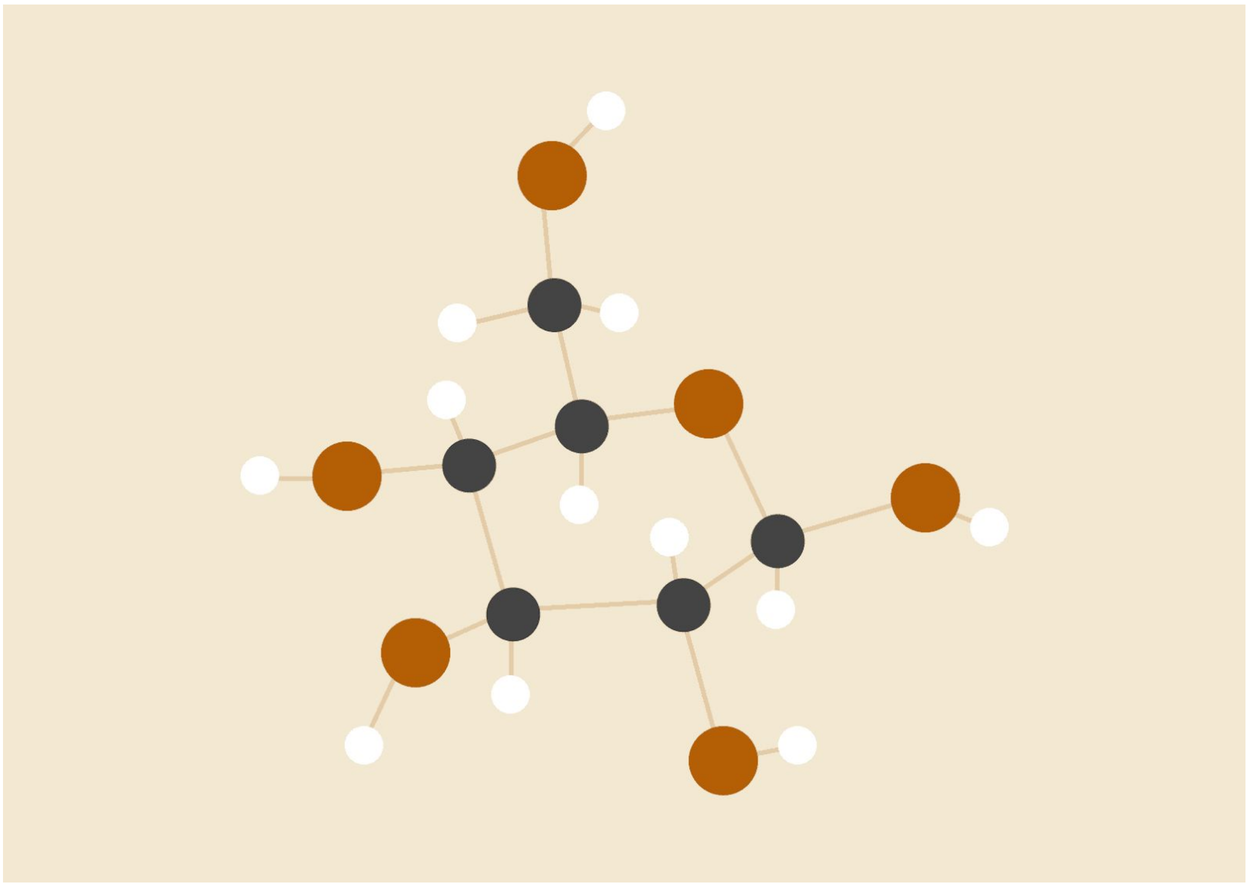
Graph Based Anomaly Detection And Description

Group 29

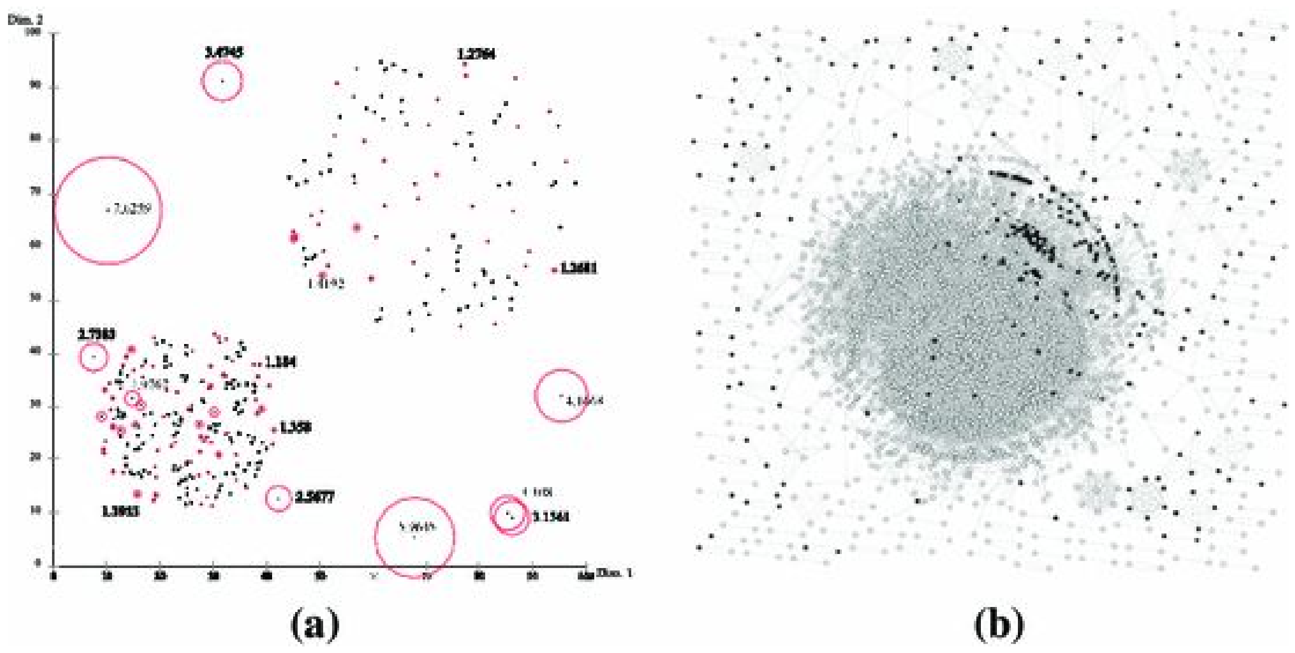


Gaurav Mahadik
&
Mandar Phapale

INTRODUCTION

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. 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. We highlight the effectiveness, scalability, generality, and robustness aspects of the method. Finally, we present several real-world applications of graph-based anomaly 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.

Source : https://arxiv.org/abs/1404.4679



WHAT IS ANOMALY DETECTION?

In data mining, anomaly detection is the identification of rare items, events or observations which raise suspicions by differing significantly from the majority of the data.

Typically the anomalous items will translate to some kind of problem such as bank fraud, a structural defect, medical problems or errors in a text. Anomalies are also referred to as outliers, novelties, noise, deviations and exceptions.

WHY GRAPHS ?

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

1.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.

2. Powerful representation:

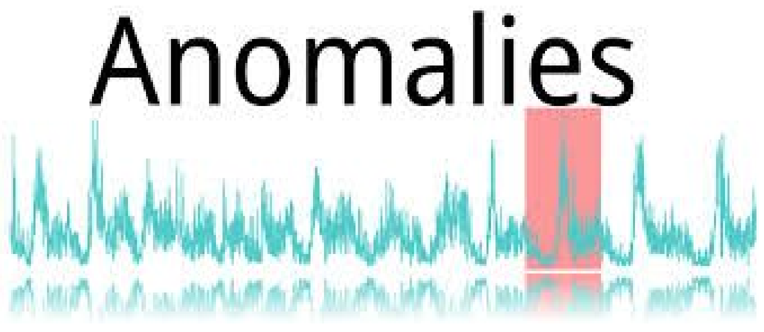
Graphs naturally represent the inter-dependencies by the introduction of links (or edges) between the related objects. The multiple paths lying between these related objects effectively capture their long-range correlations. Moreover, a graph representation facilitates the representation of rich datasets enabling the incorporation of node and edge attributes/types.

3. 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).

4. 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.



ANOMALY DETECTION CHALLENGES:

Data Specific Challenges:
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.

Problem Specific Challenges:
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.

Graph-specific challenges:
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**: Firstly, 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 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:** Secondly, 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.

ANOMALY DETECTION METHODS:

1.Anomaly Detection in Static Graphs:

1)Anomalies in plain (unlabeled) graphs:

A plain graph consists of only nodes and edges among those nodes, i.e. the graph structure.

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) Anomalies in attributed (node-/edge-labeled) 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.

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.**Anomaly Detection in Dynamic Graphs:**

The dynamic graph anomaly detection algorithms based on the type of “graph summary” or “footprint” they use, and the type of events they detect:

(i) Feature-based :

The key idea behind the feature-based methods is that similar graphs probably share certain properties, such as degree distribution, diameter, eigenvalues. 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.
- When the distance is greater than a manually or automatically defined threshold
- characterize the corresponding snapshot as anomalous.

(ii) Decomposition-based :

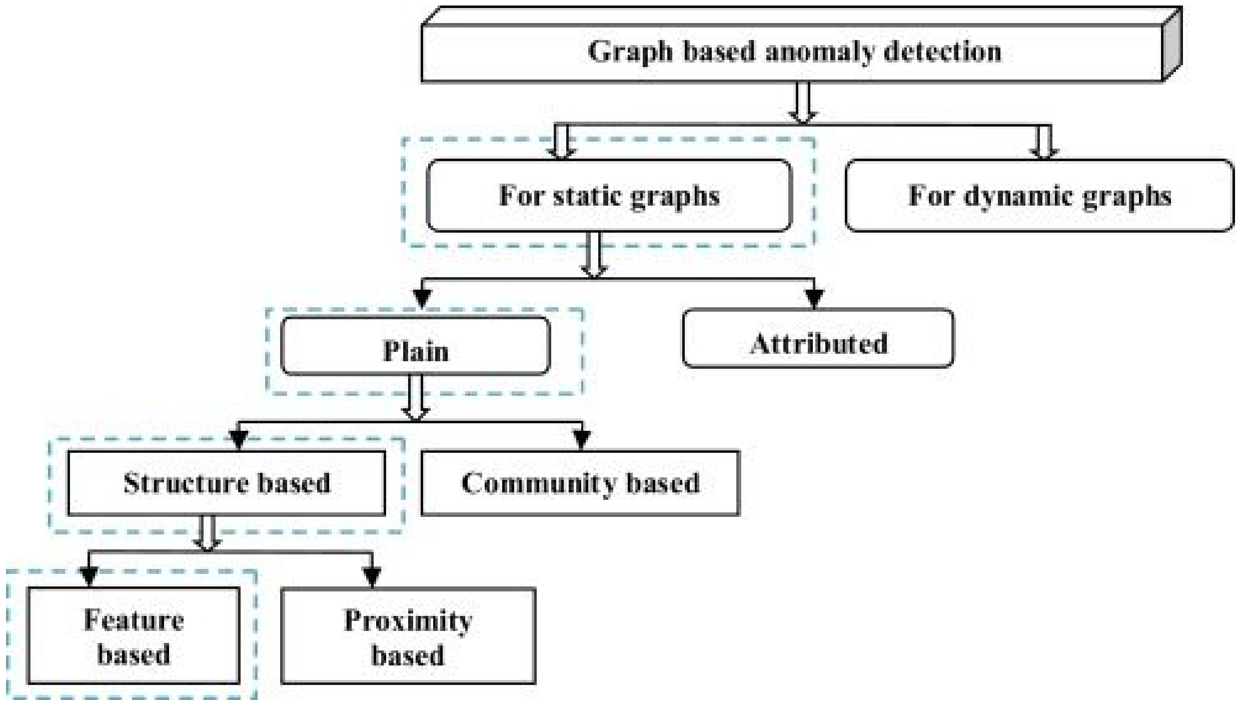
The decomposition-based approaches detect temporal anomalies by resorting to matrix or tensor decomposition of the time-evolving graphs, and interpreting appropriately selected eigenvectors, eigenvalues or singular values. The methods can be divided in two categories based on the representation of the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

(iii) Community or clustering based :

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.

(iv) Window-based :

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.



GRAPH BASED ANOMALY DESCRIPTION: INTERPRETATION AND SENSE MAKING:

1.Interpretation-friendly Graph Anomaly Detection:

The first problem is 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.

Approach : Assume we have a bipartite graph (e.g., author-conference graph), and we can represent it by its adjacency matrix 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 A = XY+R. In this factorization, the two low-rank matrices X and Y usually capture the 'normality' 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'.

2. Finding the root cause of anomalies: Interactive Graph Querying

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.

Approach: 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. The original method 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.

GRAPH BASED ANOMALY DETECTION IN REAL WORLD APPLICATIONS:

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. The following the the different real world applications of Graph Based Anomaly Detection:

1.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.

2.Anomalies in social networks

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 (i) lead the user to malicious sites that compromise the user's device, (ii) promise false rewards and make the user perform certain tasks (e.g. filling out surveys) potentially for someone else’s benefit, (iii) make the user boost the reputation of certain pages by clicking or ‘liking’ them, (iv) make the user redistribute (e.g., by sharing/re-posting), and so on.

3.Anomalies in computer networks: 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.

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: (i) 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; (ii) large number of edges makes it impractical to study the highly dynamic time-series of communications volume in tandem.

CONCLUSION

- In this project , we discussed anomaly detection, how it is different from machine learning, and then discussed different anomaly detection techniques.
- We categorized anomalies and discussed some of the techniques for detecting them.
- Our aim, however, is not to claim the superiority of graph-based methods over other detection techniques. 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.

REFERENCES

1. https://arxiv.org/abs/1404.4679
2. https://minerva-access.unimelb.edu.au/handle/11343/91689
3. https://github.com/yzhao062/anomaly-detection-resources
4. https://www.eecs.wsu.edu/~cook/pubs/kdd03.pdf
5. https://dl.acm.org/citation.cfm?id=956831