An Efficient Framework for Detecting Evolving Anomalous Subgraphs in Dynamic Networks

Minglai Shao, Jianxin Li
School of Computer Science and Engineering
Beihang University
{shaoml,lijx}@act.buaa.edu.cn

Feng Chen
Department of Computer Science
State University of New York at Albany
fchen5@albany.edu

Xunxun Chen
CNCERT
xx-chen@139.com

***Abstract*—Evolving anomalous subgraphs detection in dynamic networks is an important and challenging problem that has arisen in multiple applications and is NP-hard in general. The evolving characteristic makes most existing methods incapable to tackle this problem effectively and efficiently, as it involves huge search spaces and continuous changes of evolving connected subgraphs, especially when the data are free of distributions. This paper presents a generic efficient framework, namely dynamic evolving anomalous subgraphs scanning (dGraphScan), to address this problem. We generalize traditional nonparametric scan statistics, and propose a large class of scan statistic functions for measuring the significance of evolving subgraphs in dynamic networks. Furthermore, we make a number of computational studies to optimize this large class of nonparametric scan statistic functions. Specifically, we first decompose each scan statistic function as a sequence of subproblems with provable guarantees, and then propose efficient approximation algorithms for tackling each subproblem, while analyzing their theoretical properties and providing rigorous approximation guarantees. Extensive experiments on three real-world datasets demonstrate that our general framework performs superior over state-of-the-art methods.**

***Index Terms*—evolving anomalous subgraphs detection, dynamic networks, nonparametric scan statistics, approximation algorithm**

I. Introduction

Dynamic networks in domains ranging from computer to social media, transportation and neuroscience networks evolve temporally following the underlying network structure [1]–[5], such as malicious Botnet activity propagates in computer networks as computers get infected and recover due to software patches [6]–[8], rumors spread along friendship links in social networks [9] and traffic congestion shift spatially in road networks [10], [11]. These processes (growth of Botnet communication, spreading of rumors, moving of traffic congestion, etc.) can be summarized as the anomalous subgraphs (connected infected computers, connected users who spread rumors, connected congestion roads, etc.) evolve over time.

Intuitively, the detection of such evolving anomalous subgraphs (network processes) has the potential to expose the intrinsic mechanisms of complex dynamic networks, and has a number of important applications, such as identifying congestion processes in traffic networks helps the understanding of traffic congestion propagation and may enable improved urban planning [12], mapping a water contamination process as a growing anomalous network may help indicate the source of contamination and predict the rate and direction of its spread [13].

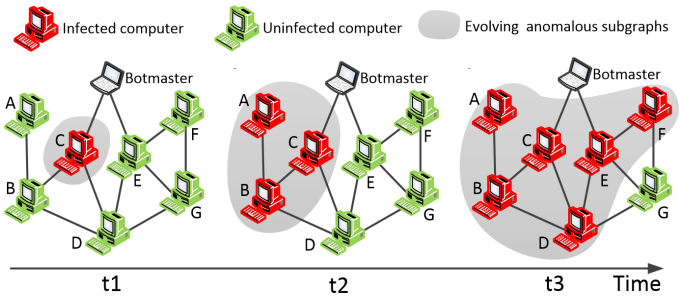


Fig. 1. An illustration of Peer-to-Peer Botnet where malicious Botnet activity propagates in computer networks as computers get infected over time. The evolving infected (anomalous) computer subgraphs $\Omega = \{\{C\}, \{A, B, C\}, \{A, B, C, D, E, F\}\}$.

This paper focuses on the problem of characterization and mining of evolving anomalous subgraphs in dynamic networks (as shown in Figure 1). While we allow the subgraphs to change in consecutive time slices to capture the evolution of the underlying phenomenon, impose a constraint on their rate of change to ensure that we capture a unique anomalous network process, fix network structure and dynamic updated vertex weights since these example applications mentioned above fall within the same setting of relatively stable network structure and dynamic attributes associated with vertices.

Unfortunately, the evolving characteristic makes most existing methods difficult to tackle the problem of evolving anomalous subgraphs detection effectively and efficiently. And this problem has been proven an NP-hard problem and still faces several important challenges [1], [4], [14], as it first involves **huge search spaces** and **continuous changes** of the large scale of real world networks evolving over long periods of time. This makes the evolving subgraphs detection become a computationally difficult task. In addition, recent works on the evolving subgraphs detection mainly focus on *parametric* methods [15]–[17]. They assume specific distributions (e.g. Poisson counts) modeling attributes varying of normal and abnormal vertices respectively, and formalize anomaly detection as a hypothesis testing problem. Nevertheless, the performance degrades when these models are incorrect [18], especially when the data are **free of distributions**. In contrast

978-1-5386-4128-6/18/$31.00 ©2018 IEEE

to traditional parametric scan statistics, such as Kulldorff statistic [19], expectation-based Poisson statistic [20] and elevated mean scan statistic [21], *nonparametric* methods do not associate specific forms of distributions with normal and abnormal vertices. Instead of distribution assumptions, they first estimate a p-value for each vertex based on empirical calibration by comparing the current attribute of the vertex with its attributes in the historical data for this vertex [2], [18], [22], [23]. This approach then maximizes a score function $\mathfrak{F}(G)$ of p-values of vertices in graph G. Typically, nonparametric scan statistics measure the significance of the collection of p-values of vertices in G over all possible connected subsets [18], [23]. However, to the best of our knowledge, there remains a lack of the work in which *nonparametric* scan statistics are used for detecting evolving anomalous subgraphs in dynamic networks.

For the sake of efficiently detecting evolving anomalous subgraphs that are free of distributions in dynamic networks, we first generalize *nonparametric* statistics, such as Berk-Jones (BJ) statisctic [24], Higher Criticism (HC) statistic [25], Davidov-Herman statistic, Tippet's statistic, and propose a large class of scan statistic functions for measuring the significance of evolving subgraphs in dynamic networks. What's more, we make a number of contributions to the computational studies for optimizing these nonparametric scan statistic functions. Therewith, an efficient framework, named as dGraphScan, is proposed. This framework can run in nearly-linear time and allows for a large class of sophisticated nonlinear scoring functions based on different nonparametric scan statistics. The main contributions of this paper are summarized as follows:

- **Formulation of the dGraphScan framework.** A general framework, namely dynamic evolving anomalous subgraphs scanning (dGraphScan), is proposed and allows for a large class of sophisticated nonlinear scoring functions based on different nonparametric scan statistics that are free of distribution assumptions, for tackling the problem of evolving anomalous subgraphs detection in dynamic networks. This contrasts with all prior methods which are developed for specific statistics. Furthermore, our approach also holds for the extensions of these functions with both vertex and edge weights.
- **Design of rigorous approximation algorithms for dynamic evolving anomalous subgraphs scanning**. We make a number of computational studies to optimize the large class of nonparametric scan statistics functions mentioned above. Specifically, we first efficiently decompose the dGraphScan as a sequence of subproblems with provable guarantees. Then approximation algorithms are proposed for solving the decomposed subproblems with the analysis of its theoretical properties. Our proposed approximation algorithms are guaranteed to converge to an optimal result and have a nearly-linear time complexity.
- **Comprehensive experiments to validate the effectiveness and efficiency of the proposed framework.** Extensive experiments are conducted to evaluate the dGraphScan on water pollution detection, traffic congestion detection and haze event detection. The results demonstrate that dGraphScan outperforms existing representative competitive methods in both performance and quality.

The rest of the paper is organized as follows. Section 2 introduces some definitions, including: dynamic network, evolving subgraphs, p-value and nonparametric scan statistics. Section 3 first generalizes nonparametric scan statistics and performs the proposed dGraphScan framework, and then presents approximation approaches. Comprehensive experiments are provided in Section 4. Finally, conclusion and future work are presented.

II. Preliminaries

This section presents several definitions, including: dynamic network, evolving subgraphs, p-value and nonparametric scan statistics.

Definition 1 (Dynamic network). *A dynamic network* $\mathbf{G} = \{\mathbb{V}, \mathbb{E}, W\}$ *is an undirected connected graph, where* $\mathbb{V} = \{v_1, ..., v_{\mathcal{N}}\}$ *is the set of vertices,* $\mathbb{E} \subseteq \mathbb{V} \times \mathbb{V}$ *refers to the set of edges, and* $W = \{w^1, ..., w^T\}$ *is a family of weight functions of the kind* $w^t : \mathbb{V} \to \mathbb{R}$ *that associate each vertex* $v \in \mathbb{V}$ *with an anomaly score. Each function* w^t *corresponds to a discrete time slice* t.

The weights of vertices quantify their time-dependent level of anomaly. We evaluate the anomaly of vertices based on the p-values in this work. Given a vertex and its observation (e. g., the average speed in road networks) at a given time, we measure the significance of this observation as its statistical p-value, according to the empirical distribution of observations on the vertex. The p-value is computed as the fraction of time slices in which an equal or higher observation is included on the same vertex [2], [22]. Moreover, we denote the p-value of a vertex v at time slice t with $p^t(v)$. Intuitively, the p-value is a measure of anomalousness within the range between 0 and 1: *the smaller the p-value of a feature value, the higher the degree of anomalousness of this feature value*. We prepare to define nonparametric statistics for evaluating the significance of p-values, which will be used to define the score functions used for measuring the degree of anomalousness of a subset of vertices and features.

Definition 2 (Evolving subgraphs). *Given a dynamic network* $\mathbf{G}$*, the evolving subgraphs* $\Omega = \{G^1, ..., G^T\}$ *is a contiguous sequence of connected subgraphs (each one in a separate time slice) of* $\mathbf{G}$ *that satisfies:*

- *Every subgraph is connected within its time slice;*
- *Two contiguous subgraphs share at least one vertex, e. g.* $V^t \bigcap V^{t+1} \neq \varnothing$, $\forall t \in \{1, 2, ..., T-1\}$, *where* $V^t \subseteq \mathbb{V}$ *is the set of vertices of* G^t *which denotes the projection of the evolving subgraphs at time* t.

Definition 3 (Nonparametric Scan Statistics). *Given a set of p-values* Θ*, nonparametric statistics (also called aggregation functions of p-values) refer to a class of scoring functions*

$\mathcal{F}(\Theta)$ that measure the joint significance of multiple p-values in Θ and have the general form:

$$\mathcal{F}(\Theta) = \varphi(\alpha, S_\alpha(\Theta), S(\Theta)) \tag{1}$$

where α is a predefined significance level of p-values; $S_\alpha(\Theta)$ refers to the number of p-values in Θ that are less than or equal to α; $S(\Theta)$ refers to the number of p-values in Θ, and the function $\varphi(\alpha, S_\alpha, S)$ satisfies two intuitive properties:
***(P1)** φ is monotonically increasing with respect to S_α;*
***(P2)** φ is monotonically decreasing with respect to α and S.*

III. Evolving Anomalous Subgraphs Scanning

This section first generalizes nonparametric scan statistics and presents a new class of evolving subgraph scan statistic functions for the problem of evolving anomalous subgraphs detection and then decomposes them as as a sequence of subproblems with provable guarantees.

A. Problem Definition

Given a dynamic network $\mathbf{G} = \{\mathbb{V}, \mathbb{E}, W\}$, to detect which evolving subgraphs are the most anomalous in this dynamic network, the general form of the nonparametric scan statistics in dynamic networks is defined as:

$$\begin{aligned} \mathfrak{F}(\Omega) = \max_{\alpha \le \alpha_{max}} \varphi(\alpha, S_\alpha(\Omega), S(\Omega)) \\ s.\,t.\ \delta(\Omega) \le B, S(\Omega) \le K \end{aligned}, \tag{2}$$

where the function φ is defined in Definition 3, $\delta(\Omega) = \sum_{t=1}^{T-1}(|V^t| + |V^{t+1}| - 2|V^t \bigcap V^{t+1}|)$ refers to the total number of the change of the evolving anomalous subgraphs Ω in adjacent time slices, B is the upper bound of the change, K is the upper bound of cardinality constraint of Ω, $S(\Omega) = \sum_{t=1}^{T} |V^t|$ refers to the total number vertices in Ω, $S_\alpha(\Omega) = \sum_{v \in \Omega} \mu(p^t(v) \le \alpha)$ is the number of p-values significant at level α. The function $\mu(\cdot) = 1$ if its input is true, otherwise $\mu(\cdot) = 0$. The significance level α can be optimized between 0 and some constant $\alpha_{max} < 1$ (0.15 by default) [22].

Through this paper, for the purpose of brevity and illustration, we consider the evolving anomalous subgraphs scan statistic function $\mathfrak{F}_{\mathrm{BJ}}(\Omega)$ (Equation (4)) based on the BJ statistic (Equation (3)) as a case study, since this statistic has been shown effective in a number of real-world applications [22], [23]. *Our proposed techniques will be applicable to other subgraph scan statistic functions as well.*

The BJ statistic, which uses the KL divergence, can be interpreted as the log-likelihood ratio statistic for testing whether the empirical p-values follow a uniform or piecewise constant distribution. Berk and Jones demonstrated that this statistic fulfills several optimality properties and has greater power than any weighted Kolmogorov statistic [24]. The BJ statistic is defined as:

$$\varphi_{\mathrm{BJ}}(\alpha, S_\alpha(\Omega), S(\Omega)) = S(\Omega) \times \mathrm{KL}(\frac{S_\alpha(\Omega)}{S(\Omega)}, \alpha), \tag{3}$$

where $\mathrm{KL}(\cdot)$ is the Kullback-Liebler divergence between the observed and expected proportions of p-values less than α. $\mathrm{KL}(\cdot)$ divergence is defined as: $\mathrm{KL}(\iota, \zeta) = \iota \log(\frac{\iota}{\zeta}) + (1 - \iota) \log(\frac{1-\iota}{1-\zeta})$. To sum up, $\mathfrak{F}_{\mathrm{BJ}}(\Omega)$ is shown as:

$$\begin{aligned} \mathfrak{F}_{\mathrm{BJ}}(\Omega) &= \max_{\alpha \le \alpha_{max}} \varphi_{\mathrm{BJ}}(\alpha, S_\alpha(\Omega), S(\Omega)) \\ &= \max_{\alpha \le \alpha_{max}} S(\Omega) \times \mathrm{KL}(\frac{S_\alpha(\Omega)}{S(\Omega)}, \alpha) \cdot \\ &\quad s.\,t.\ \delta(\Omega) \le B, S(\Omega) \le K \end{aligned} \tag{4}$$

Based on the generalized nonparametric scan statistics mentioned above, the detection of the most anomalous evolving subgraphs can be formalized as the following optimization problem:

$$\begin{aligned} \max_{\Omega} \max_{\alpha \le \alpha_{max}} \varphi(\alpha, S_\alpha(\Omega), S(\Omega)) \\ s.\,t.\ \delta(\Omega) \le B, S(\Omega) \le K \end{aligned}, \tag{5}$$

which is equivalent to:

$$\begin{aligned} \max_{\alpha \in \mathrm{U}(\mathbb{V}, W, \alpha_{max})} \max_{\Omega} \varphi(\alpha, S_\alpha(\Omega), S(\Omega)) \\ s.\,t.\ \delta(\Omega) \le B, S(\Omega) \le K \end{aligned}, \tag{6}$$

where $\mathrm{U}(\mathbb{V}, W, \alpha_{max})$ refers to the union of $\{\alpha_{max}\}$ and the set of distinct vertex p-values no more than α in dynamic network $\mathbf{G}$.

B. Problem Decomposition

Generally, for the purpose of detecting evolving subgraphs in dynamic networks, the analysis of the nonparametric scan statistic problems is difficult as it involves the evolving characteristic and a series of sophisticated nonlinear objective functions. Due to the difficulties of analyzing the aforementioned problem, we design to decompose the proposed nonparametric scan statistic problems as a sequence of subproblems with provable guarantees. The decomposition is shown below.

Theorem 1 (Problem Decomposition). *Denote $\bar{S}_\alpha(\Omega) \equiv S(\Omega) - S_\alpha(\Omega)$. Equation (6) is equivalent to the following problem:*

$$\begin{aligned} (a^*, \Omega^*) = \operatorname*{argmax}_{\alpha \in \mathrm{U}(\mathbb{V}, W, \alpha_{max})} \operatorname*{argmax}_{\Omega_\alpha \in \mathbb{S}} \varphi(\alpha, S_\alpha(\Omega), S(\Omega)) \\ s.\,t.\ \delta(\Omega) \le B, S(\Omega) \le K \end{aligned}, \tag{7}$$

where $\mathbb{S} = \{\Omega_\alpha^0, ..., \Omega_\alpha^{\sum_{t=1}^{T} \mathcal{N}}\}$ and each $\Omega_\alpha^{\mathcal{M}} \in \mathbb{S}$ refers to the solution to the following $\mathcal{M}$-budget subgraph detection problem:

$$\begin{aligned} \Omega_\alpha^{\mathcal{M}} = \operatorname*{argmax}_{\Omega} S_\alpha(\Omega) \\ s.\,t.\ \delta(\Omega) \le B, S(\Omega) \le K, \bar{S}_\alpha(\Omega) \le \mathcal{M} \end{aligned}. \tag{8}$$

Proof. Denote $\Omega_\alpha^- \equiv \{v | p^t(v) \le \alpha, v \in \Omega\}$, $\Omega_\alpha^+ \equiv \{v | p^t(v) > \alpha, v \in \Omega\}$. Each workable subgraph Ω can be decomposed into the subset of vertices Ω^+ and the subset of vertices Ω^- satisfying the conditions: $\bar{S}_\alpha(\Omega) \le \mathcal{M}$ and $\Omega = \Omega^+ \bigcup \Omega^-$. Suppose that (α^*, Ω^*) is the optimal solution to the Problem (7), and $\bar{S}_\alpha(\Omega^*) = m$, where $0 \le m \le \mathcal{N} \times T$. Then it can be easily derived that $\Omega_{\alpha^*}^m = \Omega^*$. Based on the property (P1) and (P2) mentioned in Definition 3, there

does not exist other $\Omega_{\alpha}^{\mathcal{M}}$, where $\alpha \neq \alpha^*$ or $\mathcal{M} \neq m$, such that $\varphi(\alpha, S_\alpha(\Omega_\alpha^{\mathcal{M}}), S(\Omega_\alpha^{\mathcal{M}})) > \varphi(\alpha, S_\alpha(\Omega_{\alpha^*}^{m}), S(\Omega_{\alpha^*}^{m}))$. Otherwise, this is in contradiction to the fact that (a^*, Ω^*) is the optimal solution to (7). □

IV. Approximation Algorithms

Due to the difficulties of solving the aforementioned sub-problems, we focus on finding an optimal approximation solution instead of the exact solution for the Problem (8) through the work: seek out a $\Omega^{'}$ satisfying:

$$S_\alpha(\Omega^{'}) \geq \mathcal{C} \cdot \max_{\Omega} S_\alpha(\Omega) \; s.\,t.\; \delta(\Omega) \leq B, S(\Omega) \leq K, \quad (9)$$

where $\mathcal{C} > 0$ is a constant. And a multiplicative guarantee is provided for this approximation. Moreover, *Lagrangian relaxation*, *spanning tree* and *dynamic programming* are mainly used for obtaining the best result for the Problem (9). The details of them are shown as follows.

Meanwhile, Lagrangian relation is employed to guarantee $S(\Omega) \leq K$ and is shown as follows:

$$\max_{\Omega} S_\alpha(\Omega) - \lambda S(\Omega) \quad s.\,t.\; \delta(\Omega) \leq B, S(\Omega) \leq K, \quad (10)$$

where the parameter λ controls the trade-off between the number of vertices of Ω and the approximation result. Furthermore, the Problem (10) can be rewritten as:

$$\max \sum_{v \in \Omega} (\mu(p^t(v) \leq \alpha) - \lambda) \quad s.\,t.\; \delta(\Omega) \leq B, S(\Omega) \leq K. \quad (11)$$

Moreover, under the premise of getting all different number of vertices of evolving sugraphs Ω, $\lambda \in [0, \frac{S_\alpha(\Omega)}{S(\Omega)}]$ can be got. For each $\lambda \in [0, \frac{S_\alpha(\Omega)}{S(\Omega)}]$, we also can find a corresponding $\lambda^{'} \in [0, \alpha]$ which enables the Problem (11) equivalent to:

$$\max \sum_{v \in \Omega} (\mu(p^t(v) + \lambda^{'} \leq \alpha)) \quad s.\,t.\; \delta(\Omega) \leq B, S(\Omega) \leq K. \quad (12)$$

A. Approximations with Spanning Tree

In this section, to obtain expected results efficiently for the Problem (9), we propose approximating the graphs $\{\mathbb{V}, \mathbb{E}, w^t\} \in \mathbf{G}$ from T time slices as the same tree Γ originating from a given root vertex $\tau \in \mathbb{V}$, where $t = 1, ..., T$. Then the search of the best connected evolving subgraphs Ω for the nonparametric scan statistic problem can be approximated as the search of the best sub-trees in all Γ_τ (each one in a separate time slice). In order to get Γ, we first label abnormal vertices whose p-value is no more than α and normal vertices whose p-value is more than α as 1 and 0, respectively. If $p^t(v) \leq \alpha$, denote $l^t(v) = 1$; otherwise, $l^t(v) = 0$, where $l^t(v)$ is the label of vertex v at time t. Then we denote $L(v) = l^1(v) \vee l^2(v) \vee ... \vee l^T(v)$ as the label of vertex v. Specifically, if $L(v) = 0$, the vertex v is normal in all time.

Some proposed heuristic approaches can be used to find the tree Γ based on vertex labels mentioned above, such as Random spanning tree, Breadth-first search tree, Steiner tree and Geodesic shortest path tree. The spanning trees have been successfully applied to event detection based on graphs [26], [27]. In this work, **Steiner tree** is selected owing to its outstanding comprehensive performance [26], [27]. Intuitively, a tree is good if abnormal vertices are interconnected 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. Specifically, the Steiner tree heuristic computes the steiner tree for each $\alpha \in \mathrm{U}(\mathbb{V}, W, \alpha_{max})$, computes the best sub-tree for each (9), and then returns the best solution. Based on the spanning tree and the Theorem 1, dGraphScan is presented in Algorithm 1.

Algorithm 1 dGraphScan

Input: $\mathbf{H} = 6$, $\alpha_{max} = 0.15$, dynamic network $\mathbf{G}$.
Output: the evolving anomalous subgraphs Ω^*.
For $\mathbf{h} \in \{1, ..., \mathbf{H}\}$ **do**
  Choose root vertex τ from $\{v | v \in \Omega, p^t(v) \leq \alpha_{max}\}$;
  Approximate the graphs as the trees Γ_τ;
  For $\alpha \in \mathrm{U}(\mathbb{V}, W, \alpha_{max})$ **do**
    For $\mathcal{M} = 0, ..., \bar{S}_\alpha(\mathbf{G})$ **do**
      $\Omega_\alpha^{\mathcal{M}} \leftarrow$ Approximation Algorithm $(K, \alpha, c, \gamma, B)$;
    End for
  End for
  $\Omega^{\mathbf{h}} = \mathrm{argmax}_{\Omega_\alpha^{\mathcal{M}}} \varphi(\alpha, S_\alpha(\Omega_\alpha^{\mathcal{M}}), S(\Omega_\alpha^{\mathcal{M}}))$;
End for
Calculate $\mathbf{h}^* = \mathrm{argmax}_{\mathbf{h}} \varphi(\alpha, S_\alpha(\Omega^{\mathbf{h}}), S(\Omega^{\mathbf{h}}))$;
Return $\Omega^{\mathbf{h}^*}$

Furthermore, for the sake of achieving the most desired solution to (9), a dynamic programming (DP) algorithm (as shown in Figure 2) is designed when the input dynamic network $\mathbf{G}$ is a set of trees Γ_τ with the root vertex τ.

Firstly, we introduce several notations:

- $p^{t\prime}(v)$: updated p-value of vertex v at time slice t by $p^{t\prime}(v) = p^t(v) + \lambda^{'}$.
- Γ_v: a sub-tree of $\{\mathbb{V}, \mathbb{E}, w^t\} \in \mathbf{G}$ with the root vertex v.
- $\psi^t(v)$: a Boolean value that indicates whether $p^{t\prime}(v) \leq 0$. If $p^{t\prime}(v) \leq 0$, set $\psi^t(v) = 1$; otherwise, set $\psi^t(v) = 0$.
- $\Omega_v = \{\Omega_{v^1}, \Omega_{v^2}, ..., \Omega_{v^B}\}$: candidate solutions to Γ_v, thereinto, $\Omega_{v^b} \in \Omega_v$ corresponds to $\delta(\Omega_v) = b$, where $b \in \{0, 1, ..., B\}$. Moreover, Ω_{v^b} owns the maximum $S_\alpha(\Omega_v)$ under the constraint $\delta(\Omega_v) = b$.

Then, the dynamic programming procedure from the leaf vertices to the root vertices is shown as follows.

For the leaf vertex v, we first let $\delta(v) = \sum_{t=1}^{T-1} |\psi^t(v) - \psi^{t+1}(v)| = 0, 1, ..., B$ respectively. $\delta(v)$ which is equal to $\delta(\Omega_v)$ denotes the change of vertex v in T time slices when v is the leaf vertex. Then we can obtain the corresponding candidate solutions: Ω_{v^1}, Ω_{v^2},..., Ω_{v^B} under different $\delta(v)$.

For the non-leaf vertex v, when the iteration comes to v, $v_{child} = \{v_{c1}, v_{c2}, ..., v_{cw}\}$ denotes the w child vertices of the vertex v. Every child $v_{ci} \in v_{child}$ owns $\Omega_{v_{ci}}$, where $i \in \{1, 2, ..., w\}$. Specifically, v also owns the candidate combinations corresponding to change constrains {0, 1, ..., B},

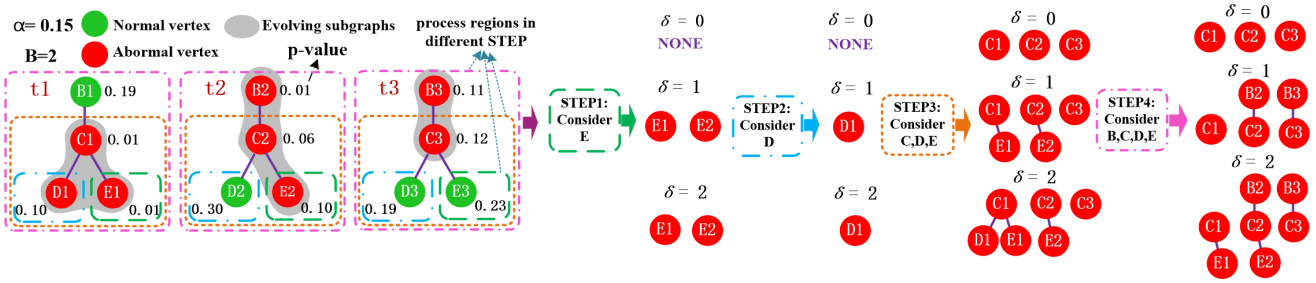


Fig. 2. An illustration of dynamic programming for finding evolving anomalous subgraphs Ω in a dynamic network. In each iteration from leaf vertices to root vertices, we select one Ω_{v^b} such that $S_\alpha(\Omega)$ is maximized under the constraint $\delta(\Omega_v) = b$ from all kinds of combination of candidate solutions to v and the candidate solutions to every child of v.

in which only v is included. Finding each of the candidate solutions to Γ_v can be reduced to an approximation 0-1 multiple-choice knapsack combinatorial optimization problem from the vertex v and v_{child}. The problem is to select one Ω_{v^b} such that $S_\alpha(\Omega_v)$ is maximized under the constraint $\delta(\Omega_v) = b$ from all kinds of combination of candidate solutions to v and the candidate solutions to v_{child} (e. g., $b = 1$, the available combinations of $\{\delta(v), \delta(\Omega_{v_{c1}}), ..., \delta(\Omega_{v_{cw}})\}$ include: $\{1, 0, ..., 0\}$, $\{0, 1, ..., 0\}$, ..., $\{0, 0, ..., 1\}$). Then all candidate solutions Ω_v of Γ_v can be found. From the leaf vertices to root vertex τ, we can obtain all candidate solutions Ω_τ of Γ_τ. Ultimately, the best result $\Omega^{'} = \max_{b\in\{0,1,...,B\}}\Omega_{\tau^b}$ can be got.

B. Approximation Solutions

In this section, we perform Algorithm 2 based on DP algorithm mentioned in Section IV.*A* over $\lambda^{'}$ to find a suitable value since the Lagrangian relaxation mentioned above just makes indirect control over $S(\Omega) \leq K$, and return the most desired $\Omega^{'}$. The approximation results are presented in Theorem 2.

Theorem 2. *Let Ω refer to the detected evolving subgraphs that potentially span T time slices. In addition, let c, $\gamma > 0$. Then Algorithm 2 can return a solution $\Omega^{'}$ that satisfies:*

$$S_\alpha(\Omega^{'}) \geq (1 + \frac{1}{c-1} - \frac{\gamma}{\alpha \cdot K}) \max_{\Omega} S_\alpha(\Omega). \tag{13}$$

And the algorithm has a time complexity $O(T\mathcal{N}B^2 \log \frac{\alpha \cdot K}{\gamma})$, where $\mathcal{N}$ is the number of vertices in $\mathbb{V}$, T refers to the total number of time slices, B is the upper bound of the change of Ω, K is the upper bound of cardinality constraint of Ω.

Proof. (1) Let Ω_l and Ω_r refer to the solutions corresponding to λ_l and λ_r respectively. Let Ω_K refer to the solution with $S_\alpha(\Omega_K) = \max_\Omega S_\alpha(\Omega)$. In the iterations, we keep two invariants $K_r \geq c \cdot K$ and $K_l < K$. From the dynamic programming, we can get a $\Omega^{'}$ satisfying $S_\alpha(\Omega^{'}) - \lambda S(\Omega^{'}) = \max_\Omega S_\alpha(\Omega) - \lambda S(\Omega)$ and

$$S_\alpha(\Omega_r) - \lambda_r S(\Omega_r) \geq S_\alpha(\Omega_K) - \lambda_r S(\Omega_K) \tag{14}$$

At the end of the iteration, we have $\lambda_l^{'} - \lambda_r^{'} \leq \varepsilon$, which can be approximated as:

$$\lambda_l - \lambda_r \leq \frac{\varepsilon \cdot S_\alpha(\Omega_K)}{\alpha \cdot K}. \tag{15}$$

Combine (14) and (15), we can get:

$$\lambda_l \leq (\frac{\varepsilon}{\alpha K} + \frac{1}{K(1-c)}) S_\alpha(\Omega_K). \tag{16}$$

Based on the dynamic programming, we also obtain:

$$S_\alpha(\Omega_l) - \lambda_l S(\Omega_l) \geq S_\alpha(\Omega_K) - \lambda_l S(\Omega_K). \tag{17}$$

which is equivalent to:

$$\begin{aligned} S_\alpha(\Omega_l) &\geq S_\alpha(\Omega_K) + \lambda_l(S(\Omega_l) - S(\Omega_K)) \\ &\geq S_\alpha(\Omega_K) + \lambda_l(-K) \end{aligned}. \tag{18}$$

Combine (16) with (18):

$$S_\alpha(\Omega_l) \geq S_\alpha(\Omega_K)(1 + \frac{1}{c-1} - \frac{\gamma}{\alpha K}). \tag{19}$$

In conclusion, we can obtain (13).

(2) In Algorithm 2, the difference $\lambda_l^{'} - \lambda_r^{'}$ initially is α and is then halved in every iteration until it reaches ε. Hence, the total number of iterations is at most: $\log \frac{\alpha}{\varepsilon} = \log \frac{\alpha \cdot K}{\gamma}$. In practical applications, α, γ can be considered as constants, this algorithm runs in time $O(X \cdot \log \frac{\alpha \times K}{\gamma})$, where X is the time complexity of the rest algorithms. For every vertex v in the whole graph, we need to find B kinds of solutions from v and its child vertices that each of them owns B candidate solutions. Moreover, every vertex spans T time slices. So we can obtain the X that is equal to $O(\mathcal{N}TB^2)$. Thus, the time complexity $O(X \cdot \log \frac{\alpha \cdot K}{\gamma})$ is approximately equal to $O(\mathcal{N}TB^2 \log \frac{\alpha \cdot K}{\gamma})$.

Algorithm 2 Approximation Algorithm

1: **Input:** K, α, c, γ, B.
2: **Output:** optimal solution to Problem (7).
3: **If** there is a $\Omega^{'}$ with $S(\Omega^{'}) \leq K$ and $\delta(\Omega^{'}) \leq B$ **then return** $\Omega^{'}$;
4: $\lambda_l^{'} = \alpha$, $\lambda_r^{'} = 0$, $\varepsilon = \frac{\gamma}{K}$;
5: **While** $\lambda_l^{'} - \lambda_r^{'} > \varepsilon$ **do**
6: $\quad \lambda_m^{'} = \frac{\lambda_l^{'} - \lambda_r^{'}}{2}$, $\quad \Omega^{'} \leftarrow \mathrm{DP}(p, \lambda_m^{'}, \alpha, B, \tau, \Gamma)$;
7: $\quad$ **If** $S(\Omega^{'}) \geq K$ and $S(\Omega^{'}) \leq c \cdot K$ **then return** $\Omega^{'}$
8: $\quad$ **If** $S(\Omega^{'}) > c \cdot K$ **then** $\lambda_r^{'} = \lambda_m^{'}$ **else** $\lambda_l^{'} = \lambda_m^{'}$.
9: **End while**
10: **Return** $\Omega^{'} \leftarrow \mathrm{DP}(p, \lambda_l^{'}, \alpha, B, \tau, \Gamma)$

V. Experiments

This section evaluates the effectiveness and efficiency of the proposed dGraphScan framework based on three real world datasets. Compared with other proposed techniques, dGraphScan outperforms in the subgraph detection.

A. Experimental Setting

Datasets: We consider water pollution detection, traffic congestion detection and haze detection as three case study scenarios.

(1) Water Pollution Dataset: The "Battle of the Water Sensor Networks" (BWSN) [28] provides a real world network with 12,527 vertices. Among them, there are 25 vertices with chemical contaminant plumes that are distributed in four different areas. The spread of these contaminant plumes on graph is simulated using the water network simulator EPANET that is used in BWSN for a period of 8 hours. For each hour, each vertex has a sensor that reports 1 if it is polluted; otherwise, reports 0. Except measuring the accuracy of subgraphs detection, we also test the robustness of subgraph detection methods to noises. We randomly selected $\mathcal{K}$ percent vertices, and flipped their sensor binary values, where $\mathcal{K}$ = 0, 2, 4, ..., 28, 30. The objective is to detect the set of polluted vertices. In order to apply nonparametric graph scan methods to this dataset, we map the sensors whose report values are 1s to the empirical p-value 0.15, and those whose report values are 0s to 1.0.

(2) Traffic Congestion Detection Dataset: This dataset is provided by 2012 Internet contest for Cloud & Mobile computing. Electronic map data and dynamic traffic data are included. The former provides the basic topological structure of urban road network, and the latter is derived from the real urban road traffic information collected from a large number of GPS sensors in the floating car (taxi). Travel time information with more than 100 thousand link chains are recorded and a connected traffic network with 10,603 vertices is constructed per hour. Based on real time speed and the standard of congestion, we can get all congestion links and then establish the ground truth. In this experiment, three traffic rush hours (7:00am, 8:00am and 9:00am) are selected and we calculate the corresponding empirical p-value for every vertex in separate time slice.

(3) Haze Event Detection Dataset: We randomly collect 10 percent of the whole Weibo data from Apr 11, 2014 to Jan 11, 2015, including 1,433,937,815 tweets. in total. After removing tweets that contain less than two terms from a dictionary of 68 terms about haze outbreaks collected from domain experts, we obtain 0.35 million tweets that are posted by 49,644 users. According to co-mentions in tweets and following relations, we construct a connected user-user network with 149,408 edges. Each user is geocoded with a province from location in profiles. For each day d and user u, we calculate the corresponding empirical p-value for each keyword using the strategy proposed in [22]. In total, we have 276 snapshot graphs, corresponding to the 276 days. Gold Standard Reports (GSR) of 9,384 official haze outbreak records (level $\geq$ 3) are collected from official websites (MEP), and an example of the GSR record is (Province = "Beijing", Country = "China", Day = "20-05-2014").

Our Method: We compare our proposed approach, named as dGraphScan, with several existing representative baseline methods. We employ 10-fold cross validation to identify the best combination of all the related parameters. Specifically, the parameter α_{max} is denoted as 0.15.

Comparison Methods: (1) We fist compare our proposed approach with five existing anomalous subgraphs detection methods, including NPHGS [22], DFS [29], Additive Graph Scan [30], NetSpot [2], and Meden [31]. (2) Then we compare the proposed method with two typical event detection methods (EventTree [27] and NPHGS) as a case study. We strictly follow the strategies recommended by the authors in their papers to estimate the related model parameters.

Performance Metrics: This study focuses on the evaluation of evolving anomalous subgraphs detection in dynamic networks for different methods.

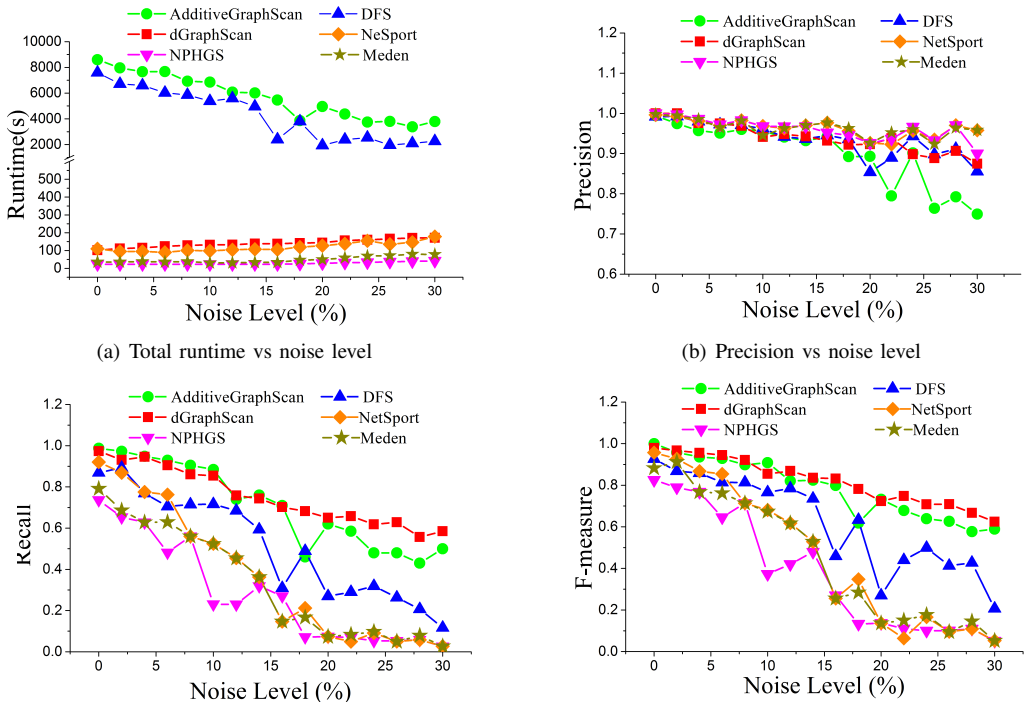
The related performance metrics include: (1) precision, (2) recall, (3) F-measure, (4) false positive rate (FPR), (5) true positive rate (TPR) for forecasting, (6) true positive rate for both detection and forecasting, (7) average lead time for forecasting, and (8) average lag time for detection.

The first three metrics are used for water pollution dataset and traffic congestion dataset in which the true anomalous subgraphs are known. The rest are used for the haze event detection dataset.

In the case study, for each method, the reported alerts are structured as tuples of (date, location), where "location" is defined at the province level. For each GSR event, we decide whether the method: (1) Had an alert in the province within 7 days before the event, which means to be "predicted"; (2) Had an alert in the province within 7 days after the event, which means to be "detected"; or (3) Had no alert in the province within 7 days before or after the event, which is "undetected".

B. Results of Evolving Subgraphs Detection

Water Pollution Detection: We run six algorithms (DFS, Additive Graph Scan, NPHGS, dGraphScan, NetSpot and Meden) on the water pollution dataset with different noise levels (0%, 2%, ..., 30%). The runtime is shown in Figure 3(a), we can see that the average runtime of Additive Graph Scan and DFS is far higher than those of the other four methods. The runtime of dGraphScan is similar to NetSpot, Meden and NPHGS, and nearly 20 times faster than the algorithm of Additive Graph Scan and DFS. Furthermore, the presion, recall and F-measure of dGraphScan mainly distributes in [0.7, 1.0] shown in Figure 3((b), (c), (d)). dGraphScan presents a better performance. For example, even if we introduce 10% noise into the dataset, dGraphScan detects truly contaminated vertices with precision greater than 0.9, the recall is near 0.8 and the F-measure is 0.9 or so. As shown in Figure 3(c), the recalls of the NPHGS, DFS, NetSpot and Meden sharply reduce from 1.0 to 0 with the increase of the noise. However, the recall of dGraphScan is similar to Additive Graph Scan



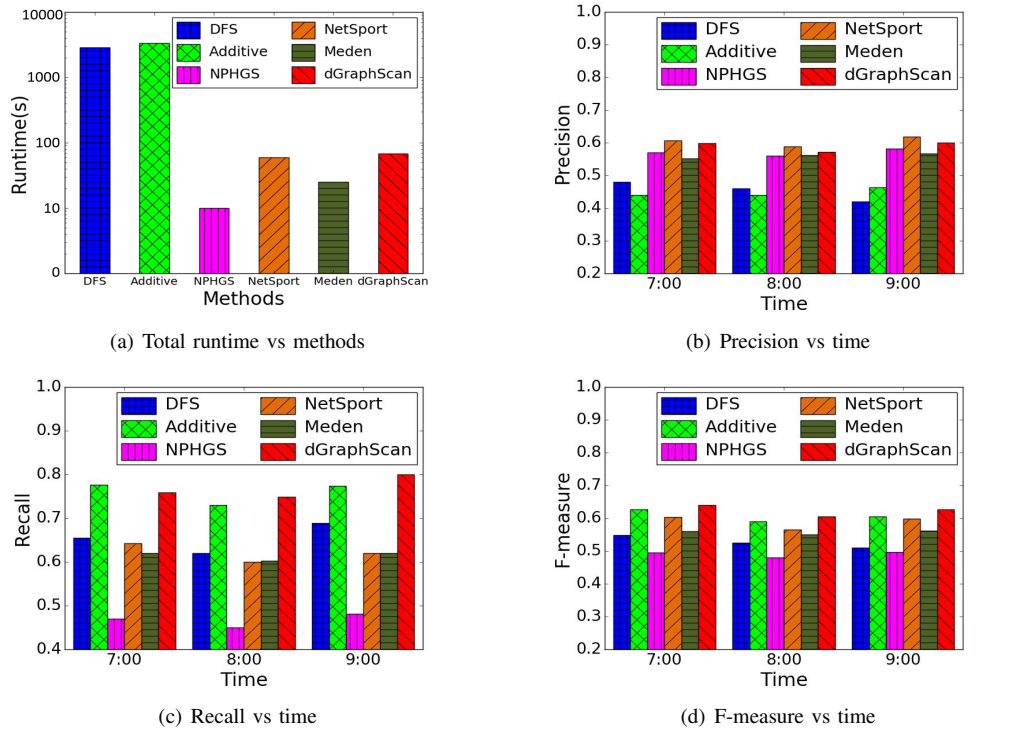
(a) Total runtime vs noise level

(b) Precision vs noise level

(c) Recall vs noise level

(d) F-measure vs noise level

Fig. 3. Comparison between dGraphScan and baseline methods based on the Water Pollution Dataset.



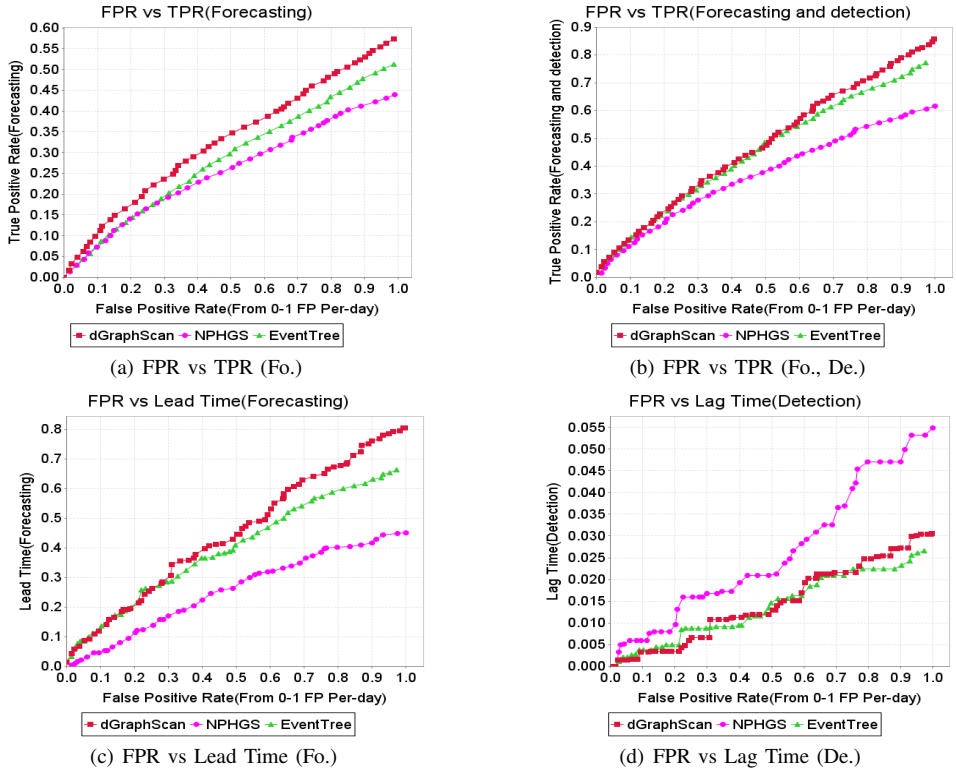
(a) Total runtime vs methods

(b) Precision vs time

(c) Recall vs time

(d) F-measure vs time

Fig. 4. Comparison between dGraphScan and baseline methods based on Traffic Congestion Dataset.



(a) FPR vs TPR (Fo.) (b) FPR vs TPR (Fo., De.)

(c) FPR vs Lead Time (Fo.) (d) FPR vs Lag Time (De.)

Fig. 5. Comparison between dGraphScan and representative event detection methods based on the Event Detection Dataset.

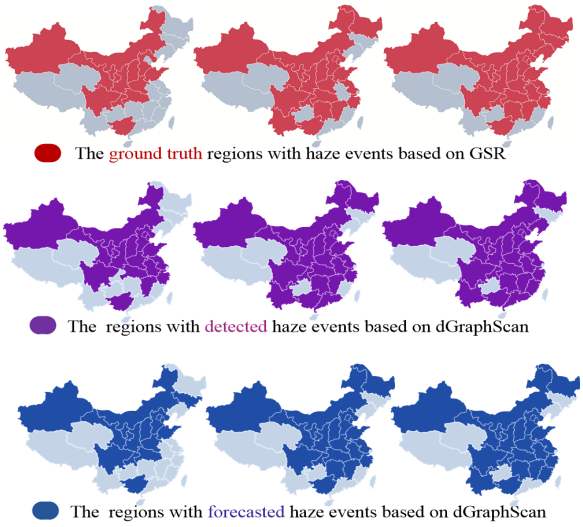


Fig. 6. An illustration of the comparison detected alert results based on dGraphScan and the ground truth of haze events from 2015-01-02 to 2015-01-04 in China. The first three, the middle three, and the last three refer to the ground truth, the detected and the forecasted results of haze events, respectively. Where each separate region prefers to a state (province of China). On a certain day, an alert of the province within the 7 days window before and after that day.

and better than those of the previous four algorithms. We also can find that dGrapgScan presents a higher F-measure than the baseline metheds in Figure 3(d). Considering the factors shown in Figure 3((b), (c), (d)), dGraphScan performs the best comprehensive accuracy nearly under different noise levels. In a word, the algorithm proposed in this paper presents the best comprehensive performance when it works on the water pollution dataset.

Traffic Congestion Detection: We run dGraphScan and baseline methods of subgraph detection to detect evolving traffic congestion subgraphs. The results, including runtime, precision, recall and F-measure, are shown in Figure 4. We can see that DFS and Additive Graph Scan also spend far more time than the other methods. In the aspect of F-measure, dGraphScan is similar to Additive Graph Scan and is better than Netspot, Meden, NPHGS and DFS. However, the runtime of Additive Graph Scan is almost 20 times slower than dGraphScan. Although the precision and recall of dGraphScan are occasional less than NetSpot and the Addictive Graph Scan respectively, dGraphScan presents a better comprehensive performance considering all aspects. As a result, dGraphScan can detect the evolving traffic congestions effectively and efficiently.

C. Case Study: Event Detection and Forecasting Based on Detected Evolving Subgraphs

For each time slice, each approach will output a detected user subgraph with an anomalousness score (the value of the objective function is maximized). A set of places are retrieved from the geocoded places of the users within this subgraph, within which each place leads to an alert with the place name, time slice, and an anomalousness score.

The results of the comparison between the proposed dGraphScan and two representative event detection methods are shown in Figure 5. And an illustration of dGraphScan results is shown in Figure 6. The Figure 5 shows that the comparison at various FPR for the target of detection and forecasting haze events. The results indicate that dGraphScan obtains much higher forecasting TPR, and much higher forecasting and detection TPR than all the baseline methods, and there is a trend that when the FPR increases, the margin between the TPR of dGraphScan and those of baseline methods consistently increases for both forecasting and detection. Specifically, based on event detection dataset, the margin for forecasting is more than 10% shown in Figure 5(a), and the margin for detection and forecasting is more than 10% shown in Figure 5(b). In addition, dGraphScan obtaines longer Lead Time, and nearly shorter Lag Time than all the baseline methods at various FPR.

VI. Conclusion and Future Work

This paper proposes an efficient framework, named dGraphScan, for evolving anomalous subgraphs detection in dynamic networks based on nonparametric scan statistics. In dGraphScan, we first generalize nonparametric scan statistics and optimize a new class of evolving subgraph scan statistic functions for the problem of evolving anomalous subgraphs detection and then present the proposed approximation algorithms. Compared with other proposed techniques, dGraphScan outperforms in evolving anomalous subgraphs detection problem. For future work, we will focus on extending dGraphscan to evolving anomalous subgraph detection in dynamic heterogeneous networks, where the vertices or edges may have different types and evolve over time.

VII. Acknowledgments

This work is supported by China 973 Fundamental R&D Program (No.2014CB340300), NSFC program (No.61472022, 61421003), SKLSDE-2016ZX-11, and partly by the Beijing Advanced Innovation Center for Big Data and Brain Computing. The corresponding author is Jianxin Li.

References

[1] M. Mongiovi, P. Bogdanov, and A. K. Singh, "Mining evolving network processes," in *ICDM*, pp. 537–546, IEEE, 2013.

[2] M. Mongiovi, P. Bogdanov, R. Ranca, E. E. Papalexakis, C. Faloutsos, and A. K. Singh, "Netspot: Spotting significant anomalous regions on dynamic networks," in *Proceedings of the 2013 SIAM International Conference on Data Mining*, pp. 28–36, 2013.

[3] S. Kumar and K. Das, "Localizing anomalous changes in time-evolving graphs," SIGMOD, 2014.

[4] J. Gao, C. Zhou, and J. X. Yu, "Toward continuous pattern detection over evolving large graph with snapshot isolation," *The VLDB Journal*, vol. 25, no. 2, pp. 269–290, 2016.

[5] P. Lee, L. V. Lakshmanan, and E. E. Milios, "Incremental cluster evolution tracking from highly dynamic network data," in *ICDE*, pp. 3–14, IEEE, 2014.

[6] S. Goel, A. Baykal, and D. Pon, "Botnets: the anatomy of a case," *Journal of Information Systems Security*, 2006.

[7] Q. Yan, Y. Zheng, T. Jiang, W. Lou, and Y. T. Hou, "Peerclean: Unveiling peer-to-peer botnets through dynamic group behavior analysis," in *INFOCOM*, pp. 316–324, IEEE, 2015.

[8] S. Ruehrup, P. Urbano, A. Berger, and A. D'Alconzo, "Botnet detection revisited: theory and practice of finding malicious p2p networks via internet connection graphs," in *INFOCOM*, pp. 3393–3398, IEEE, 2013.

[9] S. A. Myers, C. Zhu, and J. Leskovec, "Information diffusion and external influence in networks," in *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 33–41, ACM, 2012.

[10] M. Treiber and A. Kesting, "Calibration and validation of models describing the spatiotemporal evolution of congested traffic patterns," *arXiv preprint arXiv:1008.1639*, 2010.

[11] D. Helbing, M. Treiber, A. Kesting, and M. Schönhof, "Theoretical vs. empirical classification and prediction of congested traffic states," *The European Physical Journal B-Condensed Matter and Complex Systems*, vol. 69, no. 4, pp. 583–598, 2009.

[12] B. S. Kerner, H. Rehborn, M. Aleksic, and A. Haug, "Recognition and tracking of spatial–temporal congested traffic patterns on freeways," *Transportation Research Part C: Emerging Technologies*, vol. 12, no. 5, pp. 369–400, 2004.

[13] X. Ma, H. Xiao, S. Xie, Q. Li, Q. Luo, and C. Tian, "Continuous, online monitoring and analysis in large water distribution networks," in *ICDE*, pp. 1332–1335, IEEE, 2011.

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

[15] H. Djidjev, G. Sandine, C. Storlie, and S. Vander Wiel, "Graph based statistical analysis of network traffic," in *Proceedings of the Ninth Workshop on Mining and Learning with Graphs*, 2011.

[16] J. Neil, C. Hash, A. Brugh, M. Fisk, and C. B. Storlie, "Scan statistics for the online detection of locally anomalous subgraphs," *Technometrics*, vol. 55, no. 4, pp. 403–414, 2013.

[17] C. E. Priebe, J. M. Conroy, D. J. Marchette, and Y. Park, "Scan statistics on enron graphs," *Computational & Mathematical Organization Theory*, vol. 11, no. 3, pp. 229–247, 2005.

[18] D. B. Neill and J. Lingwall, "A nonparametric scan statistic for multivariate disease surveillance," *Advances in Disease Surveillance*, vol. 4, p. 106, 2007.

[19] M. Kulldorff, "A spatial scan statistic," *Communications in Statistics-Theory and methods*, vol. 26, no. 6, pp. 1481–1496, 1997.

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

[21] J. L. Sharpnack, A. Krishnamurthy, and A. Singh, "Near-optimal anomaly detection in graphs using lovasz extended scan statistic," in *Advances in Neural Information Processing Systems*, pp. 1959–1967, 2013.

[22] F. Chen and D. B. Neill, "Non-parametric scan statistics for event detection and forecasting in heterogeneous social media graphs," in *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 1166–1175, 2014.

[23] 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.

[24] R. H. Berk and D. H. Jones, "Goodness-of-fit test statistics that dominate the kolmogorov statistics," *Zeitschrift für Wahrscheinlichkeitstheorie und verwandte Gebiete*, vol. 47, no. 1, pp. 47–59, 1979.

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

[26] A. Gionis, M. Mathioudakis, and A. Ukkonen, "Bump hunting in the dark: Local discrepancy maximization on graphs," *IEEE Transactions on Knowledge and Data Engineering*, vol. 29, no. 3, pp. 529–542, 2017.

[27] P. Rozenshtein, A. Anagnostopoulos, A. Gionis, and N. Tatti, "Event detection in activity networks," in *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 1176–1185, 2014.

[28] A. Ostfeld, J. G. Uber, E. Salomons, J. W. Berry, W. E. Hart, *et al.*, "The battle of the water sensor networks (bwsn): A design challenge for engineers and algorithms," *Journal of Water Resources Planning and Management*, vol. 134, no. 6, pp. 556–568, 2008.

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

[30] S. Speakman, Y. Zhang, and D. B. Neill, "Dynamic pattern detection with temporal consistency and connectivity constraints," in *ICDM*, pp. 697–706, IEEE, 2013.

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