Author Manuscript

Author Manuscript

Author Manuscript

Author Manuscript

HHS Public Access

Author manuscript

KDD. Author manuscript; available in PMC 2017 July 13.

Published in final edited form as:

KDD. 2016 August ; 2016: 805–814. doi:10.1145/2939672.2939765.

Ranking Causal Anomalies via Temporal and Dynamical Analysis on Vanishing Correlations

Wei Cheng[1], **Kai Zhang**[1], **Haifeng Chen**[1], **Guofei Jiang**[1], **Zhengzhang Chen**[1], and **Wei Wang**[2]

[1]NEC Laboratories America

[2]Department of Computer Science, University of California, Los Angeles

Abstract

Modern world has witnessed a dramatic increase in our ability to collect, transmit and distribute real-time monitoring and surveillance data from large-scale information systems and cyber-physical systems. Detecting system anomalies thus attracts significant amount of interest in many fields such as security, fault management, and industrial optimization. Recently, invariant network has shown to be a powerful way 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. To address these limitations, in this paper 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. Extensive experiments on synthetic datasets, bank information system datasets, and coal plant cyber-physical system datasets demonstrate the effectiveness of our approach.

Keywords

causal anomalies ranking; label propagation; nonnegative matrix factorization

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

CCS Concepts

•Security and privacy → Pseudonymity, anonymity and untraceability;

Author Manuscript Author Manuscript Author Manuscript Author Manuscript

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, application soft-wares, servers, to storage, networking devices, etc.

A central task in running these large scale distributed systems is to automatically monitor the system status, detect anomalies, and diagnose system fault, so as to guarantee stable and high-quality services or outputs. Significant research efforts have been devoted to this topic in the literatures. For instance, Gertler et al. [9] 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, 13]. 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, the system audit events and the network traffic statistics, and sometimes sensory measurements in physical systems.

The invariant model is a successful example [13, 14] for large-scale system management. It focuses on discovering stable, significant dependencies between pairs of system components that are monitored through time series recordings, so as to profile the system status and perform subsequent reasoning. A strong dependency between a pair of components is called *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, how to locate true, causal anomalies can still be a challenging task due to the following reasons. First, system faults are seldom isolated. Instead, starting from the root location/component, anomalous behavior will propagate to neighboring components [13], and different types of system faults can trigger diverse propagation patterns. Second, monitoring data often contains a lot of noises due to the fluctuation of complex operation environments.

Author Manuscript Author Manuscript Author Manuscript Author Manuscript

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 the egonet based method proposed by Ge et al. [8], and the loopy belief propagation (LBP) based method proposed by Tao et al. [22]. 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, neither 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 [8] 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 [22] 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 a good evidence of the causal anomaly. This is because, although one broken edge can indicate that one (or both) of 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 [13]. Therefore the percentage of broken edges could give false evidences. 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 there exists a clear evidence of fault propagation on node ⓘ, an ideal algorithm should rank ⓘ higher than ⓗ. Third, existing methods usually consider static broken network instead of multiple broken networks at successive time points together. While we believe that, jointly analyzing temporal broken networks can help 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 prorogating 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.

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. 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 the proposed methods.
2. We further develop efficient algorithms which 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 localize 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.

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

Author Manuscript Author Manuscript Author Manuscript Author Manuscript

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 evaluate the proposed methods on both synthetic datasets and two real datasets, including the bank information system and the coal plant cyber-physical system datasets. Experimental results demonstrate the effectiveness of our methods.

2. BACKGROUND AND PROBLEM DEFINITION

In this section, we first introduce the technique of the invariant model [13] 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 [10] 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 $\hat{y}(t;\boldsymbol{\theta})$ be the prediction of $y(t)$ using the ARX model parametarized by $\boldsymbol{\theta}$, which can then be written as

$$\hat{y}(t;\boldsymbol{\theta}) = a_1 y(t-1) + \cdots + a_n y(t-n) + b_0 x(t-k) + \cdots + b_m x(t-k-m) + d \quad (1)$$

$$= \varphi(t)^\top \boldsymbol{\theta}, \quad (2)$$

where $\boldsymbol{\theta} = [a_1,\ldots,a_n, b_0,\ldots,b_m, d]^\top \in \mathbb{R}^{n+m+2}$, $\varphi(t) = [y(t-1),\ldots,y(t-n), x(t-k),\ldots,x(t-k-m), 1]^\top \in \mathbb{R}^{n+m+2}$. 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-square fitting. In real-world applications such as anomaly detection in physical systems, $0 \le n, m, k \le 2$ is a popular choice [6, 13]. We can define the "goodness of fit" (or *fitness score*) of an ARX model as

$$F(\boldsymbol{\theta}) = 1 - \sqrt{\frac{\sum_{t=1}^{N} |y(t) - \hat{y}(t,\boldsymbol{\theta})|^2}{\sum_{t=1}^{N} |y(t) - \bar{y}|^2}}, \quad (3)$$

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 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.

Author Manuscript Author Manuscript Author Manuscript Author Manuscript

To track vanishing correlations, we can use the techniques developed in [6, 15]. At each time point, we compute the (normalized) residual $R(t)$ between the measurement $y(t)$ and its estimate by $\hat{y}(t; \boldsymbol{\theta})$ by

$$R(t)=\frac{|y(t)-\hat{y}(t;\boldsymbol{\theta})|}{\varepsilon_{\max}}, \quad (4)$$

where $\varepsilon_{\max}$ is the maximum training error $\varepsilon_{\max} = \max_{1\leq t\leq N}|y(t) - \hat{y}(t; \boldsymbol{\theta})|$. If the residual exceeds a prefixed threshold, then we declare the invariant as "broken", i.e., 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 $\mathcal{G}_I$ be the invariant network with n nodes. Let $\mathcal{G}_b$ be the broken network for $\mathcal{G}_I$. 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 $\mathcal{G}_I$ 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 **A**, 1 is used to denote that the correlation exists between two time series, and 0 denotes the lack of correlation; while for **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 $\mathcal{G}_I$ 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 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 $\mathcal{G}_I$. To explicitly model this spreading process on the network, we have employed the label propagation technique [16, 24, 26]. Suppose that the (unknown) root cause anomalies are denoted by the indicator vector **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 **r**, whose entries tell us how severe each node of the network has been impaired. The propagation from **e** to **r** can be modeled by the following optimization problem

Author Manuscript Author Manuscript Author Manuscript Author Manuscript

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

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

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

where In is the identity matrix, $\tilde{\mathbf{A}}=\mathbf{D}^{-1/2}\mathbf{A}\mathbf{D}^{-1/2}$ is the degree-normalized version of **A**. Similarly we will use $\tilde{\mathbf{P}}$ as the degree-normalized **P** in the sequel. The first term in Eq. (5) is the *smoothness constraint* [26], 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 trade-off 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 [26]

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

which establishes an explicit, closed-form solution between the initial configuration **e** and the final status **r** through propagation.

To encode the information of the broken network, we propose to use **r** to reconstruct the broken network $\mathcal{G}_b$. The intuition is illustrated in Figure 2. If there exists a broken link in $\mathcal{G}_b$, e.g., $\tilde{\mathbf{P}}_{ij}$ 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'll 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 $\left|\left|(\mathbf{r}\mathbf{r}^{\top})\circ\mathbf{M}-\tilde{\mathbf{P}}\right|\right|_F^2$. Here, ○ is element-wise operator, and **M** is the logical matrix of the invariant network $\mathcal{G}_I$ (1 with edge, 0 without edge). Let $\mathbf{B}=(1-c)(\mathbf{I}_n-c\tilde{\mathrm{A}})^{-1}$, by substituting **r** we obtain the following objective function.

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

Considering that the integer programming in problem (7) is NP-hard, we relax it by using the ℓ_1 penalty on **e** with parameter **τ** to control the number of non-zero entries in **e** [23]. Then we reach the following objective function.

Author Manuscript Author Manuscript Author Manuscript Author Manuscript

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

3.2 Learning Algorithm

In this section, we present an iterative multiplicative updating algorithm to optimize the objective function in (8). The objective function is invariant under these updates if and only if **e** are at a stationary point [17]. 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 Eq. (9) will monotonically decrease the objective function in Eq. (8) until convergence.

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

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.

Author Manuscript Author Manuscript Author Manuscript Author Manuscript

3.3 Theoretical Analysis

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, τ 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
  $\mathbf{D} \leftarrow diag(\mathbf{D}_{11}, ..., \mathbf{D}_{ii})$;
  $\tilde{\mathbf{A}} \leftarrow \mathbf{D}^{-1/2}\mathbf{A}\mathbf{D}^{-1/2}$;
  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 Eq. (9);
  until *convergence*;
end

Algorithm 1.
Ranking Causal Anomalies (RCA)

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 $L = \left|\left|(\mathbf{B}\mathbf{e}\mathbf{e}^{\top}\mathbf{B}^{\top}) \circ \mathbf{M} - \tilde{\mathbf{P}}\right|\right|_F^2 + \tau \mathbf{1}_n^{\top}\mathbf{e}$. Obviously, **B, M** and $\tilde{\mathbf{P}}$ are symmetric matrix. Let $\mathbf{F} = (\mathbf{B}\mathbf{e}\mathbf{e}^{\top}\mathbf{B}^{\top}) \bigcirc \mathbf{M}$, then

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

It follows that

Author Manuscript Author Manuscript Author Manuscript Author Manuscript

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

and thereby

$$\frac{\partial\left|\left|\mathbf{F}-\tilde{\mathbf{P}}\right|\right|_{F}^{2}}{\partial\mathbf{e}_{m}}=4\mathbf{B}^{\top}[(\mathbf{F}-\tilde{\mathbf{P}})\circ\mathbf{M}](\mathbf{B}\mathbf{e}). \quad (12)$$

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

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

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

$$\nabla_{\mathbf{e}}L\circ\mathbf{e}=0 \quad (14)$$

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

3.3.2 Convergence—We use the auxiliary function approach [17] to prove the convergence of Eq. (9) in Theorem 1. We first introduce the definition of auxiliary function as follows.

Definition 3.1. *Z(h, ĥ)* is an auxiliary function for *L(h)* if the conditions

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

are satisfied for any given *h, ĥ* [17].

Lemma 3.1. If *Z* is an auxiliary function for *L*, then *L* is non-increasing under the update [17].

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

Theorem 2. Let *L*(**e**) denote the sum of all terms in *L* containing **e**. The following function

Author Manuscript Author Manuscript Author Manuscript Author Manuscript

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

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

Theorem 2 can be proven in a similar way as [7] by validating $Z(\mathbf{e}, \hat{\mathbf{e}}) \geq L(\mathbf{e})$, $Z(\mathbf{e}, \mathbf{e}) = L(\mathbf{e})$, and the Hessian matrix $\nabla\nabla_{\mathbf{e}}Z(\mathbf{e}, \hat{\mathbf{e}}) \succeq \mathbf{0}$. Due to space limitation, we omit the details.

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

$$\hat{\mathbf{e}}\frac{\overline{4\mathbf{B}^\top[\mathbf{M}\circ(\mathbf{B}\hat{e}}}{\top\mathbf{B}^\top)]\mathbf{B}\hat{e}+\tau\mathbf{1}_n)^{\frac{1}{4}}}, \quad (18)$$

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 **e**, we have $L(\mathbf{e}^0) = Z(\mathbf{e}^0, \mathbf{e}^0) \geq Z(\mathbf{e}^1, \mathbf{e}^0) \geq Z(\mathbf{e}^1, \mathbf{e}^1) = L(\mathbf{e}^1) \geq \ldots \geq L(\mathbf{e}^{Iter})$. Thus $L(\mathbf{e})$ monotonically decreases. Since the objective function Eq. (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, 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 (8) to jointly optimize **r** and **e**. The objective function is shown in the following.

Author Manuscript Author Manuscript Author Manuscript Author Manuscript

$$\min_{\mathbf{e}\geq 0,\mathbf{r}\geq 0} c\mathbf{r}^{\top}(\mathbf{I}_n - \tilde{A})\mathbf{r} + (1-c)\left|\left|\mathbf{r}-\mathbf{e}\right|\right|_F^2 + \lambda\left|\left|(\mathbf{r}\mathbf{r}^{\top})\circ\mathbf{M} - \tilde{\mathbf{P}}\right|\right|_F^2 + \tau\left|\left|\mathbf{e}\right|\right|_1 \quad (19)$$

To optimize this objective function, we can use an alternating scheme. That is, we optimize the objective with respect to **r** while fixing **e**, and vise versa. This procedure continues until convergence. The objective function is invariant under these updates if and only if **r, e** are at a stationary point [17]. Specifically, the solution to the optimization problem in Eq. (19) is based on the following theorem, which is derived from the Karush-Kuhn-Tucker (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 **e** and **r** according to Eq. (20) and Eq. (21) will monotonically decrease the objective function in Eq. (19) until convergence.

$$\mathbf{r} \leftarrow \mathbf{r}\circ\left\{\frac{\tilde{\mathbf{A}}\mathbf{r} + 2\lambda(\tilde{\mathbf{P}}\circ\mathbf{M})\mathbf{r} + (1-c)\mathbf{e}}{\mathbf{r} + 2\lambda[(\mathbf{r}\mathbf{r}^{\top})\circ\mathbf{M}|\mathbf{r}}\right\}^{\frac{1}{4}} \quad (20)$$

$$\mathbf{e} \leftarrow \mathbf{e}\circ\left[\frac{2(1-c)\mathbf{r}}{\tau\mathbf{1}_n + 2(1-c)\mathbf{e}}\right]^{\frac{1}{2}} \quad (21)$$

Based on Theorem 3, we can develop the iterative multiplicative updating algorithm for optimization similar to Algorithm 1. Due to page limit we skip the details. We refer to this ranking algorithm as R-RCA. From Eq. (20) and Eq. (21), 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 7.4, 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 **r**. To reduce the influence of the extreme values or outliers, we employ the softmax normalization on the ranking values **r**. The ranking values are nonlinearly transformed using the sigmoidal function before the multiplication is performed. Thus, the reconstruction error is expressed by $\left|\left|(\sigma(\mathbf{r})\sigma^{\top}(\mathbf{r}))\circ\mathbf{M} - \tilde{\mathbf{P}})\right|\right|_F^2$, where $\boldsymbol{\sigma}(\cdot)$ is the softmax function with:

$$\sigma(\mathbf{r})_i = \frac{e^{\mathbf{r}_i}}{\sum_{k=1}^{n} e^{\mathbf{r}_k}}, (i=1,\ldots,n). \quad (22)$$

Author Manuscript Author Manuscript Author Manuscript Author Manuscript

The corresponding objective function in Algorithm 1 is modified to the following

$$\min_{\mathbf{e}\geq 0}\left|\left|(\sigma(\mathbf{Be})\sigma^{\top}(\mathbf{Be}))\circ\mathbf{M}-\tilde{\mathbf{P}}\right|\right|_{F}^{2}+\tau\left|\left|\mathbf{e}\right|\right|_{1}. \quad (23)$$

Similarly, the objective function for Eq. (19) is modified to the following

$$\min_{\mathbf{e}\geq 0,\mathbf{r}\geq 0} c\mathbf{r}^{\top}(\mathbf{I}_{n}-\tilde{A})\mathbf{r}+(1-c)\left|\left|\mathbf{r}-\mathbf{e}\right|\right|_{F}^{2}+\lambda\left|\left|(\sigma(\mathbf{r})\sigma^{\top}(\mathbf{r}))\circ\mathbf{M}-\tilde{\mathbf{P}}\right|\right|_{F}^{2}+\tau\left|\left|\mathbf{e}\right|\right|_{1}. \quad (24)$$

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

Theorem 4. Updating **e** according to Eq. (25) will monotonically decrease the objective function in Eq. (23) until convergence.

$$\mathbf{e}\leftarrow\mathbf{e}\circ\left\{\frac{4\mathbf{B}^{\top}\boldsymbol{\Psi}(\tilde{\mathbf{P}}\circ\mathbf{M})\sigma(\mathbf{Be})}{4[\mathbf{B}^{\top}(\boldsymbol{\Psi}\sigma(\mathbf{Be})\sigma^{\top}(\mathbf{Be}))\circ\mathbf{M}]\sigma(\mathbf{Be})+\tau\mathbf{1}_{n}}\right\}^{\frac{1}{4}} \quad (25)$$

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

Theorem 5. Updating **r** according to Eq. (26) will monotonically decrease the objective function in Eq. (24) until convergence.

$$\mathbf{r}\leftarrow\mathbf{r}\circ\left\{\frac{\tilde{\mathbf{A}}\mathbf{r}+2\lambda[((\sigma(\mathbf{r})\mathbf{1}_{n}^{\top})\circ\tilde{\mathbf{P}}+\rho\boldsymbol{\Lambda})\circ\mathbf{M}]\sigma(\mathbf{r})+(1-c)\mathbf{e}}{\mathbf{r}+2\lambda[((\sigma(\mathbf{r})\circ\sigma(\mathbf{r}))\sigma^{\top}(\mathbf{r})+\sigma(\mathbf{r})(\sigma^{\top}(\mathbf{r})\tilde{\mathbf{P}}))\circ\mathbf{M}]\sigma(\mathbf{r})}\right\}^{\frac{1}{4}}, \quad (26)$$

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

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 (Eq. (23) and Eq. (24)) 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 [14]. Based on this intuition, we further develop a smoothing strategy by jointly considering the temporal broken networks. Specifically, we add a smoothing term $\left|\left|\mathbf{e}^{(t)}-\mathbf{e}^{(t-1)}\right|\right|_{2}^{2}$ 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 Eq. (27).

Author Manuscript Author Manuscript Author Manuscript Author Manuscript

$$\min_{\mathbf{e}^{(t)}\geq 0, 1\leq t\leq T} \sum_{t=1}^{T} \left[\left|\left| (\mathbf{B}\mathbf{e}^{(t)}(\mathbf{e}^{(t)})^{\top}\mathbf{B}^{\top}) \circ \mathbf{M} - \tilde{\mathbf{P}}^{(t)} \right|\right|_F^2 + \tau \left|\left| \mathbf{e}^{(t)} \right|\right|_1 \right] + \alpha \left|\left| \mathbf{e}^{(t)} - \mathbf{e}^{(t-1)} \right|\right|_2^2 \quad (27)$$

Here, $\tilde{\mathbf{P}}^{(t)}$ is the degree-normalized adjacency matrix of broken network at time point *t*. Similar to the discussion in Section 3.3, we can derive the updating formula of Eq. (27) in the following.

$$\mathbf{e}^{(t)} \leftarrow \mathbf{e}^{(t)} \circ \left\{ \frac{4\mathbf{B}^{\top}(\tilde{\mathbf{P}}^{(t)} \circ \mathbf{M})^{\top}\mathbf{B}\mathbf{e}^{(t)} + 2\alpha\mathbf{e}^{(t-1)}}{4\mathbf{B}^{\top}[\mathbf{M} \circ (\mathbf{B}\mathbf{e}^{(t)}(\mathbf{e}^{(t)})^{\top}\mathbf{B}^{\top})]\mathbf{B}\mathbf{e}^{(t)} + \tau\mathbf{1}_n + 2\alpha\mathbf{e}^{(t)}} \right\}^{\frac{1}{4}} \quad (28)$$

The updating formula for R-RCA, RCA-SOFT, and RRCA-SOFT with temporal broken networks smoothing is similar. Due to 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. EMPIRICAL STUDY

In this section, we perform extensive experiments to evaluate the performance of the proposed methods (summarized in Table 1). 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 [8, 13], and LBP [22]. For all the methods, the tuning parameters were tuned using cross validation. We use several evaluation metrics including precision, recall, and nDCG [12] to measure the performance. The 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 [12, 22]. The nDCG of the top-p ranking result is defined as $\mathrm{nDCG}_p = \frac{\mathrm{DCG}_p}{\mathrm{IDCG}_p}$, where $\mathrm{DCG}_p = \sum_{i=1}^{p} \frac{2^{\mathrm{rel}_i - 1}}{\log_2(1+i)}$, $IDCG_p$ 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.

7.1 Simulation Study

We first evaluate the performance of the proposed methods using simulations. We have followed [8, 22] in generating the simulation data.

7.1.1 Data Generation—We first generate 5000 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,

[2] http://cs.unc.edu/~weicheng/synthetics5000.csv

Author Manuscript Author Manuscript Author Manuscript Author Manuscript

500, and 1000 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 **e** in Eq. (6) and calculate **r** ($c = 0.9$). The values of the top-30 time series with largest values in **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.

7.1.2 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, 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 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 RCASOFT 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 7.4.

7.1.3 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 7.1.2. As has been discussed in Section 5, the *softmax* normalization can greatly suppress the impact of extreme values and outliers in **r**, thus improves the robustness.

Author Manuscript Author Manuscript Author Manuscript Author Manuscript

7.2 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) data set [8, 22]. The monitoring data are collected from a real-world bank information system logs, which contain 11 categories. Each category has a varying number of time series, and Table 2 gives five categories as examples. The data set contains the flow intensities collected every 6 seconds. In total, we have 1,273 flow intensity time series. The training data is 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 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 3.

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 red square. Obviously, their behaviour looks anomalous from that time point on. 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 PAGE-OUT 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, 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 4 and report the number of “DB16”-related monitors in Table 5. From Table 4, we observe that the three baseline methods only report one “DB16” related measurement in the top-12 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 4 and 5, we can see that top ranked items reported by RCA-SOFT and R-RCASOFT 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 $t = 122$,

Author Manuscript Author Manuscript Author Manuscript Author Manuscript

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 6. Due to page limit, we only show the results of mRank, gRank, RCA-SOFT and R-RCA-SOFT. By comparing the results in Tables 4 and 6, we can observe that RCA-SOFT and R-RCA-SOFT significantly outperform mRank and gRank, the latter two methods based on the percentage of broken edges are more sensitive to the anomaly prorogation.

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 7. The number of "DB16"-related monitors in the top-12 results is summarized in Table 8. From Tables 7 and 8, we observe a significant performance improvement of our methods with temporal broken networks 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.

7.3 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 data set contains time series collected through 1625 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 9451 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 algorithms in Table 9. 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 1-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, thus the resultant ranking is more meaningful.

Author Manuscript Author Manuscript Author Manuscript Author Manuscript

7.4 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 9(a) shows the value of the objective function with respect to the number of iterations on different data sets. 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 [21]. We also report the running time of each algorithm on the two real data sets in Figure 10. 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% edges are randomly selected as broken links. Using simulated data, we compare the running time of RCA, R-RCA, RCA-SOFT, and R-RCA-SOFT. Figure 9(b) plots the running time of different algorithms w.r.t. the number of nodes 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. 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. [25] 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. [13] 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 [14, 15]. These approaches generally focus on locating the faulty components, 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, 11]. For example, Akoglu et al. [1] proposed the OddBall algorithm to detect anomalous nodes in weighted graphs. Liu et al. [18] 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 provides a more dynamic and complete picture for anomaly ranking. On dynamic graphs, anomaly detection aims at detecting abnormal events [19]. Most approaches along this direction are designed to detect anomaly time-stamps in which

Author Manuscript Author Manuscript Author Manuscript Author Manuscript

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 [20]. In their approach, a set of initial suspects need to be provided; then internal relationship among these initial suspects is 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. [14] used the ratio of broken edges in the invariant network as the anomaly score for ranking; Ge et al. [8] proposed mRank and gRank to rank causal anomalies; Tao et al. [22] 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 do not take into account the global network structures, neither the global fault propagation spreading on the network. Therefore the resultant rankings can be sub-optimal.

9. 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; in the meantime, it is robust to noise and computationally efficient. Using both synthetic and real-life data sets, we show that the proposed methods outperform other competitors by a large margin.

Acknowledgments

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.

References

1. Akoglu L, McGlohon M, Faloutsos C. Oddball: Spotting anomalies in weighted graphs. PAKDD. : 410–421.
2. Akoglu L, Tong H, Koutra D. Graph-based anomaly detection and description: A survey. CoRR. 2014
3. Boyd, S., Vandenberghe, L. Convex Optimization. Cambridge University Press; 2004.
4. Breunig MM, Kriegel H-P, Ng RT, Sander J. Lof: Identifying density-based local outliers. SIGMOD. 2000:93–104.
5. Chandola V, Banerjee A, Kumar V. Anomaly detection: A survey. ACM Computing Surveys. 2009; 41(3):1–58.
6. Chen, H., Cheng, H., Jiang, G., Yoshihira, K. ICDM. 2008. Global invariants for the management of large scale information systems.
7. Ding, C., Li, T., Peng, W., Park, H. KDD. 2006. Orthogonal nonnegative matrix t-factorizations for clustering; p. 126-135.
8. Ge Y, Jiang G, Ding M, Xiong H. Ranking metric anomaly in invariant networks. TKDD. Jun.2014 8(2)
9. Gertler, J. Fault Detection and Diagnosis in Engineering Systems. Marcel Dekker; 1998.
10. System Identification (2nd Ed.): Theory for the User. 1999. Gertler: 1998.

Author Manuscript Author Manuscript Author Manuscript Author Manuscript

11. Henderson, K., Eliassi-Rad, T., Faloutsos, C., Akoglu, L., Li, L., Maruhashi, K., Prakash, BA., Tong, H. KDD. 2010. Metric forensics: a multi-level approach for mining volatile graphs; p. 163-172.
12. Jarvelin, Kalervo, Kekalainen, Jaana. Cumulated gain-based evaluation of IR techniques. TIS. 2002; 20(4):422–446.
13. Jiang G, Chen H, Yoshihira K. Discovering likely invariants of distributed transaction systems for autonomic system management. Cluster Computing. 2006; 9(4):385–399.
14. Jiang G, Chen H, Yoshihira K. Modeling and tracking of transaction flow dynamics for fault detection in complex systems. TDSC. 2006; 3(4):312–326.
15. Jiang G, Chen H, Yoshihira K. Efficient and scalable algorithms for inferring invariants in distributed systems. TKDE. 2007; 19(11):1508–1523.
16. Kim, TH., Lee, KM., Lee, SU. ECCV. 2008. Generative image segmentation using random walks with restart; p. 264-275.
17. Lee, DD., Seung, HS. NIPS. 2000. Algorithms for non-negative matrix factorization; p. 556-562.
18. Liu, C., Yan, X., Yu, H., Han, J., Yu, PS. SDM. 2005. Mining behavior graphs for ”backtrace” of noncrashing bugs; p. 286-297.
19. Rossi RA, Gallagher B, Neville J, Henderson K. Modeling dynamic behavior in large evolving graphs. WSDM. 2013:667–676.
20. Sun J, Tao D, Faloutsos C. Beyond streams and graphs: Dynamic tensor analysis. KDD ’06. 2006:374–383.
21. Sutton, RS., Barto, AG. Reinforcement Learning: An Introduction. The MIT Press; Cambridge, MA: 1998.
22. Tao, C., Ge, Y., Song, Q., Ge, Y., Omitaomu, F. ICDM. 2014. Metric ranking of invariant networks with belief propagation.
23. Tibshirani RJ. Regression shrinkage and selection via the lasso. Journal of the Royal Statistical Society, Series B. 1996; 58(1):267–288.
24. Tong, H., Faloutsos, C., Pan, J-Y. ICDM. 2006. Fast random walk with restart and its applications; p. 613-622.
25. Yemini SA, Kliger S, Mozes E, Yemini Y, Ohsie D. High speed and robust event correlation. IEEE Communications Magazine. 1996; 34:82–90.
26. Zhou, D., Bousquet, O., Lal, TN., Weston, J., Schölkopf, B. NIPS. 2003. Learning with local and global consistency; p. 321-328.

Author Manuscript Author Manuscript Author Manuscript Author Manuscript

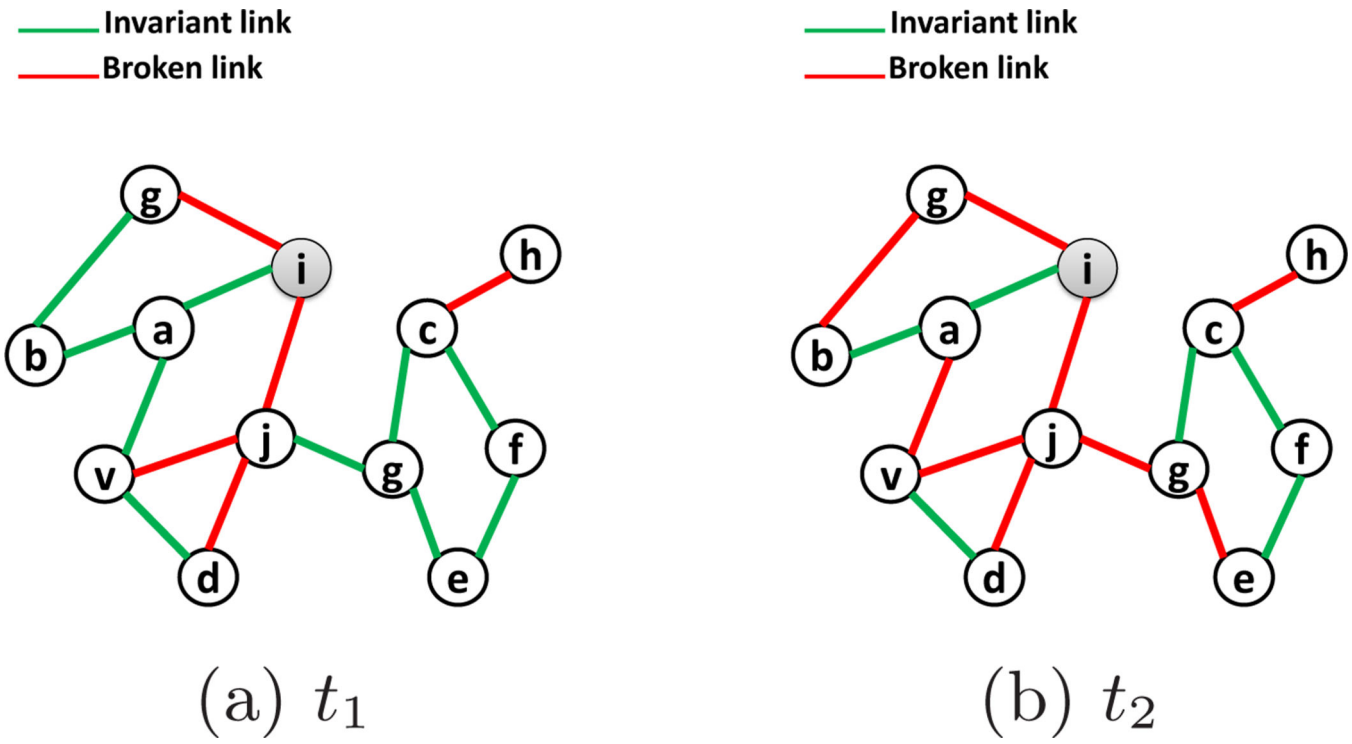


Figure 1.
Invariant network and vanishing correlations(red edges).

Author Manuscript Author Manuscript Author Manuscript Author Manuscript

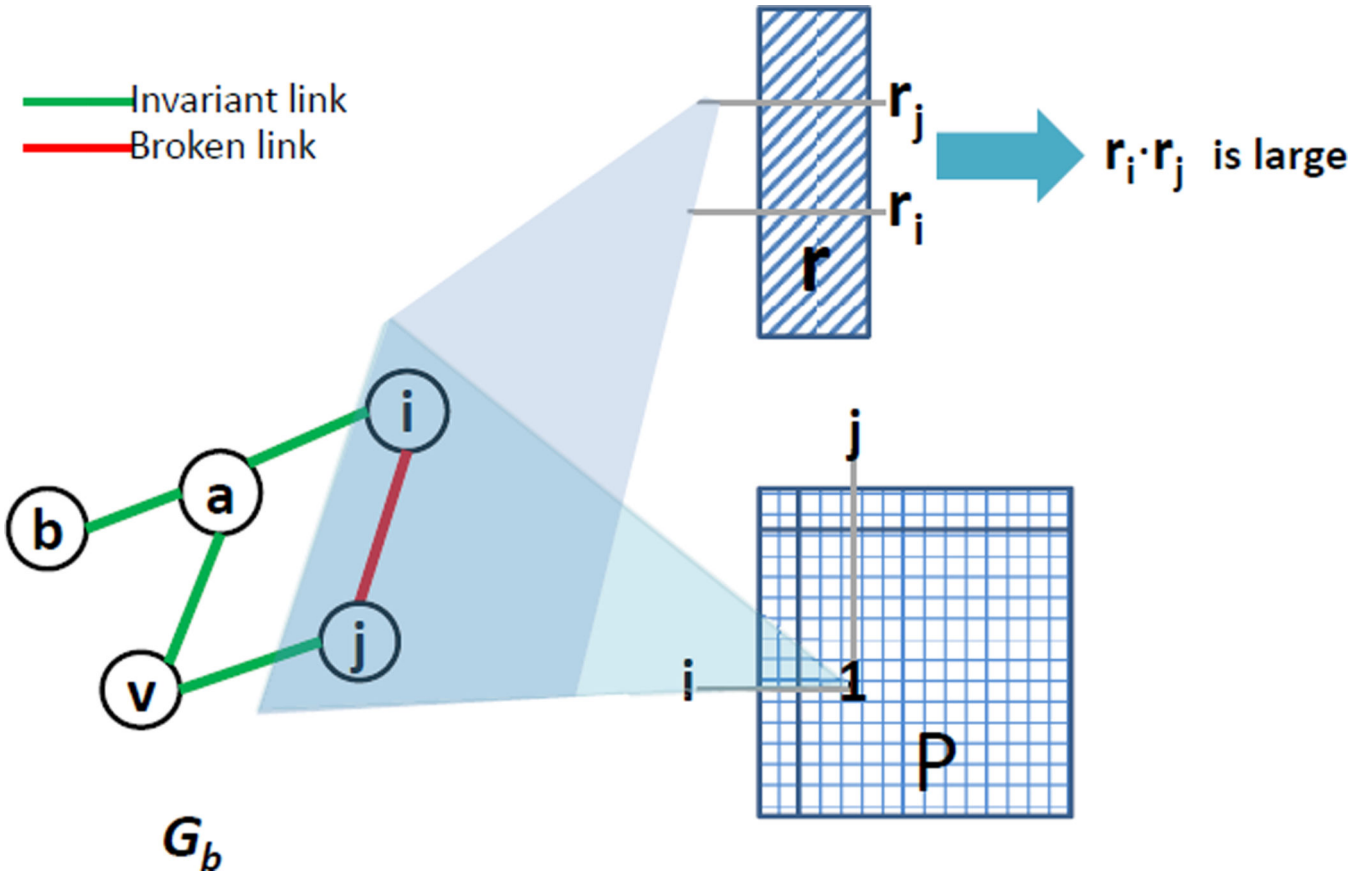


Figure 2.
Reconstruction of the broken invariant network using anomaly score vector r.

Author Manuscript Author Manuscript Author Manuscript Author Manuscript

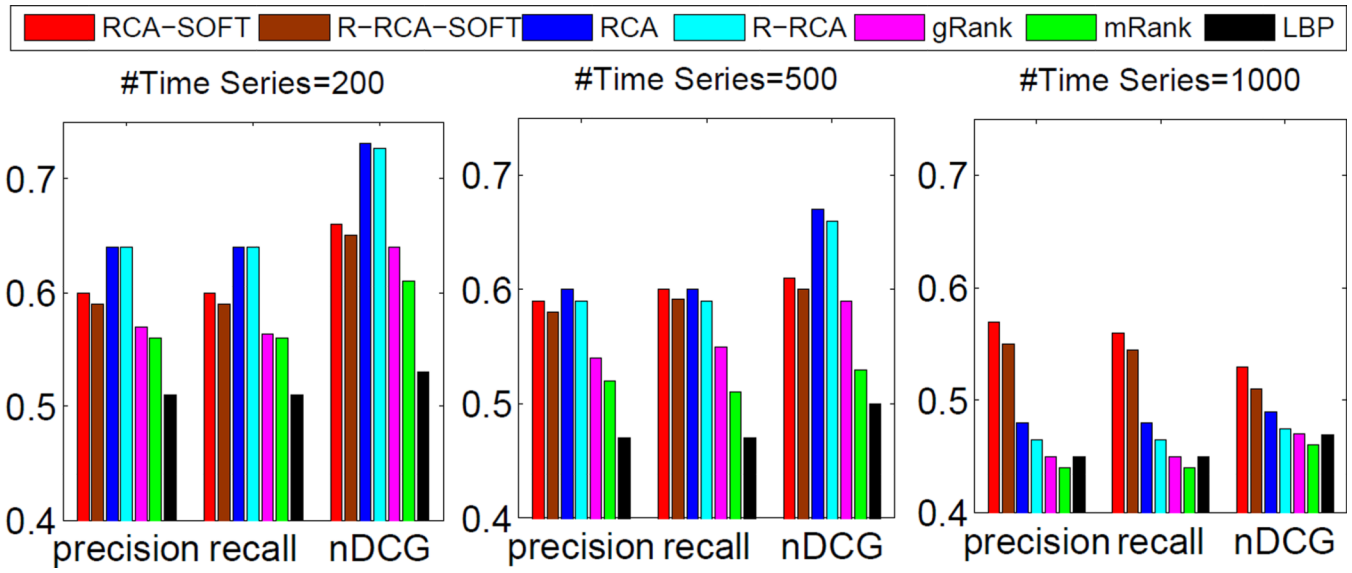


Figure 3.
Comparison on synthetic data(K, $p = 10$).

Author Manuscript Author Manuscript Author Manuscript Author Manuscript

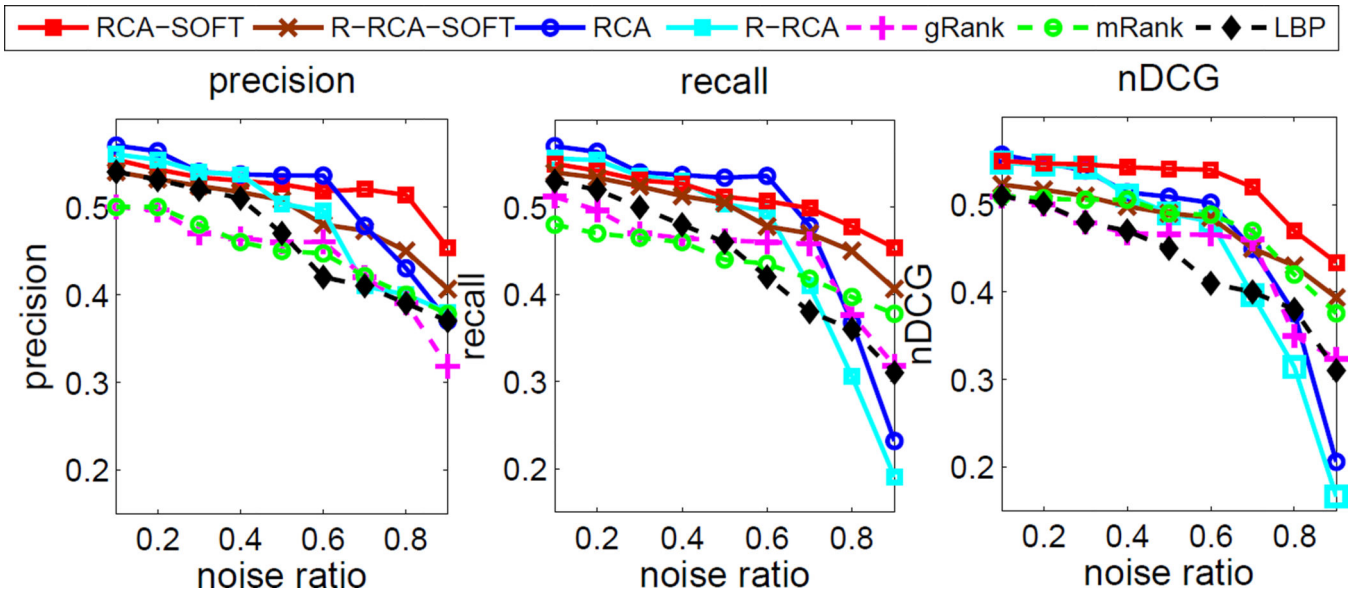


Figure 4.
Performance with different noise ratio(K, $p = 10$).

Author Manuscript Author Manuscript Author Manuscript Author Manuscript

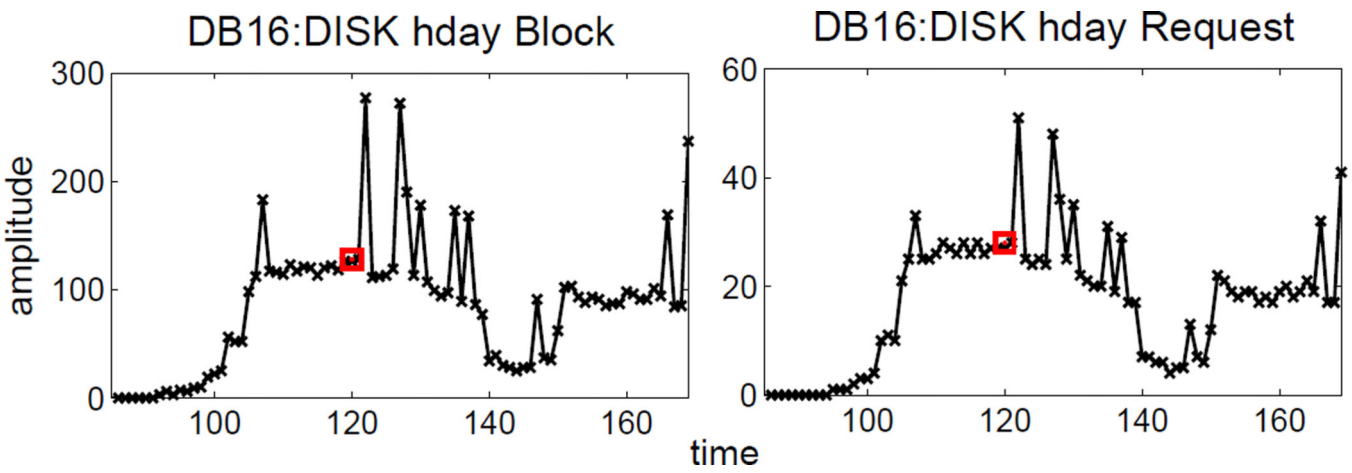


Figure 5.
Two example monitoring data of BIS.

Author Manuscript Author Manuscript Author Manuscript Author Manuscript

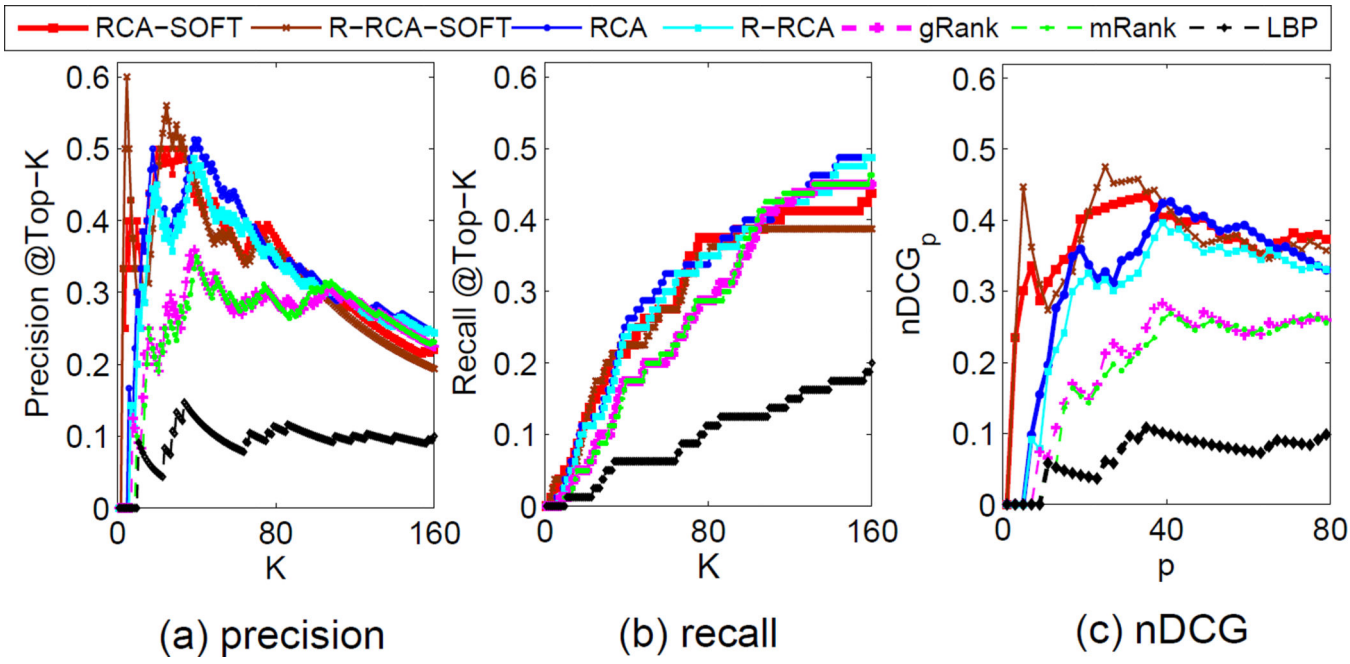
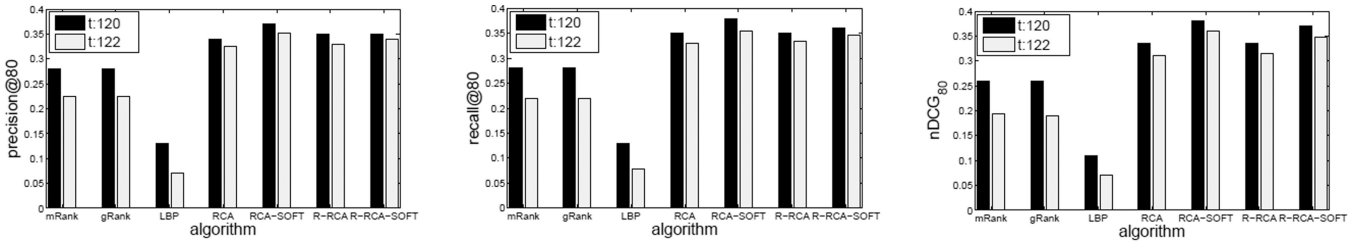


Figure 6.
Comparison on BIS data.

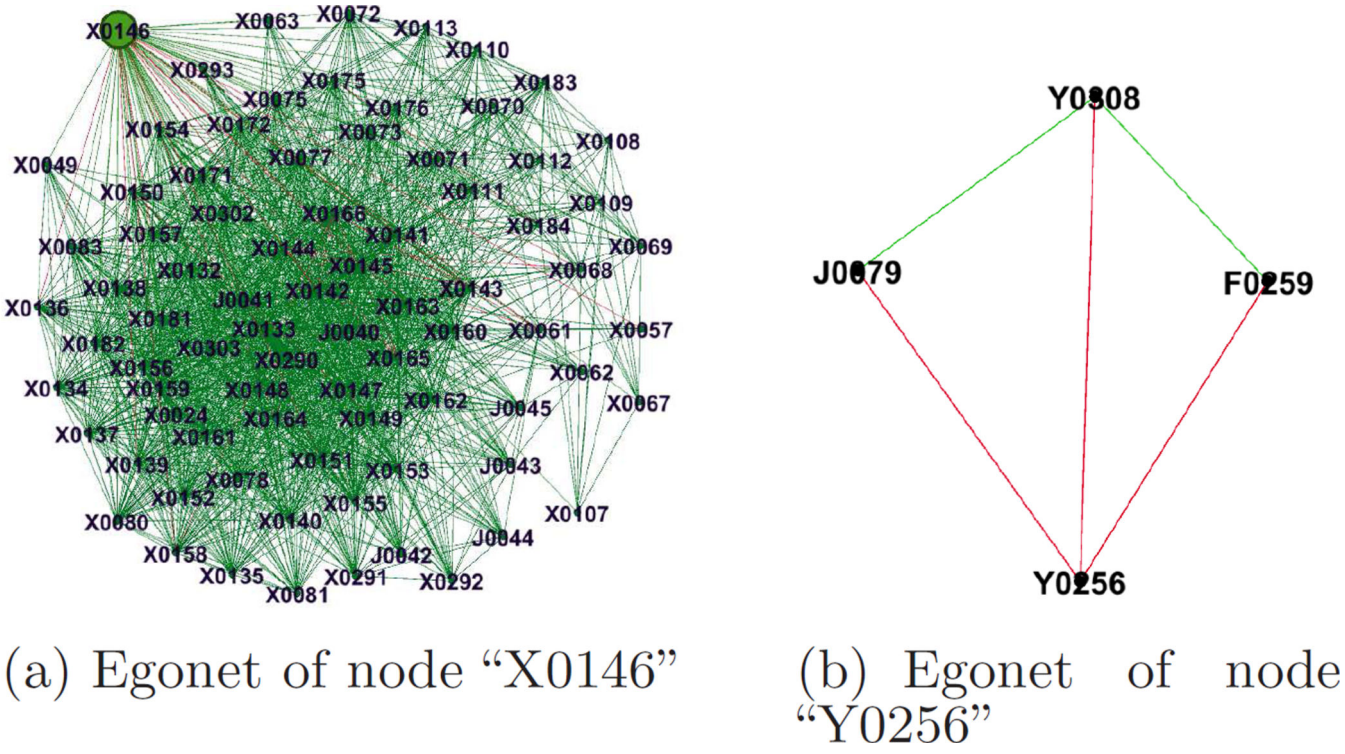


(a) precision (b) recall (c) nDCG

Figure 7.
Performance at *t*:120 v.s. *t*:122 on BIS data(*p*,*K*=80).

Author Manuscript Author Manuscript Author Manuscript Author Manuscript

Author Manuscript Author Manuscript Author Manuscript Author Manuscript



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

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

Author Manuscript Author Manuscript Author Manuscript Author Manuscript

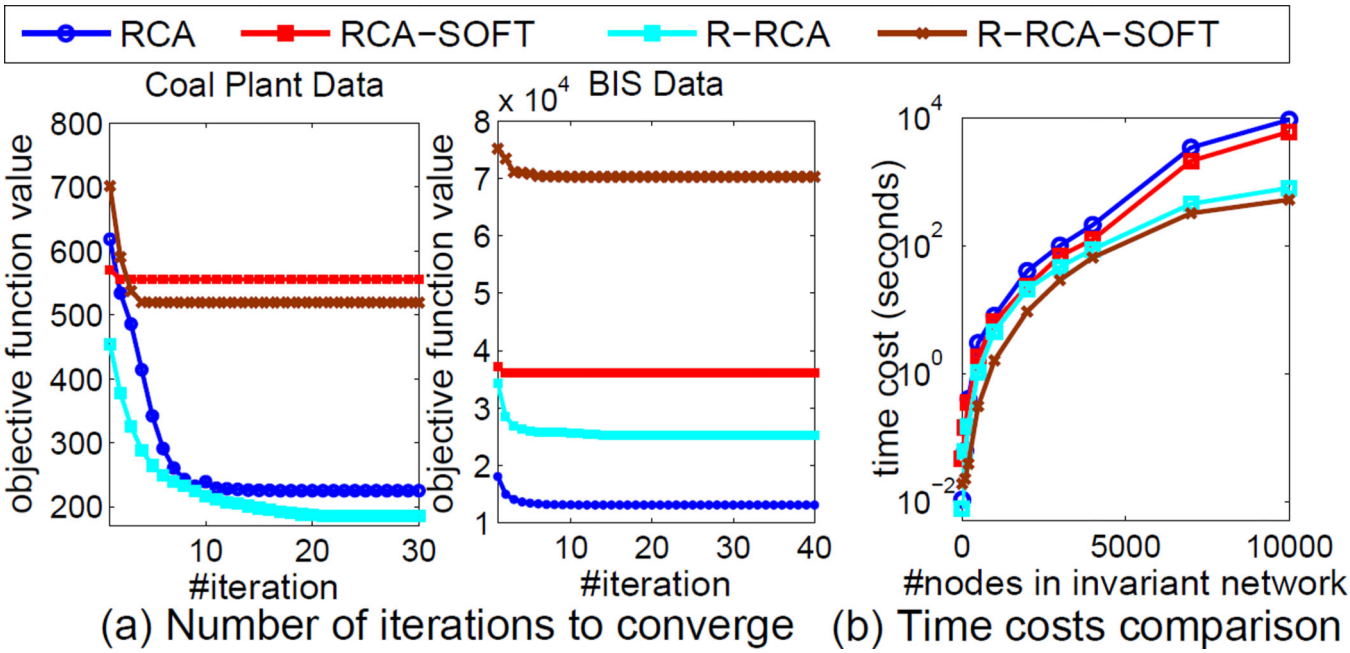


Figure 9.
Number of iterations to converge and time cost comparison.

Author Manuscript Author Manuscript Author Manuscript Author Manuscript

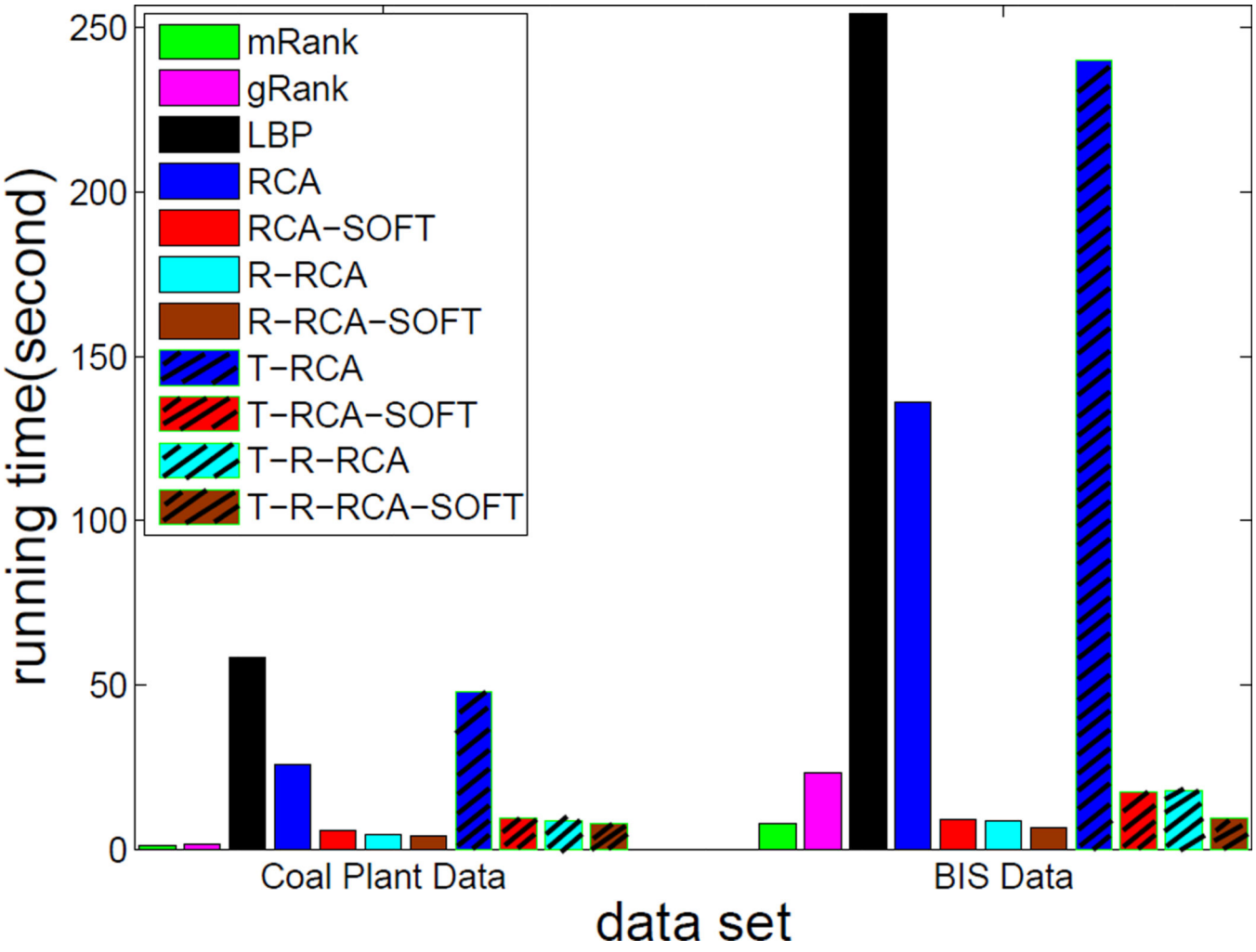


Figure 10.
Running time on real data sets.

Author Manuscript Author Manuscript Author Manuscript Author Manuscript

Table 1

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
$\mathcal{G}_I$	the invariant network
$\mathcal{G}_b$	the broken network for $\mathcal{G}_I$
$\mathbf{A}$ ($\tilde{\mathbf{A}}$) $\in \mathbb{R}^{n \times n}$	the (normalized) adjacency matrix of $\mathcal{G}_I$
$\mathbf{P}$ ($\tilde{\mathbf{P}}$) $\in \mathbb{R}^{n \times n}$	the (normalized) adjacency matrix of $\mathcal{G}_b$
$\mathbf{M} \in \mathbb{R}^{n \times n}$	the logical matrix of $\mathcal{G}_I$
$d(i)$	the degree of the i^{th} node in network $\mathcal{G}_I$
$\mathbf{D} \in \mathbb{R}^{n \times n}$	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

Author Manuscript

Author Manuscript

Author Manuscript

Author Manuscript

Table 2

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

Author Manuscript

Author Manuscript

Author Manuscript

Author Manuscript

Table 3

Data set description

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

Author Manuscript Author Manuscript Author Manuscript Author Manuscript

Table 4

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

KDD Author manuscript; available in PMC 2017 July 13.

Author Manuscript Author Manuscript Author Manuscript Author Manuscript

Table 5

Number of "DB16" related monitors in top 32 results on BIS data(r:120).

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

Author Manuscript

Author Manuscript

Author Manuscript

Author Manuscript

Table 6

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

Author Manuscript

Author Manuscript

Author Manuscript

Author Manuscript

Table 7

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

Author Manuscript

Author Manuscript

Author Manuscript

Author Manuscript

Table 8

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

Author Manuscript Author Manuscript Author Manuscript Author Manuscript

Table 9

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

KDD. Author manuscript; available in PMC 2017 July 13.