INVITED PAPER

Graph Anomaly Detection Based on Steiner Connectivity and Density

This article focuses on the important problem of anomaly detection in dynamic networks that evolve over time.

By Jose Cadena, Feng Chen, and Anil Vullikanti

ABSTRACT | Detecting "hotspots" and "anomalies" is a recurring problem with a wide range of applications, such as social network analysis, epidemiology, finance, and biosurveillance, among others. Networks are a common abstraction in these applications for representing complex relationships. Typically, these networks are dynamic-, i.e., they evolve over time. A number of methods have been proposed for anomaly detection in such dynamic network data sets, which are primarily based on changes in network properties. We provide a survey of the various formulations of anomaly detection in dynamic networks with a focus on "window-based" methods. Window-based methods first define a time window of past network snapshots to model normal behavior and then mark a snapshot as anomalous if it has significantly different patterns from those observed in the time window. We describe two classes of techniques: 1) generalizations of Steiner connectivity; and 2) dense subgraph mining. Both have been used extensively in window-based graph anomaly detection. We summarize the key problem formulations that have been studied using these approaches, and we describe details of some of the main techniques.

KEYWORDS | Approximation algorithms; dense subgraph mining; graph anomaly detection; graph mining; parameterized complexity; scan statistics

Manuscript received September 27, 2017; revised January 17, 2018; accepted February 26, 2018. Date of current version April 24, 2018. The work of J. Cadena and A. Vullikanti was supported in part by the Defense Threat Reduction Agency (DTRA) Comprehensive National Incident Management System (CNIMS) under Contracts HDTRA1-11-D-0016-0010 and HDTRA1-17-0118, and in part by the National Science Foundation (NSF) under Grants IIS-1633028 and ACI-1443054. The work of F. Chen was supported in part by NSF under Grants IIS-1750911 and IIS-1441479, and in part by the Army Research Office (ARO) under Grant W911NF1720129. *(Corresponding author: Jose Cadena.)*

J. Cadena was with the Department of Computer Science and the Biocomplexity Institute, Virginia Tech, Blacksburg, VA 24060 USA. He is now with The Machine Learning Group, Lawrence Livermore National Laboratory, Livermore, CA 94550 USA (e-mail: jcadena@bi.vt.edu).

F. Chen is with the Department of Computer Science, University at Albany–SUNY, Albany, NY 12222 USA (e-mail: fchen5@albany.edu;).

A. Vullikanti is with the Department of Computer Science and the Biocomplexity Institute, Virginia Tech, Blacksburg, VA 24060 USA (e-mail: vsakumar@vt.edu).

Digital Object Identifier: 10.1109/JPROC.2018.2813311

I. INTRODUCTION

Networks (or graphs) have become a popular abstraction for representing complex relationships in diverse applications, as diverse as social networks, systems biology, computer security, and finance [2], [31], [36], [61]. Anomaly detection is the task of finding a part of the graph (i.e., nodes, edges, subgraphs) where some "strange" or "unusual" behavior is taking place. What constitutes strange behavior depends on the nature of the network, and there has been a lot of work on considering different kinds of network properties. For instance, anomalies have been defined in terms of the edges of the network (i.e., anomalous interactions between nodes) [20], [49], [67], [78], [101], node features (i.e., members of the network who behave differently compared to other members) [4], [12], [91], and characteristics of different kinds of subgraphs [75], [94]. Anomalies have been considered in both static graphs and dynamic graphs, in which the nodes/edges or characteristics of nodes/edges (e.g., weights, features) change over time. See Fig. 1 for an illustration.

There has been extensive research on all these aspects, and there are multiple surveys that summarize the different kinds of approaches that have been considered in the literature, a lot of which are application driven. Akoglu *et al.* [5] provide a comprehensive survey of the approaches used for anomaly detection in static and temporal networks, as well as a taxonomy for this broad area of research. Shortly after, Ranshous *et al.* [82] "zoomed in" on temporal networks and gave a more detailed comparison of existing methods for these types of anomalies. However, these surveys do not go into significant detail about the technical methods used. We will not attempt to replicate these surveys. Instead, we briefly summarize the main categories of the approaches used; then, we delve deeper into two classes of methods, which rely on interesting graph theoretic properties,

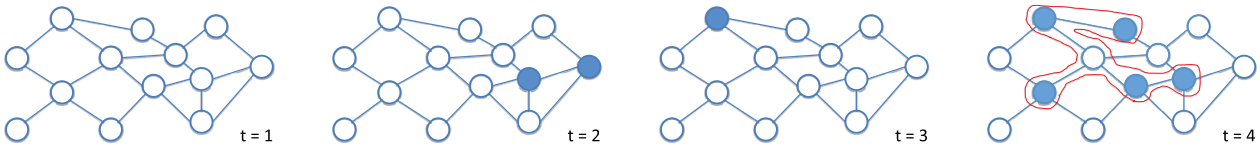

Fig. 1. ***Anomaly detection based on Steiner connectivity. Four snapshots of a network of sensors. A blue node (sensor) indicates pollution at that part of the network. However, individual sensors may become active due to noise. This is the case at times 2 and 3. However, time 4 shows a large subgraph of active sensors. This event detection problem can be cast as finding a connected subgraph with a high proportion of blue nodes–possibly connected by some white nodes. In this case, we would like to detect the graph circled in red at time 4.***

namely, 1) generalizations of Steiner connectivity; and 2) dense subgraphs. These problems are NP-hard, in general, and the methods based on these properties rely on rigorous algorithms and heuristics for finding near-optimal solutions. We describe some of the key families of techniques that have been developed and used in a number of applications. In particular, we discuss techniques based on the notion of fixed-parameter tractability for NP-hard problems [34], where the goal is to develop algorithms with running time $O(a^k q(n))$, where a is a small constant, k is a parameter, and $q(n)$ is a polynomial on n, the problem size. In other words, the complexity scales exponentially with the parameter k, but polynomially with the problem size. Such methods provide a promising way to deal with the NP-hardness of these optimization problems, but they have not been studied as much in the graph mining literature.

A. Main Categories of Graph Anomaly Research

We broadly follow the taxonomy of [5] and summarize it here. We first describe some graph theoretical notation that is used in the discussion below. A static graph $G = (V, E)$ consists of a set V of nodes (which represent entities in an application, e.g., people in a social network) and a set $E \subseteq V \times V$ of edges (which represent relationships between the nodes). In general, graphs are dynamic and change over time. We represent this by $\mathcal{G} = \{G^{(1)}, G^{(2)}, \ldots, G^{(T)}\}$, where $G^{(t)}(V^{(t)}, E^{(t)})$ is the graph at time t, also referred to as a "snapshot." In many applications, $V^{(t)} = V$ for all t, i.e., the node set is unchanged, but the edge set $E^{(t)}$ changes as a result of insertions or deletions. Each edge $e = (u, v)$ in $E^{(t)}$ has a weight $w^{(t)}(u, v)$ indicating the strength of the interaction between u and v at that time step—this can be positive or negative, in which case, it is referred to as a signed network. We drop the superscript indicating time when considering static graphs.

1) Static Graphs: Informally, the problem of anomaly detection on a static graph G calls for finding subgraphs $V' \subseteq V$ that are significantly different than most of the "normal" patterns observed in that graph. There are a number of ways to formalize such a difference. One line of work uses structural properties of a network; these methods define anomalies as subgraphs whose structure is different than the rest of the graph. For instance, the OddBall approach [4] uses a set of node-level features, such as the degree and local clustering, and it identifies a node as anomalous if the features differ significantly from the overall distribution. Another line of work leverages on the community of a graph to find anomalies. A community in a graph is loosely defined as a subset $V' \subseteq V$ of nodes that have many edges within V' (i.e., they are densely connected) and few edges to other nodes of the graph, i.e., to $V \setminus V'$. The community-based methods define anomalies as nodes or edges that do not clearly belong to any community; rather, these nodes act as "bridges" and lie in the boundary between two or more communities. An example of a community-based method is the work of Sun *et al.* [95], who consider communities based on random walks and define a link as anomalous if it connects nodes that have low likelihood of being in the same community.

Nodes or edges of a graph may also have attributes. For example, in a social network, attributes of a person (i.e., a node) would be their hometown, occupation, political and religious views, etc. This additional information may be used to define anomalous behavior. A well-known attribute-based method is the work of Noble and Cook [75]. The authors use a minimum description length (MDL) approach for finding frequent subgraphs—subgraphs with low compression cost—when each node has a label. The idea is that the opposite of "frequent" is anomalous, so graphs that are hard to compress are labeled as anomalous.

2) Dynamic Graphs: The problem of anomaly detection in a dynamic or time-evolving graph $\mathcal{G} = \{G^{(1)}, G^{(2)}, \ldots, G^{(T)}\}$ may be summarized as follows: find 1) a time stamp t or a time interval $[t_1, t_2]$, where an event or change point occurs; and 2) the subgraph $V' \subseteq V$, where this change occurs. Akoglu *et al.* [5] split the different approaches to anomaly detection in dynamic graphs into four categories.

- Feature based, which involve creating a summary for each snapshot (e.g., by converting it to a vector) and comparing consecutive snapshots using a distance function on the summaries. Distance above a certain threshold between two snapshots indicates a change point or anomaly between them [3].
- Decomposition based, which operate on the adjacency matrix representation of each snapshot, e.g., [3] and [96], or on the tensor representation of the full network time series, e.g., [11] and [58]. These methods use the eigenvectors or singular vectors of these representations to interpret anomalous events.

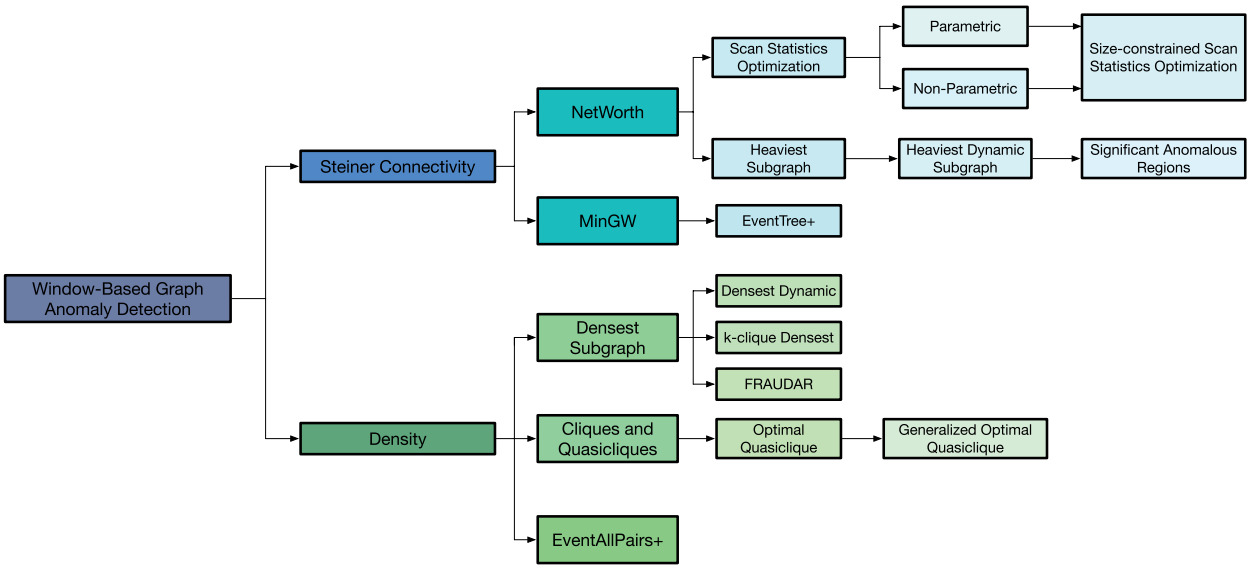

Fig. 2. ***Outline of our survey. We review problem formulations for graph anomaly detection based on Steiner connectivity and density. An arrow A → B indicates that problem B is a particular variant of problem A.***

- Community based, which define anomalies as snapshots in the time series whose community structure differs significantly from snapshots in the recent past. An example is [77], who propose a Bayesian model of community structure and a statistical test to detect change points in dynamic graphs. They use the snapshots in a time window $[t - W, t]$ to infer the community structure of the graph, and compare it with the community structure of snapshot $t + 1$, evaluating the significance by computing a Bayes factor.
- Window based, which define a time window of past snapshots to model normal behavior. Subsequent snapshots are marked as anomalous if they differ significantly from the patterns observed in the time window. Usually, in this category, the anomaly detection task is posed as an optimization problem where the goal is to find the subgraph(s) that maximizes some distance function between the current snapshot and the time window. Some examples include [27], [28], [78], [92], and [101].

B. Our Focus

We focus on the approach of window-based methods for anomaly detection in dynamic graphs, which are based on two classes of graph theoretical notions: 1) finding Steiner subgraphs with certain properties; and 2) finding dense subgraphs. Our survey consists of two parts, corresponding to these two approaches for finding anomalous subgraphs. For each part, we describe the problem formulations and briefly explain why the problems are hard in networks. We then discuss the main techniques and heuristics that have been developed for these problems and key applications to domains like biology, fraud detection, cybersecurity, public health, and bioinformatics. We finish with some conclusions and open questions. Fig. 2 provides a graphical summary of this survey.

II. ANOMALY DETECTION BASED ON STEINER CONNECTIVITY

One popular way of formalizing anomaly detection is through the family of Steiner connectivity problems, a general class of optimization problems with connectivity constraints. In Steiner connectivity problems, we have some nodes of interest—usually called terminals—that we would like to connect to each other. In order to do so, we may need to include nodes or edges that are not of interest to us—these are referred to as Steiner nodes. The challenge is to connect the terminals with as little extra overhead (i.e., few Steiner nodes) as possible. The basic Steiner subgraph formulations define a linear cost objective—e.g., minimize the sum of weights of the edges used in the solution. For anomaly detection, these formulations have been generalized to more complex objective functions in the form of network scan statistics.

The first proposed problem in the Steiner family is the minimum Steiner tree problem [52], where the goal is connecting a given set of terminal nodes in a graph using a tree of minimum cost.

Problem 1 (Minimum Steiner Tree): Given a graph $G(V, E)$ with edge costs or penalties $w : E \to \mathbb{R}^+$ and a set of terminal nodes $S \subset V$, find a connected subgraph $T(V', E')$ that includes all the terminals, i.e., $S \subseteq V'$, and minimizes the cost or total weight of the edges $\sum_{e \in E'} w(e)$.

There are many variations of this basic problem. See [48] for a compendium of Steiner connectivity formulations. Here,

we will focus on the prize-collecting versions, which have been used to model the anomaly detection task. The basic Steiner connectivity problem requires all the terminal nodes to be connected. The prize-collecting version relaxes this by considering a prize $\pi(v)$ for each node v. The objective function has two components: the sum of the prizes of the nodes not included in the solution (the "lost prize"), and the cost of the edges picked in the solution. There are two versions of these problems, depending on the specific objective function.

Problem 2 [Prize Collecting Steiner Tree (MinGW)]: Given a graph $G(V,E)$ with edge penalties $w : E \to \mathbb{R}^+$ and node penalties $\pi : V \to \mathbb{R}^+$, find a connected subgraph $T(V',E')$ that minimizes the cost of the tree plus the lost prize

$$\sum_{v \in V \setminus V'} \pi(v) + \sum_{e \in E'} w(e).$$

Problem 3 [Prize Collecting Steiner Tree (NetWorth)]: Given a graph $G(V,E)$ with edge penalties $w : E \to \mathbb{R}^+$ and node prizes $\pi : V \to \mathbb{R}^+$, find a connected subgraph $T(V', E')$ that maximizes the prize minus the cost of the tree

$$\sum_{v \in V'} \pi(v) - \sum_{e \in E'} w(e).$$

1) Anomalies as Heavy Subgraphs: The models for graph anomaly detection discussed in this section are all based on the Steiner connectivity principle of finding connected subgraphs with high prize and as little cost as possible. Nodes or edges that are deemed to be anomalous or interesting will be assigned a high prize, whereas normal or uninteresting ones will be assigned a low prize or even a penalty. The goal then becomes finding an optimal Steiner subgraph as in Problems 2 and 3. The formulations in this category differ on how the anomaly scores are assigned. In Fig. 1, we show a generic example of the anomaly detection task modeled as a Steiner connectivity problem.

A. Models and Problem Formulations

1) Graph Scan Statistics: As observed in [92], scan statistics involve formalizing a notion of "anomalousness" for a subset of data, and then scanning through the data to efficiently find a subset that optimizes an anomaly score. Originally, scan statistics were developed for disease surveillance in spatial data and involved finding simple regions, such as disks [59], [66], [71], [72], [74]. Later, scan statistics were extended to network data by considering scores for connected subgraphs.

Given a graph $G(V,E)$, we assume each node $v \in V$ has two associated values, which vary with time: 1) a baseline count $b^t(v)$, which indicates the count that we expect to see at the node v at time t, e.g., the number of people in a county corresponding to node v; and 2) an event count or observation $c^t(v)$, which indicates how many occurrences of an event of interest are seen at the node, e.g., the number of cases of a disease in a county. For simplicity, we omit time in our notation, but all the techniques presented below extend easily to network streams.

The methodology of scan statistics formalizes anomaly detection as a hypothesis testing problem. Under the null hypothesis H_0, it is "business as usual," and the event counts for all nodes are generated proportionally to their baseline counts. Under the alternative hypothesis $H_1(S)$, counts of a majority of the vertices are generated (again) with rate proportional to the baseline counts, but there exists a small connected subset $S \subseteq V$ of vertices for which the counts are generated at a higher rate than expected. Then, the goal is to find a set of vertices S that maximizes an appropriate scan statistic function $F(S)$ that compares event counts to baseline counts

$$F(S) = F(C(S), B(S), \theta)$$

where $C(S) = \sum_{v \in S} c(v)$ is the total event count or weight of S, $B(S) = \sum_{v \in S} b(v)$ is the baseline count of the set, and θ represents possible additional arguments to F.

Depending on the assumptions that are satisfied by the data, there are two broad types of scan statistics: parametric and nonparametric.

Parametric scan statistics assume that counts observed at each node are generated from some parameterized distribution [27], [29], [33], [53], [64], [66], [74], [79], [83], [99]. Common choices are distributions from the exponential family, such as Poisson or Normal, and the scan statistic is typically the log-likelihood ratio

$$F(S) = \log\left(\frac{\mathrm{P}(C(S) \mid H_1(S))}{\mathrm{P}(C(S) \mid H_0)}\right).$$

A well-known example of parametric scan statistics is the Kulldorff statistic commonly used in disease surveillance [35], [59], [60], [73], which is defined as

$$C(S)\log\left(\frac{C(S)}{B(S)}\right) + (C(V) - C(S)) \times \log\left(\frac{C(V) - C(S)}{B(V) - B(S)}\right) - C(V)\log\left(\frac{C(V)}{B(V)}\right)$$

with $\theta = (C(V), B(V))$. The extension to temporal data is easily obtained by defining $B(S)$ and $C(S)$ as the aggregate baseline and event counts over some time window T : $B(S) = \sum_{i=1}^{T} \sum_{v \in S} b^t(v)$ and $C(S) = \sum_{i=1}^{T} \sum_{v \in S} c^t(v)$. We refer to [59], [73], and [81] for discussion on the strengths and limitations of parametric scan statistics.

Nonparametric scan statistics do not assume an underlying distribution or process on the graph. Instead, they first estimate a p-value for each vertex based on empirical calibration by comparing the current features of this vertex—$c^t(v)$ and $b^t(v)$—with its features in the historical data—$c^{t-T}, \ldots, c^{t-1}(v)$ and $b^{t-T}, \ldots, b^{t-1}(v)$ for some time window size T.

Under mild assumptions [28, Th. 1], the calibrated p-values are uniform on $[0,1]$ if there is no anomalous activity among them. The problem of anomaly detection is then formalized as a hypothesis testing problem for

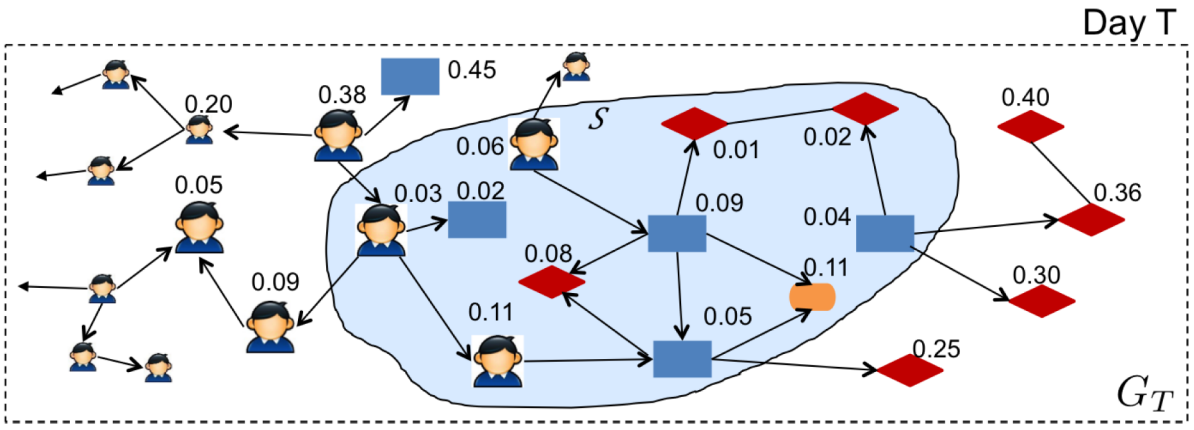

Fig. 3. ***Example of scan statistics in a Twitter network of users, tweets (blue squares), keywords (red diamonds), and locations (yellow rounded square). For a significance level of $\alpha = 0.05$, the subgraph S contains 6 out 11 significant nodes (i.e., p-value below α). Under the assumption that p-values are uniformly distributed under "normal" conditions, a scan statistic will assign a high score F(S) to this subgraph.***

testing whether the empirical p-values are uniformly distributed on $[0,1]$ [73], [81], [87]. An example of a nonparametric function is the Berk–Jones (BJ) scan statistic [18] used for civil unrest events and network intrusion detection [28], [66]. In this setting, each node v has a p-value $p(v) \in [0,1]$, and, for a significance level α, the event count $w(v)$ is 1 if $p(v) < \alpha$ (i.e., the node is significant) and 0 otherwise. This scan statistic is defined as

$$F(S) = \max_{\alpha \le \alpha_{\max}} |S| \left[\frac{W(S)}{|S|} \log\left(\frac{W(S)/|S|}{\alpha} \right) + \left(1 - \frac{W(S)}{|S|} \right) \log\left(\frac{1 - W(S)/|S|}{1 - \alpha} \right) \right]$$

with $\theta = \alpha_{max}$.

The abstract problem here is to find a connected subgraph S that maximizes one of the functions $F(S)$ described above.

Problem 4 (Network Scan Statistic Optimization): Given a graph $G(V,E)$, a score function $F(\cdot)$, and the associated counts for the model, the objective is to find a connected subset $S \subseteq V$ that maximizes $F(S)$.

Cadena *et al.* [23] show that this problem is NP-hard, by reduction from Steiner tree. Intuitively, in Problem 4, we want to connect a set of "interesting" nodes—for instance, nodes with p-value below α—while using as few noninteresting nodes as possible. We note that this hardness result contrasts with the case without any connectivity requirement, where the optimal value of the scan statistic can be computed in polynomial time because of a linear ordering property [73]. Motivated by this, a size-constrained version of the problem is introduced in [22], which can be solved using a fixed parameter tractable algorithm.

Problem 5 (Network Scan Statistic Optimization With Size Constraint): Given a graph $G(V,E)$, a score function $F(\cdot)$, the associated counts for the model, and a size parameter k, the objective is to find a connected subset $S \subseteq V$, $|S| \le k$, that maximizes $F(S)$.

We illustrate the general methodology of scan statistics in Fig. 3. The example represents a heterogeneous Twitter network in which each node has a specific type: user, tweet (blue squares), keyword (red diamonds), and geographical location (yellow rounded square). Every node has a p-value, and we have identified a potentially anomalous connected subgraph S. For a significance level of $\alpha = 0.05$, S contains 6 out 11 significant nodes (i.e., p-value below α), which is well above the expected value under the assumption of uniformly distributed p-values. A scan statistic that assumes uniformity will assign a high score $F(S)$ to this subgraph. Table 1 shows examples of commonly used scan statistics functions. We discuss applications in Section II-C.

2) Heaviest Dynamic Subgraph and Extensions: Bogdanov *et al.* [20] propose another method based on Steiner connectivity. In their setting, we are interested in finding a subgraph of anomalous or interesting edges rather than nodes.

The authors define an edge-evolving network over a set $\mathcal{T} = \{t_1, \ldots, t_2\}$ of timestamps as a tuple $(G(V,E), W, \mathcal{T})$, where 1) $G(V,E)$ is an undirected graph; and 2) $W = \{w^{t_1}, \ldots, w^{t_2}\}$ is a family of edge weight functions; $w^t(e)$ denotes the weight of an edge at time t, and this weight can be positive or negative. Informally, the weight of an edge represents its importance or anomalousness. For example, a positive edge may indicate increased interaction between two users in an online social network. Mongiovì *et al.* [69] define $w^t(e) = -\log p^t(e) / \alpha$, where $p^t(e)$ is the p-value associated with the edge at time t, and α denotes a significance level threshold. For two timestamps i, j, such that $t_1 \le i \le j \le t_2$, we say that $[i,j]$ is a subinterval of $[t_1, t_2]$ and $w^{[i,j]}(e) = \sum_{t \in [i,j]} w^t(e)$. A temporal subgraph of G is a pair $\left(G'(V', E'), [i,j]\right)$, where G' is a connected subgraph of G and $[i,j]$ is a subinterval of $[t_1, t_2]$. The score of a subgraph G' in the interval $[i,j]$ is the sum of the weights of the edges in E' during the interval *score* $\left(G', W, [i,j]\right) = \sum_{e \in E'} \sum_{k=i}^{j} w^k(e)$.

Table 1 Commonly-Used Scan Statistics Functions

Non-Parametric Scan Statistics The following definitions are by default, unless otherwise indicated: $p(v)$ refers to the p-value of node v, $N(S)=\|S\|, W(S,\alpha)=\sum_{v\in S} I(p(v)\le\alpha)$, where $I(\text{True})=1$ and $I(\text{False})=0$.	
Name	**Original Form**
Berk-Jones [18]	$F(S)=\max_{\alpha\le\alpha_{max}} N(S)KL(\frac{W(S,\alpha)}{N(S)},\alpha)$
Higher Criticism [32]	$F(S)=\max_{\alpha\le\alpha_{max}} \frac{W(S,\alpha)-N\alpha}{\sqrt{N\alpha(1-\alpha)}}$
Kolmogorov-Smirnov [102]	$F(S)=\max_{\alpha\le\alpha_{max}} \sqrt{N(S)}\cdot\left(\frac{W(S,\alpha)}{N(S)}-\alpha\right)$
Anderson-Darling [38]	$F(S)=\max_{\alpha\le\alpha_{max}} \sqrt{N(S)}\cdot\left(\frac{W(S,\alpha)}{N(S)}-\alpha\right)/\sqrt{\frac{W(S,\alpha)}{N(S)}\cdot\left(1-\frac{W(S,\alpha)}{N(S)}\right)}$
Jager-Wellner [54]	$F(S)=\max_{\alpha\le\alpha_{max}} \sqrt{N(S)}\cdot\left(1-\sqrt{\frac{N_\alpha(S)}{N(S)}}\cdot\alpha-\sqrt{(1-\frac{N_\alpha(S)}{N(S)})(1-\alpha)}\right)$
Stochastic Ordering of p-Values [8]	$F(S)=N(S)\int_0^{\alpha_{max}}\frac{(W(S,\alpha)/N(S)-\alpha)^2}{\alpha(1-\alpha)}\mathrm{d}\alpha$
Fisher's Test [41]	$F(S)=-\sum_{v\in S}\log p(v)/N(S)$
Truncated Fisher's Test	$F(S)=\max_{\alpha\le\alpha_{max}} -\frac{\sum_{v\in S} I(p(v)\le\alpha)\log p(v)}{N(S)}$
Weighted Fisher's Test	$F(S)=-\sum_{v\in S}\log(w(v)p(v))/\sum_{v\in S} w(v)$, where $w(v)$ is the predefined weight of vertex v
Stouffer's Test [93]	$F(S)=-\frac{\sum_{v\in S}\Phi^{-1}(1-p(v))}{\sqrt{N(S)}}$, where $\Phi^{-1}(\cdot)$ refers to the inverse cumulative density function of standard Gaussian distribution
Edgington's Test [37]	$F(S)=-\sum_{v\in S}\log p(v)/N(S)$
Parametric Scan Statistics The following definitions are by default, unless otherwise indicated: $C(S)=\sum_{v\in S} c(v)$, $B(S)=\sum_{v\in S} b(v)$	
Positive Elevated Mean Scan Statistic [81]	$F(S)=\sum_{i\in S} x_i/\sqrt{N(S)}$
Elevated Mean Scan Statistic [81]	$F(S)=(\sum_{i\in S} x_i)^2/N(S)$
Expectation-based Poisson Scan Statistic [73]	$F(S)=C(S)\log(C(S)/B(S))+B(S)-C(S)$
Kulldorff Scan Statistic [59]	$F(S)=C(S)\log\left(\frac{C(S)}{B(S)}\right)+(C-C(S))\log\left(\frac{C-C(S)}{B-B(S)}\right)-C\log\left(\frac{C}{B}\right)$, where $C=\sum_{v\in\mathbb{V}} c(v)$ and $B=\sum_{v\in\mathbb{V}} b(v)$
Expectation-based Gaussian Scan Statistic [73]	$F(S)=(C(S)-B(S))^2/(2B(S))$, where $\sigma(v)$ refers to the standard deviation of $c(v)$ that is calibrated based on its historical observations, $C(S)=\sum_{v\in S}(c(v)b(v))/\sigma(v)^2$, and $B(S)=\sum_{v\in S} b(v)/\sigma(v)^2$
Expectation-based Exponential Scan Statistic [73]	$F(S)=B(S)\log(B(S)/C(S))+C(S)-B(S)$, where $C(S)=\sum_{v\in S} c(v)/b(v)$, $B(S)=\|S\|$
Spatial Scan Statistic for Multinomial Data [55]	$F(S)=\sum_k\{C_k(S)\log(\frac{C_k(S)}{C(S)})+(C_k-C_k(S))\log\frac{C_k-C_k(S)}{C-C(S)}\}-\sum_k C_k\log(C_k/C)$, where $C_k(S)$ refers to the count of vertices of category k, $C(S)=\|S\|$, and $C=\|\mathbb{V}\|$

Bogdanov *et al.* model the anomaly detection task as the following optimization problems.

Problem 6 [Heaviest Dynamic Subgraph (HDS)]: Given an edge-evolving network $(G,W,\mathcal{T})$, the objective of the heaviest dynamic subgraph problem is to find a temporal subgraph $(G',[i,j])$, over all possible subgraphs G' of G and subintervals $[i,j]$ of $\mathcal{T}$, such that $score(G',W,[i,j])$ is maximized.

Problem 7 [Heaviest Subgraph (HS)]: The objective of the heaviest subgraph problem for a given $(G,w^{[t_1,t_2]})$ is to find a temporal subgraph G' of maximum score in a fixed interval $[t_1,t_2]$.

Intuitively, the goal is to find a subgraph and time interval where many of the edges have positive weight. However, as usual in Steiner connectivity problems, we are willing to include negative edges in the solution if it helps us to connect two components of high positive weight and improve on the objective function. Bogdanov *et al.* [20] note that the HS problem is equivalent to NetWorth (Problem 3). In Fig. 4, we show an example reduction from HS to NetWorth. The main idea is to replace all the adjacent positive edges by nodes of prize equal to the total edge weight and make the remaining edges have positive weight.

In the HDS problem, we are interested in finding one single subgraph and time interval that has the highest objective value. Mongiovì *et al.* consider an extension of the problem where we want to find as many nonoverlapping heavy temporal subgraphs as possible [69].

Problem 8 [Significant Anomalous Regions (SAR)]: Given an edge-evolving network $(G,W,\mathcal{T})$ and a threshold τ, the objective is to find a set of regions (temporal subgraphs) $\mathcal{R}=\{R_1,R_2,\ldots,R_k\}$ in decreasing order of scores, such that the score of each region R_i, without considering the score of positive edges overlapping with higher scoring regions, is at least τ.

3) EventTree+: Another Steiner-based method was recently proposed by Rozenshtein *et al.* [84]. The authors define activity networks, where each node v has a positive prize $\pi(v)$ and each edge e has a positive cost or distance $w(e)$. Node weights model the importance or intensity of the activity at that node, for instance, the number of posts made by a user in a social network, and we learn them from a collection of past observations. The goal is to find a subgraph that optimizes the tradeoff between prize and distance. In particular, the authors propose to find a subset of nodes $S\subseteq V$ that minimizes

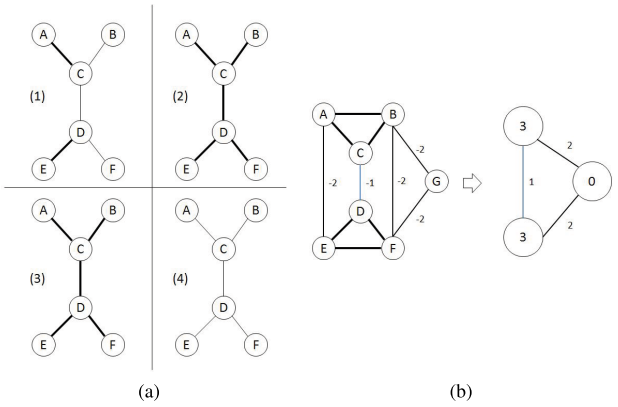

Fig. 4. ***(a) Example of an instance of HDS with four time intervals. Thick edges have weight +1 and thin edges have weight −1. The HDS includes all the edges and spans subinterval [2, 3]. For comparison, the HS in subinterval [1,1] has a score of 1 by using either (A, C) or (D, E). (b) Reduction of an instance of HS to an instance of NetWorth. Nodes A, B, and C are merged into a node of prize 3 [weights of (A, B), (B, C), and (C, A)]. Nodes D, E, and F are merged similarly. Negative edges become positive.***

$$\lambda\Pi(V\backslash S) + W(S)$$

where $\Pi(\cdot)$ and $W(\cdot)$ are functions of the total prize and distance of S, respectively, and λ is a regularization parameter that controls the tradeoff between the two functions. When $\Pi(S) = \sum_{v\in S}\pi(v)$ and $W(S)$ is the total cost of the edges used to connect S, the objective becomes MinGW. The authors refer to this particular case as EventTree+.

B. Methods and Techniques

Steiner connectivity problems are computationally challenging in general. In particular, the NetWorth problem, which all the models above are related to, cannot be approximated to a constant factor [17]. Common techniques to address this strong complexity result involve either 1) solving an easier Steiner problem similar to NetWorth; or 2) using simple fast heuristics with good empirical performance.

1) Solving MinGW Instead of NetWorth: Despite the inapproximability results for NetWorth, the MinGW problem is somewhat easier and admits an approximation algorithm. Goemans and Williamson proposed a 2-approximation[1] to the MinGW objective [44] with a time complexity of $O(n^2 \log n)$. This algorithm tends to be used as a heuristic for NetWorth. For example, Rozenshtein *et al.* [84] use it to solve EventTree+, and Bogdanov *et al.* [20] mention that the Goemans–Williamson (GW) algorithm can be used as a subroutine in their proposed method for solving the HDS problem.

[1]An algorithm for a minimization problem is an α-approximation if it can find a solution within a factor $\alpha > 1$ of the optimum solution for any instance of the problem.

We give a brief review of the primal-dual schema of Goemans and Williamson [44] for MinGW. Let $(G(V, E), w, \pi)$ be an instance of MinGW, x_e be a variable for each edge, and z_T be a variable for each subset of nodes. The rooted version of the problem, i.e., when a root node r must be included in the solution, can be formulated as the following integer program (MinGW-IP)

$$\begin{aligned}
\text{minimize} \quad & \sum_{e\in T} w_e x_e + \sum_{T\subseteq V-\{r\}} z_T \pi(T) \\
\text{subject to} \quad & \sum_{e\in\delta(S)} x(e) + \sum_{T\supseteq S} z_T \geq 1 && \forall S \subseteq V - \{r\} \\
& \sum_{T\subseteq V-\{r\}} z_T \leq 1 && \forall S \subseteq V' - \{r\} \\
& x_e, z_T \in \{0,1\} && \forall e \in E,\ T \subseteq V' - \{r\}.
\end{aligned}$$

It can be shown that this integer program is equivalent to MinGW. Intuitively, $x_e = 1$ if an edge is selected in a solution to MinGW, and $z_T = 1$ only for the set T of nodes not spanned by the edges. The first constraint ensures that the subset $S \subseteq V - \{r\}$ is either part of the solution, in which case at least one of the edges in its cut $\delta(S)$ must be selected, or else S is a subset of not-spanned nodes T. The second constraint ensures that only one subset of nodes is not spanned.

The linear relaxation (MinGW-LP) of (MinGW-IP) is obtained by replacing the constraints $x_e \in \{0,1\}$ and $z_T \in \{0,1\}$ by $x_e \in [0,1]$ and $z_T \in [0,1]$. The dual (MinGW-D) of (MinGW-LP) is

$$\begin{aligned}
\text{maximize} \quad & \sum_{S\subseteq V-\{r\}} y_S \\
\text{subject to} \quad & \sum_{S:e\in\delta(S)} y_S \leq w_e && \forall e \in E \\
& \sum_{S\subseteq T} y_S \leq \pi(T) && \forall T \subseteq V - \{r\} \\
& y_S \geq 0 && \forall S \subseteq V - \{r\}.
\end{aligned}$$

The GW algorithm tries to maximize the dual program (MinGW-D). The main idea is to grow the y_S variables as much as possible; that is, until either the first or second constraint becomes tight. The algorithm has two phases: growth and pruning. In the growth phase, we maintain a set of clusters. If a cluster has nonzero y_S, we say that it is active; otherwise, the cluster is inactive. Initially, all nodes are in active singleton clusters, except for the root, which is inactive. Let S_u denote the cluster containing node u. The algorithm maintains a quantity d_u for each node u, which captures the sum of all dual variables y_S, such that $u \in S$ over the past rounds. For any edge $e = (u, v)$, the algorithm will ensure that $d_u + d_v \leq w_e$. In a given round, the dual variables associated with all the active clusters grow at the same rate until one of the following events happen. 1) For some edge $e = (u, v)$ with S_u being active and S_v being the root component, we have $d_u + d_v = w_e$. In this case, we say that the edge (u, v) becomes tight, S_u is merged with the root cluster, and it becomes inactive. 2) For some edge $e = (u, v)$ with

S_u and S_v being distinct active clusters, we have $d_u + d_v = w_e$. In this case, we say that the edge (u, v) becomes tight, and the clusters S_u and S_v are merged to form a new cluster. 3) For some cluster T, we have $\sum_{S \subseteq T} y_s = \pi(T)$. In this case, the cluster T becomes inactive. The growth phase ends when all the clusters are inactive, and the solution returned is the set of tight edges F. In the pruning phase, we find a tree $F' \subset F$ by discarding edges whose removal does not degrade the quality of the initial solution.

As mentioned above, one drawback of this algorithm is its high running time of $O(n^2 \log n)$. There have been efforts to make it more scalable by using more efficient data structures. Cole *et al.* [30] develop an algorithm that runs in time $O(\epsilon(n+m)\log^2 n)$, where ϵ is a parameter that controls the tradeoff between running time and approximation error. The main idea is to avoid having an edge whose two endpoints are both active clusters; this technique is referred to as edge splitting by the authors. Initially, every edge (u, v) in the graph is "split" in half by adding an artificial node t between u and v; this effectively creates two edges (u, t) and (t, v). In every growth phase round, if two clusters are merged and, as a result, two active clusters become neighbors, the edge is split again ensuring that at most one of the endpoints is active. Of course, the edge cannot be split indefinitely; an edge that cannot be split is called a terminal. The user determines how many times an edge will be split via the parameter ϵ.

2) Fixed Parameter Tractable Algorithms: Numerous heuristics have been developed for Problem 4, including some based on Steiner subgraphs, which are discussed later. However, no rigorous algorithms are known, in general. We show that Problem 4 is fixed parameter tractable [34], by giving a solution to Problem 5. This means that we can find an optimal solution in time $O(a^k q(n, m))$, where a is a constant and $q(n, m)$ is a polynomial in n and m. That is, the running time is polynomial on the size of the graph, but exponential on the solution size. In contrast, a "brute force" approach would need $\binom{n}{k} = O(n^k)$ time to examine every possible connected subset of nodes of size at most k.

Cadena *et al.* [22] follow this approach to propose the only existing methods for scan statistics optimization in general graphs with rigorous theoretical guarantees. Their algorithms ColCodeNP and ColCodeP find a subgraph S with optimal score $F(S)$ of size at most k, where k is a parameter, in time $O((2e)^k m \log(n^2/\epsilon))$, where ϵ is a probability of failure. The authors also propose a preprocessing step that makes it possible to discover large anomalous subgraphs while keeping the parameter k below 10, without losing the approximation guarantees.

The algorithms of [22] rely on the color-coding technique of Alon *et al.* [7]. The idea is to color the nodes of the graph uniformly at random using $K = \{1, \ldots, k\}$ colors and restrict the search to "colorful" solutions, which are subgraphs with distinctly colored nodes. This immediately leads to an efficient algorithm because: 1) colorful solutions can be computed using a simple dynamic program; and 2) if the coloring is done randomly, there is a reasonable probability that the optimal solution is colorful. By solving the dynamic program many times with different random color assignments, the optimal solution of size k will eventually be colorful, and we will find it.

As a concrete example, we show how to use color coding to solve the parameterized version of Problem 3, i.e., maximize NetWorth over all subgraphs of at most k nodes. Let $K = \{1, \ldots, k\}$ be a color set. We define a coloring as a function $col: V \to K$ that maps nodes to colors; $col(u)$ is the color of node u. For a given set of nodes S and a color set $T \subseteq K$, we say that a subset of nodes S is colorful (with respect to T) if every node in S has a distinct color from T; that is, for all $u, v \in S$, $col(u) \neq col(v)$ and $col(u), \text{col}(v) \in T$.

We use a dynamic program to find subgraphs that are colorful and maximize the NetWorth objective. For every node v and color set $T \subseteq K$, let $\text{OPT}(v, T)$ be the NetWorth objective of the optimal tree that 1) contains v; and 2) is colorful with respect to T. When the color set is a singleton, this quantity is easy to compute

$$\text{OPT}(v, \{s\}) = \begin{cases} w(v): & col(v) = s \\ -\infty: & col(v) \neq s. \end{cases}$$

Now, for a color set T of size greater than 1, we can compute $\text{OPT}(v, T)$ recursively

$$\text{OPT}(v, T) = \max_{\substack{u:(u,v)\in E \\ T_1, T_2: T_1 \cap T_2 = T}} (\text{OPT}(v, T_1) + \max\{\text{OPT}(v, T_2) - w(v, u), 0\})$$

where the outer maximum is over all possible partitions of T into two subsets T_1 and T_2, and all possible neighbors u of v. The inner maximum denotes that we only want to connect two subtrees if the net improvement in objective value is positive. The final answer is $\text{OPT} = \max_{v \in V} \text{OPT}(v, K)$.

We can verify that this dynamic program correctly returns the best NetWorth over all connected subgraphs of size at most k, as long as the optimal subgraph is colorful. By repeating the algorithm for many random colorings, this subgraph will indeed be colorful with high probability. The probability that any tree of k nodes, in particular, the optimal tree, is colorful is

$$\frac{k!}{k^k} \geq e^{-k}$$

and the probability that it is not colorful in any of t random colorings is

$$\left(1 - \frac{1}{e^k}\right)^t.$$

For $\epsilon > 0$, let $t = -e^k \ln \epsilon$, and we can bound the probability of not finding the optimal solution

$$\left(1 - \frac{1}{e^k}\right)^{-e^k \ln \epsilon} \leq e^{(-e^{-k})(-e^k \ln \epsilon)} \leq \epsilon.$$

Theorem 2.1 (Follows From [7]): The color-coding technique yields an algorithm for the parameterized NetWorth

problem that finds the optimal NetWorth value with probability at least ϵ. The time complexity of the algorithm is $O\big((2e)^k m \log(1/\epsilon)\big)$ and the memory requirement is $O(2^k n)$.

We note that this parameterized complexity approach is also applicable to the other problem formulations presented in this section. We briefly describe the main ideas for the case of nonparametric scan statistics from [22], using the BJ scan statistic as an example. The first idea is that the BJ statistic is an increasing function of $W(S)$ if $W(S)/|S| \geq \alpha$ and $|S|$ is constant. Consider a coloring using a set K as before. For a node v and subset of colors $T \subseteq K$, we let $\mathrm{OPT}(v, T) = \max_S W(S)$, where the maximization is over all connected and colorful sets $S \subseteq V$, such that $v \in S$, $|S| = |T|$, and $\{col(u) : u \in S\} = T$. In other words, we only consider a set S if each node in the set has a distinct color from T. $\mathrm{OPT}(v, T)$ can be computed by a dynamic program with the following recurrence. For any node v and color s, $\mathrm{OPT}(v, \{s\}) = w(v)$ if $col(v) = s$, else $\mathrm{OPT}(v, \{s\}) = -\infty$. If $|T| \geq 2$

$$\mathrm{OPT}(v, T) = \max_{\substack{u:(u,v)\in E \\ T_1, T_2 \subseteq T}} \{\mathrm{OPT}((v, T_1) + \mathrm{OPT}((u, T_2)\}$$

where the maximum is over all partitions $T_1 \cup T_2 = T$ and all neighbors u of v. This can be used to obtain a bound similar to Theorem II.1 for nonparametric scan statistics. The general method can be extended to parametric scan statistics as well, but additional information needs to be maintained in the dynamic program.

Finally, Cadena *et al.* [22] design an efficient preprocessing, referred to as a refinement step, which involves compressing subsets of nodes into "supernodes." The size of a set S after refinement is determined in terms of the nodes and supernodes in it. This new size is called effective size, and, in practice, it is significantly smaller than the original size of S, making it possible to discover anomalous subgraphs with hundreds or thousands of nodes while keeping the parameter k in the single digits.

3) Heuristics: Besides the two general approaches discussed above, many problem-specific heuristics have been proposed. These algorithms are designed to be scalable and exhibit good empirical performance; however, in most cases, no quality guarantees are known for these methods.

Heuristic algorithms for parametric scan statistic optimization include a) a simulated annealing approach that is based on a concept of "non-compactness" for penalizing clusters [35]; b) the additive GraphScan algorithm, which connects clusters based on shortest path distances [92]; c) sparse learning method based on edge-lasso regularization [88]; d) spectral scan method based on graph Laplacian regularization [89]; e) submodular optimization algorithm based on Lovasz extensions [87]; and f) semidefinite programming algorithms based on linear matrix inequalities for characterizing the connectivity of subsets [6], [80], [81].

Nonparametric scan statistics in networks have only been explored recently. Chen and Neill present NPHGS, a fast iterative heuristic algorithm to optimize nonparametric scan statistics on general graphs [28]. NPHGS builds an anomalous connected subgraph S by starting at a significant node v and trying to add neighbors of v to S if it increases the target scan statistic $F(S)$. Zhou and Chen [104] take a different approach based on sparse optimization. They propose an extension of the projected gradient descent algorithm [15] that respects the connectivity constraints.

For the HDS problem, Bogdanov *et al.* [20] propose Meden. The idea behind their algorithm is to find a solution to HDS by repeatedly solving the HS problem (i.e., for a fixed subinterval). We can find a solution to HS using the GW algorithm, but the authors propose a heuristic called TopDown, with running time linear in the number of edges. Further, naively solving HS for all the possible subintervals would take time proportional to $O\left(|\mathcal{T}|^2\right)$; see Problem 6 for notation. The authors propose an aggregation scheme that makes it possible to process all the subintervals in time $O(|\mathcal{T}|\log(|\mathcal{T}|))$, with an extra overhead of $O(\log(|\mathcal{T}|))$ for running the aggregation procedure.

One could use Meden directly for the SAR problem by repeatedly running the algorithm and changing the weight of the returned edges to $-\infty$. However, this would not be very efficient. Mongiovì *et al.* [69] propose NetSpot, which efficiently alternates between finding a subgraph of high weight (for a fixed subinterval) and finding a subinterval that maximizes the score (for a fixed subgraph).

Finally, for EventTree+, in addition to using the GW algorithm, Rozenshtein *et al.* [84] propose the following heuristic. First, make a complete graph by putting an edge (u, v) between every pair of nodes with weight equal to the shortest distance between u and v in the original graph. Then, starting with an empty solution $S = \varnothing$, we add a node from $V \backslash S$ that decreases the MinGW objective the most, and we repeat until the objective does not decrease anymore.

Although some of the above methods provide quality guarantees if we relax the connectivity constraint [87], [104], these bounds do not directly apply to the original detection problem. In Table 2, we provide a summary of the different methods for anomaly detection based on Steiner connectivity.

C. Applications

The methods based on Steiner connectivity that we discussed above have been applied to a wide range of domains. One notable advantage of these methods over other graph-based approaches—such as the one in the next section—is flexibility of the discovered regions. Steiner connectivity methods can discover connected anomalous subgraphs of

Table 2 Algorithms for Anomaly Detection Based on Steiner Connectivity. n and m Are the Total Numbers of Nodes and Edges in the Input Graph, Respectively; t Is the Number of Snapshots; d Is the Maximum Depth to Explore; l Is the Number of Iterations (a Parameter); k Is the Effective Solution Size Parameter

Method	Problem	Time Complexity	Performance Bound
`AdditiveGraphScan` [92]	Nonlinear	$O(mn + n^2 \log n)$	No
`DepthFirstScan` [90]	Nonlinear	$O(n \cdot 2^d)$	No
`EdgeLasso` [88]	Quadratic	$O(l \cdot n^3)$	No
`GraphLaplacian` [89]	Quadratic	$O(l \cdot n^3)$	No
`NPHGS` [28]	Nonlinear	$O(n \log n)$	No
`GraphGHTP` [104]	Nonlinear	$O(m \log n)$	No
`ColCodeNP`, `ColCodeP` [22]	Linear, Nonlinear	$O(2^k \cdot e^k m \log \frac{n}{\epsilon})$	$(1-\epsilon)$-approximation
`TopDown` [20]	HS	$O(m)$	No
`Meden` [20]	HDS, SAR	$O(m \cdot \lvert\mathcal{T}\rvert \cdot \log^2(\lvert\mathcal{T}\rvert))$	No
`NetSpot` [69]	HDS, SAR	$O(ml)$	No
`EventTree+` [84]	MinGW	$O(n^2 \log n)$	2-approximation

arbitrary shape, in contrast with methods based on density or communities, which tend to find only denser subgraphs of more uniform shape. This is especially important in applications like biology and traffic congestion, where the target subgraphs have elongated structure. Below, we list some domains where the models that we presented are popular.

- Transportation networks: Detecting traffic bottlenecks in road or air networks using a parametric scan statistic [9], and the HDS objective [20]. The EventTree+ formulation has been used for identifying touristic hot spots in urban settings [84].
- Water distribution networks: Detecting pollution and spread of contaminant plumes using parametric scan statistics [76], [90].
- Disease outbreak detection: Early detection of disease outbreaks from information networks incorporating data from hospital emergency visits, ambulance dispatch calls, and pharmacy sales of over-the-counter drugs [90].
- Social science: Detection of crime hot spots in geographic networks using a parametric scan statistic [68]; detection and forecasting of societal events, such as civil unrest, using nonparametric scan statistics [28], [29].
- Image analysis: Detection of objects in images using a method based on Steiner connectivity [50].
- Computer networks: Detection of viruses or worms spreading from host to host in a computer network [70]; intrusions in a computer network [56].

For a summary of the applications grouped by problem formulation, see Table 3.

III. ANOMALY DETECTION BASED ON DENSITY

In a static unweighted graph $G(V, E)$, a subgraph induced by a subset of nodes $S \subseteq V$ is said to be dense if it has a "large" number of edges. Many metrics that capture this informal definition have been proposed, but we focus on three in this section. The density of a subgraph $S \subseteq V$ is defined as

$$\rho(S) = \frac{|E(S)|}{\binom{|S|}{2}}.$$

Table 3 Applications of Anomaly Detection Based on Steiner Connectivity

Name	Applications
Non-Parametric Scan Statistics	
Berk-Jones [18]	Detection of disease outbreaks, civil unrest events, and human rights events in social media graphs [28] Network intrusion detection [66] Detection of illicit activities in container shipment data [66]
Higher Criticism [33]	Detection of rare and weak effects in genomics and genetics [53] Multivariate disease outbreak detection [74]
Fisher's Test [41]	Subnetwork bio-marker detection in genetics [99] Multivariate disease outbreak detection [83] Psychological studies [79] Crime analysis [64]
Parametric Scan Statistics	
Kulldorff Scan Statistic [59]	Disease outbreak detection [60] Pattern detection in criminology [103], pediatrics [14], geriatrics [100], and psychology [65]
Elevated Mean Scan Statistic [81]	Disease outbreak detection [81]
Expectation-based Poisson Statistic [73]	Disease outbreak detection [72] Water pollution detection [92]
Heaviest Dynamic Subgraph [20]h Significant Anomalous Regions [69]s	Detection of anomalous actors in email network and traffic congestion [20] Political events in Wikipedia and traffic congestion [69]
EventTree+ [84]	Detection of urban events and touristic hot spots [84].

Here, $E(S)$ is the set of edges in the subgraph induced by S, and $\binom{|S|}{2}$ is the number of possible edges (i.e., pairwise combinations) in a set of size $|S|$. $\rho(S)$ ranges from 0, if the subgraph has no edges, to 1, if the subgraph is a clique. The second metric we consider is a relaxation of a clique, also called a quasi-clique. For a constant $\alpha \in [0,1]$, we say that a subgraph induced by $S \subseteq V$ is an α-quasi-clique if

$$f_\alpha(S) = |E(S)| - \alpha\binom{|S|}{2} \geq 0.$$

That is, S has at least a fraction α of all the possible $\binom{|S|}{2}$ edges. One weakness of these two metrics is that it is trivial to find small subgraphs with high density, such as 2-cliques—any edge will do!—or 3-cliques, which are triangles. A commonly used metric to address this weakness is the average degree, defined as

$$d(S) = \frac{|E(S)|}{|S|}.$$

Fig. 5 illustrates these definitions.

Different problem formulations to find dense subgraphs have been proposed. These formulations have been used for anomaly detection in static networks [51] and in dynamic networks by considering subgraphs in time intervals with significant changes in density, or by formalizing change in terms of density [19], [24]. We first describe some notions of density that have been used and applications based on them. We also refer to a survey by Lee *et al.* [63], which gives an overview of dense subgraph mining in a larger context.

A. Models and Problem Formulations

1) Densest Subgraph and Variants: In the densest subgraph (DS) problem, we are given a graph $G(V, E)$, and the goal is to find a subset of nodes $S \subseteq V$ that maximizes the average degree $d(S)$. In the example of Fig. 5, S is the densest subgraph. This problem can be solved optimally in polynomial time using Goldberg's flow-based algorithm [46]. There is also a linear-time greedy algorithm that yields a 1\2-approximation [13], [25], which we describe in Section III-B. In practice, this algorithm finds subgraphs with average degree close to optimal.

Many variants of the DS have been considered, including versions on directed and weighted graphs [57]. A recent application of these variants is FRAUDAR [51], a method for detecting anomalous or fraudulent product reviews and users in e-commerce sites. The authors model reviews and users as a bipartite graph—user u writes review v. Each node v is assigned a suspiciousness weight a_v indicating some prior belief that the user or review is fraudulent, and each edge (u, v) is assigned a weight $c_{u,v}$, which represents the suspiciousness of user u writing review v. Then, the goal becomes finding a subgraph S that maximizes

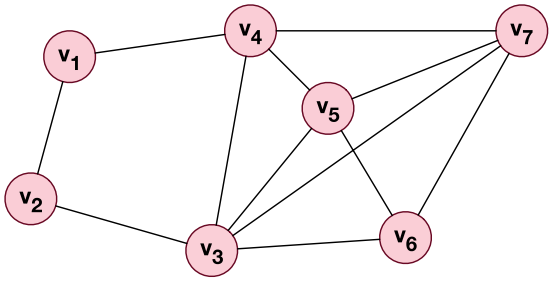

Fig. 5. ***Example illustrating the different notions of density. G(V, E) is a graph with |V| = 7. The set T = {v3, v5, v6, v7} is a 4-clique—i.e., it has density 1. For the set S = {v3, v4, v5, v6, v7}, there are nine edges with endpoints in S, so ρ(S) = 9/10 = 0.9. Therefore, T is denser than S. However, the average degree of S is d(S) = 9/5 = 1.8, whereas the set T has d(T) = 6/4 = 1.5, so S is denser with respect to the average degree measure. Finally, for α = 0.8, $f_\alpha(S) = 9 - 10\alpha = 1$ and $f_\alpha(T) = 6 - 6\alpha = 1.2$, respectively. It can be verified that T is the subset that maximizes f_α in the entire graph, the optimal quasi-clique.***

$$\frac{\sum_{v \in S} a_v + \sum_{u,v \in S} c_{u,v}}{|S|}$$

which is precisely the average degree in a weighted bipartite graph.

In real-world networks, subgraphs with maximal average degree have been found to be large—sometimes trivially spanning the entire node set V—and not very dense [98]. When we want to control the size of the subgraphs discovered, we can add a constraint k to the densest subgraph formulation. In the densest-k subgraph problem [40], the goal is to find a subset S of size k with maximum number of edges. This problem is NP-hard. When the constraint is to find a set S of size at least k, we obtain the densest-at-least-k subgraph problem; when the constraint is $|S| \leq k$, we have the densest-at-most-k subgraph problem. Both variants (also NP-hard) were proposed by Andersen and Chellapilla [10]. We note that all these formulations have been studied on directed and weighted graphs as well [57].

Recently, Tsourakakis [97] proposed the k-clique densest subgraph problem as a generalization of the densest subgraph problem. In this formulation, the goal is to find a set of nodes that maximizes $G_K(S)/|S|$, where G_k is the number of k-cliques induced by the nodes in S. For $k = 3$, we obtain the triangle-densest subgraph problem. Tsourakakis shows this latter formulation discovers graphs that are denser than the ones found by maximizing the average degree.

Finally, there are extensions to temporal networks as well. Rozenshtein *et al.* [85] extend the average degree to dynamic dense subgraphs, which they use to find interesting interaction patterns in social networks. The authors define a network stream over a set $\mathcal{T} = \{t_1, \ldots, t_2\}$ of timestamps as a tuple $(V, \mathcal{E}, \mathcal{T})$, where 1) V is a set of nodes, and 2) $\mathcal{E} = \{E^{t_1}, \ldots, E^{t_2}\}$ is a set of edges for each timestamp. The goal of the authors is to find a subset of nodes and timestamps with a high number of

interactions. For a subinterval $[i,j]$, the authors define $E^{[i,j]}$ to be the set of all edges from time i to j: $E^{[i,j]} = \cup_{k=i}^{j} E^k$. We note that so far these definitions are analogous to the edge-evolving network and temporal subgraph used by Bogdanov *et al.* for the HDS problem (Section II-A). However, Rozenshtein *et al.* extend the notion of temporal subgraph to time intervals that are not necessarily contiguous. The authors consider a time interval set, which is a collection of nonoverlapping and noncontiguous subintervals $\mathcal{T} = \{[i_1,j_1],[i_2,j_2],\ldots,[i_{|T|},j_{|T|}]\}$. We say that the span of T is the total number of timestamps covered by its subintervals: $\text{span}(T) = \sum_{k=1}^{|T|}(j_k - i_k)$. A time interval set allows for more flexibility on finding temporal periods with dense interactions; however, additional constraints on the size and span are needed to ensure that the discovered subgraphs are in fact temporally compact. These ideas lead to the following problem formulation.

Problem 9 [Dynamic Densest Subgraph (DDS)]: Given a temporal network stream $(V,\mathcal{E},\mathcal{T})$, and parameters κ and β, find a subset $S \subseteq V$ of nodes and a time interval set T such that $|T| \leq \kappa$, $\text{span}(T) \leq \beta$, and the average degree induced by S and T is maximized.

In contrast to the standard densest subgraph problem, Rozenshtein *et al.* [85] show that DDS is NP-hard, and they design efficient heuristics.

2) Cliques and Quasi-Cliques: Another set of problem formulations focuses on cliques, which are the densest subgraphs one could have. In the maximum clique problem, we are given a network, and the goal is to find the largest subgraph that is a clique [21]. However, this formulation is not very practical for two reasons. First, the maximum clique problem is hard to approximate in polynomial time to a factor of $n^{1-\epsilon}$ unless P = NP [47]. Second, finding a clique is too restrictive because all of the edges have to be present. In practice, we want to find subgraphs that are very dense, even if they are not complete.

The notion of quasi-cliques addresses these two challenges, and four formulations for finding quasi-cliques have been proposed. Abello and Resende [1] propose the maximum quasi-clique problem, where, analogous to maximum clique, we want to find the largest subgraph that is an α-quasi-clique, for some given α. They also study the maximum-k-quasi-clique problem, where the goal is to find the α-quasi-clique of size k that maximizes f_α. We are not aware of direct applications of these two problems for anomaly detection; however, they could be used as a viable alternative in an algorithm that looks for cliques as a subroutine.

The third notorious formulation is due to Tsourakakis *et al.* [98], who propose the optimal quasi-clique (OQC) problem, where the goal is to find a set of nodes that maximizes $f_\alpha(S) = E(S) - \alpha\binom{|S|}{2}$, i.e., without any size restriction. These concepts are illustrated in Fig. 5. The authors show that subgraphs with a high f_α score have high edge and triangle density, and they have small diameter—all desirable properties for dense subgraphs. Furthermore, they apply their method for finding OQCs to discover groups of correlated genes in gene coexpression networks and to discover authors with similar research interests in coauthorship graphs.

Finally, Cadena *et al.* [24] extend the definition of quasi-clique to weighted and signed networks. In their setting, every pair of nodes (u,v) has a weight $w(u,v) \in \mathbb{R}$, with $w(u,v) = 0$ representing the absence of edge (u,v). Additionally, each pair has a penalty $\alpha(u,v)$, instead of having a uniform α as in the standard α-quasi-clique definition. Then, the authors propose the generalized optimal quasi-clique (GOQC) problem, which asks for a set of nodes $S \subseteq V$ that maximizes $f_\alpha(S) = \sum_{u,v\in S}(w(u,v) - \alpha(u,v))$. By considering weights and penalties in a network stream, Cadena *et al.* also propose a problem formulation for anomaly detection in signed network streams. Let $(V,W,A,\mathcal{T})$ be a network stream over a set $\mathcal{T} = \{t_1,\ldots,t_2\}$ of timestamps. $W = \{w^{t_1},\ldots,w^{t_2}\}$ are the weights or observations at each timestamps, and $A = \{\alpha^{t_1},\ldots,\alpha^{t_2}\}$ is a set of expected values or observations. Intuitively, at timestamp t, we expect $w^t(u,v)$ to be similar in value to $\alpha^t(u,v)$, and large deviations are indicative or anomalous activity in that edge. This leads to the problem of finding a subgraph with maximum GOQC objective score for each timestamp.

Problem 10 [Event Detection in Signed Networks (EDSN)]: Given a signed network stream $(V,W,A,\mathcal{T})$, find a subset of nodes $S \subseteq V$ that maximizes

$$f_{\alpha^t}(S) = \sum_{u,v\in S}(w^t(u,v) - \alpha^t(u,v)) \tag{1}$$

for each timestamp $t \in \mathcal{T}$.

Cadena *et al.* [24] use EDSN to detect protests in political databases and congestion in highway networks. We note that, despite the extensive literature in dense subgraph mining, the problem of finding dense subgraphs in signed networks has been relatively unexplored. Most existing methods and formulations assume the weights of the graph are nonnegative, and there is no simple way to extend these methods to the signed case. There is work on community detection in signed networks [62], where communities are defined in terms of density; however, community detection and dense subgraph mining have fundamentally different objectives [63].

3) EventAllPairs+: In Section II, we described the model of Rozenshtein *et al.* [84] for event detection in activity networks. Recall that in an activity network, each node v has a positive prize $\pi(v)$, which represents the anomalousness of the node, and each edge e has a weight $w(e)$ that represents distance. The goal is to find a compact subgraph with high weight; that is, a subgraph S that maximizes $\lambda\Pi(S) - W(S)$, where $\Pi(\cdot)$ and $W(\cdot)$ are, respectively, functions of the total prize and edge weight of S. The authors consider the sum of pairwise distances for the edge weight function: $W(S) = 1\backslash 2 \sum_{u,v\in S} w(u,v)$. In order to avoid negative values on the objective function, the authors add the total pairwise distance to obtain the following objective to maximize:

$$\lambda\Pi(S) - W(S) + W(V)$$
$$= \sum_{v\in S}\pi(v) - \frac{1}{2}\sum_{u,v\in S} w(u,v) + \frac{1}{2}\sum_{u,v\in V} w(u,v).$$

The authors call this problem EventAllPairs+, and they show that it is a variant of the MaxCut problem, so similar techniques to the ones used for finding dense subgraphs can be applied in this setting. We note that, to the best of our knowledge, Rozenshtein *et al.* are the only authors to consider both Steiner connectivity and density for graph anomaly detection in the same problem formulation.

B. Methods and Techniques

In Section II-B, we described various techniques for approximating Steiner connectivity problems, such as linear programming and fixed parameter tractable algorithms. However, dense subgraph mining introduces additional computational challenges, and a different set of methods is required. For instance, the maximum clique problem is known to be W-hard [1], [34], and thus it does not admit a fixed-parameter tractable algorithm. Because of the connection with clique, parameterized complexity is ineffective for the other variants of density problems as well, such as densest subgraph and OQC. Two techniques that have been successful for these problems are 1) a greedy algorithm due to Charikar [25]; and 2) semidefinite programming [45].

1) Charikar's Greedy Algorithm: The DS problem can be solved in polynomial time by a reduction to network flow [46]. Goldberg's procedure consists of multiple invocations to a min-cut algorithm and runs in time $(n(n+m)\log(n)\log(n+m))$.

Even though this algorithm finds an optimal solution, the high running time and multiple reductions to min-cut make it impractical. Instead, most existing work on DS is based on the greedy algorithm of Charikar [25]. The main idea of the algorithm is to "peel" low degree nodes off the graph one by one and return the subgraph that has highest average degree at any point of the algorithm. More formally, we start with a graph $G_0 = G$. At iteration i of the algorithm, we remove the node that has the lowest degree in the graph G_{i-1} to obtain the graph G_i. We return the graph G_i of maximum average degree. The pseudocode is presented in Algorithm 1.

Algorithm 1 CharikarGreedy($G = (V, E)$)

$V_0 = V$, $E_0 = E$
Let $G_0 = (V_0, E_0)$
for $i = 1$ to $|V| - 1$ **do**
  $x = \arg\min_{v\in V_{i-1}} \deg(v)$
  $V_i = V_{i-1} \setminus \{x\}$, $E_i = E_{i-1} \setminus \{(x, v)|\forall v \in V_{i-1}\}$
  $G_i = (V_i, E_i)$
end for
return $\arg\max_{i=0}^{|V|-1} d(G_i)$

It can be shown that this algorithm finds a subgraph with density at least 1/2 of the optimal solution in time $O(|V| + |E|)$.

Because of its simplicity and efficiency, Charikar's procedure has been used extensively in many variants of the DS problem as well as for other metrics. The same basic greedy idea is at the core of FRAUDAR [51], the densest-at-least-k problem [57], the algorithm for DDS of Rozenshtein *et al.* [84], Tsourakakis' k-clique densest subgraph problem [97], and the OQC formulation [98]. In all these cases, except for OQC, the algorithm also preserves similar guarantees on the quality of the solution. In addition, the idea of peeling low-degree nodes sequentially has been used to find the k-core decomposition of a graph [16], which is a common subroutine in various data mining tasks [43], [86].

2) Semidefinite Programming: In their seminal paper, Goemans and Williamson [45] proposed novel techniques for designing approximation algorithms based on semidefinite programming. Their work led to significant improvements in algorithms for graph cuts and density problems. We describe the main ideas below.

A matrix $X \in \mathbb{R}^{n\times n}$ is positive emidefinite if and only if for every vector $y \in \mathbb{R}^n$, we have that $y^T Xy \geq 0$. Furthermore, there exists some matrix $V \in \mathbb{R}^{m\times n}$, where $m \leq n$, such that $X = V^T V$. We may also use the notation $X \succcurlyeq 0$ to denote that matrix X is positive semidefinite, and we assume that X is symmetric. A semidefinite program (SDP) is an optimization problem of the form

$$\begin{aligned}
&\text{maximize or minimize} && \sum_{i=1}^{n}\sum_{j=1}^{n} c_{ij}x_{ij} && \\
&\text{subject to} && \sum_{i=1}^{n}\sum_{j=1}^{n} a_{ijk}x_{ij} = b_k && \forall k \\
& && x_{i,j} = x_{ji} && \forall 1 \leq ij \leq n \\
& && X = (x_{ij}) \succcurlyeq 0. &&
\end{aligned}$$

Here, we are looking for a matrix of variables X that optimizes $\sum_{i=1}^{n}\sum_{j=1}^{n} c_{ij}x_{ij}$ under the given constraints. We note that the objective and the constraints are linear in terms of the x_{ij} variables, similar to the objective and constraints in a linear program. Also, analogously to linear programming, SDPs can be solved in polynomial time, albeit with a small error that can be ignored for all practical purposes. Equivalently, we can formulate a vector program in which the variables are vectors in $\mathbb{R}^n$. It is often more convenient to use this form to reason about approximation algorithms

$$\begin{aligned}
&\text{maximize or minimize} && \sum_{i=1}^{n}\sum_{j=1}^{n} c_{ij}(v_i \cdot v_j) && \\
&\text{subject to} && \sum_{i=1}^{n}\sum_{j=1}^{n} a_{ijk}(v_i \cdot v_j) = b_k && \forall k \\
& && v_i \in \mathbb{R}^n && \forall 1 \leq i \leq n.
\end{aligned}$$

Here, $v_i \cdot v_j$ is the inner product of the vectors, which is a scalar. The semidefinite and vector programs shown above are equivalent. To see why, we can take a solution X to the SDP and compute (approximately) the decomposition $X = V^T V$ in polynomial time. Let v_i be the ith column of V; then, $x_{ij} = v_i \cdot v_j$, and the v_i vectors are a feasible solution to the vector program. Conversely, we can take a solution to the vector program and construct a matrix V whose ith column is v_i. Let $X = V^T V$; then, X is symmetric and positive semidefinite, and $x_{ij} = v_i \cdot v_j$, so X is a feasible solution to the semidefinite program.

SDPs are used to design approximation algorithms as follows. First, the target problem is formulated as a quadratic program (QP), in which the variables are restricted to be integers. However, quadratic programming is hard, so we relax it to a semidefinite program, which we can solve efficiently. The core design challenge is rounding the variables in the SDP to integer values, as required by the QP, in such a way that we can claim an approximation guarantee on the solution. Notice that this pipeline is analogous to the rounding of linear programs to obtain approximate solutions to integer programs.

As a concrete example, we describe the algorithm of Cadena *et al.* [24] for the GOQC problem. We start with the following quadratic programming formulation for an instance of GOQC with inputs $G(V, E)$, w, and α

$$\text{(QP) maximize} \sum_{(u,s)\in E} w(u,s)\left(\frac{1 + x_u x_0 + x_s x_0 + x_u x_s}{4}\right) - \sum_{u,s\in V, u\neq s} \alpha(u,s)\left(\frac{1 + x_u x_0 + x_s x_0 + x_u x_s}{4}\right)$$

$$\text{subject to} \quad x_0, x_u \in \{-1, 1\} \quad \forall u \in V.$$

Here, each variable x_u, except for x_0, corresponds to a node $u \in V$. It can be shown that the above program is equivalent to the GOQC problem. Without loss of generality, suppose $x_0 = 1$, and let S be the set formed by the nodes for which $x_u = 1$ in the optimal solution to QP. Then, S is the optimal solution for GOQC in the graph G.
Now, consider the semidefinite relaxation of the problem

$$\text{(SDP) maximize} \sum_{(u,s)\in E} w(u,s)\left(\frac{1 + v_u v_0 + v_s v_0 + v_u v_s}{4}\right) - \sum_{u,s\in V, u\neq s} \alpha(u,s)\left(\frac{1 + v_u v_0 + v_s v_0 + v_u v_s}{4}\right)$$

$$\text{subject to} \quad v_u \cdot v_u = 1 \qquad \forall u \in V$$

$$v_0, v_u \in \mathbb{R}^{n+1} \qquad \forall u \in V.$$

After solving the SDP, the authors use the rounding approach of [26] to find a set S' as a solution to GOQC. The authors then use a local search algorithm [98] to add or remove nodes to S' until a local maxima is found. Their final solution is guaranteed to have objective value at least $O(\log n)$ of the optimal, where n is the number of nodes.

Feige *et al.* [40] proposed SDP-based algorithms for the densest-k subgraph problem. They developed a rounding procedure that yields an approximation ratio of at least $\max\left(k/2n, n^{\epsilon-1/3}\right)$, which was improved to (k/n) in subsequent work [39]. Finally, Rozenshtein *et al.* [84] reduced their EventAllPairs+ problem to an (s,t)-MaxCut problem, which can be solved using semidefinite programming with a constant approximation guarantee of 0.868.

3) Local Search Heuristic: In addition to the approximation algorithms discussed above, a local search heuristic has been successfully applied to quasi-clique-related problems. This heuristic starts with a candidate solution, usually a single node or small sets of nodes, and then we add or remove nodes to the solution until there is no improvement on the objective score. This style of algorithm appears in the work of Abello and Resende [1], Tsourakakis *et al.* [98], and Cadena *et al.* [24]. As an example, we describe the local search procedure for the OQC problem in Algorithm 2. Starting from a random vertex, the algorithm adds nodes to the solution as long as the objective score keeps increasing—b_2 becomes false. Then, we try to remove a node if it increases the score, and we repeat until either there is no improvement—b_1 becomes false—or we reach a certain number of iterations $T_{\max}$. By changing the stopping condition of the outer loop, we can obtain algorithms for the problem formulations of Abello and Resende.

Algorithm 2 LocalSearchOQC($G = (V, E), \alpha, T_{\max}$)

```
Let S = {v}, where v is chosen uniformly at random
Let b1 = TRUE
Let t = 1
while b1 and t ≤ T_max do
  Let b2 = TRUE
  while b2 do
    if there is a v ∈ V \ S such that f_α(S ∪ {v}) ≥ f_α(S)
    then S = S ∪ {v}
    else b2 = FALSE
  end while
  if there is a v ∈ S such that f_α(S \ {v}) ≥ f_α(S)
  then S = S \ {v}
  else b1 = FALSE
  t = t + 1
end while
return arg max_{S'∈{S,V\S}} f_α(S')
```

C. Applications

We briefly mentioned some applications for each formulation in Section III-B, but we summarize them here for completeness. For a summary of the applications grouped by problem formulation, see Table 4.

- Fraud detection in graphs: Hooi *et al.* [51] study the problem of finding fraudulent users and products in reviews. They propose a greedy algorithm FRAUDAR to find a subgraph S maximizing a notion of density similar to the average degree. They show that this method finds a fraudulent community in the Twitter network.

Table 4 Applications of Anomaly Detection Based on Density

Name	Applications
Densest Subgraph	
Dynamic Densest Subgraph [85]	Finding communities in temporal networks
FRAUDAR [51]	Detection of fraudulent product reviews and reviewers
Cliques and Quasicliques	
Optimal Quasiclique [98]	Finding thematic scientific groups in co-authorship networks and correlated genes in co-expression networks
Generalized Optimal Quasiclique [24]	Detecting protests in political databases and traffic incidents in highway networks
EventAllPairs+ [84]	Detection of urban events and touristic hot spots.

- Event detection in signed networks: Cadena *et al.* [24] use the GOQC problem to model events in signed temporal networks. They apply their methodology to the integrated crisis early warning system (ICEWS) data set of political events [42], and they are able to find several important civil unrest events.
- Communities in temporal interaction networks: Rozenshtein *et al.* [85] use the DDS problem to find communities in different kinds of interaction networks, including Twitter and Facebook. The temporal component is key in finding these communities.
- Thematic scientific groups: Tsourakakis *et al.* [98] apply the OQC problem to discover groups of researchers with similar interests, according to the DBLP data set.
- Correlated genes: As a second application, the same authors examine a correlation network where genes are connected by an edge if their correlation is at least 0.99. Using Algorithm 2, the authors find a clique of 14 genes highly correlated with a known tumor protein.

IV. CONCLUSION AND OPEN QUESTIONS

Network anomaly detection is an important area with a large number of applications. Formulations based on Steiner connectivity, especially using network scan statistics and density of subgraphs, provide a systematic approach for anomaly detection and have been shown to have good performance in practice. We have summarized some of the key techniques developed for these problems. In particular, we find that fixed parameter tractability is a powerful, but underexplored approach for designing efficient algorithms, and it is likely to be useful in other graph mining applications.

Extending the performance of all these methods is an important area of research. Another important issue in anomaly detection, which has not been examined adequately, is the role of uncertainty. Extending these methods to incorporate uncertainty is an important area for future research. ■

REFERENCES

[1] J. Abello, M. G. C. Resende, and S. Sudarsky, "Massive quasi-clique detection," in *LATIN 2002: Theoretical Informatics*. New York, NY, USA: Springer-Verlag, 2002, pp. 598–612.

[2] C. Aggarwal, Y. Zhao, and P. Yu, "Outlier detection in graph streams," in *Proc. IEEE Int. Conf. Data Eng. (ICDE)*, Apr. 2011, pp. 399–409.

[3] L. Akoglu and C. Faloutsos, "Event detection in time series of mobile communication graphs," in *Proc. Army Sci. Conf.*, 2010, pp. 77–79.

[4] L. Akoglu, M. McGlohon, and C. Faloutsos, "OddBall: Spotting anomalies in weighted graphs," in *Advances in Knowledge Discovery and Data Mining*. New York, NY, USA: Springer-Verlag, 2010, pp. 410–421.

[5] L. Akoglu, H. Tong, and D. Koutra, "Graph based anomaly detection and description: A survey," *Data Mining Knowl. Discovery*, vol. 29, no. 3, pp. 626–688, May 2015.

[6] C. Aksoylar, L. Orecchia, and V. Saligrama, "Connected subgraph detection with mirror descent on SDPs," in *Proc. Int. Conf. Mach. Learn.*, 2017, pp. 51–59.

[7] N. Alon, R. Yuster, and U. Zwick, "Color-coding," *J. ACM*, vol. 42, no. 4, pp. 844–856, Jul. 1995.

[8] G. Alves and Y.-K. Yu, "Accuracy evaluation of the unified P-value from combining correlated P-values," *PLoS ONE*, vol. 9, no. 3, p. e91225, 2014.

[9] B. Anbaroğlu, T. Cheng, and B. Heydecker, "Non-recurrent traffic congestion detection on heterogeneous urban road networks," *Transportmetrica A, Transp. Sci.*, vol. 11, no. 9, pp. 754–771, 2015.

[10] R. Andersen and K. Chellapilla, "Finding dense subgraphs with size bounds," in *Algorithms and Models for the Web-Graph*. New York, NY, USA: Springer-Verlag, 2009, pp. 25–37.

[11] M. Araujo *et al.*, "Com2: Fast automatic discovery of temporal ('Comet') communities," in *Advances in Knowledge Discovery and Data Mining*. New York, NY, USA: Springer-Verlag, 2014, pp. 271–283.

[12] E. Arias-Castro, E. J. Candès, and A. Durand, "Detection of an anomalous cluster in a network," *Ann. Stat.*, vol. 39, no. 1, pp. 278–304, 2011.

[13] Y. Asahiro, R. Hassin, and K. Iwama, "Complexity of finding dense subgraphs," *Discrete Appl. Math.*, vol. 121, nos. 1–3, pp. 15–26, Sep. 2002.

[14] E. Awini, P. Mattah, O. Sankoh, and M. Gyapong, "Spatial variations in childhood mortalities at the Dodowa health and demographic surveillance system site of the INDEPTH network in Ghana," *Tropical Med. Int. Health*, vol. 15, no. 5, pp. 520–528, May 2010.

[15] S. Bahmani, P. T. Boufounos, and B. Raj, "Learning model-based sparsity via projected gradient descent," *IEEE Trans. Inf. Theory*, vol. 62, no. 4, pp. 2092–2099, Apr. 2016.

[16] V. Batagelj and M. Zaversnik (2003). "An O(m) algorithm for cores decomposition of networks." [Online]. Available: https://arxiv.org/abs/cs/0310049

[17] M. Bateni, M. Hajiaghayi, and V. Liaghat, "Improved approximation algorithms for (budgeted) node-weighted Steiner problems," in *Proc. Int. Colloq. Automata, Lang., Program. (ICALP)*, 2013, pp. 81–92.

[18] R. H. Berk and A. Cohen, "Asymptotically optimal methods of combining tests," *J. Amer. Stat. Assoc.*, vol. 74, no. 368, pp. 812–814, 1979.

[19] D. Birant and A. Kut, "Spatio-temporal outlier detection in large databases," in *Proc. 28th Int. Conf. Inf. Technol. Interfaces*, Jun. 2006, pp. 179–184.

[20] P. Bogdanov, M. Mongiovì, and A. K. Singh, "Mining heavy subgraphs in time-evolving networks," in *Proc. IEEE Int. Conf. Data Mining (ICDM)*, Dec. 2011, pp. 81–90.

[21] I. M. Bomze, M. Budinich, P. M. Pardalos, and M. Pelillo, "The maximum clique problem," in *Handbook of Combinatorial Optimization*. New York, NY, USA: Springer-Verlag, 1999, pp. 1–74.

[22] J. Cadena, F. Chen, and A. Vullikanti, "Near-optimal and practical algorithms for graph

scan statistics," in *Proc. SIAM Data Mining (SDM)*, 2017, pp. 1–9.

[23] J. Cadena, F. Chen, and A. Vullikanti, "Near optimal and practical algorithms for graph scan statistics with connectivity constraints," Tech. Rep., 2018. [Online]. Available: https://tinyurl.com/yabgokac

[24] J. Cadena, A. K. Vullikanti, and C. C. Aggarwal, "On dense subgraphs in signed network streams," in *Proc. IEEE Int. Conf. Data Mining (ICDM)*, Dec. 2016, pp. 51–60.

[25] M. Charikar, "Greedy approximation algorithms for finding dense components in a graph," in *Proc. Int. Workshop Approx. Algorithms Combinat. Optim.*, Sep. 2000, pp. 84–95.

[26] M. Charikar and A. Wirth, "Maximizing quadratic programs: Extending Grothendieck's inequality," in *Proc. IEEE Symp. Found. Comput. Sci. (FOCS)*, Oct. 2004, pp. 54–60.

[27] F. Chen and D. B. Neill, "Non-parametric scan statistics for disease outbreak detection on Twitter," *Online J. Public Health Inf.*, vol. 6, no. 1, p. e155, 2014.

[28] F. Chen and D. B. Neill, "Non-parametric scan statistics for event detection and forecasting in heterogeneous social media graphs," in *Proc. ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining (KDD)*, 2014, pp. 1166–1175.

[29] F. Chen and D. B. Neill, "Human rights event detection from heterogeneous social media graphs," *Big Data*, vol. 3, no. 1, pp. 34–40, 2015.

[30] R. Cole, R. Hariharan, M. Lewenstein, and E. Porat, "A faster implementation of the Goemans-Williamson clustering algorithm," in *Proc. 12th Annu. ACM-SIAM Symp. Discrete Algorithms*, 2001, pp. 17–25.

[31] Q. Ding, N. Katenka, P. Barford, E. D. Kolaczyk, and M. Crovella, "Intrusion as (anti)social communication: Characterization and detection," in *Proc. ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining (KDD)*, 2012, pp. 886–894.

[32] D. Donoho and J. Jin, "Higher criticism for detecting sparse heterogeneous mixtures," *Ann. Stat.*, vol. 32, no. 3, pp. 962–994, Jun. 2004.

[33] D. Donoho and J. Jin, "Higher criticism for large-scale inference, especially for rare and weak effects," *Stat. Sci.*, vol. 30, no. 1, pp. 1–25, 2015.

[34] R. G. Downey and M. R. Fellows, *Parameterized Complexity*. New York, NY, USA: Springer-Verlag, 2012.

[35] L. Duczmal, M. Kulldorff, and L. Huang, "Evaluation of spatial scan statistics for irregularly shaped clusters," *J. Comput. Graph. Stat.*, vol. 15, no. 2, pp. 428–442, 2006.

[36] W. Eberle and L. Holder, "Graph-based approaches to insider threat detection," in *Proc. Workshop Cyber Secur. Inf. Intell. Res., Cyber Secur. Inf. Intell. Challenges Strategies*, 2009, Art. no. 44.

[37] E. S. Edgington, "An additive method for combining probability values from independent experiments," *J. Psychol.*, vol. 80, no. 2, pp. 351–363, 1972.

[38] F. Eicker, "The asymptotic distribution of the Suprema of the standardized empirical processes," *Ann. Stat.*, vol. 7, no. 1, pp. 116–138, 1979.

[39] U. Feige, G. Kortsarz, and D. Peleg, "The dense k-subgraph problem," *Algorithmica*, vol. 29, no. 3, pp. 410–421, Mar. 2001.

[40] U. Feige and M. Seltser, "On the densest k-subgraph problem," *Algorithmica*, vol. 29, p. 2001, 1997.

[41] R. A. Fisher, *Statistical Methods for Research Workers*. Guildford, U.K.: Genesis Publishing Pvt Ltd, 1925.

[42] D. J. Gerner, P. A. Schrodt, R. A. Francisco, and J. L. Weddle, "Machine coding of event data using regional and international sources," *Int. Stud. Quart.*, vol. 38, no. 1, pp. 91–119, 1994.

[43] C. Giatsidis, D. M. Thilikos, and M. Vazirgiannis, "Evaluating cooperation in communities with the k-core structure," in *Proc. Int. Conf. Adv. Social Netw. Anal. Mining (ASONAM)*, Jul. 2011, pp. 87–93.

[44] M. Goemans and D. P. Williamson, "A general approximation technique for constrained forest problems," *SIAM J. Comput.*, vol. 24, no. 2, pp. 296–317, 1995.

[45] M. X. Goemans and D. P. Williamson, "Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming," *J. Assoc. Comput. Mach.*, vol. 42, no. 6, pp. 1115–1145, 1995.

[46] A. V. Goldberg, "Finding a maximum density subgraph," Univ. California, Berkeley, Berkeley, CA, USA, Tech. Rep., 1984.

[47] J. Hastad, "Clique is hard to approximate within $n^{1-\epsilon}$," in *Proc. 37th Annu. Symp. Found. Comput. Sci.*, Oct. 1996, pp. 627–636.

[48] M. Hauptmann and M. Karpiński, *A Compendium on Steiner Tree Problems*. Saarbrücken, Germany: Inst. für Informatik, 2013.

[49] N. A. Heard, D. J. Weston, K. Platanioti, and D. J. Hand, "Bayesian anomaly detection methods for social networks," *Ann. Appl. Stat.*, vol. 4, no. 2, pp. 645–662, 2010.

[50] C. Hegde, P. Indyk, and L. Schmidt, "A nearly-linear time framework for graph-structured sparsity," in *Proc. Int. Conf. Mach. Learn.*, 2015, pp. 928–937.

[51] B. Hooi, H. A. Song, A. Beutel, N. Shah, K. Shin, and C. Faloutsos, "FRAUDAR: Bounding graph fraud in the face of camouflage," in *Proc. ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining (KDD)*, New York, NY, USA, 2016, pp. 895–904. [Online]. Available: http://doi.acm.org/10.1145/2939672.2939747

[52] F. K. Hwang, D. S. Richards, and P. Winter, *The Steiner Tree Problem*, vol. 53. Amsterdam, The Netherlands: Elsevier, 1992.

[53] S. K. Iyengar and R. C. Elston, "The genetic basis of complex traits: Rare variants or 'common gene, common disease?"' in *Linkage Disequilibrium and Association Mapping*. 2007.

[54] L. Jager and J. A. Wellner, "Goodness-of-fit tests via phi-divergences," *Ann. Stat.*, vol. 35, no. 5, pp. 2018–2053, 2007.

[55] I. Jung, M. Kulldorff, and O. J. Richard, "A spatial scan statistic for multinomial data," *Stat. Med.*, vol. 29, no. 18, pp. 1910–1918, 2010.

[56] A. Keen and R. J. Mayer, "Network intrusion detection," U.S. Patent 8161550, Apr. 17, 2012. [Online]. Available: http://www.google.com/patents/US8161550

[57] S. Khuller and B. Saha, "On finding dense subgraphs," in *International Colloquium on Automata, Languages, and Programming*. New York, NY, USA: Springer-Verlag, 2009, pp. 597–608.

[58] D. Koutra, E. E. Papalexakis, and C. Faloutsos, "TensorSplat: Spotting latent anomalies in time," in *Proc. 16th Panhellenic Conf. Inform. (PCI)*, Oct. 2012, pp. 144–149.

[59] M. Kulldorff, "A spatial scan statistic," *Commun. Stat., Theory Methods*, vol. 26, no. 6, pp. 1481–1496, 1997.

[60] M. Kulldorff, T. Tango, and P. J. Park, "Power comparisons for disease clustering tests," *Comput. Stat. Data Anal.*, vol. 42, no. 4, pp. 665–684, Apr. 2003.

[61] M. Kumar, R. Ghani, and Z.-S. Mei, "Data mining to predict and prevent errors in health insurance claims processing," in *Proc. ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining (KDD)*, 2010, pp. 65–74.

[62] J. Kunegis, S. Schmidt, A. Lommatzsch, J. Lerner, E. W. De Luca, and S. Albayrak, "Spectral analysis of signed graphs for clustering, prediction and visualization," in *Proc. SIAM Data Mining (SDM)*, vol. 10. 2010, p. 559.

[63] V. E. Lee, N. Ruan, R. Jin, and C. Aggarwal, "A survey of algorithms for dense subgraph discovery," in *Managing and Mining Graph Data*. New York, NY, USA: Springer-Verlag, 2010, pp. 303–336.

[64] A. Leprätre *et al.*, "Prevalence and behavioural risks for HIV and HCV infections in a population of drug users of Dakar, Senegal: the ANRS 12243 UDSEN study," *J. Int. AIDS Soc.*, vol. 18, no. 1, p. 19888, Jan. 2015.

[65] F. Margai and N. Henry, "A community-based assessment of learning disabilities using environmental and contextual risk factors," *Social Sci. Med.*, vol. 56, no. 5, pp. 1073–1085, Mar. 2003.

[66] E. McFowland, III, S. Speakman, and D. B. Neill, "Fast generalized subset scan for anomalous pattern detection," *J. Mach. Learn. Res.*, vol. 14, no. 1, pp. 1533–1561, 2013.

[67] B. Miller, N. Bliss, and P. J. Wolfe, "Subgraph detection using eigenvector L1 norms," in *Proc. Adv. Neural Inf. Process. Syst.*, 2010, pp. 1633–1641.

[68] R. Modarres and G. P. Patil, "Hotspot detection with bivariate data," *J. Stat. Planning Inference*, vol. 137, no. 11, pp. 3643–3654, 2007.

[69] M. Mongiovì, P. Bogdanov, R. Ranca, A. K. Singh, E. E. Papalexakis, and C. Faloutsos, "NetSpot: Spotting significant anomalous regions on dynamic networks," in *Proc. SIAM Data Mining (SDM)*, 2013, pp. 1–9.

[70] 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.

[71] D. B. Neill, "Fast and flexible outbreak detection by linear-time subset scanning," in *Proc. Adv. Disease Surveill.*, vol. 5. 2008, p. 48.

[72] D. B. Neill, "An empirical comparison of spatial scan statistics for outbreak detection," *Int. J. Health Geograph.*, vol. 8, p. 20, Apr. 2009.

[73] D. B. Neill, "Fast subset scan for spatial pattern detection," *J. Roy. Stat. Soc. B, Stat. Methodol.*, vol. 74, no. 2, pp. 337–360, Mar. 2012.

[74] D. B. Neill and J. Lingwall, "A nonparametric scan statistic for multivariate disease surveillance," in *Proc. Adv. Disease Surveill.*, vol. 4. 2007, p. 106.

[75] C. C. Noble and D. J. Cook, "Graph-based anomaly detection," in *Proc. 9th ACM*

SIGKDD Int. Conf. Knowl. Discovery Data Mining, 2003, pp. 631–636.

[76] D. D. Oliveira, D. B. Neill, J. H. Garrett, Jr., and L. Soibelman, "Detection of patterns in water distribution pipe breakage using spatial scan statistics for point events in a physical network," *J. Comput. Civil Eng.*, vol. 25, no. 1, pp. 21–30, 2010.

[77] L. Peel and A. Clauset, "Detecting change points in the large-scale structure of evolving networks," in *Proc. AAAI*, 2015, pp. 2914–2920.

[78] C. E. Priebe, J. M. Conroy, D. J. Marchette, and Y. Park, "Scan statistics on Enron graphs," *Comput. Math. Org. Theory*, vol. 11, no. 3, pp. 229–247, 2005.

[79] A. Przeworski, N. Cain, and K. Dunbeck, "Traumatic life events in individuals with hoarding symptoms, obsessive-compulsive symptoms, and comorbid obsessive-compulsive and hoarding symptoms," *J. Obsessive-Compulsive Rel. Disorders*, vol. 3, no. 1, pp. 52–59, 2014.

[80] J. Qian and V. Saligrama, "Efficient minimax signal detection on graphs," in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, pp. 2708–2716.

[81] J. Qian, V. Saligrama, and Y. Chen, "Connected sub-graph detection," in *Proc. Artif. Intell. Stat.*, 2014, pp. 1–69.

[82] S. Ranshous, S. Shen, D. Koutra, S. Harenberg, C. Faloutsos, and N. F. Samatova, "Anomaly detection in dynamic networks: A survey," *Wiley Interdiscipl. Rev., Comput. Stat.*, vol. 7, no. 3, pp. 223–247, 2015.

[83] J. Roure, A. Dubrawski, and J. Schneider, "A study into detection of bio-events in multiple streams of surveillance data," in *Intelligence and Security Informatics: Biosurveillance*. New York, NY, USA: Springer-Verlag, 2007, pp. 124–133.

[84] P. Rozenshtein, A. Anagnostopoulos, A. Gionis, and N. Tatti, "Event detection in activity networks," in *Proc. ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining (KDD)*, 2014, pp. 1176–1185.

[85] P. Rozenshtein, N. Tatti, and A. Gionis, "Finding dynamic dense subgraphs," *ACM Trans. Knowl. Discovery Data*, vol. 11, no. 3, pp. 27:1–27:30, Apr. 2017.

[86] A. E. Saríyüce, B. Gedik, G. Jacques-Silva, K.-L. Wu, and Ü. V. Çatalyürek, "Streaming algorithms for k-core decomposition," *Proc. VLDB Endowment*, vol. 6, no. 6, pp. 433–444, 2013.

[87] J. L. Sharpnack, A. Krishnamurthy, and A. Singh, "Near-optimal anomaly detection in graphs using Lovasz extended scan statistic," in *Proc. Adv. Neural Inf. Process. Syst.*, 2013, pp. 1959–1967.

[88] J. Sharpnack, A. Singh, and A. Rinaldo, "Sparsistency of the edge lasso over graphs," in *Proc. Artif. Intell. Stat.*, 2012, pp. 1–13.

[89] J. Sharpnack, A. Singh, and A. Rinaldo, "Changepoint detection over graphs with the spectral scan statistic," in *Proc. Artif. Intell. Stat.*, 2013, pp. 545–553.

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

[91] S. Speakman and D. B. Neill, "Fast graph scan for scalable detection of arbitrary connected clusters," in *Proc. Adv. Disease Surveill.*, 2010, p. 1.

[92] S. Speakman, Y. Zhang, and D. B. Neill, "Dynamic pattern detection with temporal consistency and connectivity constraints," in *Proc. IEEE Int. Conf. Data Mining (ICDM)*, Dec. 2013, pp. 697–706.

[93] S. A. Stouffer, E. A. Suchman, L. C. DeVinney, S. A. Star, and R. M. Williams, *The American Soldier: Adjustment During Army Life*. Princeton, NJ, USA: Princeton Univ. Press, 1949.

[94] J. Sun, C. Faloutsos, S. Papadimitriou, and P. S. Yu, "GraphScope: Parameter-free mining of large time-evolving graphs," in *Proc. ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining (KDD)*, 2007, pp. 687–696.

[95] J. Sun, H. Qu, D. Chakrabarti, and C. Faloutsos, "Neighborhood formation and anomaly detection in bipartite graphs," in *Proc. 5th IEEE Int. Conf. Data Mining (ICDM)*, Nov. 2005, pp. 1–8.

[96] J. Sun, Y. Xie, H. Zhang, and C. Faloutsos, "Less is more: Sparse graph mining with compact matrix decomposition," *Stat. Anal. Data Mining*, vol. 1, no. 1, pp. 6–22, Feb. 2008.

[97] C. Tsourakakis, "The K-clique densest subgraph problem," in *Proc. Int. Conf. World Wide Web*, 2015, pp. 1122–1132.

[98] C. Tsourakakis, F. Bonchi, A. Gionis, F. Gullo, and M. Tsiarli, "Denser than the densest subgraph: Extracting optimal quasi-cliques with quality guarantees," in *Proc. ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining (KDD)*, 2013, pp. 104–112.

[99] F. Vandin, P. Clay, E. Upfal, and B. J. Raphael, "Discovery of mutated subnetworks associated with clinical data in cancer," in *Proc. Biocomput.*, 2012, pp. 55–66.

[100] P. Vaneckova, P. J. Beggs, and C. R. Jacobson, "Spatial analysis of heat-related mortality among the elderly between 1993 and 2004 in Sydney, Australia," *Social Sci. Med.*, vol. 70, no. 2, pp. 293–304, 2010.

[101] B. Wang, J. M. Phillips, R. Schreiber, D. M. Wilkinson, N. Mishra, and R. Tarjan, "Spatial scan statistics for graph clustering," in *Proc. SIAM Data Mining (SDM)*, 2008, pp. 727–738.

[102] R. Wilcox, "Kolmogorov–Smirnov test," in *Encyclopedia of Biostatistics*. Hoboken, NJ, USA: Wiley, 2005.

[103] A. M. Zeoli, J. M. Pizarro, S. C. Grady, and C. Melde, "Homicide as infectious disease: Using public health methods to investigate the diffusion of homicide," *Justice Quart.*, vol. 31, no. 3, pp. 609–632, 2014.

[104] B. Zhou and F. Chen, "Graph-structured sparse optimization for connected subgraph detection," in *Proc. IEEE Int. Conf. Data Mining (ICDM)*, Dec. 2016, pp. 709–718.

ABOUT THE AUTHORS

Jose Cadena received the Ph.D. degree from the Department of Computer Science and the Biocomplexity Institute, Virginia Tech, Blacksburg, VA, USA.

He is a Postdoctoral Researcher in The Machine Learning Group, Lawrence Livermore National Laboratory, Livermore, CA, USA. His research interests include graph mining, machine learning for network data, combinatorial optimization, and approximation algorithms. His work has been applied to problems on the domains of social media analytics, computational epidemiology, and anomaly detection on temporal graphs.

Feng Chen received the Ph.D. degree in computer science from Virginia Tech, Blacksburg, VA, USA, in 2012.

He is an Assistant Professor of Computer Science at the University of New York at Albany —SUNY, Albany, NY, USA. His research interests include anomalous pattern detection, event detection and forecasting, graph mining, and machine learning. His research has been supported by NSF, NIH, ARO, IARPA, and the U.S. Department of Transportation.

Anil Vullikanti is an Associate Professor in the Department of Computer Science and the Biocomplexity Institute, Virginia Tech, Blacksburg, VA, USA. His research interests include the broad areas of approximation and randomized algorithms and dynamical systems, and their applications to computational epidemiology and the modeling, simulation and analysis of socio-technical systems.

Prof. Vullikanti is the recipient of the Virginia Tech College of Engineering Faculty Fellow Award 2017, and the Biocomplexity Institute of Virginia Tech Excellence in Research Award 2017, the DOE Early Career award in 2010, and the NSF CAREER award in 2009. He is an editorial board member for the *ACM Transactions on Algorithms* and the *Journal of Interconnection Networks.*