Anomaly Mining - Past, Present and Future *

Leman Akoglu

Carnegie Mellon University, Heinz College
lakoglu@andrew.cmu.edu

Abstract

Anomaly mining is an important problem that finds numerous applications in various real world domains such as environmental monitoring, cybersecurity, finance, healthcare and medicine, to name a few. In this article, I focus on two areas, (1) point-cloud and (2) graph-based anomaly mining. I aim to present a broad view of each area, and discuss classes of main research problems, recent trends and future directions. I conclude with key take-aways and overarching open problems.

Disclaimer. I try to provide an overview of past and recent trends in two areas within 4 pages. Undoubtedly, these are my personal view of the trends, which can be organized differently. For brevity, I omit all technical details and refer to corresponding papers. Again, due to space limit, it is not possible to include all (even most relevant) references, but a few representative examples.

1 Point-cloud Anomaly Mining

Point-cloud data consists of points that reside in a feature space, each of which can be seen as a d dimensional vector. Anomalous points are typically referred to as outliers, and in this section I will adopt this terminology.

Outlier mining has a very large literature, where most attention has been given to outlier detection (OD) under various settings [Aggarwal, 2013]. There exist a large pool of detectors that are distance-based, density-based, statistical-, cluster-, angle-, and depth-based, among many others [Chandola *et al.*, 2009]. Most detection models assume outliers to be scattered isolate points, while some specifically aim to detect collective outliers that can be seen as micro-clusters [Han *et al.*, 2012]. Another class of detectors target contextual outliers, which stand out within a specific context [Liang and Parthasarathy, 2016; Macha *et al.*, 2018]. These can also be seen as conditional outliers [Song *et al.*, 2007]. In addition, dynamic/streaming point-cloud OD has been studied at large [Gupta *et al.*, 2013] as outliers may often arise in settings where data is collected and monitored over time.

In the rest of this section, I discuss some of the trending classes of problems in outlier mining, organized into four lines of work as (1) user-centric OD, (2) deep learning based OD, (3) automating OD, and (4) fairness-aware OD.

1.1 User-centric Outlier Detection

User-centric outlier mining comprises two related topics: (i) explanations, and (ii) human-in-the-loop detection (HILD). Explaining the detected anomalies is crucial for settings in which outliers need to be vetted by human analysts. The purpose of vetting could be root-cause analysis/troubleshooting or sanity-checking/justification. An example to the former scenario is when the analyst identifies faults or inefficiencies in a production line or data center through OD and aims to *fix* the issues generating these outliers. The aim for the latter scenario is to distinguish statistical outliers from domain-relevant ones, where e.g. in claims auditing, not all outliers are necessarily associated with fraud. Related, HILD aims to leverage human feedback for sieving mere statistical outliers out of domain-relevant ones to eliminate false positives and thereby increase detection rate. These two problems are intertwined, since explanations could be presented to human analysts for acquiring effective feedback during HILD.

Although the vast body of work on outlier explanations is recent, the earliest example dates back several decades [Knorr and Ng, 1999], which provided what is called "intensional knowledge" by identifying minimal subspaces in which outliers stand out. Most existing work in this area are discriminative, since explanation proceeds detection that outputs (outlier/inlier) labels, and aim to identify subspaces that well-separate the outliers from the inliers [Dang *et al.*, 2014; Kuo and Davidson, 2016; Liu *et al.*, 2018]. While these have focused on providing a separate explanation for each outlier, others aim to provide explanations for *groups* of outliers [Macha and Akoglu, 2018; Gupta *et al.*, 2018] with the intent to reduce information overload on the analyst.

On the other hand, interactive OD mainly aims to leverage the (ground-truth) labels provided by a human-analyst during an auditing process to maximize the total number of true anomalies shown within a given auditing budget [Das *et al.*, 2016]. In addition to detection precision, others also factor human-effort in the overall objective [Lamba and Akoglu, 2019; Chai *et al.*, 2020].

Some of the remaining challenges in user-centric OD in-

---

*A position paper for IJCAI 2021 Early Career Spotlight Talk

clude causal explanations and novel paradigms for human-in-the-loop OD. Specifically, when the purpose is putting the explanations into action, e.g. fixing the production line, explanations should reflect a causal relation with the outcome, whereas work in this area is currently quite limited [Janzing *et al.*, 2019]. As for interactive OD, ideas at the intersection of OD and crowdsourcing could further be explored to design effective ways of leveraging human input.

1.2 Deep Learning for Outlier Detection

With the recent advent of deep neural networks (NNs), deep learning based OD has attracted attention. Main lines of work include reconstruction-based techniques (e.g. via autoencoding or generative adversarial networks), one-class classification based training, or simply using deep NNs for representation learning that is better suitable for OD. Some approaches also leverage synthesized pseudo-outlier points or estimate what-is-called pseudo-labels for existing points. I refer to one recent survey for extensive details [Pang *et al.*, 2021].

Deep NNs have real prowess for many learning problems, especially those in NLP and computer vision. However, a major obstacle to achieving similar ground-breaking performance on OD for such models appears to be model selection. Deep NNs exhibit several hyperparameters (e.g. depth, width, learning rate, drop-out rate, weight decay, number of training epochs, etc.). In the face of lack of any labels, it remains a big hurdle to effectively setting them for unsupervised OD tasks.

1.3 Automating Outlier Detection

Provided a large body of detection models exists, each with their own assumptions, a key question is which model would be most suited for a given new task. Moreover, most outlier detectors exhibit (hyper)parameters (HPs) to be set by users, e.g. number of nearest neighbors (NNs) k for NN-based methods. It is understood that most detectors are quite sensitive to the choice of their HPs [Campos *et al.*, 2016]. This is perhaps especially the case for deep NN based OD models. As such, the problem of selecting an algorithm (i.e. detection model) as well as its associated HPs is a critical one.

There has been some work on outlier model selection in recent years. AutoOD [Li *et al.*, 2020] focuses specifically on a deep autoencoder based detection model and automatically searches for an optimal neural network architecture within a predefined search space. Here they assume the availability of some labeled data that is used for validation. MetaOD [Zhao *et al.*, 2020b] employs meta-learning, transferring knowledge from historical OD tasks in the form of various models' performances to selecting a model for a given new task *without requiring any labels*.

At large, automating outlier model selection, especially for unsupervised settings, remains to be a vastly understudied, yet extremely important area.

1.4 Fairness-aware Outlier Detection

Fair data mining and OD are close cousins, as it is exactly the goal of OD to spot rare, minority samples in the data. However statistical minorities, including those associated with certain societal minorities (e.g. Hispanic), do not always reflect positive-class membership (e.g. fraud). Therefore, if

care is not taken when employing OD on data involving human subjects, OD can produce unjust outcomes.

The outlier mining community has recently routed attention to fairness-aware detection. One of the earliest work [Davidson and Ravi, 2020] focused on quantifying the fairness of any OD model's output post hoc (i.e., proceeding detection). FairLOF [Deepak and Abraham, 2020] aimed to incorporate fairness specifically to the LOF algorithm. DeepFairSVDD [Zhang and Davidson, 2021] employed adversarial training, besides the SVDD objective, to obfuscate protected group memberships. All of these aim to achieve statistical (a.k.a. group or demographic) parity (SP) as the notion of fairness. In our recent work [Shekhar *et al.*, 2021], we discussed potential sources and implications of bias in OD outcomes, and designed FairOD targeting additional fairness criteria for OD besides SP, such as treatment parity and equality of opportunity.

One of the key challenges in fair OD is the absence of any ground-truth outlier labels. Moreover, which notions and measures of fairness are suitable for OD, or the compatibility thereof, is not well understood.

2 Graph-based Anomaly Mining

Graphs are powerful representations of relational data in which entities (nodes) are linked through relationships (edges). They may exhibit various properties; nodes can be associated with labels or (multi-) attributes, edges can be directed, weighted, and have multiplicities, types, timestamps, etc. For example, consider an accounting graph database in which each graph depicts an itemized transaction; where nodes correspond to accounts, node labels depict account types (cash, revenue, etc.), edge multiplicities and weights respectively reflect the number of items and dollar amount involved, edge directions depict debit/credit flow, and finally graph-level attributes denote auxiliary information about the entire graph, such as the time, approver, etc. Alternately, consider a single bipartite graph depicting user–product review relations, in which both users and products exhibit multi-attributes (resp., e.g. username, location, etc. and brand, price, etc.), and edges have timestamps and are associated with a rating, and review text. These examples are to suggest that graphs in the real world can be quite complex.

The representational complexity (or expressiveness) of real world graphs has driven two key challenges for graph anomaly detection (GAD): (1) defining what constitutes a graph anomaly, and (2) designing general-purpose GAD solutions that can seamlessly handle graphs with any (subset of) properties without much/any modification. In the following, I discuss prevalent trends on graph anomaly detection problems, as well as graph neural network based techniques toward automated representation learning for complex graphs.

2.1 Defining Graph Anomalies

Graph anomalies can be organized into three main classes of problems: (i) node/edge-level, (ii) subgraph-level, and (iii) graph-level anomaly detection. Within each class, the definition of anomaly varies, mainly driven by two factors; the application and the available graph properties. I refer to our

survey [Akoglu *et al.*, 2015] for an outline of detection problems for static and dynamic, as well as plain and attributed graphs. I aim to give an overview of prevalent definitions and key approaches as follows.

Node/edge-level anomaly detection aims to identify rare or unusual nodes in a single input graph. *Feature-based* approaches [Akoglu *et al.*, 2010] "flatten" the graph into individual node vectors, effectively defining anomalous nodes as point-cloud outliers in this feature space. *Proximity-based* approaches define anomalies in relation to others, where the assumption is that anomalous nodes are associated with/close-by to other anomalous nodes. This is typically applied to fraud detection problems, where fraudulent users/accounts/etc. can be found nearby in the graph [Rayana and Akoglu, 2015]. Note that in this case, the definition is motivated by the application. Another category of approaches is *community-based*, which define anomalies as "misfits" with respect to the community structure. Those could be anomalous (bridge) edges between communities [Tong and Lin, 2011], or community outliers that are within well-structured communities and yet, do not exhibit similar node attributes to others in the community [Gao *et al.*, 2010; Perozzi *et al.*, 2014] (akin to contextual outliers in point-cloud OD). Note that in the latter case, the definition is informed by the availability of node attributes.

Subgraph-level anomaly detection aims to identify unusual *groups* of nodes in a single input graph. These are akin to collective outliers in point-cloud OD. *Density-based* approaches define subgraph anomalies as those with significantly higher density than the overall graph [Hooi *et al.*, 2016]. Provided edge timestamps, definition is also extended to those with high-density and also temporally-synchronized edges [Beutel *et al.*, 2013]. *Community-based* approaches leverage node attributes to define anomalous subgraphs as those communities with poor structural connectivity and/or incoherent attribute values [Perozzi and Akoglu, 2016].

Finally, **graph-level** anomaly detection aims to identify unusual graphs within a (unordered) *set* or a (often temporal) *sequence* of graphs. Most dominant category of approaches is *feature/similarity-based*, which either "flatten" the entire graph into a single feature vector, or quantify pairwise similarity between graphs through an appropriately designed graph kernel [Manzoor *et al.*, 2016]. These effectively treat anomalous graphs as point-cloud outliers in an explicit or implicit feature space. *Subgraph-based* approaches specifically define anomalous graphs as those with (the (dis)appearance of) dense subgraphs [Eswaran *et al.*, 2018] or with (the emergence of) "hotspots" [Hooi *et al.*, 2018], depending on whether anomalies are sought within a static set of graphs or time-evolving series of graphs. While the former is defined solely topologically based on edge density, the latter definition is driven by node-level signals (i.e. continuous attributes) where hotspots are defined as those groups of close-by nodes in the graph whose signal values are or have become abnormal. Another classic type of approaches is *compression-based*, which define anomalous graphs as those that cannot be compressed efficiently based on a data-driven graph encoding mechanism [Noble and Cook, 2003]. One such mechanism is graph cover by motifs (or graphlets, substructures),

wherein graphs that consist of frequent motifs can be encoded using relatively fewer bits, which is considered as the anomaly score. Note that motif-based encoding can be seen as a subgraph-based approach, discussed previously. The key distinction is that subgraph-based approaches define anomalies in terms of specific subgraphs (e.g. hotspots), whereas compression-based approaches quantify anomalies implicitly in a data-driven fashion; a graph is deemed more anomalous the less it exhibits frequent motifs (here, rare motifs can be any complement, rather than specifically defined).

A key distinction of GAD from point-cloud OD is the abundance of anomaly definitions. I tried to overview these in a two-level "catalog" here (e.g. node-level, feature-based). It would be interesting to identify other novel definitions and extend this existing catalog, ideally either based on a key driving application or otherwise with a strong justification for real-world applicability. A key challenge in this area appears to be a general-purpose, unified GAD framework that can admit graphs with any (subset of) properties. Existing methods often tend to be strongly tied to graphs with certain properties, perhaps mainly because the anomalies they aim to detect are also defined based on the type of input graph they admit. For example, an approach based on identifying dense-subgraphs in plain graphs [Beutel *et al.*, 2013] does not trivially apply to multi-attributed nodes.

2.2 Representation Learning and Deep Graph Anomaly Detection

As mentioned earlier, feature/similarity-based approaches to GAD is one of the most prevalent. To this end, a large body of work exists on graph feature extraction, graph similarity, and graph kernels. However, these are often confined to certain types of graphs they can admit and do not apply broadly (See e.g. Table 1 in [Kriege *et al.*, 2020]).

The most recent trend is graph representation learning or embedding through graph neural networks (GNNs). GNNs are appealing for GAD since they automatically learn vector representations at node- and graph-level, can be trained end-to-end provided suitable objective functions, and can readily handle graphs with various properties such as labeled/multi-attributed nodes and directed, weighted edges. As such, unsupervised embedding techniques provide a simple way for mapping a GAD problem to a point-cloud OD one.

In addition, there have been recent work on purposing GNNs specifically for GAD. The main directions include end-to-end representation learning guided by auto-encoding [Ding *et al.*, 2019] or one-class classification [Wang *et al.*, 2021], as well as defining new loss functions suitable for GAD [Zhao *et al.*, 2020a].

I have listed several advantages of GNNs above. On the other hand, there are several challenges with effectively using GNNs for GAD. First, end-to-end training necessitates suitable loss functions, which limits the solution to a few existing loss function families for GAD. Second, learned representations are hard to interpret, compared to e.g. hand-crafted features or motifs which may be more intuitive. Third, GNNs may be more complicated and slow to train. Last and perhaps most importantly, like their NN counterparts, they have a long list of (hyper)parameters. A big question is how to

tune them in the absence of labels. There is recent work on using meta-learning for effective graph anomaly detection assuming a few labels exist [Ding *et al.*, 2021]. Future work on automating model selection could unlock the true potential of GNNs for GAD.

3 Conclusions and Discussion

In this article, I aimed to present a broad overview of the past, recently trending topics and potential future directions for both point-cloud outlier detection (OD) and graph anomaly detection (GAD). In the next couple of paragraphs, I give a short summary, followed by a discussion on open problems in anomaly mining at large.

Outlier definitions seem to have settled for point-cloud OD; as global, local/contextual, and collective outliers. Great progress has also been made in terms of designing detection models, for data that is streaming, high-dimensional, distributed, etc. Recent trends include different detection paradigms, such as fairness-aware, explainable, and human-in-the-loop detection. Recent advances in deep neural networks (NNs) have also been carried over to this area for deep OD. Specifically these recent deep NN models that are heavily (hyper)parameter (HP) laden, and more broadly the plethora of available detection models, have motivated automating OD, that is, both algorithm (i.e. detector) as well as HP selection using only a few labels or no labels at all.

On the other hand, and in contrast to point-cloud OD, anomaly definitions for GAD vary widely. This is partly driven by the abundant different real-world applications to which GAD is relevant, as well as the variety of additional properties a graph can be decorated with, such as edge multiplicities, node attributes, etc. Moreover, while some definitions directly reflect the known types of anomalies (e.g., emergence or existence of dense subgraphs), others look for unknown anomalies as those outside the observed patterns (e.g. frequent motifs). This implies that the way graph patterns are defined has direct implications for GAD. Moreover, the better pattern mining would imply better anomaly detection. Deep learning, and specifically (G)NNs, is also a recent trend for GAD, for which suitable loss functions and effective model selection –esp. using no labels– are key problems.

I conclude with discussing a couple of overarching open problems for anomaly mining. The first is on **evaluation and benchmark datasets**. It is most common practice for the community to re-purpose (point or graph) classification datasets, by down/sub-sampling one or more of the classes to depict the "ground-truth" outliers [Emmott *et al.*, 2015]. This avoids having to inject synthetic anomalies, whose generation may be prone to internal biases of how certain detection models work. On the other hand, this procedure raises other concerns such as whether it creates suitable settings that align with what is being evaluated as well as human intuition, and whether it truly represents the nature of anomalies that arise in real world applications [Campos *et al.*, 2016]. Certainly, rigorous attention to designing better benchmark datasets for evaluation would help reveal the strengths and limitations of various detection models, and ultimately keep a fair and accurate record of progress [Wu and Keogh, 2020]. A promis-

ing way here is the industry putting out real world detection tasks, perhaps in the form of competitions with a leaderboard, where the data may contain some identified (but perhaps not all) true anomalies. Understandably such data may be sensitive for public domain, where privacy-preserving data release practices could be employed. Specifically for OD, distance-preserving data sketches or a pairwise-distance oracle (instead of explicit feature representations) could enable ground for data exchange.

Another overarching open challenge is adaptive **detection under adversarial settings**. Anomaly mining is used more often than not for applications involving fraud (e.g. healthcare, advertisement, tax), intrusion, etc. where intelligent attackers continuously adapt their behavior to evade detection. There exist some work on detecting specific type of (dense subgraph injection) small-scale (called "stealth") and camouflaged anomalies [Shah *et al.*, 2014; Hooi *et al.*, 2016], although these do not model the dynamic behavior of attackers. In domains such as banking and cybersecurity, models in place are observed to stop detecting any anomalies soon after (within hours) of being deployed. This calls for fast-adaptive or otherwise adversarially-robust detection algorithms.

Acknowledgments

References

[Aggarwal, 2013] Charu C. Aggarwal. *Outlier Analysis*. Springer, 2013.

[Akoglu *et al.*, 2010] Leman Akoglu, Mary McGlohon, and Christos Faloutsos. Oddball: Spotting anomalies in weighted graphs. In *PAKDD*, pages 410–421, 2010.

[Akoglu *et al.*, 2015] Leman Akoglu, Hanghang Tong, and Danai Koutra. Graph based anomaly detection and description: a survey. *ACM DAMI*, 29(3):626–688, 2015.

[Beutel *et al.*, 2013] Alex Beutel, Wanhong Xu, Venkatesan Guruswami, Christopher Palow, and Christos Faloutsos. Copycatch: stopping group attacks by spotting lockstep behavior in social networks. In *WWW*, pages 119–130, 2013.

[Campos *et al.*, 2016] Guilherme O. Campos, Arthur Zimek, Jörg Sander, Ricardo J. G. B. Campello, Barbora Micenková, Erich Schubert, Ira Assent, and Michael E. Houle. On the evaluation of unsupervised outlier detection: measures, datasets, and an empirical study. *ACM DAMI*, 30(4):891–927, July 2016.

[Chai *et al.*, 2020] Chengliang Chai, Lei Cao, Guoliang Li, Jian Li, Yuyu Luo, and Samuel Madden. Human-in-the-loop outlier detection. In *SIGMOD*, pages 19–33, 2020.

[Chandola *et al.*, 2009] Varun Chandola, Arindam Banerjee, and Vipin Kumar. Anomaly detection: A survey. *ACM computing surveys (CSUR)*, 41(3):1–58, 2009.

[Dang *et al.*, 2014] Xuan-Hong Dang, Ira Assent, Raymond Ng, Arthur Zimek, and Erich Schubert. Discriminative features for identifying and interpreting outliers. In *ICDE*, 2014.

[Das *et al.*, 2016] Shubhomoy Das, Weng-Keen Wong, Thomas G. Dietterich, Alan Fern, and Andrew Emmott. Incorporating expert feedback into active anomaly discovery. In *ICDM*, pages 853–858, 2016.

[Davidson and Ravi, 2020] Ian Davidson and S. S. Ravi. A framework for determining the fairness of outlier detection. In *ECAI*, volume 325, pages 2465–2472, 2020.

[Deepak and Abraham, 2020] P Deepak and Savitha Sam Abraham. Fair outlier detection. In *WISE*, pages 447–462, 2020.

[Ding *et al.*, 2019] Kaize Ding, Jundong Li, Rohit Bhanushali, and Huan Liu. Deep anomaly detection on attributed networks. In *SDM*, pages 594–602. SIAM, 2019.

[Ding *et al.*, 2021] Kaize Ding, Qinghai Zhou, Hanghang Tong, and Huan Liu. Few-shot network anomaly detection via cross-network meta-learning. *arXiv:2102.11165*, 2021.

[Emmott *et al.*, 2015] Andrew Emmott, Shubhomoy Das, Thomas Dietterich, Alan Fern, and Weng-Keen Wong. A meta-analysis of the anomaly detection problem. *arXiv:1503.01158*, 2015.

[Eswaran *et al.*, 2018] Dhivya Eswaran, Christos Faloutsos, Sudipto Guha, and Nina Mishra. Spotlight: Detecting anomalies in streaming graphs. In *KDD*, pages 1378–1386, 2018.

[Gao *et al.*, 2010] Jing Gao, Feng Liang, Wei Fan, Chi Wang, Yizhou Sun, and Jiawei Han. On community outliers and their efficient detection in information networks. In *KDD*, 2010.

[Gupta *et al.*, 2013] Manish Gupta, Jing Gao, Charu C Aggarwal, and Jiawei Han. Outlier detection for temporal data: A survey. *TKDE*, 26(9):2250–2267, 2013.

[Gupta *et al.*, 2018] Nikhil Gupta, Dhivya Eswaran, Neil Shah, Leman Akoglu, and Christos Faloutsos. Beyond outlier detection: Lookout for pictorial explanation. In *ECML PKDD*, 2018.

[Han *et al.*, 2012] Jiawei Han, Micheline Kamber, and Jian Pei. Outlier detection. *Data mining: Concepts and Techniques*, pages 543–584, 2012.

[Hooi *et al.*, 2016] Bryan Hooi, Hyun Ah Song, Alex Beutel, Neil Shah, Kijung Shin, and Christos Faloutsos. Fraudar: Bounding graph fraud in the face of camouflage. In *KDD*, 2016.

[Hooi *et al.*, 2018] Bryan Hooi, Leman Akoglu, Dhivya Eswaran, Amritanshu Pandey, Marko Jereminov, Larry Pileggi, and Christos Faloutsos. Changedar: Online localized change detection for sensor data on a graph. In *CIKM*, pages 507–516, 2018.

[Janzing *et al.*, 2019] Dominik Janzing, Kailash Budhathoki, Lenon Minorics, and Patrick Blöbaum. Causal structure based root cause analysis of outliers. *arXiv:1912.02724*, 2019.

[Knorr and Ng, 1999] Edwin M. Knorr and Raymond T. Ng. Finding intensional knowledge of distance-based outliers. In *VLDB*, pages 211–222, 1999.

[Kriege *et al.*, 2020] Nils M Kriege, Fredrik D Johansson, and Christopher Morris. A survey on graph kernels. *Applied Network Science*, 5(1):1–42, 2020.

[Kuo and Davidson, 2016] Chia-Tung Kuo and Ian Davidson. A framework for outlier description using constraint programming. In *AAAI*, pages 1237–1243, 2016.

[Lamba and Akoglu, 2019] Hemank Lamba and Leman Akoglu. Learning on-the-job to re-rank anomalies from top-1 feedback. In *SDM*, pages 612–620. SIAM, 2019.

[Li *et al.*, 2020] Yuening Li, Zhengzhang Chen, Daochen Zha, Kaixiong Zhou, Haifeng Jin, Haifeng Chen, and Xia Hu. AutoOD: Automated outlier detection via curiosity-guided search and self-imitation learning. *arXiv:2006.11321*, 2020.

[Liang and Parthasarathy, 2016] Jiongqian Liang and Srinivasan Parthasarathy. Robust contextual outlier detection: Where context meets sparsity. In *CIKM*, pages 2167–2172, 2016.

[Liu *et al.*, 2018] Ninghao Liu, Donghwa Shin, and Xia Hu. Contextual outlier interpretation. In *IJCAI*, pages 2461–2467, 2018.

[Macha and Akoglu, 2018] Meghanath Macha and Leman Akoglu. Explaining anomalies in groups with characterizing subspace rules. *ACM DAMI*, 32(5):1444–1480, 2018.

[Macha *et al.*, 2018] Meghanath Macha, Deepak Pai, and Leman Akoglu. Contextual outlier detection with multiple contexts: Application to ad fraud. In *ECML/PKDD*, pages 139–156, 2018.

[Manzoor *et al.*, 2016] Emaad Manzoor, Sadegh Milajerdi, and Leman Akoglu. Fast memory-efficient anomaly detection in streaming heterogeneous graphs. In *KDD*, pages 1035–1044, 2016.

[Noble and Cook, 2003] Caleb C Noble and Diane J Cook. Graph-based anomaly detection. In *KDD*, pages 631–636, 2003.

[Pang *et al.*, 2021] Guansong Pang, Chunhua Shen, Longbing Cao, and Anton Van Den Hengel. Deep learning for anomaly detection. *ACM Computing Surveys*, 54(2):1–38, March 2021.

[Perozzi and Akoglu, 2016] Bryan Perozzi and Leman Akoglu. Scalable anomaly ranking of attributed neighborhoods. In *SDM*, pages 207–215. SIAM, 2016.

[Perozzi *et al.*, 2014] Bryan Perozzi, Leman Akoglu, Patricia Iglesias Sánchez, and Emmanuel Müller. Focused clustering and outlier detection in large attributed graphs. In *KDD*, pages 1346–1355, 2014.

[Rayana and Akoglu, 2015] Shebuti Rayana and Leman Akoglu. Collective opinion spam detection: Bridging review networks and metadata. In *KDD*, pages 985–994, 2015.

[Shah *et al.*, 2014] Neil Shah, Alex Beutel, Brian Gallagher, and Christos Faloutsos. Spotting suspicious link behavior with fbox: An adversarial perspective. In *ICDM*, pages 959–964, 2014.

[Shekhar *et al.*, 2021] Shubhranshu Shekhar, Neil Shah, and Leman Akoglu. Fairod: Fairness-aware outlier detection. In *AAAI/ACM AIES*, 2021.

[Song *et al.*, 2007] Xiuyao Song, Mingxi Wu, Christopher Jermaine, and Sanjay Ranka. Conditional anomaly detection. *IEEE TKDE*, 19(5):631–645, 2007.

[Tong and Lin, 2011] Hanghang Tong and Ching-Yung Lin. Non-negative residual matrix factorization with application to graph anomaly detection. In *SDM*, pages 143–153, 2011.

[Wang *et al.*, 2021] Xuhong Wang, Baihong Jin, Ying Du, Ping Cui, Yingshui Tan, and Yupu Yang. One-class graph neural networks for anomaly detection in attributed networks. *Neural Computing and Applications*, pages 1–13, 2021.

[Wu and Keogh, 2020] Renjie Wu and Eamonn J Keogh. Current time series anomaly detection benchmarks are flawed and are creating the illusion of progress. *arXiv:2009.13807*, 2020.

[Zhang and Davidson, 2021] Hongjing Zhang and Ian Davidson. Towards fair deep anomaly detection. In *FAccT*, pages 138–148, 2021.

[Zhao *et al.*, 2020a] Tong Zhao, Chuchen Deng, Kaifeng Yu, Tianwen Jiang, Daheng Wang, and Meng Jiang. Error-bounded graph anomaly loss for gnns. In *CIKM*, pages 1873–1882, 2020.

[Zhao *et al.*, 2020b] Yue Zhao, Ryan A Rossi, and Leman Akoglu. Automating outlier detection via meta-learning. *arXiv:2009.10606*, 2020.