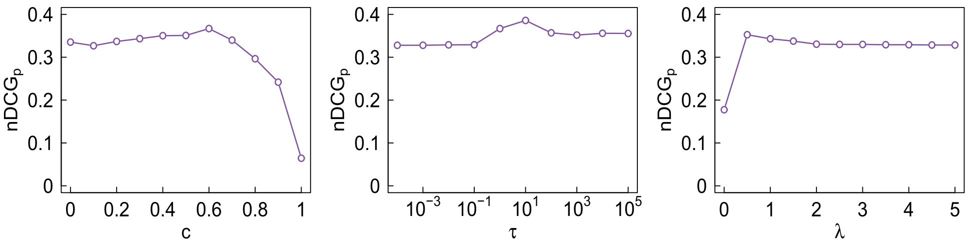
Fig. 12. Running time on real datasets.

investigating the learned weights in $\mathbf{u}$, we get the insights of W-RCA-SEMI. Figure 10(b) presents the learned weights $\mathbf{u}_i$ vs. the inconsistency of $(\mathbf{e}_i - \mathbf{v}_i)^2$ for nodes having valid $\mathbf{v}_i$'s, where the nodes are ordered by descending order of $\mathbf{u}_i$. As can be seen, W-RCA-SEMI effectively assigns small weights to large inconsistencies. Thus it can reduce the negative impacts of noisy attributes and obtain robust performance as shown in Figure 10(a).

8.5. Time Performance Evaluation

In this section, we study the efficiency of proposed methods using the following metrics: (1) the number of iterations for convergence, (2) the running time (in seconds), and (3) the scalability of the proposed algorithms. Figure 11(a) shows the value of the objective function with respect to the number of iterations on different datasets. We can observe that the objective value decreases steadily with the number of iterations. Typically less than 100 iterations are needed for convergence. We also observe that our method with *softmax* normalization takes fewer iterations to converge. This is because the normalization is able to reduce the influence of extreme values [25]. We also report the running time of each algorithm on the two real datasets in Figure 12. We can see that the proposed methods can detect causal anomalies very efficiently, even with the temporal smoothing module.

To evaluate the computational scalability, we randomly generate invariant networks with different number of nodes (with network density=10) and examine the computational cost. Here 10% of the edges is randomly selected as broken links. Using simulated data, we compare the running time of RCA, R-RCA, RCA-SOFT, and R-RCA-SOFT. Figure 11(b) plots the running time of different algorithms w.r.t. the number of nodes



(a) Varying c of RCA (b) Varying τ of RCA (c) Varying λ of R-RCA

Fig. 13. Parameter study results. The shown nDCG values are obtained by varying one parameter while fixing others.

in the invariant network. We can see that the relaxed versions of our algorithm are computationally more efficient than the original RCA and RCA-SOFT. These results are consistent with the complexity analysis in Section 4.

8.6. Parameter Study

There are three major parameters, c, τ, and λ, in the proposed RCA family algorithms. c is the tradeoff parameter controlling the propagation strength (see Section 3.1). τ is a parameter controlling the sparsity of the learned vector $\mathbf{e}$ in Equation (8). λ is used for balancing the propagation and broken network reconstruction in the relaxed RCA model in Equation (23). Next, we use the BIS dataset to study the impact of each parameter on the causal anomaly ranking accuracy.

Figure 13 shows the anomaly inference accuracy by varying each parameter in turn while fixing others. The accuracy is measured using nDCG$_\mathrm{p}$ with p equal to the number of ground-truth anomalies. Using other metrics will give similar trends, and thus they are omitted for brevity. From the figure, we observe that RCA and R-RCA perform stably in a relatively wide range of each parameter, which demonstrates the robustness of the proposed models. Specifically, the best c lies around 0.6, indicating the importance to consider sufficient fault propagations. Note that when $c = 0$ or $c = 1$, there will be no propagation or no learning of $\mathbf{e}$, respectively (see Equation (6)). For τ, its best value is around 1 and 10, which suggests a sparse vector $\mathbf{e}$, because usually there is only a small number of causal anomalies. Finally, the sharp accuracy increase by changing λ from 0 to non-zero values indicates the effectiveness of the relaxed RCA model in Equation (23). The best λ lies between 0.5 and 2, suggesting the relatively equal importances of fault propagation and broken network reconstruction in Equation (23).

9. RELATED WORK

In this section, we review related work on anomaly detection and system diagnosis, in particular along the following two categories: (1) fault detection in distributed systems and (2) graph-based methods.

For the first category, Yemini et al. [29] proposed to model event correlation and locate system faults using known dependency relationships between faults and symptoms. In real applications, however, it is usually hard to obtain such relationships precisely. To alleviate this limitation, Jiang et al. [16] developed several model-based approaches to detect the faults in complex distributed systems. They further proposed several Jaccard Coefficient-based approaches to locate the faulty components [17, 18]. These approaches generally focus on locating the faulty components, and they are not capable of spotting or ranking the causal anomalies.

Recently, graph-based methods have drawn a lot of interest in system anomaly detections [2, 5], either in static graphs or dynamic graphs [2]. In static graphs, the main task is to spot anomalous network entities (e.g., nodes, edges, subgraphs) given the graph structure [4, 10]. For example, Akoglu et al. [1] proposed the OddBall algorithm to detect anomalous nodes in weighted graphs. Liu et al. [22] proposed to use frequent subgraph mining to detect non-crashing bugs in software flow graphs. However, these approaches only focus on a single graph; in comparison, we take into account both the invariant graph and the broken correlations, which provide a more dynamic and complete picture for anomaly ranking. On dynamic graphs, anomaly detection aims at detecting abnormal events [23]. Most approaches along this direction are designed to detect anomaly timestamps in which suspicious events take place but not to perform ranking on a large number of system components. Sun et al. proposed to use temporal graphs for anomaly detection [24]. In their approach, a set of initial suspects need to be provided; then internal relationships among these initial suspects are characterized for better understanding of the root cause of these anomalies.

In using the invariant graph and the broken invariance graph for anomaly detection, Jiang et al. [17] used the ratio of broken edges in the invariant network as the anomaly score for ranking; Ge et al. [7] proposed mRank and gRank to rank causal anomalies; and Tao et al. [26] used the loopy belief propagation method to rank anomalies. As has been discussed, these algorithms rely heavily on the percentage of broken edges in egonet of a node. Such local approaches take into account neither the global network structures nor the global fault propagation spreading on the network. Therefore, the resultant rankings can be sub-optimal.

There is a number of correlation network-based system anomaly localization methods [9, 13, 14] that treat the correlation changes between system components as basic evidence of fault occurrence. Similarly to the invariant graph-based methods, these methods use the correlation changes in the egonet of each node at different time points to locate anomalous nodes. Basically, if there are more correlation changes happening in the egonet of a node, then it is more suspicious to be an anomaly. However, none of these approaches consider fault propagations. Therefore, they cannot exploit the whole structure of a network and are inferior in locating causal anomalies. Some other methods can track the eigenvectors of temporal correlation networks to detect the anomalous changes of a whole system [11, 12], but they do not rank nodes for locating causal anomalies and differ from our work in problem settings.

10. CONCLUSIONS

Detecting causal anomalies on monitoring data of distributed systems is an important problem in data-mining research. Robust and scalable approaches that can model the potential fault propagation are highly desirable. We develop a network diffusion-based framework, which simultaneously takes into account fault propagation on the network as well as reconstructing anomaly signatures using propagated anomalies. Our approach can locate causal anomalies more accurately than existing approaches; at the same time, it is robust to noise and computationally efficient. Moreover, when prior knowledge on anomalous status of nodes are available, our approach can effectively incorporate them to further enhance anomaly detection accuracy. When the prior knowledge is noisy, our approach can also automatically identify reliable information and reduce the negative impact of noise. Using both synthetic and real-life datasets, we show that the proposed methods outperform other competitors by a large margin.

REFERENCES

[1] Leman Akoglu, Mary McGlohon, and Christos Faloutsos. 2010. Oddball: Spotting anomalies in weighted graphs. In *PAKDD*. Springer, 410–421.

[2] Leman Akoglu, Hanghang Tong, and Danai Koutra. 2015. Graph based anomaly detection and description: A survey. *Data Min. Knowl. Discov.* 29, 3 (2015), 626–688.

[3] Stephen Boyd and Lieven Vandenberghe. 2004. *Convex Optimization*. Cambridge University Press.

[4] Markus M. Breunig, Hans-Peter Kriegel, Raymond T. Ng, and Jörg Sander. 2000. LOF: Identifying density-based local outliers. In *SIGMOD*. ACM, 93–104.

[5] Varun Chandola, Arindam Banerjee, and Vipin Kumar. 2009. Anomaly detection: A survey. *ACM Comput. Surv.* 41, 3 (2009), 15.

[6] Haifeng Chen, Haibin Cheng, Guofei Jiang, and Kenji Yoshihira. 2008. Exploiting local and global invariants for the management of large scale information systems. In *ICDM*. IEEE, 113–122.

[7] Yong Ge, Guofei Jiang, Min Ding, and Hui Xiong. 2014. Ranking metric anomaly in invariant networks. *ACM Trans. Knowl. Discov. Data* 8, 2 (2014), 8.

[8] Janos Gertler. 1998. *Fault Detection and Diagnosis in Engineering Systems*. CRC Press.

[9] Satoshi Hara, Tetsuro Morimura, Toshihiro Takahashi, Hiroki Yanagisawa, and Taiji Suzuki. 2015. A consistent method for graph based anomaly localization. In *AISTATS*. 333–341.

[10] Keith Henderson, Tina Eliassi-Rad, Christos Faloutsos, Leman Akoglu, Lei Li, Koji Maruhashi, B. Aditya Prakash, and Hanghang Tong. 2010. Metric forensics: A multi-level approach for mining volatile graphs. In *KDD*. ACM, 163–172.

[11] Shunsuke Hirose, Kenji Yamanishi, Takayuki Nakata, and Ryohei Fujimaki. 2009. Network anomaly detection based on eigen equation compression. In *KDD*. ACM, 1185–1194.

[12] Tsuyoshi Idé and Hisashi Kashima. 2004. Eigenspace-based anomaly detection in computer systems. In *KDD*. ACM, 440–449.

[13] Tsuyoshi Idé, Aurelie C. Lozano, Naoki Abe, and Yan Liu. 2009. Proximity-based anomaly detection using sparse structure learning. In *SDM*. SIAM, 97–108.

[14] Tsuyoshi Idé, Spiros Papadimitriou, and Michail Vlachos. 2007. Computing correlation anomaly scores using stochastic nearest neighbors. In *ICDM*. IEEE, 523–528.

[15] Kalervo Järvelin and Jaana Kekäläinen. 2002. Cumulated gain-based evaluation of IR techniques. *ACM Trans. Inf. Syst.* 20, 4 (2002), 422–446.

[16] Guofei Jiang, Haifeng Chen, and Kenji Yoshihira. 2006a. Discovering likely invariants of distributed transaction systems for autonomic system management. In *ICAC*. IEEE, 199–208.

[17] Guofei Jiang, Haifeng Chen, and Kenji Yoshihira. 2006b. Modeling and tracking of transaction flow dynamics for fault detection in complex systems. *IEEE Trans. Dependable Secure Comput.* 3, 4 (2006), 312–326.

[18] Guofei Jiang, Haifeng Chen, and Kenji Yoshihira. 2007. Efficient and scalable algorithms for inferring likely invariants in distributed systems. *IEEE Trans. Knowl. Data Eng.* 19, 11 (2007), 1508–1523.

[19] Tae Hoon Kim, Kyoung Mu Lee, and Sang Uk Lee. 2008. Generative image segmentation using random walks with restart. In *ECCV*. Springer, 264–275.

[20] Daniel D. Lee and H. Sebastian Seung. 2001. Algorithms for non-negative matrix factorization. In *NIPS*. 556–562.

[21] Ljung Lennart. 1999. *System Identification: Theory for the User*. PTR Prentice Hall, Upper Saddle River, NJ, 1–14.

[22] Chao Liu, Xifeng Yan, Hwanjo Yu, Jiawei Han, and S. Yu Philip. 2005. Mining behavior graphs for "backtrace" of noncrashing bugs. In *SDM*. SIAM, 286–297.

[23] Ryan A. Rossi, Brian Gallagher, Jennifer Neville, and Keith Henderson. 2013. Modeling dynamic behavior in large evolving graphs. In *WSDM*. ACM, 667–676.

[24] Jimeng Sun, Dacheng Tao, and Christos Faloutsos. 2006. Beyond streams and graphs: Dynamic tensor analysis. In *KDD*. ACM, 374–383.

[25] Richard S. Sutton and Andrew G. Barto. 1998. *Reinforcement Learning: An Introduction*. MIT Press, Cambridge, MA.

[26] Changxia Tao, Yong Ge, Qinbao Song, Yuan Ge, and Olufemi A Omitaomu. 2014. Metric ranking of invariant networks with belief propagation. In *ICDM*. IEEE, 1001–1006.

[27] Robert Tibshirani. 1996. Regression shrinkage and selection via the lasso. *J. R. Stat. Soc. Ser. B (Methodological)* (1996), 267–288.

[28] Hanghang Tong, Christos Faloutsos, and Jia-yu Pan. 2006. Fast random walk with restart and its applications. In *ICDM*. IEEE, 613–622.

[29] Shaula Alexander Yemini, Shmuel Kliger, Eyal Mozes, Yechiam Yemini, and David Ohsie. 1996. High speed and robust event correlation. *IEEE Commun. Mag.* 34, 5 (1996), 82–90.

[30] Guoxian Yu, Huzefa Rangwala, Carlotta Domeniconi, Guoji Zhang, and Zili Zhang. 2013. Protein function prediction by integrating multiple kernels. In *IJCAI*. AAAI Press, 1869–1875.

[31] Dengyong Zhou, Olivier Bousquet, Thomas Navin Lal, Jason Weston, and Bernhard Schölkopf. 2004. Learning with local and global consistency. In *NIPS*. 321–328.

Received October 2016; accepted January 2017