A community-aware approach for identifying node anomalies in complex networks

Helling, T.J.; Scholtes, J.C.; Takes, F.W.

Published in:
Complex Networks and Their Applications VII

DOI:
10.1007/978-3-030-05411-3_20

A community-aware approach for identifying node anomalies in complex networks

Thomas J. Helling[1], Johannes C. Scholtes[2], and Frank W. Takes[1,3]

[1] Department of Computer Science (LIACS), Leiden University
t.j.helling@umail.leidenuniv.nl
[2] Department of Data Science and Knowledge Engineering, Maastricht University,
j.scholtes@maastrichtuniversity.nl
[3] CORPNET, University of Amsterdam, takes@uva.nl

Abstract. The overwhelming amount of network data that is nowadays available, leads to an increased demand for techniques that automatically identify anomalous nodes. Examples are network intruders in physical networks or spammers spreading unwanted advertisements in online social networks. Existing methods typically identify network anomalies from a local perspective, only considering metrics related to a node and connections in its direct neighborhood. However, such methods often miss anomalies as they overlook crucial distortions of the network structure that are only visible at the macro level. To solve these problems, in this paper, the CADA algorithm is proposed, which identifies irregular nodes from a global perspective. It does so by measuring the extent to which a node connects to many different communities while not obviously belonging to one community itself. Results on synthetic and real-world data show that the incorporation of the community aspect is of critical importance, as our algorithm significantly outperforms previously suggested techniques. In addition, it scales well to larger networks of hundreds of thousands of nodes and millions of links. Moreover, the proposed method is parameter-free, enabling the hassle-free identification of anomalies in a wide variety of application domains.

Keywords: anomaly detection in networks, node anomalies, LFR benchmark, community detection

1 Introduction

An important problem in a large number of complex systems is identifying phenomena that diverge from what is considered to be normal. Doing this automatically based on data, is often referred to as anomaly detection [6]. Due to the rapidly growing amount of data that is nowadays available, methods to automatically identify data points that do not conform to expected behavior have seen increased interest in a multitude of applications, such as fraud detection [25] and intrusion detection [7].

In network data, anomaly detection deals with identifying nodes that show anomalous behavior. Most network-based anomaly detection techniques do not

label a node as anomalous, but rather assign an anomaly score to each node as an indicator of its deviance. Since the boundary of what should be considered anomalous can often not be determined in a straightforward manner, a domain expert is often involved in the final steps of the process of anomaly detection [13]. As a result, the focus of automated methods, such as the ones discussed in this paper, is to make a good initial selection of which nodes are likely anomalous.

Real-world networks obey many non-random statistical properties, such as heterogeneity of the degree distribution and the natural existence of communities; groups of nodes that are closely connected to each other [3]. This makes determining which nodes of the network are anomalies a far from trivial task. Although different types of network anomalies exist, we focus on the node anomaly. One example of the node anomaly is the node ego-network (i.e., a node and the connections between these nodes) with a star or near-star structure [1, 14].

Nodes with star structure are nodes that connect to many nodes that are not connected to each other. Such nodes are often referred to as nodes that are, roughly speaking, "not aware of the global structure" as they connect to seemingly random nodes in the network rather than closeby neighbors; they are community-agnostic. Examples of such anomalies are spammers in e-mail networks or advertisement calls in communication networks [14]. The majority of proposed anomaly detection methods approach the problem from a local perspective, not in the last place to keep methods scalable, as global methods typically take more computation time. For example, in [1] the OddBall algorithm was introduced, which proposes to fit power laws on a variety of statistics of the ego-network and then labels nodes deviating from these distributions as anomalous. However, as discussed above, connections made by anomalous nodes are typically not local, leaving a number of anomalous nodes undetected.

To solve this problem, we propose Community-Aware Detection of Anomalies (CADA), a global, community-aware and parameter-free algorithm to detect node anomalies in large real-world networks. We demonstrate with both synthetic and large-scale real-world complex network data that it effectively and efficiently identifies anomalies, outperforming previously proposed approaches.

The remainder of this paper is structured as follows. In Section 2 we provide necessary definitions and a formal problem statement. Next, in Section 3 we discuss previous work related to network-based anomaly detection. In Section 4 the proposed approach is outlined. Then in Section 5 the synthetic and real-world data sets are described. In Section 6 a set of experiments is performed to evaluate the approach. Finally, Section 7 concludes the paper.

2 Preliminaries and problem statement

This section provides various definitions of concepts and techniques, leading to the problem statement addressed in this paper.

2.1 Network terminology

A network $G = (V, E)$ consists of a set of nodes V (also called vertices or objects) and a set of edges E (also called links or connections). Throughout this paper, the direction and weight of a link between two nodes are not taken into account, although the proposed approach can easily be extended to both directed and weighted networks. The total number of nodes $|V|$ and edges $|E|$ are referred to as n and m, respectively. The degree of a node is the number of edges connected to a node, and k is the average degree over all nodes in the network. The ego-network of a node is the subgraph consisting of the direct neighborhood of that node, including the node itself, and all edges between those nodes.

Over the years, a lot of properties of real-world networks have been discovered, e.g., a power law degree distribution, the small-world phenomenon [22] and a relatively high clustering coefficient [23]. Furthermore, it was found that in real-world systems groups of nodes tend to cluster together, forming so-called communities. The process of automatically identifying a division of a network into communities is called community detection [8]. For further details on this and other network science terminology, see [3].

2.2 Problem statement

In network anomaly detection, the goal is to identify anomalous behavior in the network. We focus on the *node anomaly*, defined as a node that based on its connections seems unaware of the global network structure, recognized, e.g., by a star-like ego-network. Although relatively easily defined, the automated detection of these anomalies is far from trivial.

The aim of this paper is to devise an algorithm to compute a score $f(v)$ for each node $v \in V$ that indicates to what extent node v is a node anomaly. Based on a ranking induced by these scores, we can identify the most anomalous nodes. In Section 4 we will discuss approaches to tackle this problem.

3 Related work

This section discusses related work on the topic of anomaly detection in networks. Typically, three types of anomalies in static networks can be distinguished: node anomalies, edge anomalies, and sub-graph anomalies [2, 4]. A variety of methods have been developed, ranging from approaches that incorporate structural network features to identify parts of the network that deviate from common distributions [1, 11], to proximity-based methods that measure the relevance of nodes by randomly walking through weighted and directed networks [10, 15, 17]. Other methods attempt to identify bridge nodes in networks by clustering nodes together by either using Personalized PageRank [20], or by capturing the underlying correlations of the network by decomposing the adjacency matrix into a low-rank approximation of which the residual matrix that can be used to indicate parts of the networks that do not correspond to the expected patterns [21].

Henceforth we focus on node anomalies, nodes that are not aware of the global network structure such as intruders, spammers, or telecommunication advertisement bureaus [1, 14]. The OddBall algorithm was one of the first to detect the node anomaly by (1) extracting the number of nodes and edges from the ego-network of each node, (2) identifying a pattern of normal behaviour by defining so-called power laws on these features, and (3) identifying points that deviate from these power laws. It was found that ego-networks with an equal number of nodes and edges is a network with a star-structure, were found to deviate from the derived power law [1]. Others attempted to improve the identification of nodes with star or clique structure by introducing alternate local features of the ego-network [9, 14], incorporating features from the extended neighborhoods into the model [9, 12], or by identifying so-called hubs, nodes that do not share a certain fraction of common neighbors with any of their adjacent nodes [24]. It should be noted that the node anomaly is considerably different from merely selecting high betweenness centrality nodes in the network, because this centrality measure does not differentiate between edges in or between different parts of the network, while node anomalies typically do [12].

In this work, we aim to move away from previously approached locally operating algorithms, and take the global structure of the network into account.

4 Approach

First two approaches that already focus on identifying the node anomaly are discussed in Section 4.1. Then, the proposed community-aware approach CADA is introduced in Section 4.2. Recall, that each method assigns an anomaly score to all nodes in the network, where a higher score indicates a higher divergence from expected behavior.

4.1 Existing approaches

OddBall [1] extracts the number of nodes N_i and edges E_i of the ego-network of each node $i \in V$, to exhibit normal patterns in the data set by fitting the Egonet Density Power Law in the form of $E_i \propto N_i^{\alpha}$. Ego-networks that deviate the most from the fitted power law are considered to be an anomaly. The OddBall anomaly score $ob(i)$ of a node i is computed as follows:

$$ob(i) = \frac{Cx_i^{\theta}}{y_i} \log(y_i - Cx_i^{\theta} + 1)$$

Here, Cx_i^{θ} is the expected number of edges with x_i number of nodes, while y_i is the true number of edges. Computing this metric can be done in $O(n \cdot k^2)$ time, checking for each of the n nodes the k^2 neighbor pairs in its ego network, where k is the node degree. Although this method has the advantage of taking the natural existence of power laws in real-world network data into account, one obvious shortcoming of this method is that solely the direct neighborhood of a node is taken into consideration.

Embed [12] is a network embedding approach that preserves the local linkage structure of nodes by assigning each node i to a r-dimensional vector X_i, where X_i^r describes the relationship between node i and region r under the minimization of the objective function O, which is defined as follows.

$$O = \sum_{(i,j)\in E} ||X_i - X_j||^2 + \alpha \sum_{(i,j)\notin E} (1 - ||X_i - X_j||)^2$$

Here, α is a balancing factor between existent edges and non-existent edges. To minimize the computational cost of the algorithm, the authors propose to (1) approximately represent O by removing α and equal the number of existent and non-existent edges to balance the optimization, (2) to initialize the vectors with equal embedding values if nodes belong to the same partition according to network partitioning method METIS, and (3) reduce the number of dimensions in vectors of the embedding to $k + \beta$, where k is the average degree and β a toleration factor for the number of regions node anomalies connect to. According to the authors, $\beta = k/4$ is sufficient. After successful minimization of the network embedding, we represent the correlation of node i with r regions as follows:

$$NB(i) = y_i^1, ...y_i^r = \sum_{(i,j)\in E} (1 - ||X_i - X_j||) \cdot X_j$$

The Embed anomaly score $em(i)$ of a node i can then be computed as follows.

$$em(i) = \sum_{j=1}^{r} \frac{y_i^j}{y_i^*}$$

Here, $y_i^* = \max(y_i^1, ..., y_i^r)$. Embed runs in $O(t \cdot m \cdot (k + \beta))$, where t is the iteration threshold of gradient descent, again with m the number of edges and k the average degree. A t of 50 is sufficient. However, a drawback of the Embed algorithm is that the results are dependent on the number of dimensions r, that are chosen, and as such the approach is not parameter-free.

4.2 Proposed approach: CADA

The methods, discussed in the previous subsection, mainly focus on identifying the node anomalies from a local perspective, or are not parameter free. Here, we use community detection to also take the global perspective into account. The proposed **Community-Aware Detection of Anomalies** algorithm consists of two steps.

First, CADA assigns each node to a particular community using an out-of-the-box community detection method [8]. In this paper, we employ two well-known community detection algorithms that both scale linearly in the number of edges and as such run in $O(m)$: the Louvain algorithm and the Infomap approach. Louvain [5] is a greedy optimization method that optimizes the so-called modularity function Q, a measure for the quality of a division of a network into communities. It is based on the relation between the number of edges that

connect nodes in the same community and the expected value for the same size of the network if the same number of edges are randomly distributed. Infomap [18, 19] is an information-theoretic approach that utilizes random walks to compress the information that is needed to minimally describe how information randomly flows through the network. We refer to CADA_L or cd_L when Louvain is used as community detection method, and to CADA_I or cd_I when Infomap is used as community detection method. Both assign each node to a community, and can handle undirected and directed networks (Louvain would ignore link direction), and can incorporate weights.

The second step of CADA is to assign an anomaly score to each node, based on the communities each node connects to. The anomaly score describes to what extent the neighbors of a node belong to a diverse number of communities, while the node itself does not strongly belong to one of them. Thus, for each node i, we create a vector g_i, where g_i^c represents the number of neighboring nodes that belong to community c. g_i^* represents the maximum number of neighboring nodes that belong to the same community. We can then compute an anomaly score for each node as follows:

$$cd(i) = \sum_{j=1}^{c} \frac{g_i^j}{g_i^*}$$

5 Data

A challenge in anomaly detection in networks is that there do not exist many publicly available labeled data sets with ground truth anomalies. Therefore, we illustrate the results of our anomaly detection methods on both real-world network data sets (see Section 5.1) and synthetic network data sets (as discussed in Section 5.2). Finally, the anomaly types are discussed in Section 5.3.

5.1 Real-world network data sets

Three different real-world data sets (see Table 1) are chosen to qualitatively assess the performance of the considered anomaly detection algorithms. *Douban* (http://socialcomputing.asu.edu/datasets/Douban) is a Chinese recommendation we- bsite where a link exists between two users if they had an explicit friendship connection. *Amazon* (http://snap.stanford.edu/data/amazon0601.html) is a co-purchasing network where a link exists between two articles if these products are frequently purchased together on Amazon. *DBLP* (http://projects.csail.

Table 1. Properties of the real-world network data sets.

Data set	Descripton	Number of nodes	Number of edges
Douban	Social network	154,907	327,162
Amazon	Co-purchase network	403,394	2,443,408
DBLP	Co-authorship network	1,412,414	5,947,085

mit.edu/dnd/DBLP) is a collaboration network based on co-authorship between mostly computer scientists, extracted from the popular DBLP listing website.

5.2 Synthetic network data sets

Synthetic networks are generated using the Lancichinetti-Fortunato-Radicchi (LFR) benchmark, which allows one to generate networks of different sizes [16]. The LFR benchmark is chosen because it creates networks adhering to real-world network properties, such as degree heterogeneity and community size heterogeneity. They are generated as follows:

1. A network of n nodes is generated where each node has a degree based on power law exponent γ_1 with $k_{\min}$ and $k_{\max}$ so that the average degree is k.
2. Community sizes are generated from the power law minus exponent γ_2 with community sizes $s_{\min}$ and $s_{\max}$ so that $s_{\min} > k_{\min}$ and $s_{\max} > k_{\max}$, and the sum of community sizes is n. If these are not chosen, the community sizes will be chosen close to the degree extremes.
3. Each node is assigned to a community, as long as the community does not exceed the set community size. If the number of edges within the community exceeds the community size, the node becomes homeless. Homeless nodes are randomly assigned to a community. If the community size is exceeded then a randomly selected node of that community becomes homeless. The process terminates if all communities are completed.
4. The algorithm rewires the edges so that each node has a fraction of approximately μ internal neighbors and $1 - \mu$ external neighbors.

We generate networks with a size ranging from $1,000$ to $500,000$ nodes, and mixing parameters between 0.1 and 0.6. To provide a fair performance comparison in the experiments, the parameters are chosen as in [12], as shown in Table 2.

5.3 Anomaly types

Following the approach suggested in [12] we employ two generative processes to insert anomalies in the synthetic networks, that reflect on the node anomaly defined in Section 2.2.

Table 2. Parameter settings for generating LFR benchmark networks.

Parameter	Description	Setting
n	The number of nodes	from 10^3 to $5 \cdot 10^5$
k	Average degree	$2 \cdot n^{1.15}/n$
k_{max}	Maximum degree	$n^{1/(t_1-1)}$
γ_1	minus exponent for degree distribution	3
γ_2	minus exponent for community size distribution	2
μ_t	Mixing parameter for topology	from 0.1 to 0.6

Random anomaly is inspired by the fact that infiltrating nodes are not aware of the global network structure and therefore connect to random nodes in the network. The anomalies are injected by adding $n/100$ nodes that connect to x random existing nodes, where x is between k and k_{max}. For each inserted node, the value of x is set by drawing a value from the the same power law degree distribution as that of the synthetic network.

Replaced anomaly first generates $n + a$ nodes with the LFR benchmark. The goal is to replace a nodes to obtain $n + n/100$ nodes. We randomly select x existing nodes in the network that have a degree lower than $2 \cdot k$. An anomaly is injected by rewiring all edges from the x nodes to the new anomaly. The x nodes are then removed from the network. x ranges from $2 - 21$, with an increment of 1, until $n + n/100$ nodes are obtained.

6 Experiments

In this section, we describe how and what experiments were executed to measure the performance of CADA.

6.1 Experimental setup

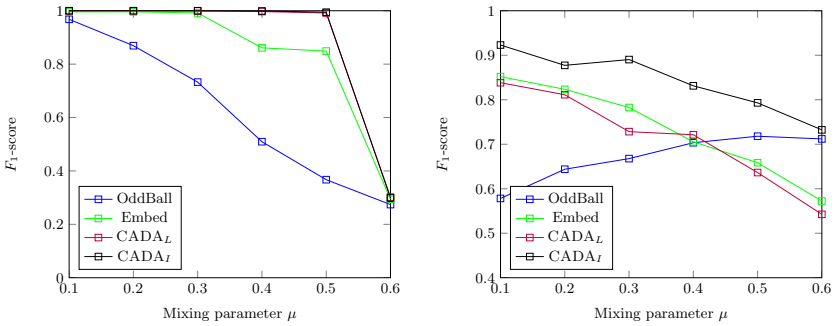
For the experiments, the parameters are set as follows. For Embed we set the number of dimensions to $\sqrt{n}$, to let the networks scale properly with the network size. Note that we flagged nodes anomalous if the anomaly score of the nodes exceeded a certain threshold Θ. To fairly evaluate and compare the algorithms, we obtained the k nodes that exceeded threshold Θ for CADA_I. Then, we extracted the top-k most anomalous nodes according to the other methods to fairly compare the most anomalous nodes. For the synthetic data sets, Θ was set to 4 to maximize performance with CADA_I. For the real-world data sets, Θ was set on 5, 5, and 9 for *Amazon*, *DBLP*, and *Douban*, respectively. Θ was chosen so that we at least covered $n/100$ nodes in the network. All reported are averages obtained from executing the experiments 10 times.

The experiments were run on a 2.8 GHz Intel Core i7 CPU with 16GB RAM, using a combination of Python and C++ algorithms. Python package *NetworkX* was used for network operations. CADA was implemented in Python and is available at https://github.com/thomashelling/cada.

6.2 Evaluation metrics

As quantitative evaluation of the algorithms on synthetic network data sets, we use the F_1-score, which is based on recall and precision. Recall is the fraction of true positives (discovered anomalies) divided by the total number of ground-truth anomalies, while precision is equal to the true positives divided by the number of nodes that are flagged anomalous. The F_1-score is then:

$$F_1 = 2 \cdot \frac{precision \times recall}{precision + recall}$$



(a) RandomAnomaly (b) ReplacedAnomaly

Fig. 1. F_1-score for different mixing parameter values for networks with 100,000 nodes.

To qualitatively compare between the different anomaly detection algorithms, we propose to measure the similarity in the 1% most anomalous nodes for each anomaly detection algorithm on the three real-world data sets. Note that this could be slightly more than 1% because nodes could have the same anomaly score, in which case we included all nodes with the same score as the node at the exact cutoff. For two sets of discovered node anomaly sets A_1 and A_2, we use the Jaccard similarity, defined as follows:

$$J(A_1, A_2) = \frac{|A_1 \cap A_2|}{|A_1 \cup A_2|}$$

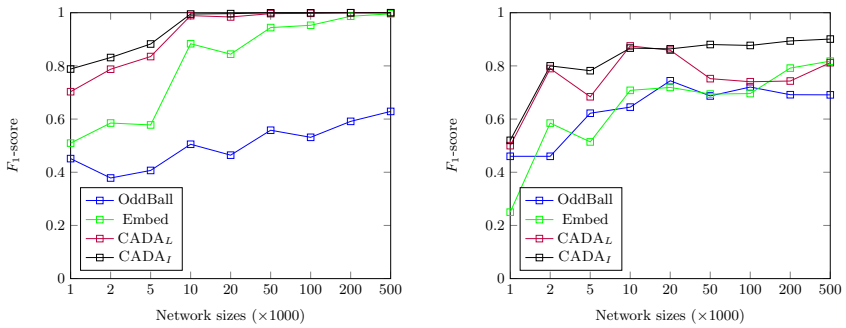
6.3 Results on synthetic data

Figure 1 shows the F_1-score of each of the methods for different values of the mixing parameter. For both anomaly types, the result is significantly higher in case of our CADA method. Embed outperforms OddBall, while OddBall is closing in on CADA and Embed once the community structure diminishes for ReplacedAnomaly. In general, results when using Infomap were higher than for Louvain, which is likely due to the resolution limit, which starts to play a larger role for higher values of the mixing parameter.

Figure 2 shows the F_1-score for a fixed value of 0.4 for the mixing parameter, but for varying sizes of the network. As the figures show, for RandomAnomaly, even for smaller size networks, CADA performs best, with Embed closing in, but only for networks larger than $200,000$ nodes. Although the difference is modest, for ReplacedAnomaly, CADA with Infomap consistently performs best.

6.4 Results on real-world data

Although on real-world data we cannot report F_1-scores, we can look at the agreement between the different methods, of which results in the form of Jaccard similarity are reported in Table 3.



(a) RandomAnomaly (b) ReplacedAnomaly

Fig. 2. F_1-score for different network sizes with fixed mixing parameter 0.4.

Zooming in on the *DBLP* dataset and the outliers found, two noteworthy findings were obtained. Two anomalies were solely identified by CADA with Infomap, but ignored by OddBall and Embed. First, we found authors that have published with many different authors, such as prof. dr. H. Vincent Poor, who was president of the IEEE Information Theory Society, and at the time of the data set has published with over 400 authors from all over the world. Second, many discovered anomalies were actually authors with the same name, such as 63 distinct authors named 'Wei Liu' that collaboratively published with over 1332 different authors. Other such authors were 'Jing Li', 'Yan Zhang', and 'Yu Zhang'. This illustrates that apart from outliers such as authors with extremely large number of publications, also errors in the underlying data can efficiently be identified using anomaly detection techniques.

6.5 Discussion

The experiments demonstrate that CADA is suitable for the purpose of identifying anomalous nodes, and appears to do so effectively on both synthetic and real-world data. Evaluation on synthetic networks with varying mixing parameters shows that the method performs consistently better compared to other methods to identify node anomalies. It reveals that methods that solely observe

Table 3. Jaccard similarity of the most anomalous nodes on *DBLP*, *Amazon*, and *Douban* (left to right). OddBall (ob), Embed (em), CADA Louvain (cd_L), and CADA Infomap (cd_I).

	em	cd_L	cd_I
ob	0.02	0.02	0.02
em	-	0.22	0.24
cd_L	-	-	0.24

	em	cd_L	cd_I
ob	0.11	0.11	0.17
em	-	0.17	0.20
cd_L	-	-	0.21

	em	cd_L	cd_I
ob	0.00	0.00	0.00
em	-	0.37	0.43
cd_L	-	-	0.34

the anomalies from a local perspective may possibly oversee the global community structure and therefore miss anomalous nodes, or require manual parameter tuning. Moreover, the community detection methods chosen scale linearly with the number of edges in the network, and therefore provide an efficient method to identify node anomalies [8]. One limitation of CADA, is that it is dependent on the community detection performance of the network, and unfortunately there is no universal method to detect communities most accurately in each network. This is illustrated in Figure 1(a) and Figure 1(b), where CADA_I consistently outperforms CADA_L. Moreover, Table 3 shows that the overlap between anomalous nodes for CADA_I and CADA_L in real-world networks still varies a lot, demonstrating that CADA relies on the performance of community detection methods to accurately flag nodes as most anomalous. However, Embed outperforms OddBall, but is dependent on the number of dimensions to accurately detect anomalies. The community detection methods can optionally be parameter free, while performance could be enhanced by uncovering smaller or bigger communities, for example using the resolution parameter in case of the Louvain algorithm.

7 Conclusions and future work

Previously proposed techniques were not parameter free or approached the network anomaly detection problem from a local perspective. In this paper we proposed the CADA algorithm, which identifies anomalous nodes based on whether they connect to many communities, while not belonging to one distinct community themself. An advantage of this community-aware approach is that it scales linearly with the number of edges. Furthermore, it is parameter free and highly effective. Experiments showed that our proposed community-aware methodology can spot anomalies in both synthetic and real-world data sets that were not discovered by previous methods. Furthermore, on synthetic benchmark datasets, CADA outperformed previous approaches.

Future work will investigate which community detection methods are most robust for node anomaly detection, and whether a hybrid method combining both global and local features, may yield more accurate and relevant anomalies.

References

1. Akoglu, L., Mcglohon, M., Faloutsos, C.: Anomaly detection in large graphs. In: In CMU-CS-09-173 Technical Report (2009)
2. Akoglu, L., Tong, H., Koutra, D.: Graph based anomaly detection and description: A survey. Data Mining and Knowledge Discovery **29**(3), 626–688 (2015)
3. Barabási, A.L.: Network science. Cambridge University Press (2016)
4. Bindu, P.V., Thilagam, P.S.: Mining social networks for anomalies: Methods and challenges. Journal of Network and Computer Applications **68**, 213–229 (2016)
5. Blondel, V.D., Guillaume, J.L., Lambiotte, R., Lefebvre, E.: Fast unfolding of communities in large networks. Journal of Statistical Mechanics: Theory and Experiment **10008**(10), 6 (2008)

6. Chandola, V., Banerjee, A., Kumar, V.: Anomaly detection: A survey. ACM Computing Surveys **41**(September), 1–58 (2009)
7. Denning, D.E.: An intrusion-detection model. In: Proceedings of IEEE Symposium on Security and Privacy, pp. 118–131 (2012)
8. Fortunato, S.: Community detection in graphs (2010)
9. Hassanzadeh, R., Nayak, R., Stebila, D.: Analyzing the effectiveness of graph metrics for anomaly detection in online social networks. In: Lecture Notes in Computer Science, vol. 7651, pp. 624–630 (2012)
10. Haveliwala, T.H.: Topic-sensitive PageRank. In: Proceedings of 11th International Conference on World Wide Web, pp. 517–526 (2002)
11. Henderson, K., Gallagher, B., Li, L., Akoglu, L., Eliassi-Rad, T., Tong, H., Faloutsos, C.: It's who you know: Graph mining using recursive structural features. In: Proceedings of 17th ACM International Conference on Knowledge Discovery and Data Mining, p. 663 (2011)
12. Hu, R., Aggarwal, C.C., Ma, S., Huai, J.: An embedding approach to anomaly detection. In: Proceedings of 32nd IEEE International Conference on Data Engineering, pp. 385–396 (2016)
13. Janssens, J.: Outlier Selection and One-class Classification. Maastricht University (2013)
14. Kaur, R., Singh, S.: A comparative analysis of structural graph metrics to identify anomalies in online social networks. Computers & Electrical Engineering **57**, 294–310 (2017)
15. Krishnan, V., Raj, R.: Web Spam Detection with Anti-Trust Rank. AIRWeb **6**, 37–40 (2006)
16. Lancichinetti, A., Fortunato, S., Radicchi, F.: Benchmark graphs for testing community detection algorithms. Physical Review E **78**(4) (2008)
17. Page, L., Brin, S., Motwani, R., Winograd, T.: The PageRank Citation Ranking: Bringing Order to the Web. World Wide Web Internet And Web Information Systems **54**(1999-66), 1–17 (1998)
18. Rosvall, M., Bergstrom, C.: Maps of random walks on complex networks reveal community structure. Proceedings of National Academy of Sciences, **105**(4), 1118–1123 (2008)
19. Rosvall, M., Bergstrom, C.T.: Multilevel compression of random walks on networks reveals hierarchical organization in large integrated systems. PLoS ONE **6**(4) (2011)
20. Sun, J., Qu, H., Chakr, D., Faloutsos, C.: Neighorhood formation and anomaly detection in bipartite graphs. Proceedings of SIAM Conference on Data Mining pp. 1–8 (2008)
21. Tong, H., Lin, C.: Non-Negative Residual Matrix Factorization with Application to Graph Anomaly Detection. In: Proceedings of SIAM Conference on Data Mining, pp. 143–153 (2011)
22. Travers, J., Milgram, S.: An Experimental Study of the Small World Problem. Sociometry **32**(4), 425 (1969)
23. Watts, D.J., Strogatz, S.H.: Collective dynamics of 'small-world' networks. Nature **393**(6684), 440–442 (1998)
24. Xu, X., Yuruk, N., Feng, Z., Schweiger, T.a.J.: SCAN: A Structural Clustering Algorithm for Networks. In: Proceedings of 13th ACM International Conference on Knowledge Discovery and Data Mining, pp. 824–833 (2007)
25. Yufeng Kou, Chang-Tien Lu, Sirwongwattana, S., Yo-Ping Huang: Survey of fraud detection techniques. In: Proceedings of IEEE International Conference on Networking, Sensing and Control, 2004, pp. 749–754 (2004)