This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2018.2868097, IEEE Transactions on Knowledge and Data Engineering

# A Nonparametric Approach to Uncovering Connected Anomalies by Tree Shaped Priors

Nannan Wu, Feng Chen, Jianxin Li, Jinpeng Huai, Baojian Zhou, Bo Li, Naren Ramakrishnan

**Abstract**—The area of anomaly detection has recently been expanded in the graph-based data. Anomalous vertices are often exhibited as a *connected* subgraph. Few works, however, have focused on connected anomalous subgraph detection because of the challenge of optimizing graph functionals under connectivity constraints. We employ Non-Parametric Graph Scan (NPGS) statistics for detecting anomalies within graph-based data. Based on the NPGS statistics, we proposed an efficient approximate approach to the connected anomalous subgraph detection problem that provides provable guarantees on performance and quality. In particular, we first decompose the problem into a sequence of subproblems, each of which can be reduced to a *Budget Price-Collecting Steiner Tree* (B-PCST) problem, and then develop efficient exact and approximate algorithms for a special category of graphs in which the anomalous subgraphs can be reformulated in a fixed tree topology. Our method has a wide variety of applications, such as disease outbreak detection, road traffic congestion detection, and event detection in social media, because the NPGS statistics is free of distribution assumptions and can be applied to heterogeneous graph data.

**Index Terms**—Nonparametric graph scan statistic, connected subgraph, Tree prior, Anomalous subgraph.

## 1 Introduction

Anomalous subgraph detection as an open problem has attracted much attention in recent years [1], [2], [3], [4], [5], [6]. We consider a graph $\mathbb{G} = (\mathbb{V}, \mathbb{E})$, where each vertex $v \in \mathbb{V}$ is associated with features values $\mathbf{x}_v \in \mathbb{R}$ (e.g., the number of infected patients in Figure 1) that follow some statistical distributions. The general goal of anomalous subgraph detection is to optimize some objective functions (e.g., $F(S)$ where $S \subseteq \mathbb{V}$) of abnormality of the feature values over all connected subsets of vertices ($S \subseteq \mathbb{V}$). To motivate this scenario, we consider the cholera outbreak problem [7] as shown in Figure 1. Suppose that we have a network of counties (i.e., *vertices*) and each vertex has a feature referring to the number of cases of cholera in that county on a given day. Suppose further that two vertices are connected by an edge if they share the boundary. We wish to identify possible cholera outbreaks at a very early stage, which requires identifying subtle patterns (e.g., *a 20% increase in the number of patients with symptoms of cholera in four local (connected) counties*) in the noisy background data. These subtle signals may not be detectable if we examine only a small part of the affected subset (e.g., *a single county*) or a larger connected subset containing many unaffected vertices (e.g., *the aggregate count for the entire state*). As a result, traditional "bottom-up" methods (*which identify and aggregate individual vertices* [8]) and "top-down" methods (*which detect anomalous global trends (bursts)*) often have low power for detecting the potentially emerging events [9], [10].

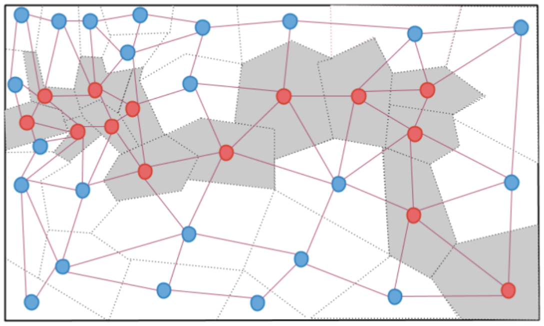


Figure 1: A potential *cholera* outbreak led to the elevated number of infected cases in counties near the river, which form an irregular shaped connected subgraph (cluster) of counties. (Redrawn from [7].)

The underlying assumption of anomalous pattern detection is that the features of a majority of vertices are generated from the same distribution representing the (typically unknown and possibly complex) normal behavior of the system; thus, we wish to detect connected or correlated subgraphs of vertices which are unexpected given the typical data distribution (e.g., Gaussian distribution, Poisson distribution). Existing methods can be categorized into two main groups, namely parametric and nonparametric methods. Parametric methods assume specific forms of distributions for features of normal and abnormal vertices respectively, and formalize anomaly detection as a hypothesis testing problem. In particular, under the alternative hypothesis ($H_1(S)$), an underlying anomalous phenomenon is characterized by the following: features of a majority of the vertices are generated from the same background distribution, and features of perhaps a small connected subset $S \subseteq \mathbb{V}$ of vertices are generated from a different distribution. The goal is to maximize an appropriate set function ($F(S)$),

- N. Wu, J. Li, J. Huai and B. Li are with School of Computer Science and Engineering, Beihang University, China 100191. E-mail: {wunannan, lijx, libo}@act.buaa.edu.cn, and huaijp@buaa.edu.cn
- F. Chen and B. Zhou are with Computer Science Department, University at Albany, SUNY, Albany, NY 12203. E-mail: {fchen5, bzhou6}@albany.edu
- N. Ramakrishnan is with Computer Science Department, Virginia Tech, Arlington, VA 22203. E-mail: naren@cs.vt.edu

typically the likelihood ratio $F(S) = \frac{\Pr(\text{Data}|H_1(S))}{\Pr(\text{Data}|H_0)}$, over all possible connected subsets $S$ (with $H_0$ being the null hypothesis). Depending on specific forms of distributions assumed, a number of methods have been proposed, including expectation-based Poisson statistic [11], Kulldorff statistic [12], elevated mean scan statistic [6], [13], and various others.

Nonparametric methods do not assume specific forms of distributions for normal and abnormal vertices. Instead, they first estimate a p-value for each vertex based on empirical calibration by comparing the current features of this vertex with its features in the historical data for the vertex [9], [14]. The empirical p-value provides an estimate of the probability that a randomly selected sample would have observed features as extreme as the current features of this vertex, under the null hypothesis that no events of interest are occurring. This approach then maximizes a score function $F(S)$ of p-values in $S$, typically nonparametric scan statistic measuring the significance of the collection of p-values in $S$, over all possible connected subsets. A number of NPGS statistic functions have been proposed in recent years, including Berk-Jones (BJ) statistic [15], Higher Criticism (HC) statistic [16], Tippet's statistic, rank truncated statistic, and various others. Note that, these nonparametric statistic functions were originally proposed to combine p-values from a set of hypothesis tests in the area of statistical meta analysis. Recent studies show that these functions can be well applied to NPGS for detecting anomalous subgraphs [9], [17], [18].

This paper focuses on nonparametric methods and considers the general optimization framework of the Non Parametric Graph Scan (NPGS) statistics:

$$\max_{S \subseteq \mathbb{V},\ S \text{ is connected}} F(S), \tag{1}$$

where $F(S)$ is a predefined NPGS statistic function. This optimization problem is hard in general. For example, the additive statistic function $F(S) := \sum_{v \in S} -\log_\alpha x(v)$ [18] can be shown to be NP-hard to optimize via reduction from the net-worth Node-Weighted Prize-Collecting Steiner Tree (PCST) problem, where $p(v) : v \to [0,1]$ maps each vertex to an empirical p-value and $\alpha$ is a predefined confidence level (e.g., 0.05). The PCST problem is known to be NP-hard and does not admit any finite approximation algorithm [19]. The hardness of the NPGS problem for non-additive statistic functions is unknown, and the non-additive property makes it difficult to prove complexity results through reductions from known discrete optimization problems.

**Related Work.** Existing algorithms for anomalous connected subgraph detection have two main groups, namely exact and approximate algorithms. 1) **Exact algorithms.** An exhaustive search algorithm, *FlexScan*, is proposed to identify the most anomalous connected subgraph within all connected subgraphs formed by a center and a connected subset of its $k-1$ neighbors [3]. By applying Linear Time Subset Scanning (LTSS [20]) to filter sub-optimal subsets, Speakman and Neill et al. improve the previous work, *FlexScan*, by designing a new branch-and-bound algorithm to graph-structured data [5]. 2) **Approximate algorithms.** For the anomalous connected subgraph, Duczmal and Assunção present a heuristic algorithm with a simulated annealing strategy [1], which is extended by incorporating regularization on the compactness of subgraphs [2]. Speakman et al. present an additive subgraph detection algorithm based on dijkstra's algorithm [21]. Rozenshtein et al. apply semidefinite programming and the GW-algorithm [22] to identify anomalous subgraphs that are compact but not necessarily connected. Chen and Neil propose a greedy algorithm based on iterative subgraph expansion and linear time subset scanning [9]. The aforementioned exact algorithms enable exact computation of the highest-scoring connected subgraphs, but become computationally infeasible if the graph size is larger than 1000. The approximate algorithms are mostly scalable to large datasets, but have no theoretical guarantees on the quality of the returned subgraphs for general graphs.

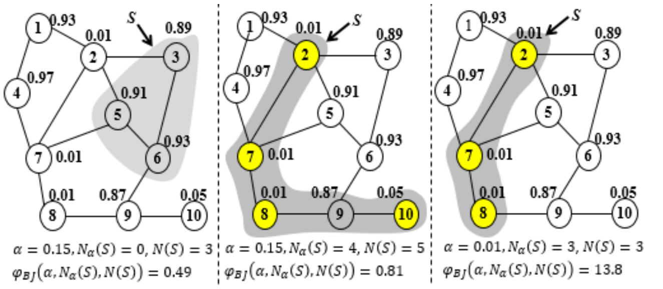
The main contributions of our work are summarized:

- **Hardness analysis.** We reformulate the NPGS problem as a sequence of B-PCST sub-problems and show that this reformulated problem is NP-hard for a large class of non-additive nonparametric statistic functions. These functions satisfy two intuitive properties on the cardinality of the input subgraph $S$ and the number of vertices in $S$ that are significant at a predefined confidence level $\alpha$.
- **Exact and approximate algorithms for a special category of tree-priors graphs.** We develop efficient algorithms to the NPGS problem that are guaranteed to find an optimal solution in worst-case $O(N^4)$ time and an $(1+\epsilon)^L$-approximate solution in worst case $O(N^3/\epsilon)$ time, respectively, when the connectivity constraint of the subgraph can be reformulated in a fixed tree topology, where $L$ refers to the depth of the tree topology.
- **Comprehensive experiments to validate the effectiveness and efficiency of the proposed techniques.** We conduct extensive experiments on a water sensor dataset and a Chinese Weibo dataset. The results demonstrate that our proposed algorithms outperform existing representative techniques for both performance and quality.
- **Real-world case studies.** We apply our proposed method to cyber-attack detection in Internet traffic networks, haze event detection in social networks, and road congested detection in road networks. By case studies, we validate our method that has wide applications in uncovering connected subgraph anomalies.

This paper is organized as follows. Section 2 reviews nonparametric graph scan statistics. Section 3 first presents the decomposition of the NPGS problem into a sequence of subproblems, the NP-hardness of the NPGS problem, and efficient approximation algorithms. Experiments on the three real-datasets are presented in Section 4, and Section 5 concludes the current work and describes the future work.

## 2 Nonparametric Graph Scan Statistics

Given a graph $\mathbb{G}(\mathbb{V}, \mathbb{E}, p)$ where $\mathbb{V} = \{v_1, ..., v_N\}$, $N$ refers to the total number of vertices, $\mathbb{E} \subseteq \mathbb{V} \times \mathbb{V}$ refers to the set of edges, and the mapping function $p : \mathbb{V} \to [0,1]$ defines a single empirical p-value corresponding to each node $v$.



$\alpha = 0.15, N_\alpha(S) = 0, N(S) = 3$
$\varphi_{BJ}(\alpha, N_\alpha(S), N(S)) = 0.49$

$\alpha = 0.15, N_\alpha(S) = 4, N(S) = 5$
$\varphi_{BJ}(\alpha, N_\alpha(S), N(S)) = 0.81$

$\alpha = 0.01, N_\alpha(S) = 3, N(S) = 3$
$\varphi_{BJ}(\alpha, N_\alpha(S), N(S)) = 13.8$

Figure 2: The BJ statistic scores of the three example subgraphs demonstrate that this score function increases with $N_\alpha(S)$ and decreases with $N(S) - N_\alpha(S)$ and $\alpha$. Yellow-colored vertices refer to the vertices whose p-values are less than or equal to $\alpha$.

About the definition for the mapping function $p$, we can refer to the recent work [9]. The general form of the Non-Parametric Graph Scan (NPGS) statistic [9], [17] is defined as:

$$F(S) = \max_\alpha F_\alpha(S) = \max_\alpha \phi(\alpha, N_\alpha(S), N(S)), \quad (2)$$

where $S \subseteq \mathbb{V}$ refers to a connected set of vertices (subgraph), $N_\alpha(S) = \sum_{v \in S} \delta(p(v) \leq \alpha)$ (i.e., $\delta(\cdot) = 1$ if its input is true, otherwise $\delta(\cdot) = 0$) is the number of p-values significant at level $\alpha$, $N(S) = \sum_{v \in S} 1$ is the total number of p-values in $S$. The significance level $\alpha$ can be optimized between 0 and some constant $\alpha_{max}$ (0.15 by default). The function $\phi(\alpha, N_\alpha(S), N(S))$ refers to a nonparametric scan statistic, i.e., a function that compares the observed number of p-values $N_\alpha(S)$ that are significant at level $\alpha$ to the expected number of significant p-values $E[N_\alpha(S)] = \alpha N(S)$, under the null hypothesis that p-values are uniformly distributed on $[0, 1]$. We assume that the function $\phi(\alpha, N_\alpha(S), N(S))$ satisfies the two intuitive properties:

(P1) $\phi$ is monotonically ***increasing*** w.r.t. $N_\alpha(S)$,
(P2) $\phi$ is monotonically ***decreasing*** w.r.t. $N(S) - N_\alpha(S)$.

These assumptions follow natually because the ratio of expected number of significant p-values $N_\alpha(S) \;/\; (N_\alpha(S) + N(S) - N_\alpha(S))\alpha$ increases with the numerator (P1), and decreases with the $(N(S) - N_\alpha(S))$ (P2). For the range of $\alpha$ in nonparametric scan statistics, its importance is discussed in [9].

This paper presents efficient algorithms for the large class of nonparametric scan statistics that satisfy the above two properties, such as the Berk-Jones (BJ) statistic [23], the Higher Criticism (HC) statistic [24], the Kolmogorov-Smirnov statistic, the Davidov-Herman statistic, and the chi-bar squared statistic. For illustration purpose, we consider the first two functions. For the simplicity, we write $N_\alpha(S)$ as $N_\alpha$ and $N(S)$ as $N$. The BJ statistic is defined as:

$$\varphi_{BJ}(\alpha, N_\alpha, N) = N \times \mathrm{KL}\left(\frac{N_\alpha}{N}, \alpha\right), \quad (3)$$

where KL is the Kullback-Liebler divergence between the observed and expected proportions of p-values less than $\alpha$:

$$\mathrm{KL}(a, b) = a \log\left(\frac{a}{b}\right) + (1 - a) \log\left(\frac{1 - a}{1 - b}\right).$$

where $a, b \in [0, 1]$, especially when $b = 0$ or $b = 1$, we have $\mathrm{KL}(a, b) = 0$. The BJ statistic can be interpreted as the log-likelihood ratio statistic for testing whether the empirical p-values follow a uniform or piecewise constant distribution. We illustrate the BJ statistic in Figure 2. Berk and Jones [23] demonstrated that this statistic fulfills several optimality properties and has greater power than any weighted Kolmogorov statistic. The HC statistic is defined as:

$$\varphi_{HC}(\alpha, N_\alpha, N) = \frac{N_\alpha - N\alpha}{\sqrt{N\alpha(1 - \alpha)}}. \quad (4)$$

The HC statistic can be interpreted as the log-likelihood ratio statistic for testing wether the empirical p-values follow a uniform or binomial distribution with the parameters $N$ and $\alpha$.

Given a selected nonparametric scan statistic function $\varphi(\alpha, N_\alpha(S), N(S))$, the detection of the most anomalous connected subgraph from $\mathbb{V}$ can be formalized as the following optimization problem:

$$\max_{S \subseteq \mathbb{V}: S \text{ is connected}} \; \max_{\alpha \leq \alpha_{max}} \phi(\alpha, N_\alpha(S), N(S)), \quad (5)$$

which is equivalent to the problem:

$$\max_{\alpha \in \mathbb{U}(\mathbb{V}, \alpha_{max})} \; \max_{S \subseteq \mathbb{V}: S \text{ is connected}} \phi(\alpha, N_\alpha(S), N(S)), \quad (6)$$

where $\mathbb{U}(\mathbb{V}, \alpha_{max})$ refers to the union of $\{\alpha_{max}\}$ and the set of distinct p-values less than $\alpha_{max}$ in $\mathbb{V}$.

## 3 Methodology

This section reformulates the NPGS problem as a sequence of subproblems, where each subproblem can be reduced to a budget prize-collecting steiner tree problem [25], and presents approximate algorithms with provable guarantees.

### 3.1 Problem Reformulation

Let $S_\alpha^- \equiv \{v \mid p(v) \leq \alpha, v \in S\}$, $S_\alpha^+ \equiv \{v \mid p(v) > \alpha, v \in S\}$. We denote a vertex $v$ as an abnormal vertex if $p(v) \leq \alpha$; otherwise, a normal vertex (i.e., abnormal set $S_\alpha^-$ and normal set $S_\alpha^+$).

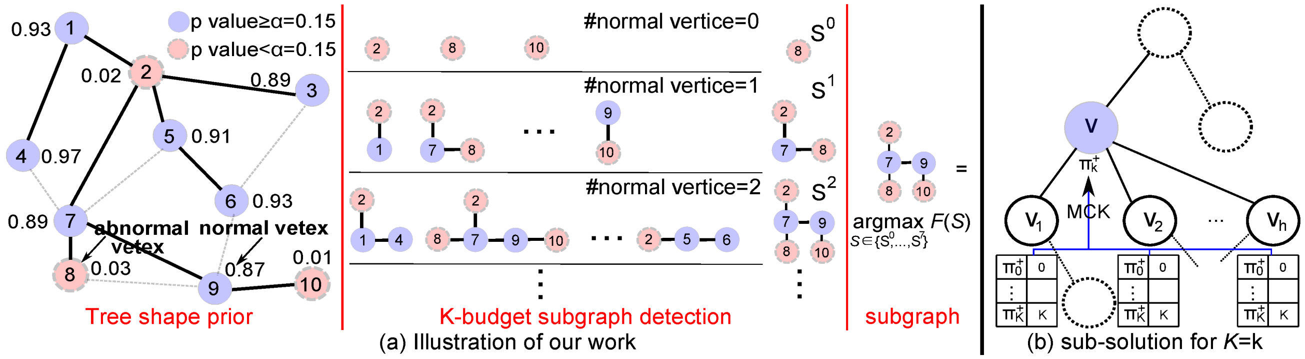
**Lemma 1.** *Given a set of normal vertices $Q \subseteq V_\alpha^+$, the NPGS problem has an additional constraint on $S_\alpha^+$:*

$$\max_{\alpha \in \mathbb{U}(\mathbb{V}, \alpha_{max})} \; \max_{S \subseteq \mathbb{V}: S \text{ is connected}} \phi(\alpha, N_\alpha(S), N(S)), \; s.t. \; S_\alpha^+ = Q \quad (7)$$

*is equivalent to the problem:*

$$\max_{\alpha \in \mathbb{U}(\mathbb{V}, \alpha_{max})} \; \max_{S \subseteq \mathbb{V}: S \text{ is connected}} N_\alpha(S), \quad s.t. \; S_\alpha^+ = Q. \quad (8)$$

*Proof.* It suffices to prove the equivalence for each $\alpha \in \mathbb{U}(\mathbb{V}, \alpha_{max})$ by contradiction. We assume that $\alpha$ is fixed and $S^*$ is the optimal solution to Problem (8), but not the optimal solution to Problem (7). It follows that there is another feasible subgraph $S^0$, such that $\phi(\alpha, N_\alpha(S^*), N(S^*)) \leq \phi(\alpha, N_\alpha(S^0), N(S^0))$. The constraint $S_\alpha^+ = Q$ (i.e., the property P2) in Problem (7) is the same as in Problem (8). According to the property P1, there must be $N_\alpha(S^*) \leq N_\alpha(S^0)$, and thus $S^*$ is not the optimal solution to Problem (8). This contradiction gives the proof. □



(a) Illustration of our work

(b) sub-solution for $K$=k

Figure 3: (a) An illustration of our work to decompose NPGS problem into a sequence of K-budget subgraph detection problems. (b) With the number of normal vertices is equal to $K$, including $v$, we aim to find a solution including more abnormal vertices. We consider assigning the value of $\pi^+_K$ in vertex $V$ as a multiple-choice knapsack problem from $\pi^+$ of children $V_{1,...,h}$. MCK refers to multiple choice knapsack.

Lemma 1 states that when we fix normal vertices in $S$, the optimal $S$ can be obtained by finding the largest number of abnormal vertices for $S$ subject to the constraint that $S$ is connected. As shown in Lemma 2, this problem can be further generalized to the situation with a budget constraint on the cardinality of normal vertices in $S$.

**Lemma 2.** *We denote $\bar{N}_\alpha(S) \equiv N(S) - N_\alpha(S)$ and present the NPGS problem with a budget constraint on the cardinality of normal vertices in S:*

$$\max_{\alpha\in\mathbb{U}(\mathbb{V},\alpha_{max})} \max_{S\subseteq\mathbb{V}:S \text{ is connected}} \phi(\alpha, N_\alpha(S), N(S)), \quad s.t.\ \bar{N}_\alpha(S)\le K \tag{9}$$

*is equivalent to the problem:*

$$\max_{\alpha\in\mathbb{U}(\mathbb{V},\alpha_{max})} \max_{S\subseteq\mathbb{V}:S \text{ is connected}} N_\alpha(S), \quad s.t. \quad \bar{N}_\alpha(S)\le K. \tag{10}$$

*Proof.* Each feasible subgraph $S$ can be decomposed to the subset of normal vertices $S^+$ and the subset of abnormal vertices $S^-$ satisfying the conditions: $N(S^+) \le K$ and $S = S^+ \cup S^-$. According to Lemma 1, for each possible $S^+$, the best subsets $S^-$ for Problem (9) and Problems (10) are identical. It follows that the best subsets $S$ for Problem (9) and Problems (10) are identical as well. □

Based on the above lemmas, we are ready to present a new reformulation of the NPGS problem that can be decomposed to simpler subproblems and efficiently approximated.

**Theorem 1 (NPGS Reformulation).** *The NPGS problem (6) is equivalent to the following problem:*

$$(\hat{\alpha}, \hat{S}) = \max_{\alpha\in\mathbb{U}(\mathbb{V},\alpha_{max})} \max_{S_\alpha\in\{S^0_\alpha,\ldots,S^N_\alpha\}} \phi(\alpha, N_\alpha(S), N(S)), \tag{11}$$

*Given the significant level $\alpha$, each set $S^K_\alpha$ is obtained by solving the following $K$-budget subgraph detection subproblem for $K = 0, \cdots, N$:*

$$S^K_\alpha = \max_{S\subseteq\mathbb{V}:S \text{ is connected}} N_\alpha(S), \quad s.t. \quad \bar{N}_\alpha(S)\le K. \tag{12}$$

*This subproblem can be reduced to the Budget node-weighted Prize Collecting Steiner Tree problem (B-PCST) [25]. Let $\mathbb{T}(\mathbb{G}) \equiv \{\mathcal{T} = (\mathbb{V}_\mathcal{T}, \mathbb{E}_\mathcal{T})\}$ denote the set of sub-trees of $\mathbb{G}$. We define $\pi_\alpha(v) = 1$ and $c_\alpha(v) = 0$ if $p(v) \le \alpha$, otherwise $\pi_\alpha(v) = 0$ and $c_\alpha(v) = 1$.*

$$\mathcal{T}^K_\alpha = \max_{\mathcal{T}\in\mathbb{T}(\mathbb{G})} \sum_{v\in\mathbb{V}_\mathcal{T}} \pi_\alpha(v), \quad s.t. \quad \sum_{v\in\mathbb{V}_\mathcal{T}} c_\alpha(v) \le K, \tag{13}$$

*where for $K = 0, \cdots, N$, each $S^K_\alpha = \mathbb{V}_{\mathcal{T}^K_\alpha}$ and $\mathcal{T}^K_\alpha$ refers to the optimum tree to Problem (13).*

*Proof.* This theorem can be proved by contradiction. Suppose $(\hat{\alpha}, \hat{S})$ is not an optimal solution to the NPGS problem. It follows that there exists a different solution $(\alpha^*, S^*)$, such that $\phi(\alpha^*, N_{\alpha^*}(S^*), N(S^*)) > \phi(\hat{\alpha}, N_{\hat{\alpha}}(\hat{S}), N(\hat{S}))$.

Let $\hat{K} := N(\hat{S}) - N_{\hat{\alpha}}(\hat{S})$ and $K^* := N(S^*) - N_{\alpha^*}(S^*)$. We first observe that $N_{\alpha^*}(\mathbb{V}_{\mathcal{T}^{K^*}_{\alpha^*}}) = N_{\alpha^*}(S^*)$; Otherwise, $\mathbb{V}_{\mathcal{T}^{K^*}_{\alpha^*}}$ will be the optimal subset, instead of $S^*$ due to the properties (P1) and (P2). This result shows that a sub-tree $\mathcal{T}$ derived from $(\alpha^*, S^*)$ must be the solution of Problem (13). Similarly, it can be shown that $N_{\hat{\alpha}}(\mathbb{V}_{\mathcal{T}^{\hat{K}}_{\hat{\alpha}}}) = N_{\hat{\alpha}}(\hat{S})$.

As $(\hat{\alpha}, \hat{S})$ is the optimal solution to the reformulated problem (11), the inequality must be true (i.e., the solution $(\hat{\alpha}, \hat{S})$ is better than all of the tuples $(\alpha, S)$ (except itself) derived from Problem (13)): $\phi(\alpha^*, N_{\alpha^*}(S^*), N(S^*)) \le \phi(\hat{\alpha}, N_{\hat{\alpha}}(\hat{S}), N(\hat{S}))$, a contradiction. Therefore, the initial assumption – $(\hat{\alpha}, \hat{S})$ is not an optimal solution to the NPGS problem – must be false. □

We illustrate our reformulation method in Figure 3. The NPGS reformulation provides two theoretical properties: NP-hardness of the NPGS problem (Theorem 2) and connection to previous work [9] (Lemma 3).

**Theorem 2 (Hardness).** *The NPGS problem (6) is NP-hard for the large class of nonparametric scan statistic functions that satisfy the properties (P1) and (P2).*

*Proof.* As shown in Theorem 1, if we consider the class of nonparametric scan statistic functions satisfying (P1) and (P2), the resulting NPGS problem can be decomposed to a sequence of K-budget subgraph detection subproblems (12). Each K-budget subgraph detection subproblem (12) can be shown to be NP-hard through a reduction from a B-PCST problem (13) in which all vertices have binary prizes and costs of 0 or 1. The NP-hardness of the NPGS problem can then be readily proved. □

**Algorithm 1:** Tree-Shape-Priors Subgraph Detection

```
   Input: Graph $\mathbb{G}(\mathbb{V}, \mathbb{E}, p)$
   Result: The most anomalous subgraph $S^*$
1  Set $\alpha_{max} = 0.15$ and $C = 5$;
2  for $c \in \{1, \cdots, C\}$ do
3      Select seed vertex $v_0$ from $\{v | v \in \mathbb{V}, p(v) \leq \alpha_{max}\}$;
4      Approximate the graph $\mathbb{G}$ as a tree $\mathcal{T}(v_0)$;
5      for $\alpha \in \mathbb{U}(\mathbb{V}, \alpha_{max})$ do
6          for $K = 0, \cdots, \bar{N}_\alpha(\mathbb{V})$ do
7              $S_\alpha^K \leftarrow \text{KBudgetSubTree}(K, \mathcal{T}_{v_0}, \alpha)$;
8          end
9          $S_\alpha = \max_{S \in \left\{S_\alpha^0, \ldots, S_\alpha^{\bar{N}_\alpha(\mathbb{V})}\right\}} \phi(\alpha, N_\alpha(S), N(S))$;
10     end
11     $S^c = \max_{\alpha \in \mathbb{U}(\mathbb{V}, \alpha_{max})} \phi(\alpha, N_\alpha(S_\alpha), N(S_\alpha))$;
12 end
13 Calculate $c^* = \max_c \phi(\alpha, N_\alpha(S^c), N(S^c))$;
14 return $S^{c^*}$
```

For the general graphs, we demonstrate in Lemma 3 that a state-of-the-art algorithm to the NPGS problem is sub-optimal in general cases.

**Lemma 3** (connection to previous work [9])**.** *The greedy algorithm proposed in a recent work [9] always returns a sub-optimal subgraph to the NPGS problem (11) when the optimal subgraph* $S^* \notin \{S_\alpha^0 \mid \alpha \in \mathbb{U}(\mathbb{V}, \alpha_{max})\}$*; and its approximation factor is unbounded in the worst case.*

*Proof.* When $S^* \notin \{S_\alpha^0 \mid \alpha \in \mathbb{U}(\mathbb{V}, \alpha_{max})\}$ as defined in Euqation (12), it means $S^*$ contains at least one normal vertex, which is in violation to the optimality condition of greedy algorithm (See Theorem 2 in [9]) that there is no "break-tire" vertex (e.g., a vertex $v$ with p-value greater than $\alpha$ and whose deletion will break the connectivity of $S^*$). The greedy algorithm will not include normal vertices in the solution, as their inclusion will decrease the objective score locally. Hence, $S^*$ will not be returned by the greedy algorithm as the final solution. Its approximation factor can be proved to be arbitrarily worse when the anomalous subgraph is composed of balanced connected components components of abnormal vertices that are connected via a small number of normal vertices. □

With reduction to the equivalent B-PCST problem (13), there is an $O(\log N)$-approximation solution by applying a polynomial-time approximation algorithm [25] of the NP-hard B-PCST problem. Detecting the optimal anomalous connected subgraph in general graphs is still difficult for the optimization methods with theoretical properties is hard to incorporate the structure of general graphs. We will present our method with nice theoretical properties by tree shaped priors in next subsections.

### 3.2 Approximations for graphs with tree shaped priors

In the preceding subsections, we have discussed the NP-hardness of the NPGS problem. However, for the general graphs, both the Big-O approximation factor and the polynomial time complexity of this approximation are not satisfactory for large graph analysis.

To design more efficient solutions to the subproblem (13), we propose to reformulate the connectivity constraint of the subgraph $S$ on a fixed topology (e.g., tree). Particularly, we approximate the graph $\mathbb{G}$ as a tree $\mathcal{T}_v$ originating at a given root vertex $v \in \mathbb{V}$, and the search of the best connected subgraph $S$ for the NPGS problem is approximated as the search of the best sub-tree in $\mathcal{T}_v$. There are several heuristics to find the tree for the input graph: (1) breadth-first tree; (2) random spanning tree; (3) steiner tree; and (4) geodesic shortest path tree. The first three tree heuristics have been successfully applied to discrepancy maximization on general graphs [26]. The fourth tree heuristic has been successfully applied to image segmentation and sensor networks [27].

**Breadth-First Tree** (BFS-Tree): A very simple way to obtain a tree for a given graph is to perform breadth-first search from the root vertex $v$. The BFS-Tree heuristic follows exactly this strategy. It selects a random set of candidate root vertices and generates a breadth-first tree for each candidate root vertex. It then computes the best sub-tree for each subproblem (12) and returns the best solution.

**Random Spanning Tree** (Random-ST): Instead of computing BFS from each candidate root vertex, we can work with a *random* tree that spans all vertices. We sample such a random tree by assigning a random weight (uniformly from [0, 1]) to every edge, and computing the minimum weight spanning tree. The Random-ST heuristic works by computing a number of such random spanning trees, computing the best sub-tree to each subproblem (13), and returning the best solution found.

**Steiner Tree** (Steiner-T): The previous two heuristics do not consider the properties (P1) and (P2) of the NPGS problem. Intuitively, a tree is good if it interconnects abnormal vertices with the least number of normal vertices. If we denote each abnormal vertex as a *terminal* vertex, and each normal vertex as a *steiner* vertex, this tree can be identified by generating the steiner tree of the input graph. The Steiner-T heuristic computes the steiner tree for each $\alpha \in \mathbb{U}(\mathbb{V}, \alpha_{max})$, computes the best sub-tree to each subproblem (13), and returns the best solution found.

**Geodesic Shortest Path tree** (Geodesic-SPT): The Geodesic-SPT heuristic allows to use a domain depending local geodesic metric and additionally to incorporate a-prior knowledge about the geometry of the subgraph of interest [27]. For the NPGS problem, we define the optimal cost (local geodesic metric) of the connecting path $\mathbf{p}$ between a fixed vertex $s$ and any vertex $x$ in the subgraph based on its nonparametric scan statistic: $\exp\{-\max_\alpha \phi(\alpha, N_\alpha(S_\mathbf{p}), N(S_\mathbf{p}))\}$, where $S_\mathbf{p}$ refers to the set of vertices in $\mathbf{p}$. Given the geodesic metric, the shortest path tree can be computed via dynamic algorithms [28].

Algorithm 1 presents the approximation algorithm to the NPGS problem based on the tree shape priors. In Step 1, $C$ refers to the number of seed root vertices ($C$ = 5 by default). Step 4 approximates the input graph $\mathbb{G}$ as a tree $\mathcal{T}(v_0)$ using one of the above four heuristics. Step 7 applies the dynamic algorithms (Section 3.3) to calculate the solution $S_\alpha^K$ to the $K$-budget subgraph detection problem (13) in the tree $\mathcal{T}(v_0)$.

```
Algorithm 2: KBudgetSubTree
  Input: Integer $K$, tree $\mathcal{T}$, and significance level $\alpha$
  Result: Optimal sub-tree to the $K$-budget subgraph
          detection problem (13)
1 Call DP($\mathcal{T}, K$) to update $\pi_l^{-v}$, $\pi_l^{+v}$, $\pi_l^v$ $s_l^v$, $n_l^v$, and $\mathcal{C}_l^v$
    for each vertex $v$ in $\mathcal{T}$ and $l = 0, \cdots, K$;
2 $v \leftarrow$ the root vertex of $\mathcal{T}$;
3 while $s_K^v = False$ do
4     $v = n_K^v$;
5 end
6 $S = GetSubTree(\mathcal{T}, v, K)$;
7 return $S$
8 Procedure GetSubTree($\mathcal{T}, v, l$)
9     $S = \phi$ ;
10    for $(v_{child}, l)$ in $C_l^v$ do
11        $S = S \cup GetSubTree(\mathcal{T}(v_{child}), v_{child}, l)$ ;
12        $S = S \cup \{v_{child}\}$ ;
13    end
14    return $S$
```

### 3.3 Dynamic algorithms for the K-Budget Subgraph Detection Subproblem (13)

When the input graph $\mathbb{G}$ is a tree $\mathcal{T}(v)$ with the root vertex $v$, we can solve the subproblem (13) optimally, using dynamic programming (DP). We first introduce a few notations:

- $\mathcal{T}(v)$: a sub-tree of $\mathbb{G}$ with the root vertex $v$.
- $\pi_l^{-v}$: the value of the best $l$-budget sub-tree to the subproblem (13) in $\mathcal{T}(v)$ that does not contain $v$.
- $\pi_l^{+v}$: the value of the best $l$-budget sub-tree to the subproblem (13) in $\mathcal{T}(v)$ that contains $v$.
- $\pi_l^v$: $\pi_l^v = \max\{\pi_l^{-v}, \pi_l^{+v}\}$.
- $s_l^v$: a boolean value that indicates if vertex $v$ belongs to the best $l$-budget sub-tree in $\mathcal{T}(v)$.
- $n_l^v$: a vertex pointer that indicates to which child of $v$ to find the best $l$-budget sub-tree, if $s_l^v = False$.
- $\mathcal{C}_l^v$: a set of tuples of the form $(v', t)$. Hereby, $v'$ is a child of $v$ and $t$ is an integer number that denotes the size of the sub-tree to be collected in $\mathcal{T}(v')$.
- $\mathcal{C}(v)$: the set of children of $v$ in $\mathcal{T}(v)$.

Algorithm 2 is the overall algorithm for the subproblem (13). Step 1 calls the dynamic programming (DP) procedure (Algorithm 3) to update the attributes for each vertex in $\mathcal{T}$. Steps 2 to 5 retrieve the root vertex of the optimal sub-tree. Step 6 calls the procedure $GetSubTree$ to retrieve the set $S$ of vertices in the optimal sub-tree. The DP procedure is described in Algorithm 3. It calculates the attributes $\{\pi_l^{-v}, \pi_l^{+v}, \pi_l^v, s_l^v, n_l^v, C_l^v\}_{l=0}^K$ for each vertex $v$. The vertices are processed from bottom to top, such that when we start to process a vertex $v$, the attributes of its child vertices have already been calculated. Specifically, Steps 2 to 11 in Algorithm 3 set initial values to the attributes of leaf vertices. The status variable $b(v)$ is a 0-1 value that indicates if the attributes of the vertex $v$ have been calculated. Steps 13 to 19 in Algorithm 3 update the attributes of a selected vertex $v$, in which its status variable $b(v)$ is 0 and status variables of its child vertices are all 1s.

```
Algorithm 3: Dynamic Programming (DP)
  Input: Tree $\mathcal{T}$ and integer $K$
  Result: Tree $\mathcal{T}$ with the updated attributes
          $\{\pi_l^{-v}, \pi_l^{+v}, \pi_l^v, s_l^v, n_l^v, C_l^v\}_{l=0}^K$ at each vertex $v$
1 $b(v) = 0, \forall v \in \mathcal{T}$;
2 for each leaf vertex $v$ of $\mathcal{T}$ do
3     $\pi_0^{-v} = 0$;
4     $l = \delta(p(v) > \alpha)$;
5     if $l$ = 1 then
6         $\pi_l^{-v} = 0$; $\pi_l^{+v} = 0$; $\pi_l^v = 0$;
7         $\pi_0^{+v} = 0$; $\pi_0^v = 0$;
8     else
9         $\pi_0^{+v} = 1$; $\pi_0^v = 1$;
10    end
11    $b(v) = 1$;
12 end
13 while $\exists v \in \mathcal{T}, \forall v_{child} \in \mathcal{C}(v), b(v) = 0, b(v_{child}) = 1$
     do
14    Update $\{\pi_l^{-v}, \pi_l^{+v}, n_l^v, \mathcal{C}_l^v\}_{l=0}^K$ via Equations (14),
        (15), and (18);
15    if $\pi_l^{-v} > \pi_l^{+v}$ then
16        $\pi_l^v = \pi_l^{-v}$; $s_l^v = False$;
17    else
18        $\pi_l^v = \pi_l^{+v}$; $s_l^v = True$;
19    end
20 end
21 return Tree $\mathcal{T}$
```

The attributes $n_l^v$ and $\pi_l^{-v}$ are computed as follows:

$$n_l^v = \arg\max_{v_i}\{\pi_l^{v_1}, \cdots, \pi_l^{v_h}\},\ \pi_l^{-v} = \pi_l^{n_l^v}, \tag{14}$$

where $\{v_1, \cdots, v_h\}$ refer to the $h$ child vertices of the vertex $v$. As illustrated in Figure 3(b), the computation of the attribute $\pi_l^{+v}$ can be reduced to a 0-1 multiple-choice knapsack (0-1 MCK) problem [29]: Given $h$ classes $\mathcal{Z}_1, \cdots, \mathcal{Z}_h$ of items to pack in a knapsack of capacity $(l - \delta(p(v) > \alpha))$, where $\mathcal{Z}_i = \{1, \cdots, K\}$. Each item $j \in \mathcal{Z}_i$ has a profit $\pi_j^{+v_i}$ and a budget $j$, and the problem is to choose at most one item from each class such that the profit sum is maximized without having the sum of budget to exceed $j$. The attribute $\pi_l^{+v}$ can then be calculated as:

$$\pi_l^{+v} = \max_{\mathbf{x}} \delta(p(v) \leq \alpha) + \sum_{i=1}^{h}\sum_{j=0}^{K} \pi_j^{+v_i} \cdot \mathbf{x}_{i,j} \tag{15}$$

subject to

$$\sum_{i=1}^{h}\sum_{j=0}^{K} j \cdot \mathbf{x}_{i,j} \quad \leq \quad l - \delta(p(v) > \alpha), \tag{16}$$

$$\sum_{j=0}^{K} \mathbf{x}_{i,j} \quad \leq \quad 1,\ i = 1, \cdots, h. \tag{17}$$

where $\mathbf{x} \in \{0, 1\}^{h \times K}$. The delta function $\delta(\cdot) = 1$ if its input is True, otherwise $\delta(\cdot) = 0$. Given the result $\mathbf{x}$ from the above problem (15), the set attribute $C_l^v$ can be calculated as:

$$\mathcal{C}_l^v = \{(v_i, j) | \mathbf{x}_{i,j} = 1\}. \tag{18}$$

The optimal solution to the 0-1 MCK problem (15) can be obtained via dynamic programming in time $O(K^2h^2)$ [30]. There is an approximation algorithm to this problem that has the approximation factor $(1+\epsilon)$ and the running time $O(Kh^2/\epsilon)$ [29].

**Theorem 3.** *1)* ***Exact Solution****: If each 0-1 MCK subproblem (15) is solved via dynamic programming [30], Algorithm 1 is guaranteed to find the optimal solution to the tree-priors-based NPGS problem with the time complexity $O(|\mathbb{U}(\mathbb{V}, \alpha_{max})| \cdot N^4)$; 2)* ***Approximate Solution****: If each 0-1 MCK subproblem (15) is solved via the approximation algorithm [29], then Algorithm 1 is guaranteed to find an approximate solution to the tree-priors-based NPGS problem with the approximation factor $(1+\epsilon)^L$, where $L$ refers to the depth of the sub-tree of $\mathbb{G}$, and the time complexity is $O(|\mathbb{U}(\mathbb{V}, \alpha_{max})| \cdot N^3/\epsilon)$.*

*Proof.* The processing of 0-1 MCK subproblems is a dominant component of Algorithm 1 in running time. For a vertex $v$ and its $h$ child vertices, we are given $h$ classes $\mathcal{Z}_1, \cdots, \mathcal{Z}_h$ of items to pack in a knapsack of capacity $(l - \delta(p(v) > \alpha))$, where $\mathcal{Z}_i = \{1, \cdots, K\}$, with the time $O(K^2h^2)$ for exact solution via dynamic programming [30] and the time $O(Kh^2/\epsilon)$ for approximate solution [29]. The total time costs to process all the vertices are hence $O(K^2N^2)$ and $O(KN^2/\epsilon)$ for calculating the exact and approximate solutions for the sub-procedure Algorithm 2 (KBudgetSubTree), respectively. KBudgetSubTree will be called $O(|\mathbb{U}(\mathbb{V}, \alpha_{max})|)$ times. Therefore, the total running times of calculating exact and approximation solutions are $O(|\mathbb{U}(\mathbb{V}, \alpha_{max})|N^4)$ and $O(|\mathbb{U}(\mathbb{V}, \alpha_{max})|N^3/\epsilon)$, respectively. Furthermore, we note that $|\mathbb{U}(\mathbb{V}, \alpha_{max})|$ can be considered as a constant as justified in [9], and thus the algorithm scales as $O(N^4)$ and $O(N^3/\epsilon)$, respectively. We can induce one-level tree $\mathcal{T}^*$ (i.e., the tree just contains one root node and the leaf nodes) from the optimal subtree. Similarly, we can induce one-level tree $\hat{\mathcal{T}}$ from the detected tree $S$ in Algorithm 2, where $S$ has at most $L$ levels and $L$ refers to the depth of the sub-tree of $\mathbb{G}$. Now we compare $\hat{\mathcal{T}}$ and $\mathcal{T}^*$, and it is a typical 0-1 MCK problem. For $\hat{\mathcal{T}}$, its leaf nodes are $(1+\epsilon)^{L-1}$ approximations for there are exact $(L-1)$-level nested 0-1 MCK problems. Thus $\hat{\mathcal{T}}$ is approximated to $\mathcal{T}^*$ with the the approximation factor $(1+\epsilon)^L$. We return the best solution from the $K|\mathbb{U}(\mathbb{V}, \alpha_{max})|$ solutions in Algorithm 1. Thus we prove the approximate solution to the tree-priors-based NPGS problem with the approximation factor $(1+\epsilon)^L$. □

The depth of the detected tree $S$ is usually less than the depth of the sub-tree of $\mathbb{G}$ for the size of $S$ is small, and so the approximation factor $(1+\epsilon)^L$ is a relaxed version.

### 3.4 Optimization

Algorithm 1 proceeds with a sequence of calls to the K-budget subtree detection algorithm (Algorithm 2) to address the subproblems (13) for different combinations of $\alpha$ and $K$. This algorithm can be further improved via the following optimization strategies:

First, instead of $\bar{N}_\alpha(\mathbb{V})$ calls to Algorithm 1, it suffices to call Algorithm 3 only once with $K = \bar{N}_\alpha(\mathbb{V})$, and the returned Tree $\mathcal{T}$ with the updated attributes $\{\pi_l^{-v}, \pi_l^{+v}, \pi_l^v, s_l^v, n_l^v, C_l^v\}_{l=0}^K$ at each vertex $v$ can be used to retrieve the sub-trees to the K-budget subgraph detection subproblems (13) for $K = 0, \cdots, \bar{N}_\alpha(\mathbb{V})$.

Second, in Algorithm 3, after the attributes of the vertex $v$ are calculated: $\{\pi_l^v, \pi_l^{-v}, \pi_l^{+v}, n_l^v, \mathcal{C}_l^v\}_{l=0}^K$ in Steps 14 to 19, we check $\pi_l^v$ based on the order $l = K, \cdots, 1$. The attributes $\{\pi_l^v, \pi_l^{-v}, \pi_l^{+v}, n_l^v, \mathcal{C}_l^v\}$ related to the $l$-budget solution can be safely removed, if at least one of the following conditions is satisfied: 1) $\pi_l^{+v} \leq \pi_{l-1}^{+v}$; 2) $\phi(\alpha, a+\pi_l^v, a+l+\pi_l^v) \leq \phi(\alpha, a+\pi_{l-1}^v, a+\pi_{l-1}^v+l-1)$; and 3) $\phi(\alpha, a+\pi_l^v, a+\pi_l^v+l) \leq \phi(\alpha, a, a)$, where $a = N_\alpha(\mathbb{V}_\mathcal{T}) - N_\alpha(\mathbb{V}_{\mathcal{T}(v)})$.

Third, denote $\mathbb{U}(\mathbb{V}, \alpha_{max}) = \{\alpha_1, \alpha_2, \cdots, \alpha_Z\}$, and assume that the $\alpha$ values are processed in the order based on the index. Suppose the current $\alpha$ is $\alpha_i$. In Algorithm 3, we maintain an additional attribute $q$ in the root $r$ that refers to an upper-bound of the number of abnormal vertices in the optimal subtree. Based on $q$, we can calculate an upperbound of the best subtree as follows: $\phi(\alpha_i, q, q)$. In the beginning, $q = N_{\alpha_i}(\mathbb{V})$. When the attributes of a vertex $v$ are calculated in Steps 14 to 20, we apply the above optimization strategy to remove unnecessary $l$-budget subtrees rooted at $v$. Suppose $\mathcal{L} = \{l^1, \cdots, l^h\}$ refers to the set of $l$-values that have been pruned. Then $q$ can be updated as follows: $q = q - (\max_{l \in \{0, \cdots, \bar{N}_{\alpha_i}(\mathbb{V})\}}\{\pi_l^v\} N_\alpha(\mathbb{V}_{\mathcal{T}(v)}) - \max_l\{\pi_l^{+v}\})$. When each time $q$ is updated, we compare the resulting upper bound $\phi(\alpha_i, q, q)$ with the best score $F_i = \max_{j \in \{1, \cdots i-1\}} \phi(\alpha_j, N_{\alpha_j}(S_{\alpha_j}), N(S_{\alpha_j}))$ calculated based on previous alpha values $\alpha_1, \cdots, \alpha_{i-1}$: If $\phi(\alpha_i, q, q) \leq F_i$, then we do not need to proceed the procedure related to $\alpha_i$.

## 4 Experiments

We evaluate the effectiveness and efficiency of our work in comparison to representative competitive methods on four real-datasets. In case studies, our findings reveal interesting applications of our method to cyber-attack detection, congested road network detection, and haze events detection.

### 4.1 Experiment Design

**Datasets:** 1) **Water Pollution Dataset.** The "Battle of the Water Sensor Networks" (BWSN) provides a real-world network of 12,527 nodes, and 25 nodes with chemical contaminant plumes that are distributed in four different areas. The spreads of these contaminant plumes on graph were simulated using the water network simulator EPANET that was used in BWSN for a period of 8 hours. Each node has a sensor that reports 1 if it is polluted; otherwise, reports 0. We randomly selected $K$ percent vertices, and flip their sensor binary values, where $K = 0, 4, 8, 10, 20, 30$, in order to test the robustness of subgraph detection methods to noises.

**2) Event Detection Dataset:** We collected 1,433,937,815 tweets (nearly 10 percent of the whole Weibo[1] data) from April 11, 2014 to January 11, 2015 (9 months). From this dataset, we selected 0.35 million Weibo tweets, which are relevant to the haze air pollution and posted by 51,940 users. According to mentions in tweets and following relations, we construct a connected user network with 158,652 edges.

[1] Weibo.com, the most popular online social networking services in China with more than 400 million users.

**3) Gov-Site Network Traffic Dataset:** An Internet security company[2] provided us with 4,270,483 logs of "***.gov.cn**" web sites browsing traffic from April 23, 2015 to May 13, 2015. We derived a network with 31,241 nodes (i.e., web sites or client IP addresses) and 59,357 edges from the traffic. For each node, we counted the number of visitation per day.

**4) Beijing-Road Network Traffic Dataset:** We obtained 0.6 billion GPS records from 12,736 taxis in Beijing, China for the whole November month, 2010, where each record consisted of the *location* and *speed*. In this paper, we focus on the main urban area in a rectangle region in Beijing where its lower left latitude and longitude are $39.77° N$ and $116.19° E$, and its upper right latitude and longitude are $40.02° N$ and $116.54° E$ respectively. We extracted the road network with 30,157 nodes and 107,720 edges in this main urban area (i.e., a "*node*" denotes a road (e.g., Xueyuan Road), and an "*edge*" denotes a cross between two roads).

**Data Preprocessing:** 1) For water pollution raw data, first we generated a connected graph of sensors by its GPS sites. We use K-nearest neighbor algorithm to generate the minimal number of edges connecting each sensor. Second, convert the sensor data to real numbers and compute the p-value for each sensor at each hour. Third, in practical setting, values of a sensor can be deviated from the true value, and thus we add noise in the sensor data. For noise level 0.02, we randomly select 0.02 times of sensors and set the new p-value of the sensor equal to 1 subtracting original p-value.

2) For haze outbreak raw data consisting of Weibo data and haze warnings issued by the state meteorological bureau, we present the preprocessing steps. a) **Vocabulary Generation:** 50 terms related to haze from domain experts; b) **Content Filtering:** we only preserve the raw tweets that match more than two terms from the vocabulary and corresponding user has location information in user profile; c) **User Geocoding:** we search for location information from the users profile. d) **User Graph:** we generate the graph based on the mentions in tweets; e) **P-value:** for day $d$, user $u$ and word $w$, we derive the frequency of $w$ in $u$ tweets at $d$, and compute the p-value for $w$ [9]. The p-value for $u$ at $d$ is the average of p values of words that is reasonable as a strong signal of haze outbreak issued with u. f) **Haze Warning:** we recorded 4279 formal haze eventsrecords (level $\geq$ 3) from the official website[3], and aggregated the records as ("Time(YYYYMMDD)", "Location(Province)").

3) For the "**.gov.cn*" site browsing data, first we generated a network by the visiting logs (i.e., an edge between $u \in \mathbb{V}$ and $v \in \mathbb{V}$ if $u$ visits $v$ or $u$ is visited by $v$). Second, for each site $v \in \mathbb{V}$, for each day $d$ in the period $T$, we computed the number of visitation activities $count_d(v) \in \mathbb{R}$. Last, for each site $v$, on the specific day $d$, its p-value $p_d(v)$ is $\sum_{t \in T, t<d} \delta(count_t(v) > count_d(v)) \ / \ \sum_{t \in T} \delta(t < d)$. Each p-value refers to the historical data.

4) For the road traffic data, we considered each road as a "*node*" and each cross as "*edges*" between roads, where each road has a tuple of GPS sites. Each taxi would report its GPS site and speed every one minute. We identified the closest road for the taxi GPS site as the road on which the taxi ran. For each road, We averaged the speeds as its speed per hour. For each road $v$, on the specific day $d$ and hour $h$, its p-value $p_d^h(v)$ is $\sum_{t \in T, t<d} \delta(speed_t^h(v) < speed_d^h(v)) \ / \ \sum_{t \in T} \delta(t < d)$. We aimed to identify the roads with smaller speeds rather than we focused on the sites with higher visitations.

**Comparison Methods:** The four existing representative anomalous subgraph detection methods are Event Tree [22], Non-Parametric Heterogeneous Graph Scan (NPHGS) [9], Linear Time Subset Scan (LTSS) [10], and Graph Laplacian Regularization (Graph-LR) [4]. Implementations of NPHGS and LTSS were obtained from the authors. EventTree and Graph-LR were replicated under the authors' instructions in their papers. Specifically, for EventTree, the authors reformulated the subgraph detection problem as unrooted prize-collecting Steiner tree problem and directly applied none-root version Goemans-Williamson (G-W) algorithm [22] to detect anomalous subgraphs. We implemented the G-W algorithm. The Graph-LR was formulated as a convex optimization problem, and we directly applied the optimization toolbox CVXOPT to implement this algorithm. We strictly followed the strategies recommended by the authors in their papers to tune the related parameters. Specifically, for EventTree and Graph-LR, we tested the set of $\lambda$ values: $\{0.1, 0.2, \cdots, 1.0, 50, 100, \cdots, 1500\}$. As EventTree requires edge weights, we define the weight of an edge in the water pipepline network as the length of the pipeline segment; and define the weight of an edge in the user-user network of the Weibo dataset as 1, for no better way to define edge weights in the networks. Two nonparametric scan statistics BJ and HC were evaluated. The parameter $\alpha_{max}$ was set to 0.15 for NPHGS and our methods. The number of seed nodes in NPHGS was set to 5 as used in the original paper, and the authors demonstrated that the setting of this parameter is not sensitive. We used 10-fold cross validation to identify the best combination of all the related parameters.

**Our Methods:** In this work, we designed a dynamic-programming algorithm to the NPGS problem with tree-shape priors (Algorithm 1). There are two versions of Algorithm 1, exact and approximate algorithms, depending on the exact or approximate solution of 0-1 MCK subproblems. In the experiments, we focus on the approximate version of Algorithm 1 due to its high scalability, which means we applied the approximation algorithm [29] to solve the 0-1 MCK subproblems. We denote this algorithm as Tree-Shape-Priors Subgraph Detection (TSPSD).

**Performance Metrics:** This work mainly employs four metrics to evaluate the performance of methods. 1) precision, 2) recall. These two metrics examine the true performance of methods in data as the noise level can be controlled accurately. 3) false positive rate (FPR), 4) true positive rate (TPR). These two metrics can be used to identify which region our method performs better than other methods and which region our method performs worse than others.

We denote $\Gamma(\mathcal{G}), \mathcal{G} \subseteq \mathbb{G}$ as the set of vertices in the subgraph $\mathcal{G}$. For a graph , the truly anomalous subgraph is $\mathcal{G}_0 \subseteq \mathbb{G}$, and for a method, the returned subgraph is $\mathcal{G} \subseteq \mathbb{G}$. Then the *precision* and *recall* are defined as follows:

$$precision = \frac{|\Gamma(\mathcal{G}) \cap \Gamma(\mathcal{G}_0)|}{|\Gamma(\mathcal{G})|} \quad recall = \frac{|\Gamma(\mathcal{G}) \cap \Gamma(\mathcal{G}_0)|}{|\Gamma(\mathcal{G}_0)|}$$

For the *event detection dataset*, we derived the gold standard

[2] An Internet security company in China with more than 0.6 billion users.

[3] http://datacenter.mep.gov.cn/

| Method | Noise Ratio (0%) | 4% | 8% | 10% | 30% |
|---|---|---|---|---|---|
| BFS-Tree (BJ) | 0.94, 0.48 (0.64) | 0.95, 0.47 (0.63) | 0.93, 0.50 (0.66) | 0.91, 0.47 (0.62) | 0.78, 0.33 (0.47) |
| Random-ST (BJ) | 0.94, 0.77 (0.84) | 0.93, 0.75 (0.83) | 0.95, 0.65 (0.77) | 0.93, 0.59 (0.71) | 0.79, 0.39 (0.53) |
| Steiner-T (BJ) | 1.00, 0.99 (**1.00**) | 0.98, 0.96 (**0.97**) | 0.95, 0.92 (**0.94**) | 0.94, 0.89 (**0.91**) | 0.77, 0.52 (0.62) |
| Geodesic-SPT (BJ) | 0.96, 0.85 (0.90) | 0.92, 0.63 (0.75) | 0.88, 0.65 (0.75) | 0.85, 0.56 (0.68) | 0.78, 0.38 (0.51) |
| EventTree | 0.97, 1.00 (0.98) | 0.89, 0.98 (0.93) | 0.70, 0.98 (0.82) | 0.42, 0.97 (0.59) | 0.09, 0.90 (0.17) |
| NPHGS (BJ) | 1.00, 0.92 (0.96) | 0.99, 0.77 (0.84) | 0.97, 0.50 (0.66) | 0.97, 0.39 (0.55) | 0.78, 0.06 (0.11) |
| LTSS (BJ) | 1.00, 1.00 (**1.00**) | 0.48, 0.96 (0.64) | 0.34, 0.92 (0.50) | 0.30, 0.90 (0.45) | 0.11, 0.70 (0.20) |
| Graph-LR | 0.93, 0.87 (0.90) | 0.95, 0.43 (0.60) | 0.89, 0.23 (0.37) | 0.68, 0.12 (0.20) | 0.97, 0.50 (**0.66**) |

Table 1: Comparison w.r.t. different noise levels in the water pollution dataset: Precision, Recall (F-Measure). The $\alpha_{max}$ is set to $0.15$, and the budget $K$ is set to 30.

| Method | FPR | TPR | TPR | Lead Time | Lag Time | Run Time |
|---|---|---|---|---|---|---|
| | (FP/Day) | (Detection) | (Forecast & Detect) | (Days) | (Days) | (Minutes) |
| TSPSD-Steiner HC (BJ) | 0.100 | **0.55** (0.49) | **0.66** (**0.66**) | **0.98** (0.97) | **3.53** (3.54) | 18 (0.3) (18 (0.3)) |
| TSPSD-Steiner HC (BJ) | 0.150 | **0.62** (0.61) | 0.70 (**0.71**) | **0.88** (0.82) | **3.92** (4.15) | 18 (0.3) (18 (0.3)) |
| TSPSD-Steiner HC (BJ) | 0.200 | **0.66** (**0.66**) | **0.74** (**0.74**) | **0.87** (0.82) | **4.00** (4.15) | 18 (0.3) (18 (0.3)) |
| NPHGS HC (BJ) | 0.100 | 0.32 (0.41) | 0.47 (0.55) | 0.72 (0.59) | 4.35 (4.70) | 3 (8) |
| NPHGS HC (BJ) | 0.150 | 0.43 (0.48) | 0.60 (**0.71**) | 0.72 (0.70) | 4.27 (4.40) | 3 (8) |
| NPHGS HC (BJ) | 0.200 | 0.50 (0.63) | 0.70 (**0.74**) | 0.71 (0.74) | 4.32 (4.12) | 3 (8) |
| EventTree | 0.100 | 0.51 | 0.65 | 0.91 | 3.71 | 7.5 |
| EventTree | 0.150 | 0.57 | 0.68 | 0.70 | 4.40 | 7.5 |
| EventTree | 0.200 | 0.60 | 0.72 | 0.81 | 4.12 | 7.5 |

Table 2: Comparison between TSPSD and Other Models on the Haze outbreak dataset. The scores of HC and BJ statistics are shown in the format: $x(y)$, where $x$ refers the score of HC, and $y$ refers to that of BJ. For 18(0.3), 18 is the overall run time and 0.3 is the detection time. The $\alpha_{max}$ is set to $0.15$, and the budget $K$ is set to 30. The value of $\alpha_{max}$ ensures that the vertices whose p-values are less than $\alpha_{max}$ are abnormal vertices. The compact $\alpha_{max}$ will lead to a high score $\varphi$.

haze event from Chinese Meteorological Bureau reports, which are structured as tuples of (*date*, *location*), where *location* is defined at the province level. For each gold standard event, we decide whether the method: 1) Had an alert in the province within 7 days before the event, which is considered to be "successfully predicted"; 2) Did not have an alert in that province with 7 days before the event, but did have an alert in that province within 7 days after the event, which is considered to be "successfully detected"; or 3) Did not trigger an alert in that province within 7 days before and after the event, which is considered to be "undetected".

### 4.2 Results: Subgraph Detection

Table 1 presents the comparison between the proposed TSPSD approach and four representative methods for the task of detecting subgraph. In this table, all measurements were averaged over the results of the water pollution dataset. We evaluates TSPSD and the four baseline methods with precision, recall and F-score metrics. At noise level 0%, 4%, 8% and 10%, TSPSD with Steiner-T prior achieved the highest F-score in detecting the contaminated water region. Even if we introduced 10% noise into the dataset, TSPSD detected 89% truly contaminated water region with the precision greater than 90%. At noise level 30%, The value of precision, recall and F-score of TSPSD with Steiner-T was comparable to the Graph-LR method but slightly lower. From the overall performance in all different noise level, TSPSD with Steiner-T performs more stable than the Graph-LR method. In other hands, mostly F-scores of TSPSD are higher than the four methods, and F-score considers both the precision and the recall to evaluate a method. TSPSD with the four tree shape priors under noise level 30% has a higher F-score than the baselines EventTree, NPHGS and LTSS. We compare TSPSD with each other TSPSD by different versions tree shape priors and nonparametric statistics BJ and HC, and find that TSPSD achieves best by Steiner-T.

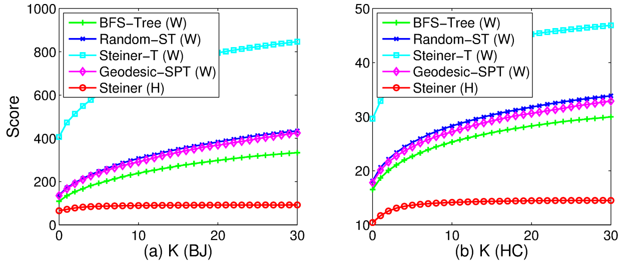


Figure 4: The average nonparametric scan statistic scores for each problem $S_{\alpha}^{K}$ in (13). (a) shows the BJ score for each tree prior in Water Pollution Dataset (W) and Haze Event Detection Dataset (H); and (b) shows the HC score.

### 4.3 Results: Event Detection

For comparable false positive rates, TSPSD achieved the highest forecasting TPR and detection TPR than the two baseline methods in Table 2. The lead time represents how long we need to predict Haze event before it actually occurs. Our method predicting haze events is earlier than baselines, and that means the larger lead time. Haze events as natural events occur usually without exceeding a half day in China. However social events (e.g., protest events), often have trigger subevents and are driven by public sentiments, and can be potentially forecasted with a large lead time (e.g., 1 to 2 weeks). It is difficult to predict Haze events before a long time for Haze events do not have these factors. For the lag time, we use the less time to detect Haze events, and that means the less lag time. Our approach performs better than baselines. Although the run time of TSPSD was little higher

| Run Time<br>(Minutes) | BFS<br>Tree | Random<br>ST | Steiner<br>Tree | Geo<br>SPT |
|---|---|---|---|---|
| OPT | 0.11 (0.11) | 0.93 (0.10) | 0.77 (0.08) | 0.62 (0.11) |
| NON-OPT | 3.16 (3.15) | 3.89 (2.81) | 2.89 (2.01) | 3.79 (3.00) |
| RunTime<br>(Minutes) | EventTree | NPHGS | LTSS | Graph<br>Laplacian |
| | 13.10 | 1.82 | 0.93 | 24.61 |

Table 3: The average run times of our proposed and baseline methods on the Water pollution dataset. The run time of our proposed method consists of two parts such as 0.93 (0.08), where 0.93 is the overall run time (including tree generation and subgraph detection) and 0.08 is the run time of the subgraph detection step. Our proposed method has two versions: 1) NON-OPT: Algorithm 1 (tree-shape-priors subgraph detection); 2) OPT: Algorithm 1 + optimizations (Subsection 3.4). We implemented both BJ and HC statistics, and their run times are equal.

than baseline methods, the time of tree generation consumes major time in overall time.

### 4.4 Parameter Tuning

For examining the sensitivity of selecting values of $K$, we plot each score $F(S_\alpha^K)$ for $K = 0, \cdots, 30$ in Figure 4 (a) and (b). We can observe that $F(S_\alpha^K)$ is stable after $K = 20$. From the scores, we can see that our approaches TSPSD-Steiner-T HC(BJ) perform best. In the Haze data set, the fewer connected users triggering Haze warnings led to the less score for BJ and HC. The results in Figure 4 (a) and (b) show that most of abnormal vertices are connected from each other with a small number of normal vertices. From Figure 2, we can observe that the different $\alpha$ values lead to the different scores $\varphi$. From Problem (11), our approaches are examined in each significant level $\alpha$ (i.e., $\alpha \leq \alpha_{max}$). We select a compact $\alpha_{max}$ to ensure that the p-values of abnormal vertices are less than $\alpha_{max}$. In the experiments, the budget $K$ and $\alpha_{max}$ are set to 30 and 0.15 respectively.

### 4.5 Runtime: Tree Shape Priors

Our proposed methods based on the four tree shape priors are compared to the four baseline methods by run time, with results shown in Table 3. All measurements were averaged over the run time of the Water Pollution Dataset. Run time of TSPSD was comparable to other methods but slightly higher in Random-St and Steiner-Tree with NON-OPT because it recomputes $\pi_l^{-v}, \pi_l^{+v}, n_l^v, \mathcal{C}_l^v$ in Algorithm 3 redundantly. When we apply optimizations (Subsection 3.4), the run time is less than NPHGS and LTSS methods. We note that the speed of detecting subgraph by TSPSD with OPT is faster 25 times than TSPSD without it. TSPSD with OPT (Subsection 3.4) performs better than all the four baseline methods.

### 4.6 Case Study in Cyber Traffic Networks

We took on the consecutive two days, May 5 and 6, 2015 in the gov-site network traffic dataset to demonstrate the performance of our method. The results for the methods TSPSD-steiner BJ (HC) are similar. We illustrated the detected cyber attack networks with major differences for TSPSD-steiner BJ (HC) in Figures 5 and 6. We just show the result by TSPSD-steiner BJ in Figure 7 on May 6, 2015.

On May 5, 2015, for TSPSD-steiner BJ (HC), the attacked sites *"www.saic.gov.cn"*, *"www.audit.gov.cn"*, *"bbs.xyw.gov.cn"*, *"www.jgjy.gov.cn"* and *"news.xyw.gov.cn"* were detected. We also discovered the major attacking sources *"X.X.171.42"*, *"X.X.148.207"* and *"X.X.42.50"*. The site *"www.saic.gov.cn"* was attacked by the two major types of actions, *Dedecms Attack* (e.g., from the attacking source *"X.X.148.207"*) and *scanner actions*. The site *"www.audit.gov.cn"* was attacked by three sources with *Dedecms Attack*, *Get SQL Inject*, and *Upload Webshell Attack*. The site *"bbs.xyw.gov.cn"* was attacked by the source *"X.X.42.50"* with *Dedecms Attack* and *Get SQL Inject*. The sites *"www.jgjy.gov.cn"* and *"news.xyw.gov.cn"* were attacked by the source *"X.X.42.50"* with the same actions. Especially, the site *"xyw.gov.cn"* was attacked by the source *"X.X.83.34"* with many types of attack actions, such as, *Infomation Leak Attack, Backup File Attack, Post SQL Inject, Get SQL Inject, scanner action, GET XSS ATTACK, Struts2 Attack, PHP Injection Attack*. These attack actions were detected by the two methods. Undoubtedly the source *"X.X.83.34"* is a typical cyber attacker. TSPSD-steiner BJ detected more attack actions than TSPSD-steiner HC.

On May 6, 2015, we illustrated the detected cyber attack network in Figure 7. There are three main attack sources *"X.X.47.149"*, *"X.X.217.93"*, *"X.X.237.185"*, and two main attacked sites *"www.saic.gov.cn"*, *"www.xyw.gov.cn"*. The site *"www.saic.gov.cn"* was attacked by *"X.X.47.149"* with *Upload Webshell Attack*, and *"X.X.217.93"* with *Dedecms Attack, scanner action*. The site *"www.xyw.gov.cn"* was attacked by *"X.X.47.149"* and *"X.X.217.93"* with the same actions. We observed that the two sites *"www.saic.gov.cn"* and *"www.xyw.gov.cn"* were also attacked on May 5, 2015. The site *"www.saic.gov.cn"* in these two days attracted more attacks from many sources. Especially, the sites *"xyw.gov.cn"* was attacked by *"X.X.83.34"* with many types of attack actions, such as, *Infomation Leak Attack, Backup File Attack, Post SQL Inject, Get SQL Inject, scanner action, GET XSS ATTACK, Struts2 Attack, PHP Injection Attack*. The same attack also occurred on May 5, 2015. There is another source *"X.X.22.194"* attacked *"www.xcghj.gov.cn"*, *"www.xinyimzj.gov.cn"* and *"www.xjgt.gov.cn"* with many same types of attack actions besides *Local File Inclusion Attack, Remote Code with Expression Language Injection Attack, Command Injection Attack, Code Injection Attack, and IIS Short Filename Attack*. Undoubtedly, the sources *"X.X.22.194"* and *"X.X.83.34"* are the typical cyber attackers. Our methods can detect the cyber attack network in the large network through the three cases in Figure 5, 6 and 7.

We have two interesting findings: *one source attacks more sites with fewer attack types, and reversely one source attacks fewer sites with more attack types*. For example, the detected main sources, such as, *"X.X.171.42"*, *"X.X.148.207"*, *"X.X.47.149"*, *"X.X.217.93"*, attacked many sites with at most 2 types of attack actions, such as, *Upload Webshell Attack*, *Dedecms Attack*. However, for the sources *"X.X.22.194"* and *"X.X.83.34"*, they attacked at most 3 sites with at least 8 types of attack actions. These findings imply that the connected anomalous subgraphs may have different attack patterns. Without considering the specific form of anomaly distribution in the traffic networks, our nonpara-

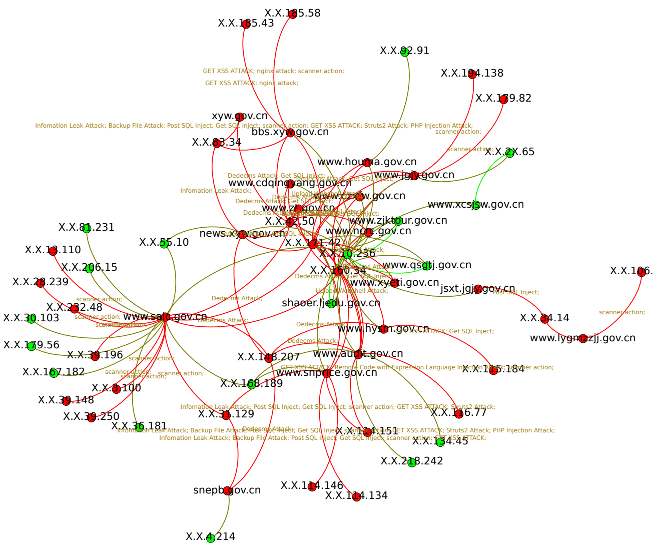


Figure 5: Attack case on May 5, 2015 by TSPSD-steiner BJ

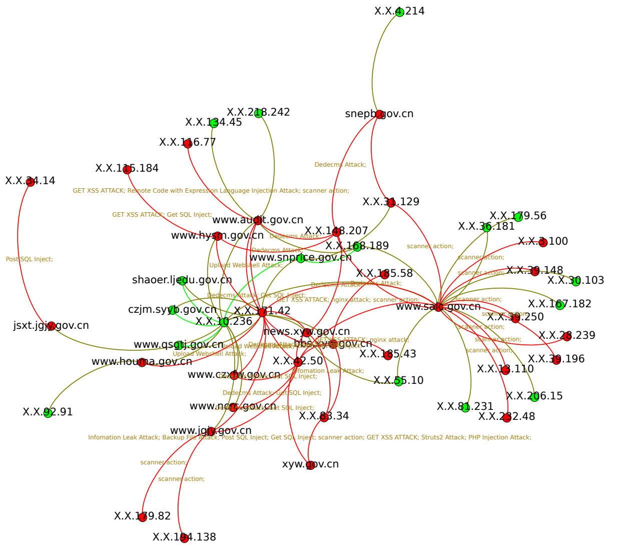


Figure 6: Attack case on May 5, 2015 by TSPSD-steiner HC

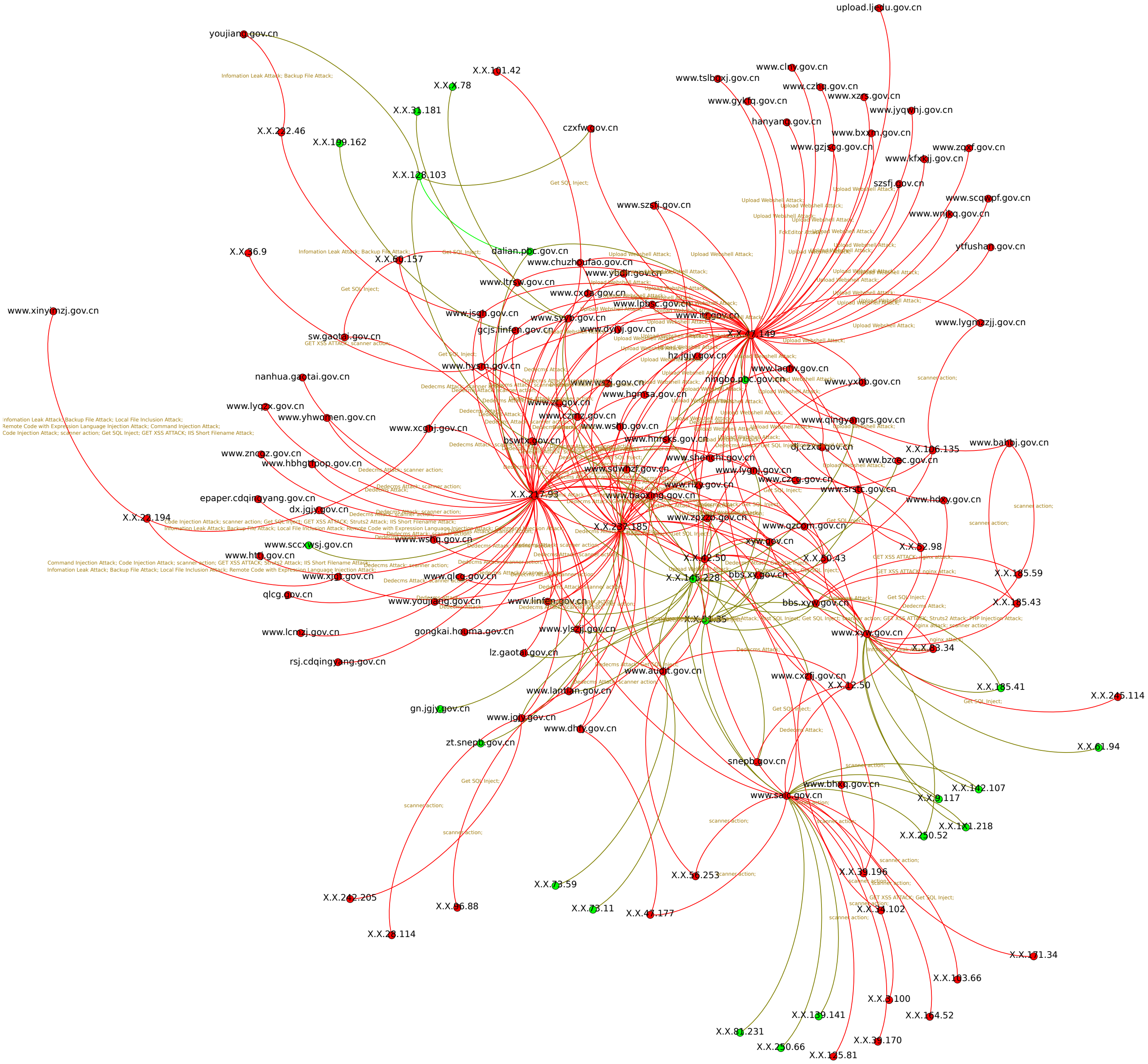


Figure 7: Attack case on May 6, 2015 by TSPSD-steiner BJ. There are attack actions if they are colored with red, and otherwise there are no attack actions.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2018.2868097, IEEE Transactions on Knowledge and Data Engineering

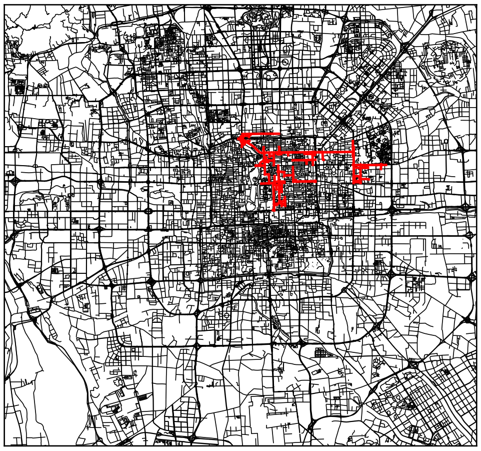


Figure 8: Congested roads (red color) on November 27, 2010, 07:00 AM - 08:00 AM, Saturday.

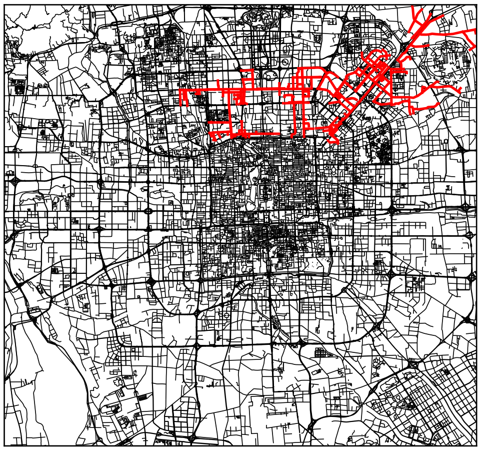


Figure 9: Congested roads on November 28, 2010, 07:00 AM - 08:00 AM, Sunday.

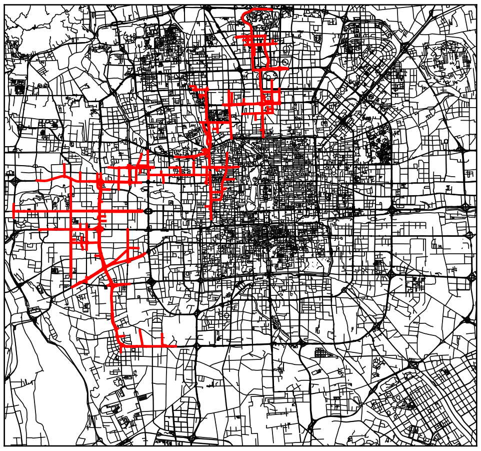


Figure 10: Congested roads on November 29, 2010, 07:00 AM - 08:00 AM, Monday.

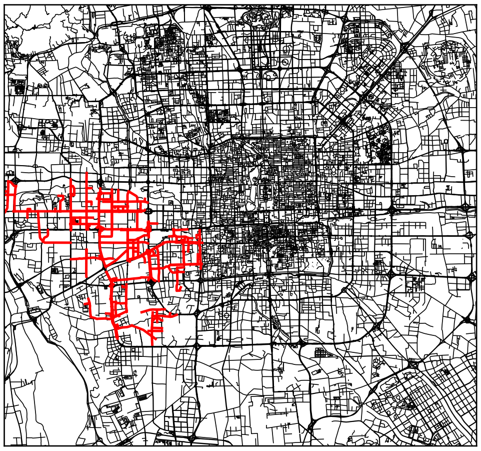


Figure 11: Congested roads on November 30, 2010, 07:00 AM - 08:00 AM, Tuesday.

metric methods have detected the different attack patterns.

### 4.7 Case Study in Social Networks

We randomly selected one day (i.e., November 27, 2014) to forecast or detect Haze events in the event detection dataset in Figure 12. We first computed the p-value for each user on this day. In the user mentioned network, we employed our methods TSPSD-steiner BJ and HC to detect the user groups (i.e., connected subgraph) about Haze events. In the groups, each user was connected to a location (i.e., province).

The report [31] stated that "*the haze events occurred within the central and eastern China, south regions of north China, north regions of Huang-huai area in China, and central Shaanxi plain between November 24-27, 2014, where the haze regions covered about one-third of China.*" In Figure 12, the lower network is a province network in China, where the blue vertices showed the wrong alerts (e.g., Shanghai), however, the yellow vertices showed the correct alerts (e.g., Beijing, Hebei, Henan). From November 24 to November 27, 2014, the Air Quality Indexes (AQI) in Beijing are 103, 177, 279 and 101 respectively (e.g., the value AQI 279 corresponds to the air quality level 5). We could observe that the subgraph detected by TSPSD-steiner HC connected to blue vertices (*wrong alerts*) are apparently less than the subgraph detected by TSPSD-steiner BJ. This observation corresponds to the results in Table 2. TSPSD-steiner HC performs better than TSPSD-steiner BJ. In this case, our methods could successfully detect or predict the Haze events in the region through social media. The main factors to our methods are the statistic (e.g., BJ and HC) and the way of computing p-value for each node. The main benefit of our nonparametric-type TSPSD methods is that there are hardly any parameters to be tuned. Although the significant level $\alpha$ is predefined (e.g., 0.15), we have proved that our result is optimal among different significant levels (i.e., $\leq \alpha$).

### 4.8 Case Study in Beijing Road Traffic Networks

We took on the consecutive four days, November 27, 28, 29 and 30, 2010, morning peak (07:00 AM - 08:00 AM), in the detected regions in Beijing, China, to demonstrate the performance of our method TSPSD-steiner BJ on congested road network detection.

The report [32] issued by the Beijing transport institute summarized the main congested roads in the morning peak (i.e., 07:00 AM - 09:00 AM, "*the south roads for east and west second ring road; the north second ring road; the south roads for east and west third ring road; the roads adhere to Wanshou road*").

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2018.2868097, IEEE Transactions on Knowledge and Data Engineering

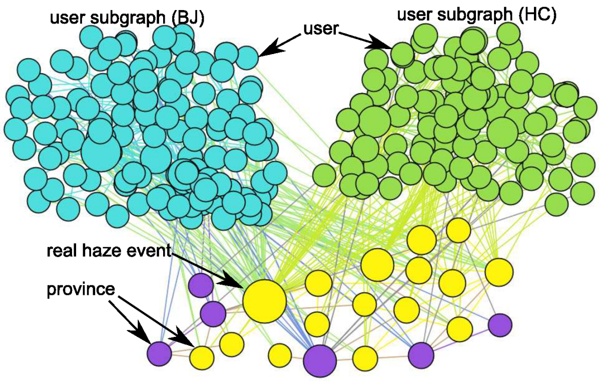
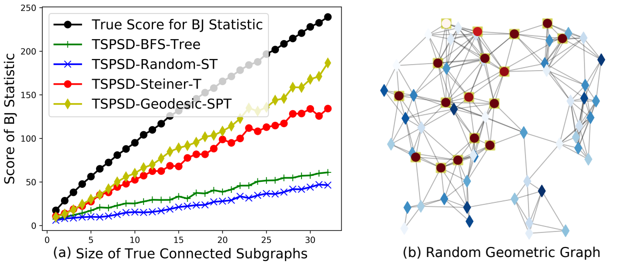


Figure 12: Haze events from Nov 27, 2014, in China. Within the 7 day window before and after that day, a yellow vertex refers to a successful forecast or detection; a blue vertex indicates an alert without a GSR record; Other color vertices consist of user subgraphs detected by TSPSD-steiner BJ or HC methods. The size of yellow and blue vertices is proportional to the count of users connected to them.

On November 27 and 28, 2010 (Saturday and Sunday, weekend), we detected the congested roads with red color in Figures 8 and 9. In Figure 8, we can observe that the main congested roads are located on the region inside the second ring road, where there are many attractions, such as the Palace Museum. At the weekend, many citizens drove to these places for entertainments, which caused to a large traffic in this region. In Figure 9, the main congested roads are destined to the Capital airport. There are occasionally associated with a fast growing traffic to the Capital airport.

On the weekday (e.g., November 29 and 30, 2010, Monday and Tuesday), the congested roads showed a significant period pattern that the congested roads got to the traffic peak by the report [32]. In Figure 10, we can observe that the congested roads are located in the west second ring road, Xi Zhi Men regions, the north third ring road, Wanshou road, and the west fourth ring road. In Figure 11, the congested roads were still the west fourth ring road and the Wanshou road, where its some nearby roads became congested in this time. The congested roads are consistent with the report [32]. We can observe that the main congested roads (e.g., the west fourth ring road and the Wanshou road) are not changed in this time for many citizens settled in the two districts (i.e., Mentougou, Fangshan), and drove to downtown for working through these roads.

From the report [32], we know that the road traffic at the weekend is remarkable larger than at the weekday. At the weekend, each road is possible to be congested, and the parametric methods will detect a large congested region for there is not a specific form of distribution to capture the variations of roads. However, our method TSPSD does not consider the specific distribution form of roads and has few parameters to be tuned. The results in Figure 10 and 11 also correspond to the congested roads in the official report [32].



(a) Size of True Connected Subgraphs

(b) Random Geometric Graph

Figure 13: (a) shows the difference between BJ scores of true subgraphs and scores achieved by our approach TSPSD (BJ) with the four tree priors on random geometric graphs; and (b) shows an example of random geometric graphs (e.g., red circle nodes denote the detected subgraph by TSPSD-Steiner-T, and yellow square nodes denote the true subgraph, where the smaller p-value is indicated by darker red, but the larger p-value is indicated by darker blue).

### 4.9 Performance Loss due to Tree Shaped Priors

As our nonparametric approach does not assume any specific forms of distributions for normal and abnormal vertices, we conducted a simulation test to evaluate the performance loss of our approach due to the tree shaped priors. We randomly generate the graphs with 64 nodes, and randomly choose anomalous connected subgraphs with the size ranging from 1 to 32. With the fixed noise level $\sigma^2 = 1$, the null hypothesis is that node values follow the normal distribution $\mathcal{N}(0,1)$, and the alternative hypothesis is that node values follow $\mathcal{N}(\mu,1)$ (e.g., $\mu = 1$) [6]. For each node, we randomly generate a sample whose size is 30 and compute its p-value. The simulation test with the same setting $K = 30, \alpha_{max} = 0.15$ is carried out 300 times.

From Figure 13(a), we can observe that scores of our approaches with Steiner and Geodesic tree priors are more closer to the true scores than the other approaches. With the size of true subgraph increasing, our approaches perform not good for the true subgraph in a tree spanned from a prior may introduce normal nodes. We randomly choose a test with 17 true nodes in Figure 13(b). Our approach TSPSD-Steiner-T detected all of the true nodes without false detected nodes, even though a node (i.e., its color is close to white) has a larger p-value.

## 5 Conclusion and Future Work

With provable guarantee based on tree shaped priors, a novel approximate algorithm is proposed to address the NPGS problem, which is reformulated as a sequence of B-PCST subproblems. Given a graph, subsets of vertices are assembled into bags, and the bags are assembled into a tree. The maximal size of bags is the *tree-width*, which describes how "tree-like" the structure of graph is [33]. For future work, we will employ the method (i.e., *tree-decomposition*) to measure how well a graph is approximated by a tree. Our work can be extended to other applications (e.g., Bitcoin fraud detection, graph-structured optimization methods).

## References

[1] L. Duczmal and R. Assunção, "A simulated annealing strategy for the detection of arbitrarily shaped spatial clusters," *Computational Statistics & Data Analysis*, vol. 45, no. 2, pp. 269–286, Mar. 2004.

[2] L. Duczmal, M. Kulldorff, and L. Huang, "Evaluation of spatial scan statistics for irregularly shaped clusters," *Journal of Computational and Graphical Statistics*, vol. 15, no. 2, pp. 428–442, 2006.

[3] K. Takahashi, M. Kulldorff, T. Tango, and K. Yih, "A flexibly shaped space-time scan statistic for disease outbreak detection and monitoring." *IJHG*, vol. 7, p. 14, 2008.

[4] J. Sharpnack, A. Singh, and A. Rinaldo, "Changepoint detection over graphs with the spectral scan statistic." in *AISTATS*, ser. JMLR Proceedings, vol. 31. JMLR.org, 2013, pp. 545–553.

[5] S. Speakman, E. McFowland III, and D. B. Neill, "Scalable detection of anomalous patterns with connectivity constraints," *JCGS*, vol. 24, no. 4, pp. 1014–1033, 2015.

[6] J. Qian, V. Saligrama, and Y. Chen, "Connected sub-graph detection," in *AISTATS, Reykjavik, Iceland, April 22-25*, 2014, pp. 796–804.

[7] G. Patil, C. Taillie *et al.*, "Geographic and network surveillance via scan statistics for critical area detection," *Statistical Science*, vol. 18, no. 4, pp. 457–465, 2003.

[8] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: A survey," *ACM Computing Surveys (CSUR)*, vol. 41, no. 3, p. 15, 2009.

[9] F. Chen and D. B. Neill, "Non-parametric scan statistics for event detection and forecasting in heterogeneous social media graphs," in *KDD '14, New York, NY, USA - Aug. 24 - 27*, 2014, pp. 1166–1175.

[10] D. B. Neill, "Fast subset scan for spatial pattern detection," vol. 74, no. 2, pp. 337–360, 2012.

[11] D. B. Neill, A. W. Moore, M. Sabhnani, and K. Daniel, "Detection of emerging space-time clusters." in *KDD*, R. Grossman, R. Bayardo, and K. P. Bennett, Eds. ACM, 2005, pp. 218–227.

[12] M. Kulldorff, "A spatial scan statistic," *Communications in Statistics: Theory and Methods*, vol. 26, pp. 1481–96, 1997.

[13] J. Sharpnack, A. Krishnamurthy, and A. Singh, "Near-optimal anomaly detection in graphs using lovasz extended scan statistic." in *NIPS*, 2013, pp. 1959–1967.

[14] E. McFowland, S. Speakman, and D. B. Neill, "Fast generalized subset scan for anomalous pattern detection," *Journal of Machine Learning Research*, vol. 14, no. 1, pp. 1533–1561, 2013.

[15] R. H. Berk and D. H. Jones, "Goodness-of-fit test statistics that dominate the kolmogorov statistics," *Probability theory and related fields*, vol. 47, no. 1, pp. 47–59, 1979.

[16] D. Donoho and J. Jin, "Higher criticism for detecting sparse heterogeneous mixtures," *Ann. Statist.*, no. 3, pp. 962–994, 06 2004.

[17] E. McFowland, S. Speakman, and D. B. Neill, "Fast generalized subset scan for anomalous pattern detection," *Journal of Machine Learning Research*, vol. 14, no. 1, pp. 1533–1561, 2013.

[18] P. Bogdanov, M. Mongiov, and A. K. Singh, "Mining heavy subgraphs in time-evolving networks." in *ICDM*, 2011, pp. 81–90.

[19] M. Bateni, M. Hajiaghayi, and V. Liaghat, "Improved approximation algorithms for (budgeted) node-weighted steiner problems," *CoRR*, vol. abs/1304.7530, 2013.

[20] D. B. Neill, "Fast subset scan for spatial pattern detection," *J. Royal Statistical Society B*, vol. 74, pp. 337–360, 2012.

[21] S. Speakman, Y. Zhang, and D. B. Neill, "Dynamic pattern detection with temporal consistency and connectivity constraints," in *ICDM'13, Dallas, TX, USA, December 7-10*, 2013, pp. 697–706.

[22] D. S. Johnson, M. Minkoff, and S. Phillips, "The prize collecting steiner tree problem: theory and practice." in *SODA*, D. B. Shmoys, Ed. ACM/SIAM, 2000, pp. 760–769.

[23] D. Donoho and J. Jin, "Higher criticism for detecting sparse heterogeneous mixtures," *Annals of Statistics*, pp. 962–994, 2004.

[24] R. H. Berk and D. H. Jones, "Goodness-of-fit test statistics that dominate the kolmogorov statistics," *Probability theory and related fields*, vol. 47, no. 1, pp. 47–59, 1979.

[25] M. Bateni, M. Hajiaghayi, and V. Liaghat, "Improved approximation algorithms for (budgeted) node-weighted steiner problems," in *ICALP*. Springer, 2013, pp. 81–92.

[26] A. Gionis, M. Mathioudakis, and A. Ukkonen, "Bump hunting in the dark: Local discrepancy maximization on graphs," in *ICDE'2015, Seoul, Korea, April 13 - April 17*, 2015, pp. 64–75.

[27] J. Sthmer, P. Schrder, and D. Cremers, "Tree shape priors with connectivity constraints using convex relaxation on general graphs." in *ICCV*. IEEE, 2013, pp. 2336–2343.

[28] P. Narvez, K.-Y. Siu, and H.-Y. Tzeng, "New dynamic algorithms for shortest path tree computation." *IEEE/ACM Trans. Netw.*, vol. 8, no. 6, pp. 734–746, 2000.

[29] M. Bansal and V. Venkaiah, "Improved fully polynomial time approximation scheme for the 0-1 multiple-choice knapsack problem," *International Institute of Information Technology Report*, 2004.

[30] D. Pisinger, "A minimal algorithm for the multiple-choice knapsack problem." *EJOR*, vol. 83, pp. 394–410, 1994.

[31] "Mep of china." [Online]. Available: www.mep.gov.cn

[32] "Beijing transport institute." [Online]. Available: www.bjtrc.org.cn

[33] I. V. Hicks, A. M. Koster, and E. Kolotoğlu, "Branch and tree decomposition techniques for discrete optimization," in *Emerging Theory, Methods, and Applications*. INFORMS, 2005, pp. 1–29.

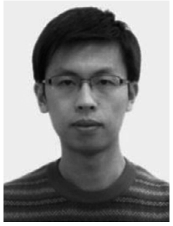
**Nannan Wu** received the BS degree in computer science from Zhengzhou University in 2009, the MS degree from Chongqing University, China, in 2012. He is working toward the PhD degree from the Department of Computer Science and Engineering, Beihang University. His research interests include graph-structured optimization and specific-shape optimization for anomalous subgraph detection in attributed networks.

**Feng Chen** received the BS degree from Hunan University in 2001, the MS degree from Beihang University, China, in 2004, and the PhD degree from Virginia Tech, all in computer science. He is an assistant professor in the Computer Science Department, SUNY Albany. He has published 70 refereed articles in major data mining venues, including ACM-SIGKDD, ACM-CIKM, ACM-GIS, and IEEE-ICDM. He holds two U.S. patents on human activity analysis filed by IBMs T.J. Watson Research Center. His research interests include statistical machine learning and data mining, with an emphasis on spatio-temporal analysis and energy disaggregation.

**Jianxin Li** received the PhD degree from Beihang University in 2008. He is a professor in the School of Computer Science and Engineering, Beihang University. He was a visiting scholar in the Machine Learning Department of CMU in 2015, and a visiting researcher of MSRA in 2011. His current research interests include data analysis and processing, distributed systems, and system virtualization.

**Jinpeng Huai** received the PhD degree in computer science from Beihang University, China, in 1993. He is a professor in the School of Computer Science and Engineering, Beihang University, China. He is an academician of the Chinese Academy of Sciences and the vice honorary chairman of China Computer Federation (CCF). His research interests include big data computing, distributed systems, virtual computing, service-oriented computing, trustworthiness, and security.

**Baojian Zhou** is a Ph.D. student from SUNY at Albany. His research interests are submodular optimization, convex optimization and graph mining.



**Bo Li** is an assistant professor at the School of Computer Science and Engineering, Beihang University. He received the Ph.D. degree in Jan. 2012. He was a visiting scholar in computer science department of University of Edinburgh in 2014. His current research interests include virtualization, system reliability and data mining etc.

**Naren Ramakrishnan** received the PhD degree in computer sciences from Purdue University, West Lafayette, IN, in 1997. He is currently the Thomas L. Phillips Professor of Engineering in the Department of Computer Science at Virginia Tech, Blacksburg, VA. His research has been supported by NSF, DHS, NIH, NEH, DARPA, IARPA, ONR, General Motors, HP Labs, NEC Labs, and Advance Auto Parts.