Data Min Knowl Disc (2015) 29:626–688
DOI 10.1007/s10618-014-0365-y

Graph based anomaly detection and description: a survey

Leman Akoglu · Hanghang Tong · Danai Koutra

Received: 5 April 2013 / Accepted: 12 June 2014 / Published online: 5 July 2014

Abstract Detecting anomalies in data is a vital task, with numerous high-impact applications in areas such as security, finance, health care, and law enforcement. While numerous techniques have been developed in past years for spotting outliers and anomalies in unstructured collections of multi-dimensional points, with graph data becoming ubiquitous, techniques for structured *graph* data have been of focus recently. As objects in graphs have long-range correlations, a suite of novel technology has been developed for anomaly detection in graph data. This survey aims to provide a general, comprehensive, and structured overview of the state-of-the-art methods for anomaly detection in data represented as graphs. As a key contribution, we give a general framework for the algorithms categorized under various settings: unsupervised versus (semi-)supervised approaches, for static versus dynamic graphs, for attributed versus plain graphs. We highlight the effectiveness, scalability, generality, and robustness aspects of the methods. What is more, we stress the importance of anomaly *attribution* and highlight the major techniques that facilitate digging out the root cause, or the 'why', of the detected anomalies for further analysis and sensemaking. Finally, we present several real-world applications of graph-based anomaly

Responsible editor: G. Karypis.

L. Akoglu (✉)
Department of Computer Science, Stony Brook University, Stony Brook, NY 11794, USA
e-mail: leman@cs.stonybrook.edu

H. Tong
Department of Computer Science, City College, City University of New York,
New York, NY 10031, USA
e-mail: tong@cs.ccny.cuny.edu

D. Koutra
Computer Science Department, Carnegie Mellon University, Pittsburgh, PA 15217, USA
e-mail: danai@cs.cmu.edu

detection in diverse domains, including financial, auction, computer traffic, and social networks. We conclude our survey with a discussion on open theoretical and practical challenges in the field.

Keywords Anomaly detection · Graph mining · Network anomaly detection · Event detection · Change point detection · Fraud detection · Anomaly description · Visual analytics

1 Introduction

When analyzing large and complex datasets, knowing what stands out in the data is often at least, or even more important and interesting than learning about its general structure. The branch of data mining concerned with discovering rare occurrences in datasets is called *anomaly detection*. This problem domain has numerous high-impact applications in security, finance, health care, law enforcement, and many others.

Examples include detecting network intrusion or network failure (Ding et al. 2012; Idé and Kashima 2004; Sun et al. 2008), credit card fraud (Bolton and Hand 2001), calling card and telecommunications fraud (Cortes et al. 2002; Taniguchi et al. 1998), auto insurance fraud (Phua et al. 2004), health insurance claim errors (Kumar et al. 2010), accounting inefficiencies (McGlohon et al. 2009), email and Web spam (Castillo et al. 2007), opinion deception and reviews spam (Ott et al. 2012), auction fraud (Pandit et al. 2007), tax evasion (Abe et al. 2010; Wu et al. 2012), customer activity monitoring and user profiling (Fawcett and Provost 1996, 1999), click fraud (Jansen 2008; Kshetri 2010), securities fraud (Neville et al. 2005), malicious cargo shipments (Das and Schneider 2007; Eberle and Holder 2007) malware/spyware detection (Invernizzi and Comparetti 2012; Ma et al. 2009; Provos et al. 2007), false advertising (Lee et al. 2010), data-center monitoring (Li et al. 2011b), insider threat (Eberle and Holder 2009), image/video surveillance (Damnjanovic et al. 2008; Krausz and Herpers 2010), and many others.

In addition to revealing suspicious behavior, anomaly detection is vital for spotting rare events, such as rare disease outbreaks or side effects in medical domain with vital applications in the medical diagnosis. As "one person's signal is another person's noise", yet another application of abnormality detection is data cleaning—i.e. the removal of erroneous values or noise from data as a pre-processing step to learning more accurate models of the data.

1.1 Outliers versus graph anomalies

To tackle the abnormality detection problem, many techniques have been developed in the past decades, especially for spotting outliers and anomalies in unstructured collections of multi-dimensional data points. On the other hand, data objects cannot always be treated as points lying in a multi-dimensional space independently. In contrast, they may exhibit *inter-dependencies* which should be accounted for during the anomaly detection process (see Fig. 1). In fact, data instances in a wide range of disciplines, such as physics, biology, social sciences, and information systems, are inherently related

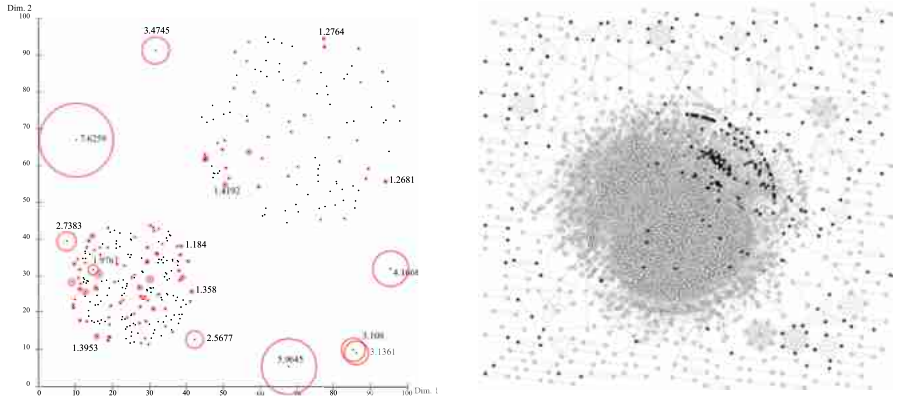


Fig. 1 Point-based outlier detection versus graph-based anomaly detection. **a** Clouds of points (multi-dimensional), **b** inter-linked objects (network)

to one another. Graphs provide a powerful machinery for effectively capturing these long-range correlations among inter-dependent data objects.

To give an illustrative example, in a reviewer-product review graph data, the extent a reviewer is fraudulent depends on what ratings s/he gave to which products, as well as how other reviewers rated the same products, to an extent how trustworthy their ratings are, which in turn again depends on what other products they rated, and so on. As can be seen, due to this long-range correlations in real-world datasets, detecting abnormalities in graph data is a significantly different task than that of detecting outlying points lying in a multi-dimensional feature space. As a result, researchers have recently intensified their study of methods for anomaly detection in structured *graph* data.

Why Graphs? We highlight four main reasons that make graph-based approaches to anomaly detection vital and necessary:

- *Inter-dependent nature of the data* As we briefly mentioned above, data objects are often related to each other and exhibit dependencies. In fact, most relational data can be thought of as inter-dependent, which necessitates to account for related objects in finding anomalies. Moreover, this type of datasets are abundant, including biological data such as the food web and protein–protein interaction (PPI) networks, terrorist networks, email and phone-call networks, blog networks, retail networks, social networks, to name but a few.
- *Powerful representation* Graphs naturally represent the inter-dependencies by the introduction of links (or edges) between the related objects. The multiple paths lying between these related objects effectively capture their long-range correlations. Moreover, a graph representation facilitates the representation of rich datasets enabling the incorporation of node and edge attributes/types.
- *Relational nature of problem domains* The nature of anomalies could exhibit themselves as relational. For example in the fraud domain, one could imagine two types of scenarios: (1) *opportunistic* fraud that spreads by word-of-mouth (if one commits fraud, it is likely that his/her acquaintances will also do so), and (2) *organized*

fraud that takes place by the close collaboration of a related group of subjects. Both of these scenarios point to relational treatment of anomalies. Another example can be given in the performance monitoring domain, where the failure of a machine could cause the malfunction of the machines dependent on it. Similarly, the failure of a machine could be a good indicator of the possible other failures of machines in close spatial proximity to it (e.g., due to excessive increase of humidity in that particular region of a warehouse).

– *Robust machinery* Finally, one could argue that graphs serve as more adversarially-robust tools. For example in fraud detection systems, behavioral clues such as log-in times and locations (e.g., IP addresses) can be easily altered or faked by advanced fraudsters. On the other hand, it may be reasonable to argue that the fraudsters could not have a global view of the entire network (e.g., money transfer, telecommunication, email, review network) that they are operating in. As such, it would be harder for a fraudster to *fit in* to this network as good as possible without knowing its entire characteristic structure and dynamic operations.

1.2 Challenges

We first discuss the very immediate challenge associated with our problem of interest. It stems from the fact that no unique definition for the problem of *anomaly detection* exists. The reason is that the general definition of an anomaly or an outlier is a vague one: the definition becomes meaningful only under a given context or application. The very first definition of an outlier dates back to 1980, and is given by Hawkins (1980):

Definition 1 (*Hawkins' Definition of Outlier, 1980*) "An outlier is an observation that differs so much from other observations as to arouse suspicion that it was generated by a different mechanism."

As one notices, the above definition is quite general and thus make the detection problem an open-ended one. As a result, the problem of anomaly detection has been defined in various ways in different contexts. In other words, the problem has many definitions often tailored for the specific application domain, and also exhibits various names such as outlier, anomaly, outbreak, event, change, fraud, detection, etc. In some applications, such as data cleaning, outliers are even called the noise—"one man's signal is another man's noise". Nevertheless, anomaly detection is one of the most evident problems in data mining with numerous applications, and the field of anomaly detection itself is well established.

Following the general definition of an outlier by Hawkins as given above, we provide a general definition for the graph anomaly detection problem as follows.

Definition 2 (*General Graph Anomaly Detection Problem*)

Given a (plain/attributed, static/dynamic) graph database,

Find the graph objects (nodes/edges/substructures) that are rare and that differ significantly from the majority of the reference objects in the graph.

For practical purposes, a record/point/graph-object is flagged as anomalous if its rarity/likelihood/outlierness score exceeds a user-defined or an estimated threshold.

In other words, an anomaly is treated as a data object or a group of objects that is rare (e.g., rare combination of categorical attribute values), isolated (e.g., far-away points in n-dimensional spaces), and/or surprising (e.g., data instances that do not fit well in our mental/statistical model, or need too many bits to describe under the Minimum Description Length principle (Rissanen 1999).

Next, we discuss the challenges associated with anomaly detection and attribution, which can be grouped into two: (1) data-specific challenges, and (2) problem-specific challenges. We also specifically highlight the challenges associated with graph-based anomaly detection.

Data-specific challenges Simply put, the challenges with respect to data are those of working with big data; namely volume, velocity, and variety of massive, streaming, and complex datasets. The same challenges generalize to graph data as well.

Scale and dynamics With the advance of technology, it is much easier than was in the past to collect and analyze very large datasets. As of today, Facebook (graph) consists of more than a billion users[1] (i.e. nodes), the Web (graph) contains more than 40 billion pages,[2] and over 6 billion users own a cell phone[3] which makes the telecommunication networks billion-scale graphs. Not only is the size of real data in tera- to peta-bytes, but also the rate at which it arrives is high. Facebook users generate billions of objects (e.g., posts, image/video uploads, etc.), billions of credit card transactions are performed every day, billions of click-through traces of Web users are generated each day, and so on. This kind of data generation can be thought of as streaming graph data.

Complexity In addition to (graph) data size and dynamicity, the datasets are rich and complex in content; including for example user demographics, interests, roles, as well as different types of relations. As such, incorporation of these additional information sources makes the graph representation a complex one, where nodes and edges can be typed, and have a long list of attributes associated with them.

As a result, methods which could scale to very large graphs, update their estimations when the graph changes over time, and that could effectively incorporate all the available and useful data sources are essential for graph-based anomaly detection.

Problem-specific challenges Additional challenges arise with respect to the anomaly detection task itself.

Lack and noise of labels One main challenge is that the data often comes without any class labels, that is, the ground truth of which data instances are anomalous and non-anomalous does not exist. Importantly, the task of manual labeling is quite challenging given the size of the data. To make things worse, even though endless human power were available, due to the complexity of certain labeling tasks, the labels are expected to be noisy and of varying quality depending on the annotator. According to Nobel laureate Daniel Kahneman "humans are incorrigibly inconsistent in making summary judgments of complex information" (Kahneman 2011). Surprisingly, they frequently give different answers when asked to evaluate the same information

[1] http://newsroom.fb.com/Key-Facts.

[2] http://www.worldwidewebsize.com/.

[3] http://huff.to/Rc2vbU.

twice. For example, experienced radiologists who evaluate chest X-rays as normal and abnormal are found to contradict themselves 20 % of the time when they see the same picture on separate occasions.

Due to challenges in obtaining labels, supervised machine learning algorithms are less attractive for the task of anomaly detection. It has been shown that humans can perform at best as good as random in labeling a review as fake or not, just by looking at its text (Ott et al. 2011) but can potentially do better by analyzing other relevant information such as the authors of the review. Likewise, a single transaction could be treated as anomalous only in relation to a history of previous transactions. These indicate that additional resources and information are needed to obtain human labels, which makes it costly to acquire them and harder and more time-consuming for the human annotators to sort through. What is more, the lack of true labels, i.e. ground truth data, also makes the evaluation of anomaly detection techniques challenging.

Class imbalance and asymmetric error The second challenge arises due to the unbalanced nature of the data; since anomalies are rare only a very small fraction of the data is expected to be abnormal. Moreover, the cost of mislabeling a good data instance versus a bad instance may change depending on the application, and further could be hard to estimate beforehand. For example, mislabeling a cancer patient as healthy could cause fatal consequences while mislabeling an honest customer as a fraudster could cause loss of customer fidelity. If learning-based techniques are to be employed, those issues regarding class imbalance and asymmetric error costs should be carefully accounted for.

Novel anomalies The third point is the wrist-fight nature of the problem setting, especially in the fraud detection domain. The more the fraudsters understand the ways the detection algorithms work, the more they change their techniques in a way to by-pass the detection and fit-in to the norm. As a result, not only the algorithms should be adaptive to changing and growing data over time, they should also be adaptive to and be able to detect novel anomalies in face of adversaries.

"Explaining-away" the anomalies Additional challenges lie in explaining the anomalies in the post-detection phase. This involves either digging out the root cause of an anomaly, telling a coherent story for the 'why' and 'how' of the anomaly, and/or presenting the results in a user-friendly form for further analysis. Most of the existing detection techniques, while doing a reasonably good job in spotting the anomalies, completely leave out this description or attribution phase and thus make it hard for humans to make sense of the outcome.

Graph-specific challenges All of the above challenges associated with the anomaly detection problem generalize to graph data. Graph-based anomaly detection, on the other hand, has additional challenges as well.

Inter-dependent objects The relational nature of the data makes it challenging to quantify the anomalousness of graph objects. While in traditional outlier detection, the objects or data points are treated as independent and identically distributed (i.i.d.) from each other, the objects in graph data have long-range correlations. Thus, the "spreading activation" of anomalousness or "guilt by associations" need to be carefully accounted for.

Variety of definitions The definitions of anomalies in graphs are much more diverse than in traditional outlier detection, given the rich representation of graphs. For example, novel types of anomalies related to graph substructures are of interest for many applications, e.g., money-laundering rings in trading networks.

Size of search space The main challenge associated with more complex anomalies such as graph substructures is that the search space is huge, as in many graph theoretical problems associated with graph search. The enumeration of possible substructures is combinatorial which makes the problem of finding out the anomalies a much harder task. This search space is enlarged even more when the graphs are attributed as the possibilities span both the graph structure and the attribute space. As a result, the graph-based anomaly detection algorithms need to be designed not only for effectiveness but also for efficiency and scalability.

1.3 Previous surveys and our contributions

There exist very comprehensive survey articles on anomaly and outlier detection in general that focus on points of multi-dimensional data instances. In particular, Chandola et al. (2009) covers outlier detection techniques, Zimek et al. (2012) focuses on outlier detection in high dimensions, and (Schubert et al. 2012) deals with local outlier detection techniques. In addition, survey and special issue journal articles that address anomaly, event, and change detection include (Chandola et al. 2012; Margineantu et al. 2010; Radke et al. 2005). Finally, due to the wide-range of application domains, fraud detection has attracted many surveys (Edge and Falcone Sampaio 2009; Flegel et al. 2010; Phua et al. 2010).

None of the previous surveys, however, discuss the anomaly detection problems in the particular context when one is confronted with large graph datasets. Further, they also do not focus, at least not directly, on graph-based detection techniques. Therefore, in this survey we aim to provide a comprehensive and structured overview of the state-of-the-art techniques for anomaly, event, and fraud detection in data represented as graphs. As such, our focus is notably different from, while being complementary to the earlier surveys. Specifically, our contributions are listed as follows.

1. Different from previous surveys on anomaly and outlier detection, we focus on abnormality detection in (large) graph datasets, using graph-based techniques.
2. We comprehensively explore unsupervised techniques that exploit the graph structure, as well as (semi-) supervised methods that employ relational learning.
3. We put the abnormality (anomaly, event, fraud) detection methods under a unifying lens, point out their connections, pros and cons (e.g., scalability, robustness, generality, etc.) and applications on diverse real-world tasks.
4. In addition to anomaly detection, we highlight the importance of explaining the detected anomalies and provide a survey of analysis tools and techniques for post-detection exploration and sense-making.

1.4 Overview and organization

We present our survey in four major parts. A general outline and a list of topics we cover are given as follows.

Table 1 Categorization of graph-based techniques in Sects. 2 and 3

	Plain	Attributed
Static	[Section 2.1]	[Section 2.2]
	Open	
Dynamic	[Section 3.2]	[Section 3.2]
		Many Open Challenges

I. *Anomaly detection in static graphs* (Sect. 2)
  (a) Anomalies in plain (unlabeled) graphs
  (b) Anomalies in attributed (node-/edge-labeled) graphs
II. *Anomaly detection in dynamic graphs* (Sect. 3)
  (a) Feature-based events
  (b) Decomposition-based events
  (c) Community- or clustering-based events
  (d) Window-based events
III. *Graph-based anomaly description* (Sect. 4)
  (a) Interpretation-friendly graph anomaly detection
  (b) Interactive graph querying and sense making
IV. *Graph-based anomaly detection in real-world applications* (Sect. 5)
  (a) Anomalies in telecom networks
  (b) Anomalies in auction networks
  (c) Anomalies in account networks
  (d) Anomalies in security networks
  (e) Anomalies in financial networks
  (f) Anomalies in opinion networks
  (g) Anomalies in the Web network
  (h) Anomalies in social networks
  (i) Anomalies in computer networks

The first part focuses on anomaly detection methods for *static* graph data, and is covered for both unlabeled (plain) and labeled (attributed) graphs. The second part focuses on change or event detection approaches for time-varying or *dynamic* graph data, based on edit distances and connectivity structure. The overview of the first two sections, along with the areas with open problems and challenges are provided in Table 1. In the third part, we stress the importance of anomaly *attribution* in revealing the root-cause of the detected anomalies and in presenting anomalies in a user-friendly form. We provide the state-of-the-art tools that could facilitate the post-analysis of detected anomalies for the crucial task of sense-making. Finally, in the fourth and last part we demonstrate graph-based anomaly detection techniques in action, where we discuss several real-world *applications* in diverse domains.

We show the general outline of our survey in Fig. 2 illustrating a sketch of the taxonomy. In the first two parts, namely static and dynamic graph anomalies, we focus on *unsupervised* techniques as well as *(semi-) supervised* approaches based on relational classification. Later in the third part, we focus on *qualitative analysis* techniques for the sense-making of spotted anomalies. Finally in part four, we present a long list of *applications* of graph-based anomaly detection in a wide range of networks, including finance, security, accounting, to name a few.

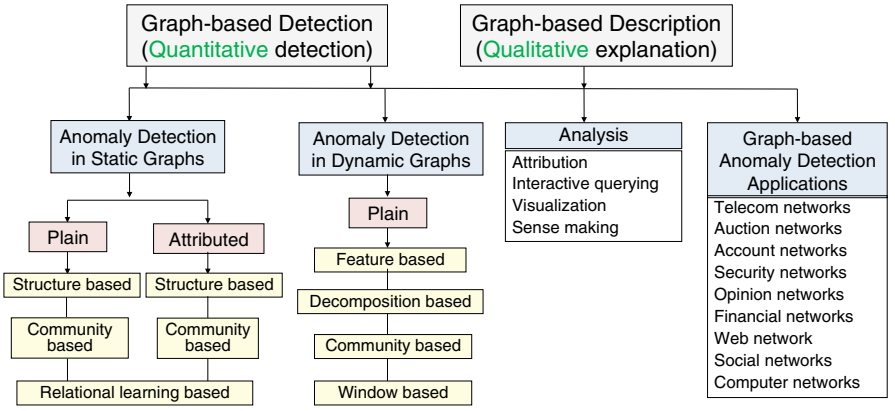


Fig. 2 Graph-anomaly detection: the outline of the survey

2 Anomaly detection in static graphs

In this section, we will address the anomaly detection in static snapshots of graphs. That is, the main task here is to spot anomalous network entities (e.g., nodes, edges, subgraphs) given the entire graph structure. We start with a very brief overview of outlier detection techniques in static clouds of data points and provide pointers for further reading. Next, we survey anomaly detection techniques for static graphs.

Overview: outliers in clouds of data points

Outlier detection deals with the problem of spotting outlying points in the (high-dimensional) feature space of data points. While not directly related, outlier detection techniques are employed in graph-based anomaly detection, for example after a graph-feature extraction step as we describe in this section. Thus it is beneficial to know of general outlier detection methods for spotting graph anomalies.

In outlier detection, some methods provide binary 0/1 classification of data points, i.e. outlier versus non-outlier, while most methods try to assign what is called an outlierness score that enables the quantification of the level of outlierness of the objects and subsequently rank the objects accordingly. For an illustration, see Fig. 1a.

There are several different ways of multi-dimensional outlier detection. The techniques can be classified into density-based (Breunig et al. 2000; Papadimitriou et al. 2003), distance-based (Aggarwal and Yu 2001; Chaudhary et al. 2002; Ghoting et al. 2008; Knorr and Ng 1998; Orair et al. 2010; Wang et al. 2011b), depth-based (Ruts and Rousseeuw 1996), distribution-based (Saltenis 2004), clustering-based (He et al. 2003; Lieto et al. 2008; Miller and Browning 2003; Wang et al. 2012c), classification-based (Abe et al. 2006; Hempstalk et al. 2008; Janssens et al. 2009), information theory-based (Ando 2007; Böhm et al. 2009; Smets and Vreeken 2011), spectrum-based (Liu et al. 2013), and subspace-based (Keller et al. 2012; Kriegel et al. 2012; Müller et al. 2010, 2012) techniques. Moreover, there exist outlier detection techniques that can work with categorical features (Akoglu et al. 2012c; Das and Schneider 2007; Smets and Vreeken 2011), or a mixture of both types of features (Otey et al. 2006) in addition to one-class classification-based approaches (Janssens et al. 2009; Pauwels and Ambekar 2011).

We refer the reader to a comprehensive survey on outlier detection for more discussion and details (Chandola et al. 2012) as well as a recent book by Aggarwal (2013) on outlier analysis with comprehensive details on these techniques.

Anomalies in static graph data

We will study anomaly detection in graph data under two settings: (1) plain graphs, and (2) attributed graphs. An attributed graph is a graph where nodes and/or edges have features associated with them. For example in a social network, users may have various interests, work/live at different locations, be of various education levels, etc. while the relational links may have various strengths, types, frequency, etc. A plain graph, on the other hand, consists of only nodes and edges among those nodes, i.e. the graph structure.

While the specific definition of the graph anomalies may vary, a general definition for the anomaly detection problem for static graphs can be stated as follows:

Definition 3 (*Static-graph anomaly detection* problem)

Given the snapshot of a (plain or attributed) graph database,
Find the nodes and/or edges and/or substructures that are "few and different" or deviate significantly from the patterns observed in the graph.

2.1 Anomalies in static plain graphs

For a given plain graph, the only information about it is its structure. This category of anomaly detection methods thus exploit the structure of the graph to find patterns and spot anomalies. These structural patterns can be grouped further into two categories: *structure-based* patterns and *community-based* patterns.

2.1.1 Structure based methods

We organize the structure-based approaches into two: feature-based and proximity-based. The first group exploits the graph structure to extract graph-centric features such as node degree and subgraph centrality, while the second group uses the graph structure to quantify the closeness of nodes in the graph to identify associations.

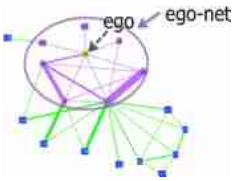
Feature-based approaches

Main idea This group of approaches uses the graph representation to extract structural graph-centric features that are sometimes used together with other features extracted from additional information sources for outlier detection in the constructed feature space. Essentially, these methods transform the graph anomaly detection problem to the well-known and understood outlier detection problem.

Graph-centric features One could use the given graph structure to compute various measures associated with the nodes, dyads, triads, egonets, communities, as well as the global graph structure (Henderson et al. 2010). These features have been used in several anomaly detection applications including Web spam (Becchetti et al. 2006) and network intrusion (Ding et al. 2012) as we will discuss in detail in Sect. 5.

The *node-level* features include (in/out) degrees, centrality measures such as eigenvector (Bonacich and Lloyd 2001), closeness (Noh and Rieger 2004), and betweenness (Freeman 1977) centralities, local clustering coefficient (Watts and Strogatz 1998), radius (Kang et al. 2011c), degree assortativity, and most recently, roles (Henderson et al. 2012). The *dyadic* features include reciprocity (Akoglu et al. 2012a), edge betweenness, number of common neighbors, as well as several other local network overlap measures (Gupte and Eliassi-Rad 2012; Liben-Nowell and Kleinberg 2003). Akoglu et al. (2010) introduce *egonet* features such as its number of triangles, total weight, principal eigenvalue, etc. as well as their pairwise correlation patterns. Henderson et al. (2011) enrich and extend the possible graph-based features with recursively aggregating existing features. The *node-group-level* features can be listed as compactness measures, such as density, modularity (Newman 2006), and conductance (Andersen et al. 2006). Finally, examples to *global* measures include number of connected components, distribution of component sizes (Kang et al. 2010), principal eigenvalue, minimum spanning tree weight, average node degree, global clustering coefficient, to name but a few.

Approaches A feature-based anomaly detection technique called ODDBALL is proposed by Akoglu et al. (2010), which extracts egonet-based features and finds patterns that most of the egonets of the graph follow with respect to those features. As such, this method can spot anomalous egonets (and hence anomalous nodes), as those that do not follow the observed patterns.



An *egonet* is defined as the 1-step neighborhood around a node; including the node, its direct neighbors, and all the connections among these nodes (an example is shown on the right figure). More formally an egonet is the induced 1-step sub-graph for each node. Given the egonets, the main question and challenge is which features to look at, as there is a long list of possible graph-based measures that can be extracted as egonet features. The paper proposes a carefully chosen subset of features (e.g., number of triangles, total weight of edges, etc.) that are (1) observed to yield patterns across a wide range of real-world graphs, and (2) fast to compute and easy to interpret.

The egonet features are then studied in pairs and several patterns in the form of power-laws are observed among strongly related features (e.g., number of neighbors and number of triangles). For a given egonet, its deviation from a particular pattern is computed based on its "distance" to the relevant power-law distribution. Each egonet then receives a separate deviation, or outlierness, score with respect to each pattern.

The multiple scores a node receives from various observed patterns brings up the question of how to combine them to obtain the final scores or final ranking. Several works (Gao and Tan 2006; Lazarevic and Kumar 2005) have proposed solutions to how to unite multiple outlierness scores. This problem is addressed in works on outlier ensembles, as discussed in Aggarwal (2012), Zimek et al. (2014).

There are several advantages of analyzing the egonet features in pairs, rather than in union. First, this facilitates the visualization of the patterns and outliers in 2-d for post-analysis. Second, the low dimensionality of the feature space helps with interpretability of the results, that is, one can tell what type of anomalies a node belongs to based on its deviation from a particular pattern, or "law". As an example, in Kang et al. (2014), the authors propose a package for visualization of billion-scale graphs by focusing on correlation plots (node features in pairs), as well as the spy plot and distribution plots for various features. The visualization tool is carefully designed to make the outliers pronounced even by a simple inspection.

Later work by Henderson et al. (2011) extends the feature base by recursively combining node-based ("local") and egonet-based (neighborhood) features. A recursive feature is defined as some aggregate value (e.g., mean, min, max) computed over any existing feature value (including recursive ones) among a node's neighbors. Intuitively, local and egonet features capture neighborhood information, whereas recursive features enable to go beyond direct neighborhood to capture more of "regional" or behavioral information. An iterative procedure with run time complexity linear in graph size is detailed in the paper to compute recursive features and prune highly correlated features on the go.

Proximity-based approaches

Main idea This group of techniques exploits the graph structure to measure closeness (or proximity) of objects in the graph. These methods capture the simple auto-correlation between these objects, where close-by objects are considered to be likely to belong to the same class (e.g., malicious/benign or infected/healthy).

Approaches Measuring the importance of the nodes in a graph is one of the most widely studied graph problems. PageRank (Brin and Page 1998) is one of the most popular algorithms which is based on random walks. A random walk on the (unweighted) graph jumps randomly from node to node. If currently present on a node u, a random walk in the next step jumps to one of its neighbors with equal probability $1/d_u$ where d_u is the degree of node u. The stationary probability distribution of the random walk on the graph is then considered to rank the nodes by their "importance".

This walk is known to converge if the transition matrix, the entries of which denote the jump probabilities between neighboring nodes, is stochastic, aperiodic, and irreducible (Feller 1968). On an undirected graph, the stationary probability of a random walk at node u is directly proportional to its degree d_u, and is independent of the starting node. On directed graphs, it is probable that the irreducibility condition, which states that there is a non-zero probability of going from any one node to any other, will be unmet (e.g., in the existence of sink nodes and multiple strongly connected components). To resolve these issues, a random restart of the walk is performed with a certain probability $\alpha \in (0, 1)$ (a.k.a. the damping factor), where the restart node is chosen at random.

A widely used graph-closeness measure that is also based on random walks but with an extension of restarts to a particular node is the Personalized PageRank (PPR) (Haveliwala 2003). Given a restart node q and the parameter α consider the random walk with restart, starting at node q, such that at any step when currently present at a node u, it chooses any of its neighbors with equal probability $(1-\alpha)/d_u$, and returns to the restart node q with probability α. The stationary probability at any node v of the random walk with restart is defined as the PPR score of v with respect to the restart node q. A more general version of this measure can be given for a set Q of restart nodes, where their restart probabilities sum to α. This type of PageRank computation is often referred to as the Biased PageRank. The stationary distribution of probabilities indicates the proximity (or closeness) of each node in the graph with respect to the (set of) restart node(s), and is higher for the nodes that have many, short, and high weighted paths to the restart node(s).

Another graph proximity measure that quantifies the closeness of two nodes in the graph is SimRank (Jeh and Widom 2002), which computes similarity of the structural context in which the graph objects occur, based on their relationships with other objects. It is often thought as measuring how soon two random surfers starting from the two nodes are expected to meet each other by randomly walking "backwards" in the graph. Several variants of SimRank are also proposed by Antonellis et al. (2008), Zhao et al. (2009), Chen and Giles (2013).

Finally, many link prediction approaches essentially quantify the similarity or closeness of pairs of nodes in the graph. Several such measures of varying computational complexity exist. The simple ones include the Jaccard proximity, which is the normalized number of common neighbors of the two nodes. Others include the total number of paths or node-disjoint paths. The slightly more complex Katz measure Katz (1953) counts all the paths weighted inversely proportional to the path length. For a well documented list and evaluation of these as well as other measures, we refer the reader to Liben-Nowell and Kleinberg (2003).

2.1.2 Community based methods

Main idea The cluster or community-based methods for graph anomaly detection rely on finding densely connected groups of "close-by" nodes in the graph and spot nodes and/or edges that have connections across communities. In fact, the definition of anomaly under this setting can be thought of as finding "bridge" nodes/edges that do not directly belong to one particular community.

Approaches Methods that exploit communities or proximity of nodes in the graph to spot (node) anomalies in bipartite graphs include (Sun et al. 2005). Several real-world data can be represented with bipartite graphs where the bridge nodes reveal interesting phenomena. Examples include publication networks: authors versus (unusual) papers written by authors from different research communities; P2P networks: users versus (cross-border) files; financial trading networks: stocks versus (cross-sector) traders; and customer-product networks: users versus ("cross-border") products.

The two main problems addressed in Sun et al. (2005) are (P1) how to find the community of a given node, which is also referred as the "neighborhood" of a node, and (P2) how to quantify the level of the given node to be a bridge node. For (P1),

the authors use random-walk-with-restart-based PPR scores (Haveliwala 2003) of all the nodes with respect to the given node, where those nodes with high PPR scores constitute the neighborhood of a node. On similar lines, for (P2) the pairwise PPR scores among all the neighbors of the given node are aggregated by averaging to compute a so-called "normality" score of a node. Intuitively, nodes with low normality scores have neighbors with low pairwise proximity to one another. This suggests that the neighbors lie in different, separate communities, which makes the given node resemble a bridging node across communities.

AUTOPART (Chakrabarti 2004) is based on the notion that nodes with similar neighbors are clustered together, and the edges that do not belong to any structure constitute anomalies (e.g., cross-cluster bridge edges). Similarly, nodes that have many cross-connections to multiple different communities are considered not to belong to any particular cluster and thus also constitute anomalies. For finding communities in a graph, the algorithm re-organizes the rows and columns of the adjacency matrix into a few homogeneous blocks (of either low or high density). These blocks have the property of containing nodes that are more densely connected together than with the rest of the nodes in the graph—which is the underlying idea in clustering. Chakrabarti (2004) develops a parameter-free, iterative algorithms based on the Minimum Description Length principle (Rissanen 1999) for rearranging the rows and columns, as well as for finding the best number of blocks or node groups automatically without requiring any user input.

Another method that aims to spot (node and edge) anomalies based on graph communities (Tong and Lin 2011) relies on matrix factorization. Matrix factorization has been used to address several problems ranging from dimensionality reduction (Ambai et al. 2011; Nikulin and Huang 2012) to (graph) clustering (Kuang et al. 2012; Wang et al. 2012b). The factorization of a data matrix $\mathbf{A}$ is often formulated as $\mathbf{A} = \mathbf{X} \times \mathbf{Y} + \mathbf{R}$, where $\mathbf{X}$ and $\mathbf{Y}$ are the low rank factors and $\mathbf{R}$ denotes the residual matrix. In traditional non-negative matrix factorization (NMF), there exists additional constraints on the non-negativity of both $\mathbf{X}$ and $\mathbf{Y}$, which for example aids in determining the communities. Different from this traditional approach, the main idea for finding anomalies is to waive these original constraints but instead enforce non-negativity constraints on the *residual* matrix for interpretability (hence the name NRMF). The approach proves effective in spotting "strange" connections, such as port-scanning-like or ddos-like activity, bridging connections, as well as bipartite-core structures with the help of the non-negative residual matrix.

The "bridge" nodes and/or edges can be seen as intrusive connectors and/or connections that cross the community boundaries in computer security. For example, Ding et al. (2012) regards intrusion as entering a community to which one does not belong, and looks for communication that does not respect the community boundaries. Analysis shows that cut-vertices (vertices the removal of which disconnects the graph into components) correspond well with ground-truth traffic sources that attempted an intrusion, by sending malicious or unwanted traffic. This work essentially shows one of the real-world applications that community-based anomaly detection methods prove to be effective.

Other community-based network outlier detection methods directly focus on network clustering, and in the process, spot hubs and outliers as a by-product (Sun et al.

2010; Xu et al. 2007). To find network clusters, SCAN (Xu et al. 2007) exploits the neighborhood of vertices; vertices sharing many neighbors are grouped into the same clusters. As such, vertices that are bridging many clusters are labeled as hubs, whereas those that cannot be assigned to any community are flagged as outliers. To overcome the issue of selecting the minimum similarity threshold parameter of Xu et al. (2007), Sun et al. (2010) proposes a novel clustering framework called GSKELETONCLU that also aims to find hubs and outliers as byproduct of the graph clustering.

2.2 Anomalies in static attributed graphs

For certain kinds of data, it is possible to have a richer graph representation, in which nodes and edges exhibit (non-unique) attributes. Examples to such graphs include social networks with user interests as attributes, transaction networks with time, location, and amount as attributes, cargo shipments with visited ports, financial information, type of transported goods as attributes, and so on.[4]

This category of anomaly detection methods on attributed graphs exploit the structure as well as the coherence of attributes of the graph to find patterns and spot anomalies. These methods can also be grouped into two: *structure-based* and *community-based* methods. In a nutshell, the structure-based methods exploit frequent substructure and subgraph patterns to spot deformations in these patterns, while community-based methods aim to spot what is called community-outliers that do not exhibit the same characteristics as the others in the same community.

2.2.1 Structure based methods

Main idea Structure based approaches mainly aim to identify substructures in the graph that are rare structurally, i.e. connectivity-wise, as well as attribute-wise. As such, inverse of frequent attributed subgraphs are sought out. The differences from these normative substructures are quantified in various ways as we describe below.
Approaches One of the earliest works on attributed graph anomaly detection by Noble and Cook (2003) addresses two related problems: (P1) the problem of finding unusual substructures in a given graph, and (P2) the problem of finding the unusual subgraphs among a given set of subgraphs, in which nodes and edges contain (non-unique) attributes. Main insight to solve these problems is to look for structures that occur infrequently, which are roughly opposite to what is called the "best substructures". Intuitively, best substructures are those that occur frequently in the graph and thus can compress the graph well. An information-theoretic formulation based on the Minimum Description Length (MDL) principle (Rissanen 1999) that trades off between compression quality and the size of such substructures (as the entire graph is the best compressor) is devised as an objective.

The main idea for detecting unusual substructures (P1) is to define a measure that is inversely related to the MDL-based measure defined for the best substructures and rank substructures by this new measure. Similarly, the main idea for finding the unusual

[4] We will use the words 'attribute' and 'feature' interchangeably throughout text.

subgraphs (P2) is to define a measure that penalizes those subgraphs containing few common (i.e. best) substructures, making them more anomalous.

The methods by Noble and Cook (2003) essentially build on frequent subgraphs with categorical attributes. On the other hand, most often datasets come with a mix of both numerical and categorical attributes, e.g., dollar amounts in transaction data and number of (e.g., Ping, SYN, etc.) requests in network log data. Treating each numerical value as a distinct attribute loses ordering and closeness information. To address this problem (Davis et al. 2011) proposed *discretizing* the numerical attributes, where the majority "normal" values are assigned the same single categorical attribute, and all other values are assigned their "outlierness" score. Several discretization mechanisms, e.g., based on fitting probability density functions, k-NNs, outlier detection (in particular LOF Breunig et al. 2000), and clustering (CbLOF He et al. 2003), have been studied. We also include other discretization techniques that could apply under this setting such as SAX (Lin et al. 2003), MDL-binning (Kontkanen and Myllymki 2007), and minimum entropy discretization (Fayyad and Irani 1993).

Later work by Eberle and Holder (2007) follows a different insight to look for anomalies than the previous work. Rather than focusing on infrequent substructures, they go after those substructures that are *very similar to, though not the same as, a normative (i.e. best) substructure*. A statement by United Nations Office on Drugs and Crime corroborates this insight: "The more successful money-laundering apparatus is in imitating the patterns and behavior of legitimate transactions, the less the likelihood of it being exposed."

Using the insight that an intruder would make at most a certain number of changes to blend in with the normal data instances and lower their chances of being detected glaringly, the work by Eberle and Holder (2007) formulates three types of anomalous cases based on modification, insertion, and deletion. They formulate various anomaly scores that use both (in)frequency and modification cost (the lower, the more anomalous). We note that the anomalies are assumed to consist of only one type of anomaly, which is prone to missing e.g., a deletion followed by a modification.

On similar lines, Liu et al. (2005) use subgraphs of attributed graphs for detecting non-crashing software bugs. In this type of application domain, every execution of a software program is represented as an attributed graph called behavior graph, where nodes denote functions (attributed with function names), and (directed) edges depict function calls or function transitions. Different from previous methods discussed so far, the idea here is to train a classification model that takes as input positive and negative behavior graphs for correct and incorrect executions, respectively. First, (closed) frequent subgraphs are extracted from a set of behavior graphs, which are then used as features in training a classification model.

The pattern-based (e.g., frequent substructures) anomaly detection techniques as described above make them interpretable and amenable for post-analysis by domain experts to reveal the root cause. Moreover, these methods are quite generally defined such that they can be applied on various types of data and scenarios where the data can be represented as attributed (sub)graphs (like the software execution flow-graphs). On the other hand, this generality comes at a cost of high false positive rates, as not all rare occurrences can be attributed to anomalous cases. Furthermore several user-specified thresholds, such as the amount of alteration threshold or subgraph

frequency threshold, make it hard to trade off false positive and false negative rates by users.

2.2.2 *Community based methods*

Main idea These approaches aim to identify those nodes in a graph, often called the community outliers, the attribute values of which deviate significantly from the other members of the specific communities that they belong to. For example, a smoker in a community of vastly non-smoker baseball players is an example of a community outlier. As such, communities are analyzed based on both link and attribute similarities of the nodes they consist of. While some methods aim to detect outliers simultaneously with detecting the communities in the graph, some perform the outlier detection as a second step after performing the attributed graph clustering.

Approaches Gao et al. (2010a) differentiates graph-based community outlier detection from three closely related problems; namely, global outlier detection that only considers node attributes, structural outlier detection that only considers links (e.g., Xu et al. 2007 as is discussed in the previous section), and local outlier detection that only considers attribute values of direct neighbors. While interesting on their own right, these three types of methods are prone to miss outliers in the unison of these—outliers with respect to other community members' attributes. They develop a unified probabilistic model that simultaneously finds communities as well as spot community outliers. The unsupervised learning algorithm called CODA alternates between the two steps of parameter estimation (fixed cluster assignment), and inference for cluster assignments (fixed parameters). As with the nature of such learning algorithms, the good initialization of clusters at the beginning is a crucial step for the algorithm to reach a good solution. Moreover, the convergence of the algorithm is not guaranteed. One way that is used to find a good initialization is to employ a graph clustering algorithm to find a first-cut good quality clustering based on only the link structure, which also helps with faster convergence.

Müller et al. (2013) propose a node outlier ranking technique in attributed graphs called GOUTRANK. Different from Gao et al. (2010a), their main insight into community outlier detection is the fact that the complex anomalies could be revealed in only a subset of relevant attributes (a.k.a. subspaces). This becomes more apparent especially in high dimensional feature spaces due to the curse of dimensionality (Beyer et al. 1999). Roughly speaking, all objects appear to be sparse and dissimilar in high dimensions, or in other words, all the distances between pairs of objects look similar causing all the objects to be equally (dis)similar to one another. In their work, they also consider quantifying the degree of deviation for each node-outlier which is beyond binary detection. As such, they address two main challenges associated with community outlier detection in attributed graphs; the selection of subgraphs and subspaces, and the scoring of nodes in multiple subspace clusters.

Recently, Perozzi et al. (2014) proposed a new formulation, called FOCUSCO, on finding user-driven cluster or community outliers in graphs with node attributes. Given an initial set of nodes provided by a user, the approach first identifies a subset of attributes, i.e. an attribute subspace, that the given nodes agree on (called "focus attributes") and then finds clusters of densely connected nodes in the graph that also

agree on this attribute space (called "focused clusters"). The focus attributes are interpreted as properties that make the cluster nodes "click". Based on these focused clusters, an outlier is defined as a node which belongs to a cluster structurally but deviates from it in focus attributes. In other words, nodes that are tightly connected to many other nodes in a cluster but that do not exhibit similar focus attributes constitute the outliers. The authors develop an algorithm that extracts focused clusters and their respective outliers simultaneously. The detection of outliers in this setting is mainly geared by user preference and the description of outliers is achieved via the specific focus attributes that they violate.

Finally, there is a large body of other related work that mainly addresses the problem of attributed graph clustering—without focusing on outlier detection, including Akoglu et al. (2012b), Boden et al. (2012a,b), Günnemann et al. (2010, 2012). These methods could form the basis for community outlier detection in a post-processing step, as opposed to integrated clustering and outlier detection in one algorithm as with the techniques discussed above. During post-processing, nodes that could not be assigned to a "large enough" community (e.g., singletons or micro-clusters) could be analyzed further, or the nodes the removal from a community of which increases a "fitness" score of the community can be flagged as abnormal.

2.2.3 Relational learning based methods

Main idea This group consists of network-based collective classification algorithms the main idea of which is to exploit the relationships between the objects to assign them into classes, where the number of classes is often two: anomalous and normal. Different from proximity-based approaches which aim to quantify auto-correlations among graph objects, these algorithms are often more complex and thus can model and exploit more complex correlations between the graph objects.
Approaches Classification is the problem of assigning class labels to, or shortly labeling, data instances based on their observed attributes. Anomaly detection can be formulated as a classification problem, when one has a representative labeled data available. For example, determining whether a Web page is spam or non-spam based upon the words that appear in it and identification of benign/malicious web pages, fraud/legitimate transactions, etc. can all be thought of as two-way classification problems. When the labeled data size is reasonably large, one can employ fully supervised classification, where the labeled data is used for model learning. When labeled data is scarce, but still available, one can employ semi-supervised classification, where the learning is done by simultaneously using labeled and unlabeled data.

In traditional statistical machine learning approaches, the instances are often assumed to be i.i.d and often the learning algorithms ignore the dependencies among data instances. Relational classification, on the other hand, is the task of inferring the class labels of a network of objects simultaneously or collectively. The underlying assumption in relational classification is that the relationships between objects carry important information for classifying the objects, such as two linked Web pages. In many cases, there is a simple auto-correlation between the objects, where the linked objects are likely to have the same labels (e.g., spam pages link to other spam pages, infected people are linked to other infected people). In other cases, more complex

correlations may be exhibited (e.g., fraudsters trade with honest people and not with other fraudsters).

There exist a large amount of research on relational classification methods (Friedman et al. 1999; Jensen et al. 2004; Lu and Getoor 2003; Macskassy and Provost 2003; Neville and Jensen 2000, 2003; Neville et al. 2003; Taskar et al. 2002). Generally, these methods exploit one or more of the following input:

1. the class labels of its neighbors, and
2. the node attributes (features),
3. the attributes of the node's neighbors.

We note that although it is possible that some methods described in this section are amenable to use only the first type of information, i.e. nodes' class labels, and need not exploit node attributes, most methods are easily generalizable to incorporating node attribute information, if available. Thus, we cover these methods in this section that is attributed to anomaly detection in attributed graphs, and remark that some methods do apply to plain graphs as well.

Relational classification methods can be categorized into local and global methods (Sen et al. 2008). The local algorithms build local predictive models for the class of a node in the network and use often iterative inference procedures to collectively classify the unlabeled objects. The second group of algorithms define a global formulation of class dependencies and use inference algorithms to solve for the assignments that would maximize the joint probability distribution.

The techniques for the local methods can differ in both the local models and the inference methods that they use. Chakrabarti (2007) use Naive Bayes models for the local attributes of the object and the class labels of the neighbor objects. They then use mean field relaxation labeling for the inference. Neville and Jensen (2000) also use a Naive Bayes model for the attributes, but they use an iterative classification algorithm (ICA) for inference. In later work, they investigate the use of relational dependency networks (RDNs) and the inference algorithm is based on Gibbs sampling (Neville and Jensen 2003). Lu and Getoor (2003) use logistic regression as a local model and ICA for inference but they explore various ways of aggregation that can be used for the class labels of the related objects. For sparsely labeled networks, Gallagher et al. (2008) propose ways to infer "ghost" edges based on graph closeness to improve classification performance.

As for the global methods, Friedman et al. (1999) use probabilistic relational models (PRMs) as a (full joint) model and then use Loopy Belief Propagation (LBP) (Yedidia et al. 2003) for the inference. Taskar et al. (2002) use relational Markov networks (RMNs) as a (full joint) model and also use LBP for inference. Macskassy and Provost (2003) propose a simple baseline algorithm called (probabilistic) weighted-vote relational network (wv-RN) classifier where they use only the class labels of objects for classification; they infer the class label of an object by taking a weighted average of the potentially inferred class labels of the related objects iteratively. Other global formulations are based on Markov logic networks (MLNs).

All in all, the relational inference algorithms mentioned above can be listed as

– Iterative classification algorithm (ICA)
– Gibbs sampling

– Loopy belief propagation
– Weighted-vote relational network classifier

All these algorithms are fast, iterative, approximate inference algorithms, since exact inference is known to be NP-hard in arbitrary networks (Cooper 1990). Moreover, convergence is not guaranteed for any of them. Node ordering for updates (e.g., random, diversity-based) may alter the classification results. For local methods, additional challenges include feature construction and local classification. For feature construction one has to decide whether to consider in-, out-, or both-neighbors, and aggregation method of neighbor labels (e.g., max, mode, count), as well as choice of neighbors to consider (e.g., all, top-k most confidently labeled). With respect to local classification, one requires training data, and has to choose the classifier type (e.g., Naive Bayes, logistic regression, k-NN, SVM).

With respect to scalability, these methods mostly rely on message passing or information aggregation over neighbors and thus scale linearly with number of edges in the graph. Recently, techniques to speed up inference for massive graphs, especially based on LBP, have been proposed by Kang et al. (2011a), Koutra et al. (2011).

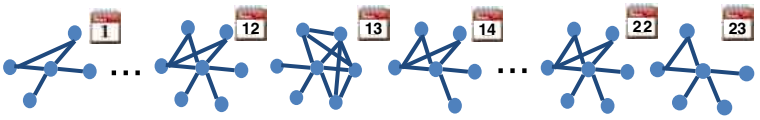
Before concluding this section on graph anomaly detection techniques in static graphs, we provide a summary and qualitative comparison of the detection algorithms presented in this section in Table 2.

3 Anomaly detection in dynamic graphs

3.1 Overview: event detection in time series of data points

In the literature, there is abundance of work on event detection on data series: statistical quality control (Montgomery 1997); the famous auto-regressive moving average model used for predictions (Box and Jenkins 1990); a drift detection method (Gama et al. 2004); a chart-based approach for monitoring temporal, medical data (Grigg et al. 2003); change detection in categorical data (Bay and Pazzani 1999); StreamKrimp, an MDL-based algorithm (Leeuwen and Siebes 2008); detection of disease outbreaks (Wong et al. 2005). A nice tutorial that covers event detection in data series is Neill and Wong (2009), a survey on outlier detection for temporal data is Gupta et al. (2013), and an extension of the survey that includes techniques for time series, data streams, graphs and spatio-temporal data is given in a very recently published book (Gupta et al. 2014).

3.2 Event detection in time series of *Graph* data



This section provides an overview of the anomaly detection algorithms that have been proposed for *dynamic* or *time-evolving* graphs (i.e. sequences of static graphs), the

Table 2 Qualitative and quantitative comparison of anomaly detection algorithms for *static* graphs

Algorithm	Graphs						Output format	Visualization
	Weighted	Unweighted	Attributed	Plain	Linear	Parameter-free		
OddBall (Akoglu et al. 2010; Henderson et al. 2011)	✓	✓	✗	✓	✗	✓	[0, ∞] node anomaly scores	Pairwise feature scatter plots, egonets
Sun et al. (2005)	✓	✓	✗	✓	✗	✓	[0, 1] node normality scores	Score distribution
Autopart (Chakrabarti 2004)	✗	✓	✗	✓	✓	✓	Binary edge classification	Adjacency matrix organized by node clusters
NNrMF (Tong and Lin 2011)	✓	✓	✗	✓	✓	✓	Binary edge/node classification	Residual matrix
Ding et al. (2012)	✓	✓	✗	✓	✗	✓	Binary node classification	Egonets
SCAN (Xu et al. 2007)	✗	✓	✗	✓	✓	✗	Binary node classification	Clustering with hub & outlier nodes
gSkeletonClu (Sun et al. 2010)	✓	✓	✗	✓	✗	✓	Binary node classification	Clustering with hub&outlier nodes
Subdue (Noble and Cook 2003)	✗	✓	✓	✗	✗	✗	Substructure anomaly score $\in \mathbb{N}$	Graph substructures

Table 2 continued

Algorithm	Graphs						Output format	Visualization
	Weighted	Unweighted	Attributed	Plain	Linear	Parameter-free		
SUBDUE (Noble and Cook 2003)	✗	✓	✓	✗	✗	✗	[0, 1] subgraph anomaly score	Subgraphs
SUBDUE (Eberle and Holder 2007)	✗	✓	✓	✗	✗	✗	[0, ∞] subgraph anomaly score	Modified subgraphs
Liu et al. (2005)	✗	✓	✓	✗	✗	✗	Binary graph classification	Graphs with traced-back crashing points
CODA (Gao et al. 2010a)	✓	✓	✓	✗	✓	✗	Binary node classification	Graph clustering with community outlier nodes
GOUTRANK (Müller et al. 2013)	✗	✓	✓	✗	✓	✗	[0, ∞] node anomaly scores	Subspace clustering and outlier nodes

The first four columns refer to the type of graphs that an algorithm can be applied to (with or without weights on the edges, with or without attributes for the nodes); "Linear" holds true for those methods that have time complexity linear in the number of edges of the input graph (and more otherwise); "Parameter-free" methods correspond to those that do not expect any user-specified input parameters; "Output format" corresponds to the output type/format of the method (e.g., anomaly scores and their ranges, binary output/classification e.g., anomalous or not); and "Visualization" refers to the graphical means used—if any—to present the anomalous instances to the user (e.g., distribution plots, graph with the anomalous nodes/edges annotated)

evolution of which as well as their communities have been studied by several research groups (Backstrom et al. 2006; Leskovec et al. 2005). In addition, Aggarwal and Subbian (2014) provides a comprehensive survey on evolutionary network analysis. The anomaly detection problem for dynamic graphs, which is the main focus of our survey, is also known as temporal anomalous pattern detection, event detection, change-point detection, and is commonly defined as follows:

Definition 4 (*Dynamic-Graph Anomaly Detection Problem*)

Given a sequence of (plain or attributed) graphs,
Find (i) the timestamps that correspond to a *change* or *event*, as well as
(ii) the top-k nodes, edges, or parts of the graphs that contribute most to the change (*attribution*).

Depending on the application domain, the requirements of the algorithms vary, but among the most usual desired properties are:

- *Scalability*. As instructed by the size and volume of the graphs that are produced daily, ideally, the algorithms should be linear or sub-linear on the size of the input graphs. In the dynamic setting, an additional, desired property is that the algorithm should be linear on the size of the *update* of the input graphs.
- *Sensitivity to structural and contextual changes*. The anomaly detection methods should be able to discern structural differences between the input graphs under comparison (e.g., missing/new edges, missing/new nodes, changes in the weights of the edges), as well as changes in other properties of the graphs, such as labels of the nodes or edges.
- *Importance-of-change awareness*. The algorithms should be sensible to the type and extent of change. Changes in "important" nodes, edges or other graph attributes should result in greater anomaly scores, than changes in less important structures.

A brief overview of the anomaly detection algorithms for time-evolving graphs is given in Bilgin and Yener (2006). However, the abundance of time-evolving graphs in the recent years has led to increasing interest in them, and subsequently new research has been carried out in this area. In the following subsections, we classify the dynamic graph anomaly detection algorithms based on the type of "graph summary" or "footprint" they use, and the type of *events* they detect: (1) feature-based (e.g., nodes, edges, edge weights), (2) decomposition-based, (3) community or clustering-based, and (4) window-based.

3.2.1 Feature based events

Main idea The key idea behind the feature-based methods is that similar graphs probably share certain properties, such as degree distribution, diameter, eigenvalues (Kang et al. 2011b; Watts 1999). The general approach in detecting anomalous timestamps in the evolution of dynamic graphs can be summarized in the following steps:

- Extract a "good summary" from each snapshot of the input graph.
- Compare consecutive graphs using a distance—or equivalently, similarity-function. A nice survey on similarity measures is given in Cha (2007).

– When the distance is greater than a manually or automatically defined threshold (or conversely, the similarity is smaller than a threshold), characterize the corresponding snapshot as anomalous.

When it comes to comparing consecutive graphs, there is no definite answer about the graph features that one should compare among the various timestamps. The novelty of each proposed algorithm lies in the "graph summary" it constructs, the distance/similarity function it uses, as well as the way it defines and chooses the threshold to flag an instance as anomaly. The majority of feature-extraction-based algorithms derive just a similarity score between two input graphs, without doing attribution; in other words, the algorithms usually cannot detect the nodes or regions of the graphs that changed most.

Approaches Shoubridge et al. (2002) and Bunke et al. (2006b) propose several "graph footprints" and metrics for monitoring communication networks:

- Maximum Common Subgraph (MCS) distance of the adjacency or the "2-hop" matrices (=square of adjacency matrix),
- Error correcting graph matching distance (Shoubridge et al. 1999), which refers to the number of edit operations needed to convert a graph to another, and the costs of each operation may vary,
- Graph Edit Distance (GED), which is a simplification of the previous distance, where only topological changes are allowed (i.e. no changes in edge weights),
- Hamming distance for the adjacency matrices of the graphs, which essentially counts the number of different entries in the matrices,
- variations of edge-weight distances,
- λ-distance of the adjacency, the "2-hop", or Laplacian matrices, which is defined as the differences in the whole graph spectra, or the top-k eigenvalues of the respective matrices. Peabody (2003) also proposes the λ-distance of the Normalized Laplacian matrices.

At this point, it is worth mentioning that although we consider λ-distance a graph-feature-based anomaly detection technique, it can be also classified as decomposition-based technique, since the extraction of the eigenvalues of a matrix is done by its decomposition (SVD Golub and Van Loan 1996, PCA Pearson 1901, LSI Deerwester et al. 1990, CUR Drineas et al. 2006).

Shoubridge et al. (2002) and Bunke et al. (2006b) use the metrics for tracking sudden changes in communication networks for performance monitoring. The best approaches, in terms of change awareness, are the GED and MCS, both of which are NP-complete, but the former approach can be simplified given the application and it becomes linear on the number of nodes and edges in the graphs. In Shoubridge et al. (2002), the graph symmetric difference and difference in the vertex neighborhood subgraphs are proposed for change attribution.

The authors in Bunke et al. (2006b) also go beyond the simple features, such as nodes, edges and weights, and introduce also more complex graph distance functions; the modality distance is defined as the Euclidean distance between the Perron vectors of the input graphs. Moreover, the authors propose the median graph distance; the median graph was first introduced by Dickinson et al. (2002), and it is the graph that minimizes the sum of the edit distances to all the graphs in the sequence.

Two variations of GED with simple and non-linear cost functions for the allowed operations, which also accommodate the weights of the input graphs is given in Kapsabelis et al. (2007), and used for accurate monitoring of dynamic computer networks. More details about the graph edit distance can be found in the survey (Gao et al. 2010b).

In Bunke et al. (2006a), the authors do not only compute the distances between consecutive graph instances, but all the pairwise distances (GED), and then apply an offline multidimensional scaling (MDS) procedure; each graph is represented by a point in the 2d-plane, and the distances between the points reflect their structural distances. This way the authors provide a nice, graphical representation of the changes that occur in a time-evolving graph; points that deviate from the mass of points correspond to anomalous timestamps or events.

Gaston et al. (2006) detect abnormal changes in time-evolving communication graphs using the diameter distance—i.e. the difference in the graph diameter—which is defined as the greatest of the longest shortest paths for all vertices.

One of the early works in this category was conducted by Pincombe (2005). The main idea of this work is to extract a single feature from each graph instance, and then, by using an appropriate metric, compare this feature in consecutive time ticks. Next, the resulting time series of the feature distances is modeled as an auto-regressive moving average process (ARMA) (Box and Jenkins 1990), and the residuals (deviations from the model) are evaluated. The instances whose residuals exceed a threshold are considered anomalous. Briefly, ARMA is a model for describing time series by using two polynomials (the first for auto-regression, the second for moving average); it is widely used for predicting values in time series. Among the 10 metrics that Pincombe used—weight, maximum common subgraph (MCS) weight/edge/vertex, graph/median edit, modality, diameter, entropy, spectral distance—, the MCS edge, MCS vertex, edit, median and entropy were able to detect the anomalies that were introduced in a time-evolving IP traffic dataset. Recently, another work that detects anomalies in time series (*not graph data*), was introduced by Zhu and Sastry (2011). Their approach uses a General Likelihood Ratio (GLR) test based on Kalman filter for estimating the parameters of Auto-regressive Integrated Moving Average (ARIMA). The main insight remains the same; the detection of anomalies is based on the residuals of the filter, but in this case the monitoring of the residuals is done with the GLR test. Since this work is not used on graph data, we do not elaborate more here; however, it appears to be a nice alternative for the approach used in Pincombe (2005).

Along the same lines, the authors in Papadimitriou et al. (2008) introduce five graph similarity functions for directed, time-evolving web graphs: vertex/edge overlap similarity, vertex ranking, vertex/edge vector similarity, sequence similarity, and signature similarity. Among these metrics, the one that performs best in terms of change detection in web graphs is the Signature Similarity (SS), which is based on the SimHash algorithm. This algorithm uses as features the nodes and edges of the input graphs, weighted appropriately by their PageRank.

Berlingerio et al. (2012) use a graph similarity approach for discontinuity detection in daily instances of social networks. In a nutshell, NetSimile consists of three phases: (1) Feature Extraction. The focus is on local and egonet-based features (e.g., number of neighbors, clustering coefficient, average of neighbors' degrees); (2) Fea-

ture Aggregation. The node×features matrix of the first phase is converted to a single "signature" vector that consists of the median, mean, standard deviation, skewness and kurtosis of each extracted feature over all the nodes in the graph; (3) Comparison. The signature vectors are compared using the Canberra Distance, and a single similarity score is produced for consecutive timestamps of the graph sequence. The days that have low similarity score with the surrounding days are characterized as anomalous.

Another recent work, Koutra et al. (2013b), proposes a complex graph-feature-based similarity approach, DELTACON, for discontinuity detection, which enjoys several desired properties. The intuition behind the method is to compare the pairwise node affinities of consecutive snapshots of the graph sequence. These node affinities are computed in this work by a fast variant of Belief Propagation (Koutra et al. 2011). The matrices of pairwise node similarity matrices are then compared using the Matusita Distance (which is related to the Euclidean Distance), and the distance is finally transformed to similarity. A faster algorithm that avoids computing all the pairwise similarity scores is also proposed, and it is based on the idea of finding the similarity of all the graph nodes to non-overlapping groups of nodes (instead of each node individually). Once the time series of the consecutive-graph similarities is obtained, Quality Control with Individual Moving Range (Montgomery 1997) is used to spot the anomalous daily ENRON-graph instances.

In contrast to the most of the previous works that detect anomalous *graph* instances, the following algorithms spot anomalous *nodes* in a graph sequence.

The key idea in Akoglu and Faloutsos (2010) is the following: *A node is anomalous at some time frame, if its "behavior" deviates from its past "normal behavior"*. The authors build the "behavior" of the nodes by extracting various egonet node features (e.g., weighted and unweighted in- and out-degree, number of neighbors, number of triangles) from each snapshot of the graph sequence, and create a correlation matrix of node "behaviors" at each time window using Pearson's correlation coefficient. For each correlation matrix (one per time window), the principal eigenvector, which has one entry per node, is computed. By placing all the corresponding entries of the eigenvectors in a vector, the "eigen-behavior" vector of each node is obtained, and compared against its typical "eigen-behavior", which is found by using averaging in the past time windows or SVD. The similarity between the "behaviors" is evaluated using the Euclidean dot-product. For low similarity between a node's "behavior" and its past "behaviors", the corresponding time window is reported as anomalous.

Last but not least, the work of Rossi et al. (2012) builds on top of ROLX (Henderson et al. 2012)—an NMF and MDL-based role extraction algorithm– to develop an algorithm that recursively extracts structural global and node features, and determine the nodes' roles (e.g., centers of stars, bridge nodes) over time. The authors use the method for understanding and tracking the network dynamics and evolution, but propose comparing the obtained node feature vectors over time in order to detect anomalous patterns. Another similar approach, DBMM (Rossi et al. 2013), that builds on top of ROLX combines feature extraction, matrix decomposition, and a window-based analysis to model the node behavior in temporal graphs, predict future behaviors and spot anomalies. First, the NMF and MDL-based role extraction algorithm computes the node group memberships. Then, by taking into account k previous time steps, a role transition model per node is generated. The approach does not detect anomalous graph

instances, but anomalous *n*odes per time step in decreasing order of anomalousness; the anomaly score of each node is defined as the difference between its estimated and true mixed membership.

3.2.2 Decomposition based events

Main idea The decomposition-based approaches detect temporal anomalies by resorting to matrix or tensor decomposition of the time-evolving graphs, and interpreting appropriately selected eigenvectors, eigenvalues or singular values. The methods can be divided into two categories based on the representation of the graphs: matrices versus tensors.

Approaches We will first discuss the matrix-oriented approaches. These include the λ-distance (Bunke et al. 2006b; Peabody 2003; Shoubridge et al. 2002), and the algorithms proposed in Akoglu and Faloutsos (2010) and Rossi et al. (2013) that were presented in Sect. 3.2.1. All of these approaches use graph features generated by SVD, eigenvalue decomposition or NMF, and, thus, can be also classified as decomposition-based anomaly detection techniques.

An additional work that handles each graph in the sequence separately by its matrix representation is Idé and Kashima (2004) (also window-based approach), which aims at monitoring multi-tier Web-based systems. Conceptually, the method first extracts the principal eigenvector from the adjacency matrix of each graph; this is referred to as activity vector. Then, by applying SVD on the matrix that consists of the past activity vectors in a time window w, the typical activity vector is found, and the similarity between the current and typical activity vectors is computed as the cosine of the angle between them. The next step of the algorithm is to define the parameters of the von Mises-Fisher probability distribution Fisher et al. (1993) of the anomaly metric, and the threshold for characterizing a graph as anomalous or normal; the latter is found using an online algorithm. It is worth mentioning that the activity vector per node enables attribution, i.e. detection of the individual nodes that contributed most to the change in a particular graph instance. Based on Idé and Kashima (2004), the authors in Ishibashi et al. (2010) detect uncommon traffic patterns in communication graphs. The novelty of their approach lies in the way the adjacency matrix of the network is created: instead of encoding the connectivity/communication patterns between the hosts, the cells hold the similarity between them, a property that is computed based on the overlap between their destination hosts.

SVD is not the only tool used by the decomposition-based detection algorithms. On the contrary, the last decade, several improvements on SVD have been proposed, including the CUR matrix approximation (Drineas et al. 2006), the Compact Matrix Decomposition (CMD) (Sun et al. 2008), and Colibri-S (Tong et al. 2008). A pictorial comparison of the four methods is given in Fig. 3. Given a set of 2-d data points, SVD constructs an optimal subspace using all the data points (full circles); CUR samples data points allowing for duplicates and linear redundancy (full circles), and approximates the original points based on them. CMD improves on CUR by sampling without substitution, while Colibri-S also guarantees that no linear redundancy exists in the sampled data points. Table 3 provides a qualitative comparison of the four approaches. Although SVD is optimal in both norm-2 and Frobenius norm, it is inefficient time and

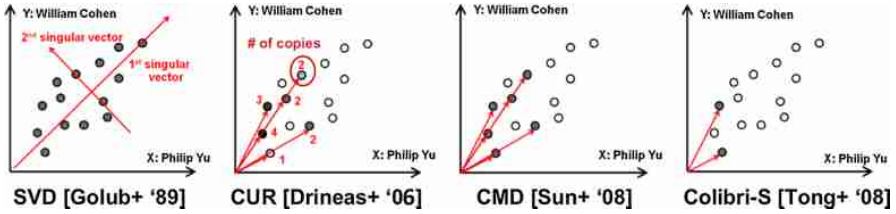


Fig. 3 Illustration of qualitative differences between matrix decompositions used for anomaly detection in dynamic graphs

Table 3 Qualitative comparison of matrix decomposition methods: SVD, CUR, CMD, Colibri-S

	SVD	CUR/CMD	Colibri-S
Quality	✓	✓	✓
Efficiency	✗	✓	✓
Interpretation	✗	✓	✓
Dynamic graphs	✗	✗	✓

space-wise. Moreover, the singular vectors do not have an intuitive interpretation since they describe the data in a rotated space, and the SVD of a matrix cannot be readily updated for dynamic or streaming graphs. CUR and CMD are much more efficient than SVD, and highly interpretable. Finally, Colibri-S is even more efficient in time and space, inherits the previous methods' interpretability, and additionally provides for efficient updates for dynamic graphs.

CMD (Sun et al. 2008) has been applied for anomaly detection in dynamic graphs: the low-rank approximations of the sparse input graphs are used as their summaries. The reconstruction error of each graph from its summary is tracked over time, and, if it changes significantly at some time tick, the corresponding graph is deemed as anomalous.

Now we move on to the second category of decomposition-based event detection methods, which use tensors instead of matrices for the representation of the graphs. Streaming Tensor Analysis (STA) (Sun et al. 2006) is applied for anomaly detection to a computer network described by a source-destination-port graph. The authors introduce the tensor data structure, instead of a simple matrix, because they describe the networks with more entities than just source and destination. Similarly to (Sun et al. 2008), the main idea behind the proposed algorithm is to decompose the stream of tensors into projection matrices (one for each mode of the tensor), and incrementally update the latter matrices over time. If the incremental update leads at some point to high reconstruction error, then the tensor of that time stamp is considered anomalous.

More recently, three more tensor-based approaches were proposed by Koutra et al. (2012), Papalexakis et al. (2012), and Araujo et al. (2014). The first work simply uses the PARAFAC tensor decomposition; the second develops a fast, sampling-based, parallelizable decomposition algorithm for sparse tensors; the third, COM2, relies on tensor decomposition (PARAFAC) to obtain scores for time-evolving communities, and then applies MDL to find the "important" communities, and control their expansion (community size). In all three papers, for temporal anomaly detection, the first two dimensions of the tensors hold the information of the adjacency matrix, additional dimensions are used for attributes or extra entities, and the last dimension corresponds

to the time. The detection of outlier groups of nodes at specific time stamps consists of observing different than 'usual' behavioral patterns in the factors of the decomposition (e.g., sudden increase in the interactions between nodes, bursty or bot-like behavior).

3.2.3 Community/cluster based events

Main idea The main idea of the community or clustering-based approaches is, instead of monitoring the changes in the whole network, to monitor graph communities or clusters over time and report an event when there is structural or contextual change in any of them.

Approaches Being a building block for many applications, clustering, and the related, but not identical, problem of community detection, have been studied thoroughly in the data mining and theory communities: METIS (Karypis and Kumar 1995), one of the first partitioning algorithms that were developed, followed by its parallel implementation ParMETIS (Karypis and Kumar 1996); frequent subgraph mining (Kuramochi and Karypis 2001); spectral clustering (Ng et al. 2001; Shi and Malik 1997); evolutionary clustering (Chakrabarti et al. 2006); the Newman's algorithms for community detection in complex systems (Newman 2004, 2006; Newman and Girvan 2004); co-clustering for concurrent clustering of the rows and the columns of the adjacency matrix of a graph (Chakrabarti 2004; Dhillon et al. 2003), and its distributed variants (Papadimitriou and Sun 2008); dynamic community detection algorithms Tantipathananandh and Berger-Wolf (2009); Tantipathananandh et al. (2007), Tantipathananandh and Berger-Wolf (2011), and empirical comparison of methods for network community detection (Leskovec et al. 2010).

GraphScope (Sun et al. 2007a) is an MDL-based, parameter-free algorithm for discovering node partitions in streaming, directed, bipartite graphs, and monitoring their evolution over time in order to detect events or changes. The partitions consist of "similar" nodes in the sense that splitting a partition leads to higher encoding cost of the adjacency matrix. The algorithm iteratively searches for the best source and destination partitions in each graph snapshot, until further partitioning does not lead to additional decrease of the encoding cost. Then, "similar" snapshots are merged into a segment and compressed together; on the other hand, "dissimilar" consecutive snapshots lead to the creation of a new segment, and declaration of a change-point. A closely related tensor and MDL-based approach is Com2 (Araujo et al. 2014), which tracks "important" communities over time, as described in Sect. 3.2.2. Another approach that also uses node partitioning in order to identify structural anomalies in streaming graphs is GOutlier (Aggarwal et al. 2011), where the focus is on undirected, unipartite graphs. A reservoir sampling method is applied to create several node partitions and develop a structural edge generation model per partition, which describes the likelihood fit of an edge. Each edge in the incoming graph is characterized by its composite likelihood fit, which is defined as its median likelihood fit across all node partitions. Then, the graph's outlier score is represented by the geometric mean of all the composite edge likelihood fits, and the graph is reported as anomalous if its score is t standard deviations below the average outlier score of the graphs seen so far.

A slightly different approach than the ones described above is the Bayesian anomaly detection method presented in Heard et al. (2010). The authors focus on detecting

anomalous regions in social networks using a two-stage Bayesian approach. At the first step of the method, the anomalousness of each edge is computed by modeling the interactions between each pair of nodes as a counting process. Also, at every graph instance, a p value—based on the Bayesian learning of the count distributions—is calculated for every existent edge and used in order to decide whether it is anomalous or not. The algorithm treats the graph sequence as a stream; it detects changes in the new graphs based on the history (sequential analysis), but also updates the history in light of the new instance (retrospective analysis). This step bears similarities with the methodology followed in Aggarwal et al. (2011). However, the second and last step of the approach in Heard et al. (2010) is different; it essentially applies clustering techniques on the small subgraph consisting of the anomalous nodes and edges of the first step, so that locally anomaly regions are discovered.

A probabilistic modeling approach to change-point detection proposed in Peel and Clauset (2014) uses the generalized hierarchical random graphs (GHRG) to model the community structure of real-world networks. The GHRC model decomposes the nodes of the graph into a collection of nested groups, the relationships of which are represented by a dendogram. This representation captures the community structure at all scales. The change-points are identified by significant changes in the parameters of the fitted model through a generalized likelihood ratio test.

Finally, Gupta et al. (2012) introduce the novel problem of detecting nodes which, over time, behave differently from the rest community members; those nodes are called *evolutionary community outliers*. The approach, ECOUTLIER, consists of two parts: matching the time-evolving communities (which are detected in each graph instance by applying state-of-the-art techniques), and detecting the evolutionary community outliers. To solve the problem, an optimization framework that applies a coordinated descent algorithm is used to match the communities over time by appropriately weighting the contribution of the outlier nodes. It operates on pairs of consecutive timestamps of graphs, and returns a ranked list of community outliers.

3.2.4 Window based events

Main idea The last category of time-evolving graph anomaly detection algorithms encompasses methods that are bound to a time window in order to spot anomalous patterns and behaviors in the input graph sequence. Essentially, a number of previous instances are used to model the "normal" behavior, and the incoming graph is compared against those in order to characterize it as normal or anomalous.

Approaches In Priebe et al. (2005), the authors apply scan statistics (as well known as "moving window analysis") to detect graph snapshots that have unusually high connectivity compared to the past. In general, scan statistics are used for detecting clusters of events in time and space (Glaz et al. 2001; Kulldorff 1997; Naus 1982). Essentially, a local statistic is computed for each time window, and the maximum statistic within each window is called scan statistic; if the scan statistic exceeds a threshold, the corresponding time frame is deemed outlier. In this work, the locality statistic used on the disjoint, weekly snapshots of the ENRON who-emails-whom graph is the number of edges in the k-step neighborhood of each node, where $k = 0, 1, 2$. This work is followed by a similar, scan-statistics-based approach in Neil

(2011), where model-based locality statistics are computed in paths and stars, instead of k-step neighborhoods. The method aims at spotting anomalies in computer networks, and the considered shapes are motivated by hacker attacks seen in real networks.

More recently, Mongiovi et al. (2013) tackled the problem of detecting contiguous regions in graphs that are anomalous over time by relating it to the NP-hard problem of finding the Heaviest Dynamic Subgraph (HDS). For each weighted graph in the input sequence, the anomalousness of each edge is computed as its statistical p value using the empirical distribution of the edge weights; lower p value corresponds to higher anomalousness. The proposed iterative algorithm, which solves approximately the HDS problem, alternates between the detection of the subgraph that maximizes the anomaly score for a given interval (spatial), and the detection of time interval that maximizes the score for a given subgraph (temporal). The output of the method is the regions that are more anomalous than a user-defined threshold. An interesting connection is observed between this work and Heard et al. (2010); the approach in the latter paper can be used to compute the anomaly score of each edge, and then the algorithm in Mongiovi et al. (2013) can be applied to detect regions that demonstrate anomalous behaviors.

As mentioned in Sect. 3.2.2, the method described in Idé and Kashima (2004) can also be considered window-based, as the current activity of each node is compared against its activity in the past w time ticks. Similarly, Rossi et al. (2013) belongs to this category as well, since it models the role transitions of the nodes by taking into account the transitions from a number of previous time steps. In addition, the probabilistic graph model fitting approach by Peel and Clauset (2014) of Sect. 3.2.3 is also a window-based one, where the generalized likelihood ratio test is applied over a sliding window of fixed length w to detect if any changes have occurred with respect to the fitted model.

3.3 Discussion

In the previous sections, we review the works in the literature that deal with the problem of graph anomaly detection over time. No matter which type of events are detected, the notion of graph or subgraph/community/cluster similarity usually comes into play at some step of the algorithms. Although the material that follows is not specifically designed for graph anomaly detection, it is closely related to it, as it gives alternative ways of computing the similarity between graphs, or, equivalently, their adjacency matrices.

– **Edit distance/graph isomorphism** One approach to graph comparison when the correspondence between the nodes in *not* known is graph isomorphism. The underlying idea is that two graphs are similar if they are isomorphic (Pelillo 1999), or one is isomorphic to a subgraph of the other (Chartrand et al. 1998; Ullmann 1976), or they have isomorphic subgraphs. The drawback of this approach is that the exact versions of the algorithms are exponential and, thus, not readily applicable to the continuously increasing in size and volume graphs. The graph edit distance (Bunke 1999), which has been mentioned in Sect. 3.2.1, is a generalization of the graph isomorphism problem.

- **Iterative methods** The assumption behind the iterative methods is that "two nodes are similar if their neighborhoods are also similar". In each iteration, the nodes exchange similarity scores and this process ends when convergence is achieved. Several successful algorithms belong to this category: the similarity flooding algorithm (Melnik et al. 2002) applies in database schema matching; SimRank (Jeh and Widom 2002) measures the self-similarity of a graph, ie. it assesses the similarities between all pairs of nodes in one graph; the algorithm proposed by Zager and Verghese (2008) introduces the idea of coupling the similarity scores of nodes and edges in order to compute the similarity between two graphs when the node correspondence is unknown. Bayati et al. (2013) develop two approximate sparse graph matching algorithms using message passing algorithms, and specifically Belief Propagation. Finally, Koutra et al. (2013a) design an alternating projected gradient descent algorithm for efficiently aligning big *bipartite* graphs by exploiting the structural properties of the input graphs.
- **Feature extraction** A number of graph similarity functions, which have been used for graph clustering, classification and applications other than change-point detection, have been proposed in the literature. The research directions in this category include: algebraic connectivity (Fiedler 1973; Wilson and Zhu 2008), a spectral method that has been studied thoroughly; an SVM-based approach on global feature vectors (Li et al. 2011a); social networks similarity (Macindoe and Richards 2010) which is based on graph features that are of value from the social viewpoint; computing edge curvatures under heat kernel embedding (Elghawalby and Hancock 2008); comparison of the number of spanning trees (Kelmans 1976); fast random walk graph kernel for unlabeled (Kang et al. 2012) or labeled graphs (Kashima et al. 2003); graph kernels (Vishwanathan et al. 2010), which are used for computing the similarity between graphs (not nodes). We should note that graph kernels cannot do attribution—i.e. detect the nodes that contribute most to a change in the graph sequence.

As in Sect. 2, we close this section by comparing the dynamic-graph anomaly detection algorithms qualitatively, as well as quantitatively in Table 4.

Choosing one of the algorithms presented in the previous sections for an anomaly detection application is not an easy task nor is there a unique appropriate algorithm; among the things that one should consider when choosing an algorithm are: the type of application (e.g., traffic, communication, computer network), the type of data at hand (e.g., weighted, unweighted, attributed), whether the correspondence between the nodes in consecutive graph snapshots is known or not, the time and parameter requirements, as well as the target of the application (detection of anomalous graph instance, subgraph, or node). Table 4 can help refine the algorithms that can be applied in each case. The reader should bear in mind that, in many cases, applying multiple change-point detection techniques is meaningful, as it contributes to the discovery of different types of anomalies.

Concluding remarks: static & dynamic graph anomaly detection

Evaluation To finalize Sects. 2 and 3, we discuss evaluation methodologies of the graph-based anomaly detection approaches that have been employed in the literature

Table 4 Qualitative and quantitative comparison of anomaly detection algorithms for *dynamic* graphs

Algorithm	Unweighted	Weighted	Plain	Attributed	Linear	Parameter-free	Output format	Node corresp.	Attribution	Visualization (plot over time)
MCS (Bunke et al. 2006b; Shoubridge et al. 2002)	✓	✓	✓	✗	✗	✓	[0, 1]	✗	✗	Consec. graph dist. scores
HD (Bunke et al. 2006b; Shoubridge et al. 2002)	✓	✓	✓	✗	✓	✓	[0, 1]	✗	✗	Consec. graph dist. scores
ECGM (Bunke et al. 2006b; Shoubridge et al. 1999, 2002)	✓	✓	✓	✗	✗	✓	[0, –)	✗	✗	Consec. graph dist. scores
GED (Bunke et al. 2006b; Shoubridge et al. 2002)	✓	✗	✓	✗	✓	✓	$[0, \#nodes + \#edges]$	✗	✓	Spy plot of graph difference
λ-distance (Bunke et al. 2006b; Shoubridge et al. 2002)	✓	✓	✓	✗	✗	✓+	$[0, \infty)$	✓	✗	Consec. graph dist. scores
GED_w (Kapsabelis et al. 2007)	✓	✓	✓	✗	✓	✓	$[0, \infty)$	✓	✗	Consec. ged scores
Diameter distance (Gaston et al. 2006)	✓	✓	✓	✗	✗	✓	$[0, \infty)$	✓	✗	Consec. graph diameter distance
MDS (Bunke et al. 2006a)	✓	✗	✓	✗	✗	✗	Pairwise dist.	✓	✗	MDS + consec. ged scores
VEO (Papadimitriou et al. 2008)	✓	✓	✓	✗	✓	✓+	[0, 1]	✓	✗	Consec. graph sim. scores
Vertex Ranking (Papadimitriou et al. 2008)	✓	✓	✓	✗	✓	✓+	[0, 1]	✓	✗	Consec. graph sim. scores
Vertex/edge vector sim. (Papadimitriou et al. 2008)	✓	✓	✓	✗	✓	✓+	[0, 1]	✓	✗	Consec. graph sim. scores

Table 4 continued

Algorithm	Unweighted	Weighted	Plain	Attributed	Linear	Parameter-free	Output format	Node corresp.	Attribution	Visualization (plot over time)
Sequence Sim. (Papadimitriou et al. 2008)	✓	✓	✓	✗	✓	✓+	[0, 1]	✓	✗	Consec. graph sim. scores
Signature Sim. (Papadimitriou et al. 2008)	✓	✓	✓	✗	✓	✓+	[0, 1]	✓	✗	Consec. graph sim. scores
Akoglu and Faloutsos (2010)	✗	✓	✓	✗	✗	✓	Z-scores	✓	✓	node Z-scores
NetSimile (Berlingerio et al. 2012)	✓	✗	✓	✗	✓	✓	[0, 1]	✗	✗	Consec. graph sim.scores
DeltaCon (Koutra et al. 2013b)	✓	✓	✓	✗	✓	✓+	[0, 1]	✓	✗	Consec. graph sim. scores
Role- dynamics (Rossi et al. 2012)	✓	✓	✓	✓	✓	✓	Role memberships	✓	✓	Role memberships
DBMM (Rossi et al. 2013)	✓	✓	✓	✗	✓	✓	Role memberships	✓	✓	Role memberships
Eigen- space based (Idé and Kashima 2004)	✓	✓	✓	✗	✗	✗	Dissim. score [0, 1]	✓	✓	Sim. scores & activ. vector change
STA (Sun et al. 2006)	✓	✓	✓	✓	✗	✓+	Reconstruction err.	✓	✓	Reconstruction error
CMD (Sun et al. 2008)	✓	✓	✓	✗	✗	✓+	Reconstruction err. (SSE)	✓	✓	Reconstruction error
ParCube (Papalexakis et al. 2012)	✓	✓	✓	✓	✗	✓	Factors	✓	✓	Factors over time
GraphScope (Sun et al. 2007a)	✓	✗	✓	✗	✗	✓	Reordered mat. spy plot	✓	✓	Encoding cost over time

Table 4 continued

Algorithm	Unweighted	Weighted	Plain	Attributed	Linear	Parameter-free	Output format	Node corresp.	Attribution	Visualization (plot over time)
COM2 (Araujo et al. 2014)	✓	✓	✓	✗	✓	✓	Tensor decomp.	✓	✓	Tensor decomp. over time
GOUTLIER (Aggarwal et al. 2011)	✓	✓	✓	✗	✓	✓	Likelihood [0,1]	✓	✓	Likelihood over time
BAYESIAN APPROACH (Heard et al. 2010)	✓	✓	✓	✓	✗	✗	p values	✓	✓	Predictive p value
ECOUTLIER (Gupta et al. 2012)	✓	✓	✓	✗	✓	✗	Community memberships	✓	✓	Community memberships
SCAN STATISTICS (Priebe et al. 2005)	✓	✗	✓	✗	✓	✓	Scan statistics	✓	✓	Scan stat. & vertex scores
SCAN STATISTICS (Neil 2011)	✓	✓	✓	✗	✓	✓	Scores of regions	✓	✓	Scan stat.
NETSPOT (Mongiovi et al. 2013)	✗	✓	✓	✗	✓	✗	Scores of regions	✓	✓	Scores of regions

The first four columns refer to the type of input graphs (with or w/o weights on the edges, with or w/o attributes (or labels) for the nodes); "Linear" holds true for those methods that have time complexity linear in the size of the input graphs (and false otherwise); "Parameter-free" methods correspond to those that do not expect any user-specified input parameters (+: parameter can be set, but is not required); "Output format" corresponds to the output type/format of the method (e.g., anomaly scores and their ranges); "Node corresp." is true if the algorithm assumes that the correspondence between the nodes of the graph sequence is known; "Attribution" holds true if the algorithm spots nodes/edges/regions of graph that are anomalous (and false if it detects anomalous graph instances); and "Visualization" refers to the graphical means used—if any—to present the anomalous instances to the user (e.g., distribution plots, graph with the anomalous nodes/edges annotated)

thus far. Recall that ground truth data is often inexistent in the anomaly detection scenarios, thus, various methods in the literature have been evaluated in several different ways which we describe next.

- *Internal evaluation* This kind of evaluation mechanism uses the anomalousness scores of objects assigned by a given method to statistically quantify their extremity, e.g., by computing their p values under the empirical distribution of scores of all objects. This evaluation is internal, since the scores are dependent on the specific method and can be as diverse as likelihoods, compression costs, distances, etc., and may not necessarily directly tied with the external purpose of the anomaly detection.
- *Qualitative evaluation* Unlike the previous approach which is quantitative, qualitative evaluation employs informal procedures. One approach is to try to explain away the detected anomalies through a story related to a real-world scenario. Another approach is to incorporate domain knowledge to exploit and make sense of the detected anomalies. This latter methodology is often used in medicine, where the anomalies may help in knowledge discovery and help with diagnosis.
- *Synthetic graph generation* A mechanism that is well resorted to is synthetic data generation. In graph-based anomaly detection, several methods create realistic graphs using graph generators such as preferential attachment Barabási and Albert (1999), Forest Fire Leskovec et al. (2005), random-typing graphs Akoglu and Faloutsos (2009) (power-law graphs), and the Waxman (Internet AS topology graphs) Medina et al. (2001) models. Often, the kind of anomalies are directly injected to the synthetic graphs. Sometimes the graph structure can also be modified by randomly rewiring edges or swapping node attributes. The methods are then evaluated by their precision and recall in recovering the created anomalies. Synthetic graphs also help with evaluating the behavior of the proposed methods, such as their accuracy and scalability with changing graph characteristics, such as size and degree-distribution.
- *Anomaly injection* The injection of synthetic anomalies has been discussed above. This is similar, only this time the anomalies are injected into the real-world graphs. One challenge in this version of anomaly injection is that the evaluation based on precision and recall becomes tricky, as it would be severe to call the anomalies detected other than the injected ones as false positives, given that the original graph may also contain same type of anomalies.
- *Validation by external source* Another evaluation approach relies on multiple information sources that are consistent with each other in identifying the anomalies. In such a setting, one or more sources are used for the actual anomaly detection task. The detected anomalies are then tried to be validated or justified based on the rest of the unused information sources. For example, one may only use the graph structure to detect opinion spam and find out fake reviewers, and then use their temporal behavioral information, such as number of reviews written in a day, to see if the detected reviewers also exhibit suspicious behavior.

Summary Finally to summarize, we create Table 5 including the various methods discussed this far under different categorization schemes such as static and dynamic, and plain and attributed graphs. Interestingly, we were unable to find any examples of methods that aims to find anomalies in dynamically changing attributed graphs. We

Table 5 Categorization of graph-based techniques discussed in Sects. 2 and 3

	Plain	Attributed
Static	[Section 2.1]	[Section 2.2]
	AUTOPART (Chakrabarti 2004)	SUBDUE (Noble and Cook 2003)
	Sun et al. (2005)	Liu et al. (2005)
	SCAN (Xu et al. 2007)	SUBDUE (Eberle and Holder 2007)
	ODDBALL (Akoglu et al. 2010)	CODA (Gao et al. 2010a)
	GSKELETONCLU (Sun et al. 2010)	(Davis et al. 2011)
	NRMF (Tong and Lin 2011)	GOUTRANK (Müller et al. 2013)
	Ding et al. (2012)	FOCUSCO (Perozzi et al. 2014)
	NETRAY (Kang et al. 2014)	
	Open	
Dynamic	[Section 3.2]	[Section 3.2]
	Shoubridge et al. (1999)	STA (Sun et al. 2006)
	Shoubridge et al. (2002)	BAYESIAN APP. (Heard et al. 2010)
	Dickinson et al. (2002)	ROLE- DYNAMICS (Rossi et al. 2012)
	EIGENSPACE- BASED (Idé and Kashima 2004)	TENSORSPLAT (Koutra et al. 2012)*
	SCAN STATISTICS(Pincombe 2005)	PARCUBE (Papalexakis et al. 2012)*
	SCAN STAT. (Priebe et al. 2005)	COM2 (Araujo et al. 2014)*
	Bunke et al. (2006b)	
	MDS Bunke et al. (2006a)	
	Gaston et al. (2006)	
	GED_W (Kapsabelis et al. 2007)	
	GRAPHSCOPE (Sun et al. 2007a)	**Many Open Challenges**
	Papadimitriou et al. (2008)	
	Akoglu and Faloutsos (2010)	
	CMD (Sun et al. 2008)	
	Ishibashi et al. (2010)	
	GOUTLIER (Aggarwal et al. 2011)	
	ECOUTLIER (Gupta et al. 2012)	
	NETSIMILE (Berlingerio et al. 2012)	
	DELTACON (Koutra et al. 2013b)	
	NETSPOT (Mongiovi et al. 2013)	
	DBMM (Rossi et al. 2013)	

* Not applied in attributed graphs, but it is possible to admit labels/attributes

foresee that this would require novel definitions of anomalies in such a setting as well as necessitate the identification of real world scenarios in which such definitions come alive. Moreover, we notice that methods on static graphs strictly either deal with plain or attributed versions of graphs. It would be interesting to build methods that can work with both; which apply to plain graphs but also can use side (attribute) information if available. We classify those areas of research as open problems in our categorization, and point them out as possible avenues for future exploration.

4 Graph-based anomaly description: interpretation and sense-making

Like many other real applications, the ground-truth for graph anomaly either does not exist or is very difficult or costly to obtain. Consequently, the end analysts often have to spend much post-processing time to validate the detection results. For example, according to a recent DARPA BAA,[5] it is estimated that an intelligence agent can only perform 60 initial reviews on average for the so-called insider threat detection. This, coped with the facts that many graph anomaly detection algorithms (including insider threat detection) still have a high false positive rate, makes it extremely challenging and time consuming to identify at least one true positive in such applications. On the other hand, it is usually much more persuasive for an ordinary user if the detection algorithm can tell not only which instance is abnormal, but also why it looks so different from the majority, normal examples.

To address these issues, graph anomaly attribution has been attracting more and more research attention in the recent years. In this section, we will review two main types of techniques. The first group aims to make the detection of each individual instance more 'interpretable', which is usually done by encoding the so-called interpretation-friendly properties into the traditional graph anomaly detection algorithms. For this category, we will mainly use matrix-factorization based graph anomaly approach as an example. The second group tries to answer the following question. Given a set of initial suspects (e.g., the top ranked instances from a graph anomaly detection algorithm), how can we find and characterize the internal relationship among them so that we can better understand the root cause of such anomalies? For this category, we will introduce interactive graph querying and sense making.

Definition 5 (*Graph-based anomaly description problem*)

Given a set of anomalies of graph entities (nodes and edges)
Interpret and explain the detection of the individual anomalies,
Find and characterize the associations among the anomalies.

4.1 Interpretation-friendly graph anomaly detection

Main idea Here, we consider the first problem of how to make the detection of each *individual* instance (e.g., nodes, edges) more interpretable. The main idea is to *encode* the so-called interpretation friendly property into the traditional graph anomaly detection algorithms. We will present the matrix based graph anomaly detection methods.
Approaches Suppose we have a bipartite graph (e.g., author-conference graph), and we can represent it by its adjacency matrix $\mathbf{A}$ with the rows being authors, columns being conferences and non-zero elements meaning the corresponding authors who have published papers in the corresponding conferences. In the matrix-based graph anomaly approaches, we start with factorizing the adjacency matrix as $\mathbf{A} = \mathbf{X}\mathbf{Y}' + \mathbf{R}$. In this factorization, the two low-rank matrices $\mathbf{X}$ and $\mathbf{Y}$ usually capture the 'normality'

[5] https://www.fbo.gov/utils/view?id=2f6289e99a0c04942bbd89ccf242fb4c.

of the graphs (e.g., clusters, communities, etc); while the residual **R** measures the deviation from such 'normality', and thus is often a good indicator of 'anomaly'.

The different matrix-based graph anomaly detection approaches differ in the way they get these matrices. SVD/PCA is one of the most popular choices, where the columns of **X** and **Y** are the singular vectors (up to a scalar by the singular values) of the original matrix **A**. While it is mathematically optimal in the sense that it minimizes the reconstruction error in both the L2 and the Frobenius norm, it is not necessarily good for interpretation. We give two examples below to make such matrices less abstract and therefore more interpretable/consumable to the end analysts.

First, note that the singular vectors are usually the linear combination of *all* the columns/rows of the original adjacency matrix, which are not always easy for interpretation. More recently, the so-called *example-based low-rank approximations* have started to appear, such as CX/CUR (Drineas et al. 2006), CMD (Sun et al. 2007b) and Colibri (Tong et al. 2008). All of these methods use the actual columns and rows of the adjacency matrix **A** to form **X** and **Y**. The benefit is that they provide an intuitive as well as sparse representation, since **X** and **Y** are directly sampled from the original adjacency matrix. The cost of such kind of decomposition is that the approximation is often sub-optimal compared to SVD. We refer the readers to Sect. 3.2.2 for the detailed description of these methods. Also see Fig. 3 for a pictorial comparison.

Another interpretation-friendly property that has been recognized widely in the recent years is *non-negativity* since negative values are usually hard to interpret. NMF methods (Lee and Seung 2000) which restrict the entries in **X** and **Y** to be non-negative have attracted a lot of research attention. By imposing such non-negativity constrains on the *factorized matrices*, NMF provides a more interpretable way for data mining tasks, e.g., clustering, community detection, etc. Note that although the NMF has been studied largely in the context of such applications (e.g., clustering), we would expect that it is also beneficial for graph anomaly detection, since it helps improve the interpretation of graph normality.

In the context of graph anomalies, it is often the case that anomalies on graphs correspond to some actual behaviors/activities of certain nodes. For instance, we might flag an IP source as a suspicious port-scanner if it *sends packages* to a lot of destinations in an IP traffic network (Sun et al. 2007b); an IP address might be under the DDoS (distributed denial-of-service) attack if it *receives packages* from many different sources (Sun et al. 2007b); a person is flagged as 'extremely multi-disciplinary' if s/he *publishes papers* in many remotely related fields in an author-conference network (Akoglu et al. 2010); in certain collusion-type of fraud in financial transaction network, a group of users always *give good ratings* to another group of users in order to artificially boost the reputation of the target group (Chau et al. 2006), etc. If we map such behaviors/activities (e.g., 'sends/receives packages', 'publishes papers', 'gives good ratings', etc) to the language of matrix factorization, it also suggests that the corresponding entries in *the residual matrix* **R** should be non-negative. In order to capture such activities, non-negative residual matrix factorization (NrMF) has been proposed in Tong and Lin (2011, 2012), which explicitly requires those elements in the residual matrix **R** to be non-negative if they correspond to actual links in the original graphs. This in turn highly adds to the ease of interpretation. Figure 4 presents a visual com-

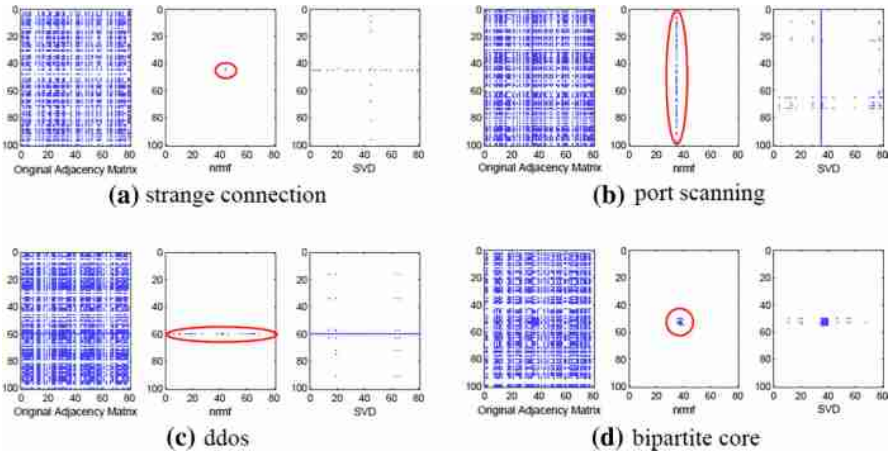


Fig. 4 Visual comparison between NrMF and SVD. For each type of graph anomalies, the *first column* is the adjacency matrix of the original graph, the *second* and *third columns* are the residual matrices by NrMF and that by SVD, respectively

parison between NrMF and the standard SVD on four typical graph anomalies (Tong and Lin 2011, 2012).

For feature-based graph anomaly detection, visualization in the (sub)space of the feature is also a very natural and powerful tool to improve the interpretation of graph anomalies (Akoglu et al. 2010; Kang et al. 2014). By making the abnormal graph nodes 'standing out' in these low-dimensional plots, the end-user could have an intuitive understanding on which graph feature(s) makes them different from normal.

4.2 Finding the root-cause of anomalies: interactive graph querying

Main idea Next, we consider the second problem of finding and characterizing the internal *relationships* among the anomalies so that we can better understand the root cause of such anomalies. We will introduce interactive graph querying. The main idea is to find a concise *context* where detected graph anomalies are linked to each other (see Fig. 5 for an illustration). Note that while extremely useful in graph anomaly detection, these techniques themselves have a much broad applicability.

Approaches connection subgraphs is one of the earliest works along this line, which is defined as a small subgraph of a large graph that best captures the relationship from a source node to a target node (Faloutsos et al. 2004). The original method in Faloutsos et al. (2004) is based on the so-called delivered current. By interpreting the graph as an electric network, applying $+1$ voltage to one query node and setting the other query node 0 voltage, it aims to choose the subgraph which delivers maximum current between the query nodes. Koren et al. (2006) propose using cycle-free effective conductance based method for this problem by only considering the top-k simple (i.e. cycle-free) paths from the source to the target. Ramakrishnan et al. (2005) further apply the delivered current based method to multi-relational graphs.

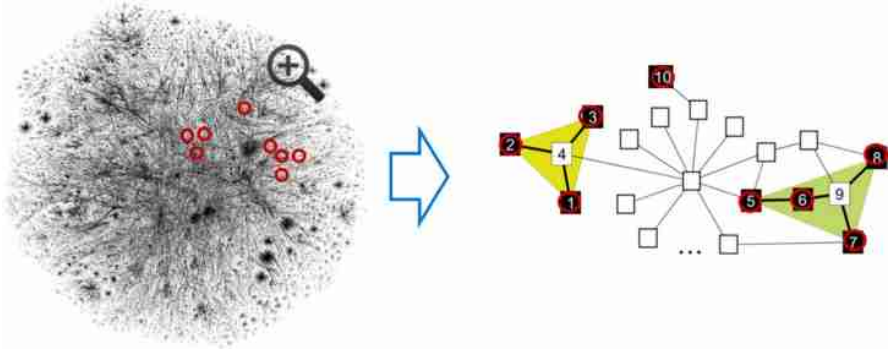


Fig. 5 The main idea of interactive graph querying. The *left*, given a set of detected abnormal nodes (*red circles*) from a given large graph (*black*). *Right*, the desired output which shows a concise summarization of these abnormal nodes (e.g., how they are further grouped into a few clusters, how the abnormal nodes within each group are linked to each other, etc) (Color figure online)

Note that in all these works, they deal with pairwise source-target queries. *Center-Piece Subgraphs* (CePS) (Tong and Faloutsos 2006) generalizes this by considering the following settings: Given Q query nodes in a social network (e.g., a set of top-ranked authors in a co-authorship network), find the node(s) and the resulting subgraph, that have strong connections to all or most of the Q query nodes. This provides an intuitive tool to identify the potential root cause of graph anomaly detection results. For example, in the context of law enforcement, given a set of initial suspects, we may want to find other persons who have strong connections to all or most of the existing suspects, who might be the master criminal mind. The discovered path(s) in the resulting subgraph also provides an intuitive explanation on how/why the master mind connects to the individual suspects.

All the above works we have introduced in this subsection so far, assume, explicitly or implicitly, some specific connectivity structure among the query nodes. CePS provides certain degree of flexibility by allowing the so-called k-SoftAnd, where we only require the center-piece nodes to have strong connections to k-out-of-Q query nodes. But the end users still need to specify such a parameter k which is not necessarily an easy task for applications like graph anomaly detection. To address this issue, Dot2Dot (Akoglu et al. 2013b; Chau et al. 2012) proposes to find 'right connections', that is, given a set of query nodes (e.g., the top-k ranked nodes in graph anomaly detection), it groups them into one or more groups and within each group, it finds the simple connections to characterize the relationship within that group. This problem itself is NP-Hard, and the authors propose efficient parameter-free algorithms to find approximate solutions. In the example of top-k ranking list from some graph anomaly detection algorithm, Dot2Dot not only automatically groups the detected anomalies one or more groups and each group could correspond to a specific type of anomalies; but also provides some explanations why they belong to the same group and what is the possible root cause for that group of anomalies. Moreover, in the case there is a false positive node in the top-k ranking list (e.g., a node which is far away from all

the other, true positive, nodes in the top-k ranking list) by automatically treating it as a group by itself.

5 Graph-based anomaly detection in real-world applications

Next we shift our focus to real-world fraud and spam scenarios. Several different techniques have been developed for fraud and spam detection in many real world scenarios including frequent pattern mining (Jindal et al. 2010), behavioral monitoring (Fawcett and Provost 1999), supervised learning (Phua et al. 2004), and so on. In this section, we will motivate and focus on *graph-based* detection techniques for real-world applications and particularly highlight their advantages. However, the purpose of our survey is not to suggest the superiority of graph-based techniques over other detection methodologies. Rather, we introduce the available tools focusing on those that exploit graphs. It would be up to the application developers to carefully choose what tools suit their needs best as different approaches may achieve different performances depending on the application. For a general survey on various fraud detection techniques, we refer the reader to Bolton and Hand (2002) and Phua et al. (2010).

We highlight two main advantages of graph-based fraud detection techniques as we discussed in Sect. 1; relational nature of the problem domain and adversarial robustness. The former intuitively refers to the fact that fraud often occurs in two different ways, (1) by word of mouth where the acquaintances of a fraudster can be considered as more likely to also commit fraud, and (2) by collaboration where closely related parties come together to commit fraud. In both scenarios, the relational "closeness" can be exploited with graph-based detection techniques. The latter, robustness to adversaries, relates to the difficulty imposed on the attacker to break the detection method. One can think that the graph-based representation of the domain in which fraud is committed is fully available only to the system administrators. In other words, it is often the case that the fraudsters only have a limited view of the operational graph in which they act. Therefore, it becomes harder for them to carefully cover their traces so as to "fit in" the global behavioral patterns of this graph.

In this part of the survey, we cover a wide range of applications including telecom fraud (Cortes et al. 2002), auction fraud (Pandit et al. 2007), accounting fraud (McGlohon et al. 2009), securities fraud (Neville et al. 2005), opinion spam (Dai et al. 2012; Wang et al. 2011a), trading fraud (Li et al. 2010), network intrusion (Ding et al. 2012; Idé and Kashima 2004), and Web spam and malware detection (Becchetti et al. 2006; Benczúr et al. 2005; Castillo et al. 2007; Gyöngyi et al. 2004; Kang et al. 2011a; Krishnan and Raj 2006; Wu et al. 2006).

5.1 Anomalies in telecommunication networks

While there are many types of telecommunications fraud, one of the most prevalent is known as the subscription fraud. In this type of fraud, the fraudster often acquires an account using false identity with the intention of using the service for free and not making any payments.

One of the earliest studies that proves the graph-based methods effective in telecommunications fraud detection is done by Cortes et al. (2002), who mainly use linkage analysis together with temporal and calling volume information. In particular, they build and maintain subgraphs around each phone account which they name as the "communities of interest" (COI) of the account. The COI mainly contains the other phone accounts that are most related to the given account in terms of dynamically weighted measures that consider the call quantity and durations between these parties over time. Using these informative subgraphs updated daily, two discriminative properties are observed. Firstly, fraudulent phone accounts are found to be linked; fraudsters either directly call each other or they call the same phone numbers which puts them in close proximity in the COIs. A second observation shows that it is possible to spot new fraudulent accounts by the similarity of their COIs to previously flagged fraudulent COIs—this is due to detected and disconnected fraudsters by the phone operator creating new accounts and exhibiting similar calling habits, which are effectively captured by their COIs.

These graph-based linking methods provide powerful machinery on top of previously used signature-based methods (Cortes and Pregibon 2001; Cortes et al. 2000), where few simple measures such as extensive late night activity and long call durations have been taken as indicators for fraudulent behavior.

5.2 Anomalies in auction networks

Auction sites such as eBay, uBid, bidz, and Yahoo! Shopping are attractive targets for auction fraud, which constituted about 25 % of the complaints to Federal Internet Crime Complaint Center (IC3) in the U.S. in 2008 (Federal Bureau of Investigation (FBI) 2009). The majority of online auction fraud occurs as non-delivery fraud (∼33 %), where the seller fails to deliver/ship the purchased goods to the buyer.

Chau et al. (2006) developed one of the very first graph-based methods to spot fraudsters committing auction fraud and showed the effectiveness of their method on a large crawl of eBay data. The motivation to use graph-based methods in that domain is the insufficient solutions based on the individual's features, such as age, geo-location, login times, session history, etc. which are "easy" to fake. As we discussed earlier in this section, as well as in Sect. 1.1, the intuition is that as the fraudsters have only a local view of the auction graph, it is "harder" for them to alter their behavior and still be able to "fit in" this graph at large without knowing all the patterns of interactions.

The analysis of the fraudsters' behavior reveals that in order to game the feedback and reputation system, fraudster create additional accounts or "roles" called accomplices. Thus, fraudsters exhibit two roles:

– *accomplice* trades with honest users, looks legitimate

– *fraudster* trades with accomplices to "sell" (cheap) items and receive good feedback to boost reputation, and occasionally commits fraud with honest users when reputation is high enough to convince them

Accomplices and fraudster do not necessarily interact among each other. Moreover, honest users trade among themselves as well as with accomplices that also look like honest users. As such, there is quite a bit heterophily among the labels of neighboring nodes: with fraudsters mostly linked to accomplices and occasionally to honest users, accomplices linked to both fraudsters and honest users serving as middle-men, and honest users mostly linked to other honest users and accomplices.

Using the insights of these interaction characteristics, Pandit et al. (2007) developed a relational classification model based on RMNs that can capture these complex correlations (in particular heterophily) among the node labels (honest, accomplice, fraudster), and used LBP for inference.

5.3 Anomalies in accounting networks

Accounting fraud involves the task of spotting high-risk accounts with suspicious transactions behavior. Many existing techniques for detection rely on (noisy) domain knowledge and rule-based signals, for example, based on large number of returns, many late postings, round-dollar entries, etc.

Based on the insight that closely related accounts by their transaction relations would be more likely to have the same labels (risky vs. non-risky), McGlohon et al. (2009) use relational classification to detect accounting fraud. Here, unlike the heterophily observation in Chau et al. (2006), the homophily (auto-correlation) of neighboring class labels is assumed. Similarly, a RMN representation is developed and LBP is used for inference.

One of the representational powers of global joint models like RMNs, in addition to their ability to capture complex correlations, is the fact that they can integrate prior knowledge if available. In this particular application, the prior knowledge (probability) of accounts being risky translates to prior belief potentials in the RMN representation. In fact, McGlohon et al. (2009) use the previously used (noisy) domain knowledge based on rule-based flags to estimate the prior beliefs. These beliefs are then propagated in the network where some of them are corroborated and some may be discarded. Their results showed that through this type of graph-based validation, the detection (true positive) rate improved significantly over the rule-based methods for the same (small) false positive rate.

5.4 Anomalies in security networks

Relational learning has also been used in securities fraud detection where the task is to spot securities brokers that are likely to commit fraud and other violations of securities regulations in the future. While previous methods used handcrafted rules based on information intrinsic to the brokers such as the number and type of past violations, Neville et al. (2005) exploited relational information such as social, professional, and

organizational relationships (e.g., past co-worker) among the brokers. In fact, this is one of the applications where the likelihood of committing fraud is highly dependent on social phenomena: communicated and encouraged by word-of-mouth by people who wish to commit fraud that relational methods are excellent at spotting.

In particular, Neville et al. (2005) use a subgraph representation for each of the securities brokers. Each subgraph includes, in addition to the target broker, various types of other objects (e.g., firms, disclosures), as well as links that represent relationships between these objects (e.g., employment links between a broker and a branch, filing links of disclosures on the broker), and attributes on these objects and links. They then learn relational probability trees (Neville et al. 2003) which exploits (aggregated) relational features of those subgraphs to model the distribution of the class labels, showing that the learned models rank brokers in a manner consistent with the subjective ratings of experienced examiners, and better than handcrafted rules.

5.5 Anomalies in opinion networks: deception and fake reviews

Review sites such as Yelp, TripAdvisor, Amazon, etc., are attractive targets for opinion spam. Opinion spam exhibits itself as hype or defame spam, where (often paid) fraud reviewers write fake reviews to untruthfully boost or damage a vendor's reputation, respectively and cause unjust perception of the services by future customers.

This problem has been approached by three different methodologies, based on (1) behavioral analysis (Feng et al. 2012b; Jindal and Liu 2008; Jindal et al. 2010; Xie et al. 2012), (2) language stylometry analysis to spot deception (Feng et al. 2012a; Ott et al. 2011), and (3) relational analysis and network effects to exploit connections among fraudulent reviewers (Akoglu et al. 2013a; Wang et al. 2011a, 2012a). More specifically, with respect to (1) and (2), Jindal and Liu (2008), Jindal et al. (2010) extract behavioral features such as review length, posting times, time order of reviews (whether first posted review or not), etc. in addition to rule-based mining to spot suspicious reviewers. Feng et al. (2012b) study the distributional patterns in rating behaviors, while (Xie et al. 2012) focus on temporal reviewing behaviors to detect fake review(er)s. As for language-based detection, Ott et al. (2011) unearth the excessive usage of superlatives, self-referencing, rate of misspell, and agreement words in fake reviews as important clues.

With respect to graph-based detection (3), Wang et al. (2011a) developed a propagation algorithm to capture the relationships between reviewers, reviews, and stores (or products, services). The method defines a trustiness score for each reviewer, reliability score for each store, and a honesty score for each review. These scores are defined in terms of one another: reviewer trustiness is a (non-linear) function of his/her reviews' honesty scores, store reliability is a function of the trustiness of the reviewers writing reviews for it, and finally review honesty is a function of the reliability of the store it is written for as well as the trustiness of the reviewers who have also written reviews for the same store it was written for. The algorithm randomly initiates these scores, and updates them iteratively until some convergence criterion is reached. This is similar in design to the HITS algorithm by Kleinberg (1998) where the authoritativeness and hubness scores of Web pages, which are defined in terms of linear functions of

each other, are updated iteratively. On the other hand, the algorithm is not guaranteed to converge, and cannot exploit extra knowledge such as textual clues or behavioral information but is complementary to these previous methods.

Most recently, Akoglu et al. (2013a) exploited relational classification for opinion spam detection. In particular, they developed a relational model based on RMNs that can capture the correlations between reviewers and stores, and used LBP for inference. One main difference from earlier network classification based methods is the signed nature of the opinion network, in which the reviewers are connected to stores (or products) with positive (+) or negative (−) links that capture the sentiment of their reviews (e.g., like/dislike). The signed links affect the label correlations: e.g., while a fraudulent reviewer is likely to link to a low-quality store with a − link (unjustly boosting its reputation), it is less likely for him/her to link to a high-quality store with a + link; although this latter case occurs where fraudulent users occasionally write truthful reviews to camouflage their otherwise fraudulent activities, which is accounted for in the RMN model.

5.6 Anomalies in financial trading networks

Li et al. (2010) use graph-based substructures and their efficient detection to spot potential fraudulent cases in trading networks. These cases consist of a group of traders that trade among each other in certain ways so as to manipulate the stock market. More specifically, the group of traders may perform transactions on a specific stock among themselves for some amount of time during which the overall shares of the target stock in their trading accounts increase and they end up producing a large volume of transactions on this stock. After the stock price goes up, these traders start selling the acquired shares to the public producing excessive volume of transactions to traders other than themselves.

These two different behaviors of a group of traders within consecutive time windows are formulated in graph-based terms. In the former, in which excessive buying of the stock occurs, the in-link weights are expected to be quite high (these are called blackhole patterns), while in the latter selling stage the out-link weights highly exceed in-links' (these are called volcano patterns). These two fraudulent trading behaviors are formally defined and formulated in graph-theoretic terms, and efficient algorithms are developed to detect such patterns quickly in very large and dynamically changing financial trading networks.

5.7 Anomalies in the Web network: spam and malware

One suitable way to define Web spam is any attempt to get an unjustifiably favorable relevance or importance score for some Web page, considering the page's true value. One of the main techniques in combating spam and malware on the Web has been to use trust and distrust propagation over the graph structure. These techniques assume that a link between two pages on the Web signifies trust between them; i.e. a link from page i to page j is a conveyance of trust from page i to page j. Moreover, if the target page is known to be a spam page, then they consider the trust judgment of

the source page as invalid, in which case the source page is penalized for trusting an untrustworthy page.

One of the earliest methods in improving the PageRank algorithm to combat Web spam is TrustRank (Gyöngyi et al. 2004), which employs the idea of propagating trust from a set of highly trusted seed sites. Initially human experts select a list of seed sites that are well-known and trustworthy on the Web. Each of these seed sites is assigned an initial trust score. A biased PageRank is then used to propagate these trust scores to the descendants of these sites. The amount of trust decreases with distance from the seed set and the number of outgoing links from a given site.

Anti-TrustRank (Krishnan and Raj 2006) can be thought of as the dual of TrustRank that performs propagation starting from known bad pages and propagate distrust instead of trust. The intuition used in this work is that the pages pointing to spam pages are very likely to be spam pages themselves. Anti-Trust is propagated in the reverse direction along incoming links, starting from a seed set of spam pages.

Wu et al. (2006) also point out several issues regarding TrustRank's assumptions, such as the fact that it looks at outgoing links and divides trust propagated to children by their count, which causes two equally trusted pages (but with different number of children) propagate different trust scores to their children. Moreover the children accumulate trust by simply summing the trust scores from their parents. Instead, they use different splitting and accumulation techniques. In addition, they employ both trust and distrust propagation, and finally assign a weighted score of the two.

One of the main challenges of these methods discussed so far is that they all expect a manually labeled seed set of good or bad pages. Benczúr et al. (2005) propose a novel way to overcome this challenge and fully automate the process. Their idea is to look at the distribution of PageRank scores of neighbors for each node (i.e. Web page in the graph), which is expected to be power-law distributed given the overall PageRank score distribution being power-law and the self-similarity of the Web. For those nodes where the PageRank distribution of their neighbors deviate significantly from power-law, they assign a "penalty". Similar to Anti-TrustRank, a new PageRank biased by the penalty scores gives the Spam-Rank scores.

Link-based spam detection (Becchetti et al. 2006) looks at different graph-based measures which are then used as features to train classification models. The graph features include PageRank, TrustRank scores, degree, assortativity (i.e. degree correlation), fraction of reciprocal edges, average degree of neighbors, etc. These type of link-based features are complementary to other techniques that use content-based features, such as the number of words, number of hyperlinks, text redundancy, etc. (Canali et al. 2011; Ntoulas et al. 2006), and content-free features, such as URL-based-only host and lexical features (Ma et al. 2009).

The work called 'Know your neighbors' (Castillo et al. 2007) makes use of various types of features in tandem to learn classifiers and furthermore, use the graph structure to "smooth" the classification results. Main idea is to extract features that are (a) link-based (edge-reciprocity, assortativity, TrustRank score, ratio of TrustRank to Pagerank, radius, neighborhood growth rate for increasing number of hops, etc.); and (b) content-based (compression ratio, entropy of n-grams, etc.). Using these features they learn classifiers (decision-trees) and then smooth the classifier scores using the graph structure. In particular, they use three different ways to exploit the Web graph:

(1) clustering where all the nodes in the same cluster is relabeled by the majority of the initial labeling, (2) random-walk-with-restart where probabilities are set to normalized spamicity scores from the classifier (similar to Anti-TrustRank), and (3) stacking where a set of extra features for each object are added to the classification over iterations by combining the predictions for the related objects in the graph (this indeed is an ensemble method).

5.8 Anomalies in social networks

Related to the previous section, another group of malware detection methods focuses on social malware in social networks such as Facebook. Such malware is also called socware. Socware consists of any posts appearing in one's news feed in social media platforms such as Facebook and Twitter that (1) lead the user to malicious sites that compromise the user's device, (2) promise false rewards and make the user perform certain tasks (e.g., filling out surveys) potentially for someone else's benefit, (3) make the user boost the reputation of certain pages by clicking or 'liking' them, (4) make the user redistribute (e.g., by sharing/re-posting), and so on.

To combat socware, Rahman et al. (2012) propose a classification framework that exploits "social-context-aware" features, such as message similarity of posts across different users who shared [or made to share as in (4) above] a particular post, the size of the propagation of the post in the network, the total 'like' and comment counts of other network users on the post, etc. in addition to other content-based features. In another study, Gao et al. (2012) perform online spam filtering on social networks using incremental clustering, based on features that also include network-level features such as sender's degree and the interaction history between users. These methods rely on learning classifiers based on collective feature sets (including graph-based features). On the other hand, it would be interesting to see if unsupervised methods that directly focus on graph mining could help in identifying online socware, by studying the propagation-based dissemination of socware in the network.

5.9 Anomalies in computer networks: cyber-attacks and intrusion

Most graph-based network intrusion detection methods focus on the dynamically growing and changing nature of the network graph. In this graph, the nodes represent the agents in the networks, such as ad/file/directory servers and client nodes, and edges represent their communications over the network (note that the edges may be weighted, capturing volume or frequency). The insight behind tracking the dynamic nature of the network graph is the assumption that the communication behavior of a compute node would change when under attack.

There exist two main challenges associated with tracking large communications networks and the necessity to consider their relational characteristics: (1) large number of compute nodes makes it impractical to monitor them individually, moreover the behavior of the nodes may be dependent on each other and thus monitoring them in isolation would bypass their correlations; (2) large number of edges makes it impractical to study the highly dynamic time-series of communications volume in tandem.

For these reasons, Idé and Kashima (2004) monitor what is called the "activity" vector of nodes. The activity score of a node is computed collectively; if a node links to many active nodes, its activity score is high. With this definition, the activity vector essentially becomes the principal eigenvector of the adjacency matrix that depicts the communication graph. They track this vector over time by measuring the change in its direction and magnitude and develop online thresholding techniques to decide when to flag a change as a significant event. These events may correspond to network attacks as well as failures and other network configuration changes.

Sun et al. (2008) exploit matrix decomposition to capture the norm of network activity. They employ a sparse and efficient (both in time and storage) method called CMD to decompose the adjacency matrix of the network graph and use relative sum-square-error of reconstruction as a measure of change to track over time for newcoming snapshots of the network graph. They observe that this new measure of change detects events that total volume monitoring misses.

Another graph-based method Ding et al. (2012) considers analysis of network communities, as we discussed in Sect. 2.1. Simply reput, the idea is to monitor cross-community communication behavior to spot network intrusion. Intuitively, communications that cross community boundaries, considered as anti-social, are suspicious and can be treated as signal of attack. The ROC curves show that methods based on this insight achieve over 90 % accuracy in detection, however with a somewhat high false alarm rate of about 50 % in ground-truth data with malicious attacks.

Finally, while not directly focusing on network intrusion, Iliofotou et al. (2007, 2011) use graph based network traffic representations, called traffic dispersion graphs, to analyze, monitor, visualize, and classify network traffic.

6 Conclusions and open challenges

Summary In this survey, our aim has been to provide a comprehensive overview of graph-based techniques for anomaly, event, and fraud detection, as well as their use for post-analysis and sense-making in explaining the detected abnormalities. Following our taxonomy in Fig. 2, we surveyed quantitative detection and qualitative explanation/attribution techniques as two main parts. The detection methods are further categorized into three groups: (1) anomaly detection in static graphs; (2) event detection in dynamic graphs; and (3) fraud detection in real-world scenarios. The first two groups (anomalies and events) consist of *general* abnormality definitions and their detection techniques proposed mainly by the data mining community. The third group (fraud scenarios) consists of *specialized* techniques for particular fraud types as observed in the real world and mostly involve (machine) learning approaches. Furthermore, the attribution techniques highlight graph-based tools for analysis, visualization, monitoring, exploration, and sense-making of the anomalies.

Conclusions One of the main messages we aimed to convey has been the expressiveness of graphs in capturing real world phenomena, which makes them a very powerful machinery for abnormality detection. In particular, we emphasized that (1) data instances are often inter-dependent and exhibit long-range correlations, (2) the

anomaly detection problem is often relational in nature (e.g., opportunistic or organized fraud), and (3) robust, hard-to-circumvent machinery is essential in the arms race with the attackers in fraud scenarios. As such, graphs prove to be effective in all these aspects.

Our aim, however, is not to claim the superiority of graph-based methods over other detection techniques. On the other hand, our goal is to highlight the advantages of graphs, and provide a comprehensive list of available algorithms and tools that exploit graphs to build anomaly detection solutions. We believe that those would prove complementary to other types of techniques and should most probably be used in tandem for better detection performance. In fact, it is at the discretion of the practitioners to decide what type of scenarios best describe the problems they are dealing with, as well as what tools best fit their needs.

Open challenges While there has been tremendous amount of work in developing graph-based algorithms and machinery for graph-based abnormality detection and attribution, we believe there is still more work that needs to be done. In this part, we provide a discussion of open challenges which we group in two parts: *theoretical* and *practical* research challenges.

Theoretical research challenges While there has been considerable amount of work on static graphs, there still remain problems in the study of dynamic graphs.

- *Anomaly detection on attributed dynamic graphs* While static attributed graphs have been exploited in abnormality detection, there exists only a few works on spotting anomalies by exploiting dynamic *attributed* graphs (see Table 5). It is certainly of interest to develop definitions and formulations for abnormalities in such settings, and explore and identify where they could find applications in the real world.
- *The history/trace of dynamic updates* While most techniques for dynamic graphs consider and work with edge/node updates, there exists no work that exploits the *history* of the updates. For example, imagine a Web page having a link to a malicious Web page in the past which is later removed. While from the change point of view this is an edge removal, the existence of such a link in the history of the page should be taken into account in making future evaluations, rather than treating and committing to the change as a simple edge removal.
- *Choosing the 'right' time window/granularity* Many algorithms for time-evolving graphs require a time-window for feature extraction or computation of the normal graph/node activity; one of the open questions is how to choose this window in order to discover the different types of outliers in the graph sequences. Would it make sense to set it to a day, or a week, or a month, based on the respective periodicities that have been reported for human activities/botnet attacks etc.? Would another time granularity serve for detecting the existent anomalies? Or, would a combination of time granularities work best?

Moreover, while there has been considerable work from *algorithmic* point of view in abnormality detection, there still remains problems from *systems* perspective.

- *Adversarial robustness* Most methods in the data mining and machine learning community focus on detection performance while ignoring *adversarial robustness*.

It is of high interest, from the practitioner's point of view, to understand the adversarial robustness of a new algorithm; i.e. how easy is it to break the algorithm, or what is the minimum amount of knowledge or computational power the attacker needs to have access to, in order to camouflage his/her bad activities.
- *The cost of graph anomaly detection* Most methods ignore the *cost* aspects of information. These costs, on the other hand, may exhibit themselves in various forms with varying levels, e.g., cost in measurement and monitoring exerted on the system; cost in being exposed to certain types of attacks exerted on the users; and cost in getting around of the algorithms exerted on the adversaries (which also relates to the above). These varying costs should be accounted for differently in algorithm development.
- *Scalable real-time discontinuity detection* One of the most important future challenges is to develop scalable approaches for real-time discontinuity detection, i.e. for streaming graphs. Specifically, research should focus on algorithms that are linear, or, even better, sub-linear to the input.

Practical research challenges Challenges from the practitioner's point of view, which could also be posed as research problems, include the following.
- *Finding the X-factor* It is often hard to predict what would boost a detection algorithm's performance the most; e.g., better priors or better and/or more (human) labels in learning algorithms, better parameter tuning, creating frameworks that combine multiple algorithms working in parallel or sequentially, choosing the appropriate algorithms for the framework, or simply having more data.
- *Evaluation* Due to the challenges associated with collecting true labels related to cost and annotator noise, ground truth data is often inexistent. As such, various works employ different approaches as was discussed in the Concluding Remarks after Sects. 2 and 3, such as anomaly injections and qualitative analysis. Thus far, there is no standard for evaluating (graph) anomaly detection methods.
- *Graph construction* Often times the data does not form a network as it is the case in computer networks. Rather, it is up to the practitioners to build a network representation of their data in order to use graph-based techniques. In such cases, it is often hard to anticipate what source of data is best to use in graph construction.
- *Anomaly detection on multi-graphs* On the contrary to above, it may be the case that there is more than one network available, capturing different aspects of relations (e.g., friendship network and telecommunication network among the same individuals). While possibly beneficial, how to exploit all available networks and fuse clues from all these sources for anomaly detection remains an open area.
- *Balance between attribution and 'novelty' detection* By anomaly attribution, we essentially want to attribute the detected anomalies to known, human-understandable evidences (e.g., the known frauds, the known rule-based meta detectors, etc). This might contradict to some anomaly detection tasks where the goal is to find 'novel' patterns beyond users' current understandings. More research needs to be done in the direction of how to balance between the attribution and the ability of the detection algorithm to find 'novelty'.
- *Augmented graph anomaly detection* When there is an explicit network representation, it may also be possible to introduce/remove *latent* edges, for example edges

based on (e.g., text, time-series, correlation) similarities or domain knowledge (e.g., known irrelevant types of edges).

Acknowledgments This material is based upon work supported by the Army Research Office (ARO) under Cooperative Agreement Numbers W911NF-14-1-0029 and W911NF-09-2-0053, the Defense Advanced Research Projects Agency (DARPA) under Contract Numbers W911NF-11-C-0088, W911NF-11-C-0200 and W911NF-12-C-0028, the National Science Foundation (NSF) under Grant Nos. IIS-1217559 and IIS1017415, by Region II University Transportation Center under the Project number 49997-33-25, and the Stony Brook University Office of Vice President for Research. Any findings and conclusions expressed in this material are those of the author(s) and do not necessarily reflect the position or the policy of the U.S. Government and the other funding parties, and no official endorsement should be inferred. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation here on.

References

Abe N, Zadrozny B, Langford J (2006) Outlier detection by active learning. In: Proceedings of the 12th ACM international conference on knowledge discovery and data mining (SIGKDD), Philadelphia, PA, pp 504–509

Abe N, Melville P, Pendus C, Reddy CK, Jensen DL, Thomas VP, Bennett JJ, Anderson GF, Cooley BR, Kowalczyk M, Domick M, Gardinier T (2010) Optimizing debt collections using constrained reinforcement learning. In: Proceedings of the 16th ACM international conference on knowledge discovery and data mining (SIGKDD), Washington, DC. ACM, pp 75–84

Aggarwal C, Subbian K (2014) Evolutionary network analysis: a survey. ACM Comput Surv 47(1):10. doi:10.1145/2601412

Aggarwal CC (2012) Outlier ensembles. In: ACM SIGKDD explorations

Aggarwal CC (2013) Outlier analysis. Springer, New York Incorporated

Aggarwal CC, Yu PS (2001) Outlier detection for high dimensional data. In: Proceedings of the ACM international conference on management of data (SIGMOD), Santa Barbara, CA. ACM, pp 37–46

Aggarwal CC, Zhao Y, Yu PS (2011) Outlier detection in graph streams. In: Proceedings of the 27th international conference on data engineering (ICDE), Hannover, Germany, pp 399–409

Akoglu L, Faloutsos C (2009) RTG: a recursive realistic graph generator using random typing. Data Min Knowl Discov 19(2):194–209

Akoglu L, Faloutsos C (2010) Event detection in time series of mobile communication graphs. In: Proceedings of army science conference

Akoglu L, McGlohon M, Faloutsos C (2010) OddBall: spotting anomalies in weighted graphs. In: Proceedings of the 14th Pacific-Asia conference on knowledge discovery and data mining (PAKDD), Hyderabad, India, pp 410–421

Akoglu L, de Melo POSV, Faloutsos C (2012a) Quantifying reciprocity in large weighted communication networks. In: Proceedings of the 16th Pacific-Asia conference on knowledge discovery and data mining (PAKDD), Kuala Lumpur, Malaysia

Akoglu L, Tong H, Meeder B, Faloutsos C (2012b) PICS: parameter-free identification of cohesive subgroups in large attributed graphs. Proceedings of the 12th SIAM international conference on data mining (SDM), Anaheim, CA. SIAM/Omnipress, pp 439–450

Akoglu L, Tong H, Vreeken J, Faloutsos C (2012c) Fast and reliable anomaly detection in categorical data. In: Proceedings of the 21st ACM conference on information and knowledge management (CIKM), Maui, Hawaii, pp 415–424

Akoglu L, Chandy R, Faloutsos C (2013a) Opinion fraud detection in online reviews using network effects. In: Proceedings of the 7th international AAAI conference on weblogs and social media (ICWSM), Ann Arbor, MI

Akoglu L, Vreeken J, Tong H, Duen HC, Tatti N, Faloutsos C (2013b) Mining connection pathways for marked nodes in large graphs. In: Proceedings of the 13th SIAM international conference on data mining (SDM), Texas-Austin, TX

Ambai M, Utama NP, Yoshida Y (2011) Dimensionality reduction for histogram features based on supervised non-negative matrix factorization. IEICE Trans Inf Syst 94-D(10):1870–1879

Andersen R, Chung F, Lang K (2006) Local graph partitioning using pagerank vectors. In: Proceedings of the 47th annual IEEE symposium on foundations of computer science. IEEE Computer Society, pp 475–486

Ando S (2007) Clustering needles in a haystack: an information theoretic analysis of minority and outlier detection. In: Proceedings of the 7th IEEE international conference on data mining (ICDM), Omaha, NE, pp 13–22

Antonellis I, Garcia-Molina H, Chang C-C (2008) Simrank++: query rewriting through link analysis of the click graph. In: Proceedings of the 34nd international conference on very large data bases (VLDB), Auckland, New Zealand, pp 408–421

Araujo M, Papadimitriou S, Günnemann S, Faloutsos C, Basu P, Swami A, Papalexakis E, Koutra D (2014) Com2: fast automatic discovery of temporal (comet) communities. In: Proceedings of the 18th Pacific-Asia conference on knowledge discovery and data mining (PAKDD), Tainan, Taiwan

Backstrom L, Huttenlocher D, Kleinberg J, Lan X (2006) Group formation in large social networks: membership, growth, and evolution. In: Proceedings of the 12th ACM international conference on knowledge discovery and data mining (SIGKDD), Philadelphia, PA. ACM, pp 44–54

Barabási A-L, Albert R (1999) Emergence of scaling in random networks. Science 286:509–512

Bay SD, Pazzani MJ (1999) Detecting change in categorical data: mining contrast sets. In: Proceedings of the 5th ACM international conference on knowledge discovery and data mining (SIGKDD), San Diego, CA. ACM Press, pp 302–306

Bayati M, Gleich DF, Saberi A, Wang Y (2013) Message passing algorithms for sparse network alignment. ACM Trans Knowl Discov Data 7(1):3:1–3:31

Becchetti L, Castillo C, Donato D, Leonardi S, Baeza-Yates R (2006) Link-based characterization and detection of Web Spam. In: Second international workshop on adversarial information retrieval on the web (AIRWeb)

Benczúr AA, Csalogány K, Sarlós T, Uher M (2005) Spamrank: fully automatic link spam detection. In: Proceedings of the first international workshop on adversarial information retrieval on the web

Berlingerio M, Koutra D, Eliassi-Rad T, Faloutsos C (2012) Netsimile: a scalable approach to size-independent network similarity. CoRR, abs/1209.2684

Beyer K, Goldstein J, Ramakrishnan R, Shaft U (1999) When is "nearest neighbor" meaningful? In: International conference on database theory, pp 217–235

Bilgin CC, Yener B (2006) Dynamic Network Evolution: Models, Clustering, Anomaly Detection. Rensselaer Polytechnic Institute, Troy, NY

Boden B, Günnemann S, Hoffmann H, Seidl T (2012a) Mining coherent subgraphs in multi-layer graphs with edge labels. In: Proceedings of the 18th ACM international conference on knowledge discovery and data mining (SIGKDD), Beijing, China. ACM, pp 1258–1266

Boden B, Günnemann S, Seidl T (2012b) Tracing clusters in evolving graphs with node attributes. In: Proceedings of the 21st ACM conference on information and knowledge management (CIKM 2012), Maui, USA

Böhm C, Haegler K, Müller NS, Plant C (2009) CoCo: coding cost for parameter-free outlier detection. In: Proceedings of the 15th ACM international conference on knowledge discovery and data mining (SIGKDD), Paris, France. ACM, pp 149–158

Bolton RJ, Hand DJ (2001) Unsupervised profiling methods for fraud detection. In: Proceedings of conference credit scoring and credit control VII, pp 5–7

Bolton RJ, Hand DJ (2002) Statistical fraud detection: a review. Stat Sci 17(3):235–255

Bonacich P, Lloyd P (2001) Eigenvector-like measures of centrality for asymmetric relations. Soc Netw 23(3):191–201

Box GEP, Jenkins G (1990) Time series analysis. Forecasting and Control, Holden-Day, Incorporated

Breunig MM, Kriegel H-P, Ng RT, Sander J (2000) Lof: identifying density-based local outliers. In: Proceedings of the ACM international conference on management of data (SIGMOD), Dallas, TX. ACM, pp 93–104

Brin S, Page L (1998) The anatomy of a large-scale hypertextual web search engine. Comput Netw 30(1–7):107–117

Bunke H (1999) Error correcting graph matching: on the influence of the underlying cost function. IEEE Trans Pattern Anal Mach Intell 21(9):917–922

Bunke H, Dickinson PJ, Humm A, Irniger C, Kraetzl M (2006a) Computer network monitoring and abnormal event detection using graph matching and multidimensional scaling. In Proceedings of 6th industrial conference on data mining (ICDM), pp 576–590

Bunke H, Dickinson PJ, Kraetzl M, Wallis WD (2006b) A graph-theoretic approach to enterprise network dynamics (PCS). Birkhauser, Basel

Canali D, Cova M, Vigna G, Kruegel C (2011) Prophiler: a fast filter for the large-scale detection of malicious web pages. In: Proceedings of the 19th international conference on World Wide Web (WWW), Hyderabad, India. ACM, pp 197–206

Castillo C, Donato D, Gionis A, Murdock V, Silvestri F (2007) Know your neighbors: web spam detection using the web topology. In: Proceedings of the 30th international conference on research and development in information retrieval (SIGIR), Amsterdam. ACM, pp 423–430

Cha S-H (2007) Comprehensive survey on distance/similarity measures between probability density functions. Int J Math Models Methods Appl Sci 1(4):300–307

Chakrabarti D (2004) Autopart: parameter-free graph partitioning and outlier detection. In: Proceedings of the 8th European conference on principles and practice of knowledge discovery in databases (PKDD), Pisa. Italy. Springer, New York, pp 112–124

Chakrabarti D, Kumar R, Tomkins A (2006) Evolutionary clustering. In: Proceedings of the 12th ACM international conference on knowledge discovery and data mining (SIGKDD), Philadelphia, PA. ACM, pp 554–560

Chakrabarti S (2007) Dynamic personalized pagerank in entity-relation graphs. In: Proceedings of the 16th international conference on World Wide Web (WWW), Alberta, Canada, pp 571–580

Chandola V, Banerjee A, Kumar V (2009) Anomaly detection: a survey. ACM Comput Surv 41:15:1–15:58

Chandola V, Banerjee A, Kumar V (2012) Anomaly detection for discrete sequences: a survey. IEEE Trans Knowl Data Eng 24(5):823–839

Chartrand G, Kubicki G, Schulz M (1998) Graph similarity and distance in graphs. Aequ Math 55(1–2):129–145

Chau DH, Pandit S, Faloutsos C (2006) Detecting fraudulent personalities in networks of online auctioneers. In: Proceedings of the 10th European conference on principles and practice of knowledge discovery in databases (PKDD), Berlin, Germany, pp 103–114

Chau DH, Akoglu L, Vreeken J, Tong H, Faloutsos C (2012) Tourviz: interactive visualization of connection pathways in large graphs. In: Proceedings of the 18th ACM international conference on knowledge discovery and data mining (SIGKDD), Beijing, China, pp 1516–1519

Chaudhary A, Szalay AS, Moore AW (2002) Very fast outlier detection in large multidimensional data sets. In Proceedings of the ACM SIGMOD workshop on research issues in data mining and knowledge discovery (DMKD), Madison, WI

Chen H.-H, Giles CL (2013) ASCOS: an asymmetric network structure context similarity measure. In: IEEE/ACM international conference on advances in social networks analysis and mining (ASONAM), Niagara Falls, Canada

Cooper GF (1990) The computational complexity of probabilistic inference using Bayesian belief networks. Artif Intell 42(2–3):393–405

Cortes C, Pregibon D (2001) Signature-based methods for data streams. Data Min Knowl Discov 5(3):167–182

Cortes C, Fisher K, Pregibon D, Rogers A (2000) Hancock: a language for extracting signatures from data streams. In: Proceedings of the 6th ACM international conference on knowledge discovery and data mining (SIGKDD), Boston, MA. ACM, pp 9–17

Cortes C, Pregibon D, Volinsky C (2002) Communities of interest. Intell Data Anal 6(3):211–219

Dai H, Zhu F, Lim E-P, Pang HH (2012) Detecting anomalies in bipartite graphs with mutual dependency principles. In: Proceedings of the 12th IEEE international conference on data mining (ICDM), Brussels, Belgium. IEEE Computer Society, pp 171–180

Damnjanovic U, Virginia FA, Izquierdo E, Martínez JM (2008) Event detection and clustering for surveillance video summarization. In: 9th international workshop on image analysis for multimedia interactive services. IEEE Computer Society, pp 63–66

Das K, Schneider JG (2007) Detecting anomalous records in categorical datasets. In: Proceedings of the 13th ACM international conference on knowledge discovery and data mining (SIGKDD), San Jose, CA. ACM, pp 220–229

Davis M, Liu W, Miller P, Redpath G (2011) Detecting anomalies in graphs with numeric labels. In: Proceedings of the 21st ACM conference on information and knowledge management (CIKM), Glasgow, Scotland. ACM, pp 1197–1202

Deerwester S, Dumais ST, Furnas GW, Landauer TK, Harshman R (1990) Indexing by latent semantic analysis. J Am Soc Inf Sci 41(6):391–407

Dhillon IS, Mallela S, Modha DS (2003) Information-theoretic co-clustering. In: Proceedings of the 9th ACM international conference on knowledge discovery and data mining (SIGKDD), Washington, DC. ACM, pp 89–98

Dickinson P, Bunke H, Dadej A, Kraetzl M (2002) Median graphs and anomalous change detection in communication networks. In: Information, decision and control. Final Program and Abstracts, pp 59–64

Ding Q, Katenka N, Barford P, Kolaczyk ED, Crovella M (2012) Intrusion as (anti)social communication: characterization and detection. In: Proceedings of the 18th ACM international conference on knowledge discovery and data mining (SIGKDD), Beijing, China. ACM, pp 886–894

Drineas P, Kannan R, Mahoney MW (2006) Fast monte carlo algorithms for matrices iii: computing a compressed approximate matrix decomposition. SIAM J Comput 36(1):184–206

Eberle W, Holder LB (2007) Discovering structural anomalies in graph-based data. In: Proceedings of the international workshop on mining graphs and complex structures at the 7th IEEE international conference on data mining (ICDM), Omaha, NE. IEEE Computer Society, pp 393–398

Eberle W, Holder LB (2009) Graph-based approaches to insider threat detection. In: Proceedings of the 5th annual cyber security and information intelligence research workshop (CSIIRW). ACM, p 44

Edge ME, Falcone Sampaio PR (2009) A survey of signature based methods for financial fraud detection. Comput Secur 28(6):381–394

Elghawalby H, Hancock ER (2008) Measuring graph similarity using spectral geometry. In: Proceedings of the 5th international conference on image analysis and recognition (ICIAR), pp 517–526

Faloutsos C, McCurley KS, Tomkins A (2004) Fast discovery of connection subgraphs. In: Proceedings of the 10th ACM international conference on knowledge discovery and data mining (SIGKDD), Seattle, WA, pp 118–127

Fawcett T, Provost FJ (1996) Combining data mining and machine learning for effective user profiling. In: Proceedings of the 2nd AAAI international conference on knowledge discovery and data mining (KDD), Portland, OR. AAAI Press, pp 8–13

Fawcett T, Provost FJ (1999) Activity monitoring: noticing interesting changes in behavior. In: Proceedings of the 5th ACM international conference on knowledge discovery and data mining (SIGKDD), San Diego, CA. ACM, pp 53–62

Fayyad UM, Irani KB (1993) Multi-interval discretization of continuous-valued attributes for classification learning. In: Proceedings of the 5th international joint conference on artificial intelligence (IJCAI), Chambery, France. Morgan Kaufmann, pp 1022–1029

Federal Bureau of Investigation (FBI) (2009) Online auction fraud

Feller W (1968) An introduction to probability theory and its applications. Wiley, New York

Feng S, Banerjee R, Choi Y (2012a) Syntactic stylometry for deception detection. In: Proceedings of the 50th annual meeting of the association for computational linguistics (ACL), Jeju Island, Korea

Feng S, Xing L, Gogar A, Choi Y (2012b) Distributional footprints of deceptive product reviews. In: Proceedings of the 6th international AAAI conference on weblogs and social media (ICWSM), Dublin, Ireland

Fiedler M (1973) Algebraic connectivity of graphs. Czechoslov Math J 23(98):298–305

Fisher NI, Lewis T, Embleton BJJ (1993) Statistical analysis of spherical data. Cambridge University Press, Cambridge, MA

Flegel U, Vayssire J, Bitz G (2010) A state of the art survey of fraud detection technology. In: Insider threats in cyber security, volume 49 of advances in information security. Springer, Berlin, pp 73–84

Freeman LC (1977) A set of measures of centrality based upon betweenness. Sociometry 40:35–41

Friedman N, Getoor L, Koller D, Pfeffer A (1999) Learning probabilistic relational models. In: Proceedings of the 11th international joint conference on artificial intelligence (IJCAI), Stockholm, Sweden, pp 1300–1309

Gallagher B, Tong H, Eliassi-Rad T, Faloutsos C (2008) Using ghost edges for classification in sparsely labeled networks. In: Proceedings of the 14th ACM international conference on knowledge discovery and data mining (SIGKDD), Las Vegas, NV. ACM, pp 256–264

Gama J, Medas P, Castillo G, Rodrigues P (2004) Learning with drift detection. In: SBIA Brazilian symposium on artificial intelligence. Springer, Berlin, pp 286–295

Gao H, Chen Y, Lee K, Palsetia D, Choudhary A (2012) Towards online spam filtering in social networks. In: Proceedings of the 19th annual network & distributed system security symposium

Gao J, Tan P-N (2006) Converting output scores from outlier detection algorithms into probability estimates. In: Proceedings of the 6th IEEE international conference on data mining (ICDM), Hong Kong, China, pp 212–221

Gao J, Liang F, Fan W, Wang C, Sun Y, Han J (2010a) On community outliers and their efficient detection in information networks. In: Proceedings of the 16th ACM international conference on knowledge discovery and data mining (SIGKDD), Washington, DC. ACM, pp 813–822

Gao X, Xiao B, Tao D, Li X (2010b) A survey of graph edit distance. J Pattern Anal Appl 13(1):113–129

Gaston ME, Kraetzl M, Wallis WD (2006) Using graph diameter for change detection in dynamic networks. Aust J Comb, 299–311

Ghoting A, Parthasarathy S, Otey ME (2008) Fast mining of distance-based outliers in high-dimensional datasets. Data Min Knowl Discov 16(3):349–364

Glaz J, Naus J, Wallenstein S (2001) Scan Statistics. Springer

Golub GH, Van Loan CF (1996) Matrix computations, 3rd edn. Johns Hopkins University Press, Baltimore, MD

Grigg OA, Farewell VT, Spiegelhalter DJ (2003) Use of risk-adjusted cusum and rspert charts for monitoring in medial contexts. Stat Methods Med Res

Günnemann S, Färber I, Boden B, Seidl T (2010) Subspace clustering meets dense subgraph mining: a synthesis of two paradigms. In: Proceedings of the 10th IEEE international conference on data mining (ICDM), Sydney, Australia. IEEE Computer Society, pp 845–850

Günnemann S, Boden B, Seidl T (2012) Finding density-based subspace clusters in graphs with feature vectors. Data Min Knowl Discov 25(2):243–269

Gupta M, Gao J, Sun Y, Han J (2012) Integrating community matching and outlier detection for mining evolutionary community outliers. In: Proceedings of the 18th ACM international conference on knowledge discovery and data mining (SIGKDD), Beijing, China. ACM, pp 859–867

Gupta M, Gao J, Aggarwal CC, Han J (2013) Outlier detection for temporal data: a survey. IEEE Trans Knowl Data Eng 99(PrePrints):1. ISSN 1041–4347

Gupta M, Gao J, Aggarwal CC, Han J (2014) Outlier detection for temporal data. Synthesis lectures on data mining and knowledge discovery. Morgan & Claypool Publishers

Gupte M, Eliassi-Rad T (2012) Measuring tie strength in implicit social networks. In: Proceedings of the ACM conference on web science, Evanston, IL. ACM, pp 109–118

Gyöngyi Z, Garcia-Molina H, Pedersen J (2004) Combating web spam with trustrank. In: Proceedings of the 30th international conference on very large data bases (VLDB), Canada, Toronto, pp 576–587

Haveliwala TH (2003) Topic-sensitive pagerank: a context-sensitive ranking algorithm for web search. IEEE Trans Knowl Data Eng 15(4):784–796

Hawkins D (1980) Identification of outliers. Chapman and Hall, London

He Z, Xiaofei X, Deng S (2003) Discovering cluster-based local outliers. Pattern Recognit Lett 24(9–10):1641–1650

Heard NA, Weston DJ, Platanioti K, Hand DJ (2010) Bayesian anomaly detection methods for social networks. Ann Appl Stat 4:645–662

Hempstalk K, Frank E, Witten IH (2008) One-class classification by combining density and class probability estimation. In: Proceedings of the European conference on machine learning and principles and practice of knowledge discovery in databases (ECML PKDD), Antwerp, Belgium. Springer, Berlin

Henderson K, Eliassi-Rad T, Faloutsos C, Akoglu L, Li L Maruhashi K, Prakash BA, Tong H (2010) Metricforensics: a multi-level approach for mining volatile graphs. In: Proceedings of the 16th ACM international conference on knowledge discovery and data mining (SIGKDD), Washington, DC, pp 163–172

Henderson K, Gallagher B, Li L, Akoglu L, Eliassi-Rad T, Tong H, Faloutsos C (2011) It's who you know: graph mining using recursive structural features. In: Proceedings of the 17th ACM international conference on knowledge discovery and data mining (SIGKDD), San Diego, CA. ACM, pp 663–671

Henderson K, Gallagher B, Eliassi-Rad T, Tong H, Basu S, Akoglu L, Koutra D, Faloutsos C, Li L (2012) RolX: structural role extraction & mining in large graphs. In: Proceedings of the 18th ACM international conference on knowledge discovery and data mining (SIGKDD), Beijing, China, pp 1231–1239

Idé T, Kashima H (2004) Eigenspace-based anomaly detection in computer systems. In: Proceedings of the tenth ACM SIGKDD international conference on knowledge discovery and data mining, proceedings of the 10th ACM international conference on knowledge discovery and data mining (SIGKDD), Seattle, WA. ACM, pp 440–449

Iliofotou M, Pappu P, Faloutsos M, Mitzenmacher M, Sumeet S, Varghese G (2007) Network monitoring using traffic dispersion graphs. In: Proceedings of the 7th ACM SIGCOMM conference on internet measurement, San Diego, CA. ACM, pp 24–26

Iliofotou M, Kim H, Faloutsos M, Mitzenmacher M, Pappu P, Varghese G (2011) Graption: a graph-based P2P traffic classification framework for the internet backbone. Comput Netw 55(8):1909–1920
Invernizzi L, Comparetti PM (2012) Evilseed: a guided approach to finding malicious web pages. In: IEEE symposium on security and privacy, pp 428–442
Ishibashi K, Kondoh T, Harada S, Mori T, Kawahara R, Asano S (2010) Detecting anomalous traffic using communication graphs. In: Telecommunications: the infrastructure for the 21st century (WTC), pp 1–6
Jansen BJ (2008) Click fraud. IEEE Comput 40(7):85–86
Janssens JHM, Flesch I, Postma EO (2009) Outlier detection with one-class classifiers from ML and KDD. In: Proceedings of the 8th international conference on machine learning and applications (ICMLA), Miami Beach, FL. IEEE Computer Society, pp 147–153
Jeh G, Widom J (2002) SimRank: a measure of structural-context similarity. In: Proceedings of the 8th ACM international conference on knowledge discovery and data mining (SIGKDD), Edmonton, Alberta, pp 538–543
Jensen D, Neville J, Gallagher B (2004) Why collective inference improves relational classification. In: Proceedings of the 10th ACM international conference on knowledge discovery and data mining (SIGKDD), Seattle, WA, pp 593–598
Jindal N, Liu B (2008) Opinion spam and analysis. In: Proceeding of the 1st ACM international conference on web search and data mining (WSDM), pp 219–230
Jindal N, Liu B, Lim E-P (2010) Finding unusual review patterns using unexpected rules. In: Proceedings of the 19th ACM conference on information and knowledge management (CIKM), Toronto, Canada. ACM, pp 1549–1552
Kahneman D (2011) Thinking, fast and slow. Farrar, Straus and Giroux
Kang U, McGlohon M, Akoglu L, Faloutsos C (2010) Patterns on the connected components of terabyte-scale graphs. In: Proceedings of the 10th IEEE international conference on data mining (ICDM), Sydney, Australia, pp 875–880
Kang U, Chau DH, Faloutsos C (2011a) Mining large graphs: algorithms, inference, and discoveries. In: Proceedings of the 27th international conference on data engineering (ICDE), Hannover, Germany. IEEE Computer Society, pp 243–254
Kang U, Papadimitriou S, Sun J, Tong H (2011b) Centralities in large networks: algorithms and observations. In: Proceedings of the 11th SIAM international conference on data mining (SDM), Mesa, AZ, pp 119–130
Kang U, Tsourakakis CE, Appel AP, Faloutsos C, Leskovec J (2011c) Hadi: mining radii of large graphs. ACM Trans Knowl Discov Data 5: 8:1–8:24. ISSN 1556–4681
Kang U, Tong H, Sun J (2012) Fast random walk graph kernel. In: Proceedings of the 12th SIAM international conference on data mining (SDM), Anaheim, CA
Kang U, Lee J.-Y., Koutra D, Faloutsos C (2014) Net-Ray: visualizing and mining web-scale graphs. In: Proceedings of the 18th Pacific-Asia conference on knowledge discovery and data mining (PAKDD), Tainan, Taiwan
Kapsabelis KM, Dickinson PJ, Dogancay K (2007) Investigation of graph edit distance cost functions for detection of network anomalies. In: Proceedings of the 13th Biennial computational techniques and applications conference, CTAC-2006, volume 48 of ANZIAM journal, pp C436–C449
Karypis G, Kumar V (1995) Metis-unstructured graph partitioning and sparse matrix ordering system, version 2.0. Technical report, University of Minnesota, Department of Computer Science
Karypis G, Kumar V (1996) Parallel multilevel k-way partitioning scheme for irregular graphs. In: Proceedings of the 1996 ACM/IEEE conference on supercomputing (CDROM), Supercomputing '96. IEEE Computer Society
Kashima H, Tsuda K, Inokuchi A (2003) Marginalized kernels between labeled graphs. In: Proceedings of the twentieth international conference on machine learning. AAAI Press, pp 321–328
Katz L (1953) A new status index derived from sociometric analysis. Psychometrika 18(1):39–43
Keller F, Müller E, Böhm K (2012) Hics: high contrast subspaces for density-based outlier ranking. In: Proceedings of the 28th international conference on data engineering (ICDE), Washington, DC, pp 1037–1048
Kelmans AK (1976) Comparison of graphs by their number of spanning trees. Discrete Math 16(3):241–261
Kleinberg JM (1998) Authoritative sources in a hyperlinked environment. In: Proceedings of the 5th Annual ACM-SIAM symposium on discrete algorithms (SODA), San Francisco, CA, pp 668–677
Knorr EM, Ng RT (1998) Algorithms for mining distance-based outliers in large datasets. In: Proceedings of the 24th international conference on very large data bases (VLDB), New York City, NY, pp 392–403

Kontkanen P, Myllymki P (2007) MDL histogram density estimation. J Mach Learn Res Proc Track 2:219–226

Koren Y, North SC, Volinsky C (2006) Measuring and extracting proximity in networks. In: Proceedings of the 12th ACM international conference on knowledge discovery and data mining (SIGKDD), Philadelphia, PA, pp 245–255 (2006)

Koutra D, Ke T-Y, Kang U, Chau DH, Pao H-KK, Faloutsos C (2011) Unifying guilt-by-association approaches: theorems and fast algorithms. In: Proceedings of the European conference on machine learning and principles and practice of knowledge discovery in databases (ECML PKDD), Greece, Athens, pp 245–260

Koutra D, Papalexakis E, Faloutsos C (2012) Tensorsplat: spotting latent anomalies in time. In: 16th panhellenic conference on informatics (PCI)

Koutra D, Tong H, Lubensky D (2013a) Big-Align: fast bipartite graph alignment. In: Proceedings of the 13th IEEE international conference on data mining (ICDM), Dallas, Texas

Koutra D, Vogelstein J, Faloutsos C (2013b) Deltacon: a principled massive-graph similarity function. In: Proceedings of the 13th SIAM international conference on data mining (SDM), Texas-Austin, TX

Krausz B, Herpers R (2010) MetroSurv: detecting events in subway stations. Multimed Tools Appl 50(1):123–147

Kriegel H-P, Kröger P, Schubert E, Zimek A (2012) Outlier detection in arbitrarily oriented subspaces. In: Proceedings of the 12th IEEE international conference on data mining (ICDM). Brussels, Belgium, pp 379–388

Krishnan V, Raj R (2006) Web spam detection with anti-trust rank. In: Proceedings of the 2nd international workshop on adversarial IR on the Web at the 29th international conference on research and development in information retrieval (SIGIR), Seattle, WA, pp 37–40

Kshetri N (2010) The economics of click fraud. IEEE Secur Priv 8(3):45–53

Kuang D, Park H, Ding CHQ (2012) Symmetric nonnegative matrix factorization for graph clustering. In: Proceedings of the 12th SIAM international conference on data mining (SDM), Anaheim, CA, pp 106–117

Kulldorff M (1997) A spatial scan statistic. Commun Stat Theory Methods 26:1481–1496

Kumar M, Ghani R, Mei Z-S (2010) Data mining to predict and prevent errors in health insurance claims processing. In: Proceedings of the 16th ACM international conference on knowledge discovery and data mining (SIGKDD), Washington, DC. ACM, pp 65–74

Kuramochi M, Karypis G (2001) Frequent subgraph discovery. In: Proceedings of the 2001 IEEE international conference on data mining, proceedings of the 1st IEEE international conference on data mining (ICDM), San Jose, CA, Washington, DC, USA. IEEE Computer Society, pp 313–320

Lazarevic A, Kumar V (2005) Feature bagging for outlier detection. In: Proceedings of the 11th ACM international conference on knowledge discovery and data mining (SIGKDD), Chicago, IL, pp 157–166

Lee DD, Sebastian HS (2000) Algorithms for non-negative matrix factorization. In: Proceedings of the 14th annual conference on neural information processing systems (NIPS), Denver, CO, pp 556–562

Lee K, Caverlee J, Webb S (2010) Uncovering social spammers: social honeypots + machine learning. In: Proceedings of the 33rd international conference on research and development in information retrieval (SIGIR), Switzerland, Geneva, pp 435–442

Leeuwen M, Siebes A (2008) Streamkrimp: detecting change in data streams. In: Proceedings of the European conference on machine learning and principles and practice of knowledge discovery in databases (ECML PKDD), Antwerp, Belgium. Springer, Berlin, pp 672–687

Leskovec J, Kleinberg J, Faloutsos C (2005) Graphs over time: densification laws, shrinking diameters and possible explanations. In: Proceedings of the 11th ACM international conference on knowledge discovery and data mining (SIGKDD), Chicago, IL. ACM, pp 177–187

Leskovec J, Lang KJ, Mahoney M (2010) Empirical comparison of algorithms for network community detection. In: Proceedings of the 19th international conference on World Wide Web (WWW), Raleigh, NC, New York, NY, USA, ACM, pp 631–640

Li G, Semerci M, Yener B, Zaki MJ (2011a) Graph classification via topological and label attributes. In: Proceedings of the 9th international workshop on mining and learning with graphs (MLG), San Diego, USA

Li L, Liang C.-JM, Liu J, Nath S, Terzis A, Faloutsos C (2011b) Thermocast: a cyber-physical forecasting model for data centers. In: Proceedings of the 17th ACM international conference on knowledge discovery and data mining (SIGKDD), San Diego, CA. ACM

Li Z, Xiong H, Liu Y, Zhou A (2010) Detecting blackhole and volcano patterns in directed networks. In: Proceedings of the 10th IEEE international conference on data mining (ICDM), Sydney, Australia. IEEE Computer Society, pp 294–303

Liben-Nowell D, Kleinberg JM (2003) The link prediction problem for social networks. In: Proceedings of the 12th ACM conference on information and knowledge management (CIKM), New Orleans, LA, pp 556–559

Lieto G, Orsini F, Pagano G (2008) Cluster analysis for anomaly detection. In: Proceedings of the 2nd international conference on complex, intelligent and software intensive systems (CISIS), Barcelona, Spain, volume 53 of advances in soft computing. Springer, Berlin, pp 163–169

Lin J, Keogh E, Lonardi S, Chiu B (2003) A symbolic representation of time series, with implications for streaming algorithms. In: Proceedings of the ACM SIGMOD workshop on research issues in data mining and knowledge discovery (DMKD), San Diego, CA. ACM, pp 2–11

Liu B, Xiao Y, Cao L, Hao Z, Deng F (2013) Svdd-based outlier detection on uncertain data. Knowl Inf Syst 34(3):597–618

Liu C, Yan X, Yu H, Han J, Philip SY (2005) Mining behavior graphs for "backtrace" of noncrashing bugs. In: Proceedings of the 5th SIAM international conference on data mining (SDM), Newport Beach, CA

Lu Q, Getoor L (2003) Link-based classification. In: Proceedings of the 20th international conference on machine learning (ICML), Washington, DC

Ma J, Saul LK, Savage S, Voelker GM (2009) Beyond blacklists: learning to detect malicious web sites from suspicious URLs. In: Proceedings of the 15th ACM international conference on knowledge discovery and data mining (SIGKDD), Paris, France. ACM, pp 1245–1254

Macindoe O, Richards W (2010) Graph comparison using fine structure analysis. In: International conference on privacy, security, risk and trust (SocialCom/PASSAT), pp 193–200

Macskassy S, Provost F (2003) A simple relational classifier. In: Proceedings of the KDD-workshop on multi-relational data mining (MRDM), Washington, DC, pp 64–76

Margineantu DD, Wong W-K, Dash D (2010) Machine learning algorithms for event detection. Mach Learn 79(3):257–259

McGlohon M, Bay S, Anderle MG, Steier DM, Faloutsos C (2009) Snare: a link analytic system for graph labeling and risk detection. In: Proceedings of the 15th ACM international conference on knowledge discovery and data mining (SIGKDD), Paris, France, pp 1265–1274

Medina A, Lakhina A, Matta I, Byers JW (2001) BRITE: an approach to universal topology generation. In: Proceedings of the IEEE 9th international symposium on modeling, analysis and simulation of computer and telecommunication systems. IEEE Computer Society

Melnik S, Garcia-Molina H, Rahm E (2002) Similarity flooding: a versatile graph matching algorithm and its application to schema matching. In: Proceedings of the 18th international conference on data engineering (ICDE), San Jose, CA

Miller DJ, Browning J (2003) A mixture model and em-based algorithm for class discovery, robust classification, and outlier rejection in mixed labeled/unlabeled data sets. IEEE Trans Pattern Anal Mach Intell 25(11):1468–1483

Mongiovi M, Bogdanov P, Ranca R, Singh AK, Papalexakis EE, Faloutsos C (2013) Netspot: spotting significant anomalous regions on dynamic networks. In: Proceedings of the 13th SIAM international conference on data mining (SDM), Texas-Austin, TX

Montgomery DC (1997) Introduction to statistical quality control, 3rd edn. Wiley, New York

Müller E, Schiffer M, Seidl T (2010) Adaptive outlierness for subspace outlier ranking. In: Proceedings of the 19th ACM conference on information and knowledge management (CIKM), Toronto, Canada. ACM, pp 1629–1632

Müller E, Assent I, Sanchez PI, Mülle Y, Böhm K (2012) Outlier ranking via subspace analysis in multiple views of the data. In: Proceedings of the 12th IEEE international conference on data mining (ICDM), Brussels, Belgium. IEEE Computer Society, pp 529–538

Müller E, Sánchez PI, Mülle Y, Böhm K (2013) Ranking outlier nodes in subspaces of attributed graphs. In: Proceedings of the 4th international workshop on graph data management: techniques and applications

Naus JI (1982) Approximations for distributions of scan statistics. J Am Stat Assoc 77(377):177–183

Neil J (2011) Scan statistics for the online detection of locally anomalous subgraphs. PhD thesis, University of New Mexico

Neill DB, Wong W.-K (2009) A tutorial on event detection tutorial. In: ACM international conference on knowledge discovery and data mining (SIGKDD)

Neville J, Jensen D (2000) Iterative classification in relational data. In: Proceedings of the AAAI workshop on learning statistical models from relational data. AAAI Press, pp 13–20
Neville J, Jensen D (2003) Collective classification with relational dependency networks. In: Proceedings of the 9th ACM international conference on knowledge discovery and data mining (SIGKDD), Washington, DC
Neville J, Jensen D, Friedland L, Hay M (2003) Learning relational probability trees. In: Proceedings of the 9th ACM international conference on knowledge discovery and data mining (SIGKDD), Washington, DC
Neville J, Simsek O, Jensen D, Komoroske J, Palmer K, Goldberg HG (2005) Using relational knowledge discovery to prevent securities fraud. In: Proceedings of the 11th ACM international conference on knowledge discovery and data mining (SIGKDD), Chicago, IL, pp 449–458
Newman MEJ (2004) Detecting community structure in networks. Eur Phys J B 38:321–330
Newman MEJ (2006) Modularity and community structure in networks. Proc Natl Acad Sci 103(23):8577–8582
Newman MEJ, Girvan M (2004) Finding and evaluating community structure in networks. Phys Rev E 69(2):026113
Ng AY, Jordan MI, Weiss Y (2001) On spectral clustering: analysis and an algorithm. In: Advances in neural information processing systems. MIT Press, pp 849–856
Nikulin V, Huang T-H (2012) Unsupervised dimensionality reduction via gradient-based matrix factorization with two adaptive learning rates. J Mach Learn Res Proc Track 27:181–194
Noble CC, Cook DJ (2003) Graph-based anomaly detection. In: Proceedings of the 9th ACM international conference on knowledge discovery and data mining (SIGKDD), Washington, DC, pp 631–636
Noh JD, Rieger H (2004) Random walks on complex networks. Phys Rev Lett 92:118701
Ntoulas A, Najork M, Manasse M, Fetterly D (2006) Detecting spam web pages through content analysis. In: Proceedings of the World Wide Web conference. Edinburgh, Scotland, pp 83–92
Orair GH, Teixeira CHC, Wang Y, Meira W Jr, Parthasarathy S (2010) Distance-based outlier detection: consolidation and renewed bearing. Proc VLDB Endow 3(2):1469–1480
Otey ME, Ghoting A, Parthasarathy S (2006) Fast distributed outlier detection in mixed-attribute data sets. Data Min Knowl Discov 12(2–3):203–228
Ott M, Choi Y, Cardie C, Hancock JT (2011) Finding deceptive opinion spam by any stretch of the imagination. In Proceedings of the 49th annual meeting of the association for computational linguistics (ACL), Portland, OR, pp 309–319
Ott M, Cardie C, Hancock JT (2012) Estimating the prevalence of deception in online review communities. In: Proceedings of the 21st international conference on World Wide Web (WWW). Lyon, France. ACM, pp 201–210
Pandit S, Chau DH, Wang S, Faloutsos C (2007) Netprobe: a fast and scalable system for fraud detection in online auction networks. In: Proceedings of the 16th international conference on World Wide Web (WWW), Alberta, Canada
Papadimitriou P, Dasdan A, Garcia-Molina H (2008) Web graph similarity for anomaly detection. J Internet Serv Appl 1(1):1167
Papadimitriou S, Sun J (2008) DisCo: distributed co-clustering with map-reduce: a case study towards petabyte-scale end-to-end mining. In: Proceedings of the 8th IEEE international conference on data mining (ICDM), Pisa, Italy. IEEE Computer Society, pp 512–521
Papadimitriou S, Kitagawa H, Gibbons PB, Faloutsos C (2003) Loci: fast outlier detection using the local correlation integral. In: Proceedings of the 19th international conference on data engineering (ICDE), Bangalore, India. IEEE Computer Society, pp 315–326
Papalexakis EE, Faloutsos C, Sidiropoulos ND (2012) Parcube: sparse parallelizable tensor decompositions. In: Proceedings of the European conference on machine learning and principles and practice of knowledge discovery in databases (ECML PKDD). Bristol, UK, pp 521–536
Pauwels EJ, Ambekar O (2011) One class classification for anomaly detection: support vector data description revisited. In: Proceedings of the 11th IEEE international conference on data mining (ICDM), vol 6870, Vancouver, Canada, pp 25–39
Peabody M (2003) Finding groups of graphs in databases. Master's thesis, Drexel University
Pearson K (1901) On lines and planes of closest fit to systems of points in space. Philos Mag 2(6):559–572
Peel L, Clauset A (2014) Detecting change points in the large-scale structure of evolving networks. CoRR, abs/1403.0989

Pelillo M (1999) Replicator equations, maximal cliques, and graph isomorphism. Neural Comput 11(8):1933–1955
Perozzi B, Akoglu L, Sanchez PI, Müller E (2014) Focused clustering and outlier detection in large attributed graphs. In: ACM special interest group on knowledge discovery and data mining (SIG-KDD)
Phua C, Alahakoon D, Lee V (2004) Minority report in fraud detection: classification of skewed data. SIGKDD Explor 6(1):50–59
Phua C, Lee VCS, Smith-Miles K, Gayler RW (2010) A comprehensive survey of data mining-based fraud detection research. CoRR, abs/1009.6119
Pincombe B (2005) Anomaly detection in time series of graphs using arma processes. ASOR Bull 24(4): 2–10
Priebe CE, Conroy JM, Marchette DJ, Park Y (2005) Scan statistics on enron graphs. Comput Math Organ Theory 11(3):229–247. ISSN 1381–298X
Provos N, McNamee D, Mavrommatis P, Wang K, Modadugu N (2007) The ghost in the browser: analysis of web-based malware. In: Proceedings of the 1st workshop on hot topics in understanding botnets (HotBots)
Radke RJ, Andra S, Al-Kofahi O, Roysam B (2005) Image change detection algorithms: a systematic survey. IEEE Trans Image Process 14(3):294–307
Rahman MS, Huang T.-K., Madhyastha HV, Faloutsos M 2012) Efficient and scalable socware detection in online social networks. In: Proceedings of the 21st USENIX conference on Security symposium (Security). USENIX Association, pp 32–32
Ramakrishnan C, Milnor W, Perry M, Sheth A (2005) Discovering informative connection subgraphs in multi-relational graphs. In: SIGKDD explorations special issue on link mining
Rissanen J (1999) Hypothesis selection and testing by the MDL principle. Comput J 42:260–269
Rossi RA, Gallagher B, Neville J, Henderson K (2012) Role-dynamics: fast mining of large dynamic networks. In: Proceedings of the 21st international conference on World Wide Web (WWW), Lyon, France, WWW '12 Companion. ACM, pp 997–1006
Rossi RA, Gallagher B, Neville J, Henderson K (2013) Modeling dynamic behavior in large evolving graphs. In: Proceeding of the 6th ACM international conference on Web search and data mining (WSDM), pp 667–676
Ruts I, Rousseeuw PJ (1996) Computing depth contours of bivariate point clouds. Comput Stat Data Anal 23(1):153–168
Saltenis V (2004) Outlier detection based on the distribution of distances between data points. Informatica (Lithuanian Academy of Sciences) 15(3):399–410
Schubert E, Zimek A, Kriegel H-P (2012) Local outlier detection reconsidered: a generalized view on locality with applications to spatial, video, and network outlier detection. Data Mining Knowl Discov 28(1): 190–237. doi:10.1007/s10618-012-0300-z
Sen P, Namata G, Bilgic M, Getoor L, Gallagher B, Eliassi-Rad T (2008) Collective classification in network data. AI Mag 29(3):93–106
Shi J, Malik J (1997) Normalized cuts and image segmentation. IEEE Trans Pattern Anal Mach Intell 22:888–905
Shoubridge P, Kraetzl M, Ray D (1999) Detection of abnormal change in dynamic networks. In: Information, decision and control, 1999. IDC 99. Proceedings. pp 557–562
Shoubridge P, Kraetzl M, Wallis WD, Bunke H (2002) Detection of abnormal change in a time series of graphs. J Interconnect Netw 3(1–2):85–101
Smets K, Vreeken J (2011) The Odd One Out: identifying and characterising anomalies. In: Proceedings of the 11th SIAM international conference on data mining (SDM), Mesa, AZ, pp 804–815
Sun H, Huang J, Han J, Deng H, Zhao P, Feng B (2010) gskeletonclu: density-based network clustering via structure-connected tree division or agglomeration. In: Proceedings of the 10th IEEE international conference on data mining (ICDM), Sydney, Australia. IEEE Computer Society, pp 481–490
Sun J, Qu H, Chakrabarti D, Faloutsos C (2005) Neighborhood formation and anomaly detection in bipartite graphs. In: Proceedings of the 5th IEEE international conference on data mining (ICDM), Houston, TX. IEEE Computer Society, pp 418–425
Sun J, Tao D, Faloutsos C (2006) Beyond streams and graphs: dynamic tensor analysis. In: Proceedings of the 12th ACM international conference on knowledge discovery and data mining (SIGKDD), Philadelphia, PA, pp 374–383

Sun J, Faloutsos C, Papadimitriou S, Yu PS (2007a) GraphScope: parameter-free mining of large time-evolving graphs. In: Proceedings of the 13th ACM international conference on knowledge discovery and data mining (SIGKDD), San Jose, CA. ACM, pp 687–696

Sun J, Xie Y, Zhang H, Faloutsos C (2007b) Less is more: compact matrix decomposition for large sparse graphs. In: Proceedings of the 7th SIAM international conference on data mining (SDM), Minneapolis, MN

Sun J, Xie Y, Zhang H, Faloutsos C (2008) Less is more: sparse graph mining with compact matrix decomposition. Stat Anal Data Min 1(1): 6–22. ISSN 1932–1864

Taniguchi M, Haft M, Hollmen J, Tresp V (1998) Fraud detection in communication networks using neural and probabilistic methods. Acoust Speech Signal Process 2:1241–1244

Tantipathananandh C, Berger-Wolf T (2009) Constant-factor approximation algorithms for identifying dynamic communities. In: Proceedings of the 15th ACM international conference on knowledge discovery and data mining (SIGKDD), Paris, France. ACM, pp 827–836

Tantipathananandh C, Berger-Wolf T (2011) Finding communities in dynamic social networks. In: Proceedings of the 11th IEEE international conference on data mining (ICDM), Vancouver, Canada. IEEE, pp 1236–1241

Tantipathananandh C, Berger-Wolf T, Kempe D (2007) A framework for community identification in dynamic social networks. In: Proceedings of the 13th ACM international conference on knowledge discovery and data mining (SIGKDD), San Jose, CA, New York, NY, USA, ACM, pp 717–726

Taskar B, Abbeel P, Koller D (2002) Discriminative probabilistic models for relational data. In: Proceedings of the 18th conference on uncertainty in artificial intelligence (UAI), Alberta, Canada, pp 485–492

Tong H, Faloutsos C (2006) Center-piece subgraphs: problem definition and fast solutions. In: Proceedings of the 12th ACM international conference on knowledge discovery and data mining (SIGKDD), Philadelphia, PA, pp 404–413

Tong H, Lin C-Y (2011) Non-negative residual matrix factorization with application to graph anomaly detection. In: Proceedings of the 11th SIAM international conference on data mining (SDM), Mesa, AZ, pp 143–153

Tong H, Lin C-Y (2012) Non-negative residual matrix factorization: problem definition, fast solutions, and applications. Stat Anal Data Min 5(1):3–15

Tong H, Papadimitriou S, Jimeng S, Yu PS, Faloutsos C (2008) Colibri: fast mining of large static and dynamic graphs. In: Proceedings of the 14th ACM international conference on knowledge discovery and data mining (SIGKDD), Las Vegas, NV, pp 686–694

Ullmann JR (1976) An algorithm for subgraph isomorphism. J ACM 23(1):31–42

Vishwanathan SVN, Schraudolph NN, Kondor RI, Borgwardt KM (2010) Graph kernels. J Mach Learn Res 11:1201–1242

Wang G, Xie S, Liu B, Yu PS (2011a) Review graph based online store review spammer detection. In: Proceedings of the 11th IEEE international conference on data mining (ICDM), Vancouver, Canada, pp 1242–1247

Wang G, Xie S, Liu B, Yu PS (2012a) Identify online store review spammers via social review graph. ACM Trans Intell Syst Technol 3(4):61

Wang L, Rege M, Dong M, Ding Y (2012b) Low-rank kernel matrix factorization for large-scale evolutionary clustering. IEEE Trans Knowl Data Eng 24(6):1036–1050

Wang X, Wang X, Wilkes DM (2012c) A minimum spanning tree-inspired clustering-based outlier detection technique. In: Proceedings of the 12th IEEE international conference on data mining (ICDM), Belgium, Brussels, pp 209–223

Wang Y, Parthasarathy S, Tatikonda S (2011b) Locality sensitive outlier detection: a ranking driven approach. In: Proceedings of the 27th international conference on data engineering (ICDE), Hannover, Germany, pp 410–421

Watts DJ (1999) Small worlds. Princeton University Press, Princeton, NJ

Watts DJ, Strogatz SH (1998) Collective dynamics of 'small-world' networks. Nature 393(6684):440–442. ISSN 00280836

Wilson RC, Zhu P (2008) A study of graph spectra for comparing graphs and trees. J Pattern Recognit 41(9):2833–2841

Wong W.-K., Moore A, Cooper G, Wagner M (2005) What's strange about recent events (wsare): an algorithm for the early detection of disease outbreaks. J Mach Learn Res 6:1961–1998. ISSN 1532–4435

Wu B, Goel V, Davison BD (2006) Propagating trust and distrust to demote web spam. In: Proceedings of the workshop models of trust for the Web (MTW) at the 15th international World Wide Web Conference (WWW), Edinburgh, Scotland, volume 190 of CEUR workshop proceedings

Wu R-S, Ou C-S, Lin HY, Chang S-I, Yen DC (2012) Using data mining technique to enhance tax evasion detection performance. Expert Syst Appl 39(10):8769–8777

Xie S, Wang G, Lin S, Yu PS (2012) Review spam detection via temporal pattern discovery. In: Proceedings of the 18th ACM international conference on knowledge discovery and data mining (SIGKDD), Beijing, China, pp 823–831

Xu X, Yuruk N, Feng Z, Schweiger TAJ (2007) Scan: a structural clustering algorithm for networks. In: Proceedings of the 13th ACM international conference on knowledge discovery and data mining (SIGKDD), San Jose, CA. ACM, pp 824–833

Yedidia JS, Freeman WT, Weiss Y (2003) Understanding belief propagation and its generalizations. In: Exploring AI in the new millennium. Morgan Kaufmann Publishers Inc, pp 239–269

Zager L, Verghese G (2008) Graph similarity scoring and matching. Appl Math Lett 21(1):86–94

Zhao P, Han J, Sun Y (2009) P-rank: a comprehensive structural similarity measure over information networks. In: Proceedings of the 18th ACM conference on information and knowledge management (CIKM), Hong Kong, China. ACM, pp 553–562

Zhu B, Sastry S (2011) Revisit dynamic arima based anomaly detection. In: International conference on privacy, security, risk and trust (Social-Com/PASSAT), pp 1263–1268

Zimek A, Schubert E, Kriegel H-P (2012) A survey on unsupervised outlier detection in high-dimensional numerical data. Stat Anal Data Min 5(5):363–387

Zimek A, Campello RJGB, Sander J (2014) Ensembles for unsupervised outlier detection: challenges and research questions. A position paper. SIGKDD Explor Newsl 15(1):11–22