Anomaly Discover: A New Community-based Approach for Detecting Anomalies in Social Networks

Hedia Zardi, Hajar Alrajhi
Department of Computer Science-College of Computer, Qassim University, Buraydah 51452, Saudi Arabia

***Abstract*—In this paper, a new method called Anomaly Discover is provided for detecting anomalies in communities with mixed attributes (binary, numerical and categorical attributes). Our strategy tries to identify unusual users in Online Social Networks (OSN) communities and score them according to how far they deviate from typical users. Our ranking is based on both users' attributes and network structure. Moreover, for effective anomaly detection, the context-selection process is performed for choosing relevant attributes that demonstrate a strong contrast between normal and abnormal users. So the anomaly score is defined as the degree of divergence in the network structure as well as a context-specific subset of attributes. To assess the efficacy of our model, we used real and artificial networks. We then compared the outcomes to those of two state-of-art models. The outcomes show that our model performs well since it outperforms other models and can pick up anomalies that competing models miss.**

Keywords*—Anomaly detection; community anomaly; anomaly ranking; social networks; relevant attributes

I. Introduction

In data analysis, anomaly detection (also called outlier detection) is identifying items that raise suspicions by differing from the majority of the data. The rising popularity of social networks has attracted some malicious users who have been abusing them. Because of this, it becomes essential on social networks to identify anomalous users. It aims to find users whose activities deviate significantly from regular users, which affects widely different areas such as the detection of social email senders, and detection of fake accounts.

For analyzing relationships in networks, graph-based anomaly detection (GBAD) approaches were used to detect abnormal patterns. In fact, social networks often have data objects linked to each other, so they can be represented as graph networks. The nodes in graph G(N, E) refer to individuals, whereas the edges refer to relationships. The nodes and connecting edges make up a simple graph. On the other hand, nodes and/or edges with related features such as work status, individual ages, type of interaction, and duration, make up the attributed graph(also known as the labeled graph). The attributed networks combine a topological structure with a rich set of features.

The anomaly detection methods in plain graphs analyze the interactions between nodes and employ the network structure to extract graph-centric features and quantify the nodes' closeness. However, the anomaly detection methods in attributed graphs employ the graph's structure and the coherence of attributes as auxiliary information to find patterns and identify anomalies. Unfortunately, few research has considered attributed graphs. In addition, most of these methods focus only on numerical node attributes; but, real-world networks' node attributes are made up of many attribute types. Therefore, the approaches that consider the network structure and users' attributes, depend on the inclusion of all of a given network's attributes. This makes them inadequate for application to today's networks, which feature ever-growing numbers of attributes. Furthermore, the existence of irrelevant attributes in these networks is inevitable and obstructs anomaly detection. Thus, the context selection process of selecting relevant attributes that show a high level of contrast between normal and anomalous users is crucial for effective anomaly detection.

The community-based approaches are well-known graph-based strategies suggested to address the challenge of anomaly detection [1], [2], [3], [4], [5], [6], [7], [8], [9], [10], [11]. These approaches are based on detecting groups of nodes that are densely connected to each other in the graph and to different communities. Actually, anomalous nodes under this setting can be defined as identifying "bridge" nodes that do not have a direct relationship to a specific community [4].

The aim of this work is to present a new approach to the detection of anomalous users in social networks. Our approach is a community-based approach in attributed graphs which considers both the nodes' attributes and the network structure. A node behavior, which would be considered normal across the entire network, may appear as anomalies within the community context. For this reason, our approach is based on the selected context of relevant attributes. In our approach's first step, we start with the detection of the different communities in the network. In the second step, the nodes will be ranked based on the network structure and attributes similarity. The ranking scores given to the nodes represent the anomaly degree for each node, where a high score indicates an anomalous node, and a low score indicates a normal node.

This paper is organized as follows. In Section II, a summary of the state-of-the-art that reviews the existing works in the field of anomaly detection in a social network is provided. Section III is dedicated to the details of our proposed method for anomaly detection. In Sections IV and V, the performance of our approach is examined and the results obtained by our experiments are presented. A summary and our recommendations for future works are provided in Section VI.

II. Related Works

Due to the great interest in the discovery of anomalies in recent times, a variety of algorithms have resulted. We will discuss some of the suggested methods in this section. These methods can be divided into two groups according to the basic idea of the approach, structure-based approaches and community-based approaches for unlabeled (plain) and labeled (attributed) graphs.

A. Structure-based Approaches

The concept behind structure-based approaches [12], [13], [14] is that to inspect both normal and anomalous nodes' characteristics, the structural properties are checked. To find abnormal nodes, specific graph matrices are calculated. Akoglu et al. [12] proposed a feature-based model called OddBall. This method finds patterns that the majority of the graph's egonets follow in relation to the egonet-based features it extracts.The 1-step neighborhood surrounding a node, which includes the node, is referred to as a node's egonet. The main eigenvalue of the weighted adjacency matrix of the egonet, the weight of the egonet, the number of each node's neighbours, and the number of egonet's edges are all examples of egonet features. These features are then analyzed for each egonet, and based on how much they deviate from a particular pattern, an outliers score is given. In [13], the authors use the same previous approach, but with a different metric called a brokerage, which is defined as how many times a node bridges a connection between two other nodes when there is no direct link between them. In [14], the neural network model is used to construct graph-centric attributes to identify the nodes as abnormal or normal. Degree centrality and closeness centrality, in addition to betweenness centrality, were used singly and in combination to improve the model's accuracy. Anomalous nodes have a greater degree of proximity centrality, and betweenness centrality.

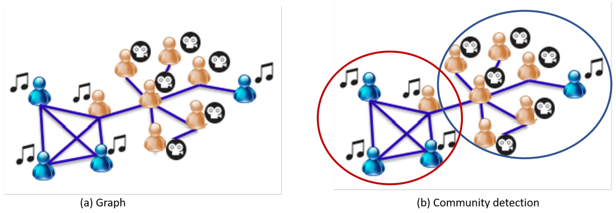
B. Community-based Approaches

Community approaches are based on the concept of finding groups or communities of densely connected nodes in a network and then detecting the abnormal nodes within these communities. Anomalous nodes are described as nodes that do not show the same characteristics compared to other nodes belonging to the same community or that cannot be assigned to any community [3]. So, in the first stage, Community-based approaches start with the community detection process that groups the nodes into groups that contain dense relationships inside those groups and a few connections between other groups (see Fig. 1).

The second stage consists of detecting community anomalies by finding the nodes that do not deserve to be in this community. An anomaly is a node that has different properties compared to the members of its community and that is not very connected to them. Some approaches identify the nodes as either anomalous users or normal users (see Fig. 2). Some other approaches give a score to each user that determines the degree of its abnormality.

Community-based approaches are more effective at spotting anomalies in attributed graphs. In these approaches, the context of the node is specified by the community because it should share qualities with other nodes within its community. A node is considered abnormal when it deviates from these typical characteristics. Community-based approaches can be divided into approaches for plain graphs and approaches for attributed graphs:

- Community-based approaches in plain graphs [4], [5] only depend on the structural information of the network. They look at how nodes are related and use the network structure to extract useful information.
- Detecting community anomalies in attributed graphs requires considering both the nodes' attributes and the network structure [9], [10], [11], [1]. Some approaches use the entire attributes space, which is a disadvantage because they are subjected to the curse of dimensionality, which comes with a slew of issues, including longer runtime. Some other approaches select a set of relevant attributes (called context) to filter the full attributes space and select only relevant ones that show a high level of contrast between normal and anomalous users.


(a) Graph

(b) Community detection

Fig. 1. Community detection in the graph.

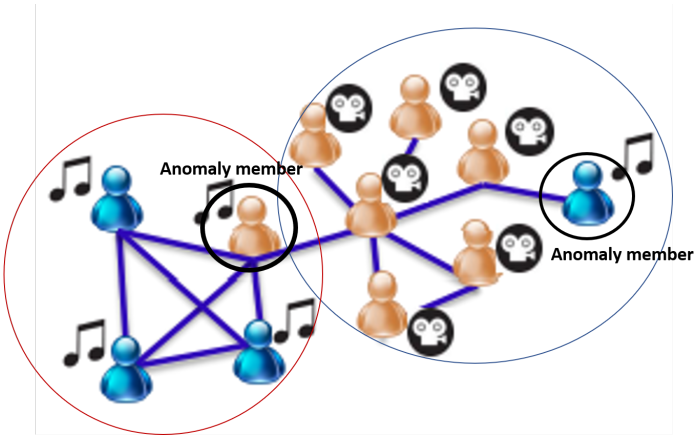


Fig. 2. Community-based anomaly detection in the graph.

III. "Anomaly Discover" Approach

A. Basic Idea

In this work, we present the "Anomaly Discover" approach which is a community-based method for detecting anomalous nodes in social networks. These community anomalies are the nodes that are entrenched within certain graph communities and have deviating attribute values, making them undetectable by global techniques that look for deviation throughout the whole graph node range. As a result, we concentrate on anomalous nodes that differ from their peers in terms of the structure of the graph and the attributes of the node. While some nodes are strongly connected to their communities, the attribute values can vary significantly amongst nodes, for that, we examine these two factors. By introducing context selection, we concentrate on a subset of important attributes. Context selection reduces the issue of unrelated attributes spreading the full-attributes space and concealing anomalous nodes. Context selection is another technique used to reduce the algorithm's time complexity. There are two phases to in our methodology. Firstly, the network is split into communities. A community is a collection of users (that is nodes) who are closely connected and who have similar attributes. So, nodes that lack these features should be regarded as abnormal. So this phase aids in the detection of aberrant nodes that render the community definition invalid. We adopt a modularity-based strategy to split the network since it is fast and can handle enormous networks. In the second phase, for each node, the anomaly score is calculated. By considering both attribute and structure information, this score is utilized to rank nodes depending on the degree of their abnormality. For that, the abnormal nodes are given higher rankings than the usual nodes, which are given lower rankings. Two components make up our score: (i) a structure-based score that considers the graph structure and (ii) an attribute-based score that considers attribute data.

B. Approach Description

In our model, the first phase involves obtaining the communities, and the second phase involves associating the nodes with the anomaly scores. We outline each phase of our method in this section.

1) First phase: Community detection: We divide the graph into communities in the first phase to identify the most linked nodes. In our model, we apply the Louvain Algorithm [15], which is a well-known modularity-based strategy for detecting communities in graphs. We employ the Louvain algorithm because of its speed and efficacity. This approach initially allocates each node to a distinct community in order to maximize modularity gain, and then iteratively moves each node to its neighbor. It comes to a halt when further modularity gains are no longer attainable. Gains in modularity are calculated as follows:

$$\delta Q = [\frac{\sum_{in} + 2k_{i,in}}{2m} - (\frac{\sum_{tot} + k_i^2}{2m})] - [\frac{\sum_{in}}{2m} - (\frac{\sum_{tot}}{2m})^2 - (\frac{k_i}{2m})^2] \tag{1}$$

where $\sum_{in}$ represents the sum weights of the edges inside C, $\sum_{tot}$ represents the sum of weights of the edges incident to nodes in C, k_i represents the sum of weights of the edges incident to node i, $k_{i,in}$ represents the sum of weights of the edges from i to other nodes in C and the sum of the weights of all the network's edges are represented by m. Then, when this value is determined for all communities i is linked to, the community that resulted in the greatest modularity increase is assigned to i. In the absence of an increase, i stays inside its original community. All nodes are subjected to this process regularly and sequentially until no more increases in modularity are possible. Once this local modularity maximum is attained, the first stage is finished.

In the second step of the algorithm, each node is combined into a single community, and a new network made up of nodes from the first phase's communities is created. Next, the first phase is applied once more to the new network.

2) Second phase: Anomaly scoring: In this step, we characterize the anomalous nodes within the communities using the attribute and structural data. As a result, the node's anomaly score must take into account its deviation within the necessary attributes as well as its community relationships. This score introduces new issues, as it requires combining data from both of the components described in this section: the variance in relevant attribute values of and community connections. The attribute-based score and the structure-based score, that are explained in the following subsections, make up our anomalous score.

Structure-based score: In structure-based scoring, we use a similarity measure to examine the node's relevance. We also look at how connected a node is to the members of its community. Within a community anomaly nodes are consequently less comparable, connected, and influential. In this work, the Jaccard similarity measure and node connectivity are used to calculate the structure-based score, as follows:

$$StrAnomaly(v) = 1 - Jacc(v) * con(v) \tag{2}$$

Where $Jacc(v)$ is the Jaccard index of node v and $con(v)$ is the connectivity of node v. A metric known as node connectivity is used to evaluate how strongly connected a node is to its surrounding neighbourhood. A node produces a high value when its connection to its community is higher than the average connection for the community. The following is a description of the node connectivity:

$$con(v) = \frac{nb(v)}{\frac{\sum_{j=1}^{C} nb(v_j)}{|C|}} \tag{3}$$

where nb represents neighbors of the node within the community, and $|C|$ represents the nodes' number within the community. The node with high community connectivity has a high score $con(v)$ and vice versa. Anomaly nodes have a low connectivity score compared to normal nodes, which are high in connectivity.

The Jaccard Similarity measure is used to measure the similarity of the node to its community in a graph and hence similar and dissimilar nodes can be detected. This similarity index is used to identify anomalies nodes in the community. The Jaccard index for each node in a community is the sum of the edges the node has with other nodes in the similar community and all other nodes in the graph. In other words, the intersection of the number of node neighbors in the graph over the union of the number of nodes in the similar community. The following is how the Jaccard index is calculated:

$$Jacc(v) = \frac{|ng(v) \cap cv(v)|}{|ng(v) \cup cv(v)|} \tag{4}$$

Where $ng(v)$ represents the neighbors of the node v in the graph, and $cn(v)$ represents all nodes in the same community with node v. The value of the Jaccard index should range in: $0Jacc(v)1$. A Jaccard index of 1 means the node is fully connected to other nodes in the similar community and vice versa.

Attribute-based score: The objective of the attribute-based score is to quantify the difference in a node's attributes from those of other nodes within the similar community considering subset of related attributes that are chosen in accordance with the context. Considering that many real-world networks are heterogeneous, this attribute similarity function facilitates the combination of attributes of various types. We employ Yi Yang's original Unsupervised Discriminative Feature Selection (UDFS) algorithm to identify relevant attributes [16]. Its goal is to select the data representation features with the highest discriminating. The algorithm optimises the features and produces a ranking and weighting of the features. Additionally, to filter the entire space of attributes, they are ranked according to their UDFS scores as follows:

$$min_{W^TW=1}Tr(W^TMW)+\gamma||W||_{2,1} \tag{5}$$

where ω^i is denoted by the i-th row of W, i.e., $W = [\omega^1, ...\omega^d]^T$, the objective function represented in eq. 5 can also be expressed as:

$$min_{W^TW=1}Tr(W^TMW)+\gamma\sum_{i=1}^{d}||\omega^i||_2 \tag{6}$$

As a result, a new representation of χ_i employing only a select few features is provided for a datum $\chi_i, \chi_i = W^Tx_i$. As an alternative, we can rank each feature $f_i|_{(i=1)}^{d}$ based on $||\omega^i||_2$ in descending order and choose the features with the highest rankings. As a consequence, we'll select a subset of the most crucial attributes with size N, where N is a parameter that our algorithm takes into account as input. Given the context (relevant attributes), we can estimate the attribute-based score, which can be defined as the average of all the scores for the relevant attributes. This attribute-based score is defined as:

$$AttrAnomaly(v)=\frac{\sum_{r=1}^{n}S(v,a_r)}{n}, \forall a_r \in A \tag{7}$$

where $S(v, a_r)$ represents the attribute score of node v for the attribute a_r, $\forall a_r \in A$, and in the set A of relevant attributes the number of attributes is represented by n, which is define as:

$$S(v,a_r)=\frac{\sum_{j=1}^{C}d(v,v_j)}{|C|}, \forall v_j \in C \tag{8}$$

where v_j stands for the other nodes within the community, the nodes' number in the community is denoted by $|C|$, and the distance between the nodes v and v_j is represented by $d(v, v_j)$, which is assigned either to zero or one. When $d(v, v_j)$ is zero, this means that the distance between the these nodes is equal to or less than the mean distance (Md), otherwise, it is set at one.

$$d(v_j,v_j)=\begin{cases}0 & \text{if}|a_r(v_i)-a_r(v_j)|\le Md(C_{(v)},a_r)\\ 1 & \text{Otherwise.}\end{cases}$$

where $a_r(v_i)$ represents the the attribute a_r's value in the node v_i attribute vector, and the mean distance of attribute a_r of node v is denoted by $Md(C_{(v)}, a_r)$ within the community which is illustrated as follows:

$$Md(C_v,a_r)=\frac{\sum_{i,j=1}^{C}(a_{r(v_i)}-a_{r(v_j)})}{p} \tag{9}$$

where the distance between the node v_i and the node v_j for the attribute a_r is represented by $(a_{r(v_i)} - a_{r(v_j)})$, and in each community the number of node pairs is represented by p. This distance for binary attributes is determined by the *simple matching coefficient* between the nodes vi and vj for the a_r attribute:

$$(a_{r(v_i)}-a_{r(v_j)})=\frac{\sum_{k=1}^{d}\left(a_{rk(v_i)}\wedge a_{rk(v_j)}\right)\vee\left(\neg a_{rk(v_i)}\wedge\neg a_{rk(v_j)}\right)}{d}$$

The *Jaccard similarity index* between the "1-of-N" binary encodings of $(a_{r(v_i)}$ and $a_{r(v_j)})$ gives the distance for categorical attributes; in other words:

$$(a_{r(v_i)}-a_{r(v_j)})=\frac{\sum_{k=1}^{d}a_{rk(v_i)}\wedge a_{rk(v_j)}}{\sum_{k=1}^{d}a_{rk(v_i)}\vee a_{rk(v_j)}}$$

The*Euclidean distance* between $(a_{r(v_i)}$ and $a_{r(v_j)})$ is what determines this distance for numeric characteristics; that is,

$$(a_{r(v_i)}-a_{r(v_j)})=\frac{1}{1+\sqrt{\sum_{k=1}^{d}\left(a_{rk(v_i)}\wedge a_{rk(v_j)}\right)^2}}$$

where d is the attribute's a_r dimensions, $(a_{rk(v_i)}$ is the value of the attribute's a_r k-th coordinate for node v_i, and $\neg$, $\wedge$ and $\vee$ are the logical operators for NOT, AND, and OR, accordingly.

Anomaly score:: To rank nodes in every community and recognize anomalous nodes, we next use both the structure-based and attribute-based scores to get an aggregate anomaly score (those with a higher anomaly score). Calculate each node v anomaly score as below:

$$AnomalyScore(v)=StrAnomaly(v)+(1-\alpha)AttAnomaly(v) \tag{10}$$

where the weight used to regulate the relative importance of attribute-based anomaly and the structure-based anomaly is represented by α.

C. Algorithm

First, we use the Louvain algorithm to partition the graph into communities (line 1). Then, we determine each node's Jccarad similarity (line 3). Before assigning the structure-based score, we iterate over each node to determine its connectedness(line 6). Afterward, we determine each attribute's UDFS score (line 8), and we choose the N attributes corresponds to lowest score of UDFS to be the relevant attributes (line 9). Then, for each community, we iterate, comparing the distance between the node and its community nodes with the community mean distance using the relevant attributes (lines 10-16). Finally, the anomalous score of the nodes from G is returned (line 20) by combining the structure-based score and attributes-based score (lines 17–19). The name (AnomalyDiscover), an integration of the words community and anomaly, refers to our algorithm for community-based anomaly detection.

Algorithm 1 Anomaly Discover

```
Require: $G : (V, E)$, $A$: Attributes, $N$ : number of select attributes
Ensure: Ranking of all $v \in V$
 1: C ← Louvain Method(G)
 2: Initialize empty vectors $StrAnomaly$, $AttrAnomaly$, $AnomalyScore$ for all $v \in V$
 3: Jacc ← Jaccard similarity index for all $v \in G$ (Eq. 4)
 4: For each $v \in G$ do
 5:     Compute connectivity(v) (Eq. 3)
 6:     Compute StrAnomaly(v) (Eq. 2)
 7: End For
 8: For all $a_r \in A$ calculate UDFS score (Eq. 6)
 9: A' ← subset of $N$ attributes with the lowest UDFS score
10: For each community $C_k$ in $C$ do
11:     Md ← mean values of attributes from A' in $C_k$ (Eq. 9)
12:     For each $v_i$ in $C_k$ do
13:         $s_{vi}$ ← a dictionary containing for each attribute $a_r$ of $v_i$ its anomaly score
14:         Compute AttrAnomaly(v) (Eq. 7)
15:     End For
16: End For
17: For each $v \in G$ do
18: Calculate the anomaly score using the Eq. 10
19: End For
20: Return AnomalyScore
```

D. Complexity Analysis

First, the graph is partitioned into communities by applying the Louvain algorithm, which has a running-time cost $O(vlogv)$ for the number of graph nodes. After that, the Jaccard similarity measure is calculated, and that costs $O(v + e)$, where the number of edges in the graph is represented by e. Next, the StrAnomaly score is calculated with the nodes' number linear cost. The context is then defined using the UDFS score, which has a cost of $O(mv^2)$, where m represents the total attributes' number. Consequently, the computational complexity of the Anomaly Discover model is $O(max(mv^2, v + e))$. When all of the graph nodes are allocated to one community, which happens when a quadric analysis is carried out for each community, this is the worst-case scenario. As a result, the algorithm performs better on a real network with a large number of communities.

IV. Experimental Evaluation of Performances

To study the performance of the "Anomaly Discover" method, we compared it with two well-known algorithms witches are CODA and ConSub that we briefly introduce in this section:

- CODA [17] is one of the most popular models used for anomaly detection in social network communities. In this model, community detection and anomalous node identification are done in a single step. It utilizes the entire attribute set of nodes.
- ConSub's [7] concept is a statistically-based selection of a subset of all attributes of the nodes. This subset demonstrates dependencies inside the graph structure. To find abnormal nodes in the communities, a subset of attributes is chosen and used with the DistOut distance-based outlier model.

A. Evaluation Measures

In order to evaluate the performance of the community-anomaly detection model and establish its validity on synthetic and real datasets, we compare the acquired nodes' ranks of the model with the ground truth. The Area Under the Curve (AUC) is one of the most important performance metrics for anomaly ranking and classification models. AUC measures how well the model can differentiate between two classes; a greater value of AUC means a more effective model. In a machine learning classification task, comparing the actual classes to the predicted classes of the mode. Hence, the results can be categorized into four groups: true positives, false positives, true negatives, and false negatives. True positives are actual anomalies that the model predicted correctly as anomalies, while true negatives are actual normals that the model predicted correctly as normal. On the other hand, false positives are actual normals that the model predicted as an anomaly, while, false negatives are actual anomalies that the model predicted as normal. Specificity is the proportion of correctly identified negatives, whereas sensitivity is the proportion of correctly detected positives. We provide several thresholds in the classification model and to create the ROC curve, sensitivity (also referred to as the true positive rate) is plotted against the false positive rate which is calculated as (1-Specificity).

So, the optimum model is the one that reliably detects all positives and all negatives at a specified threshold value while still obtaining the highest levels of specificity and sensitivity. The top-left portion of the ROC plot contains the greatest value. The area under the curve (top-left corner) consequently represents the ROC curve's ability to reach the highest level of specificity and sensitivity.

The model's runtime is the second metric considered in this evaluation since it's crucial to see if the model can accurately identify the community abnormality in a timely manner. If a shorter runtime is possible, it is better if the outcomes are high-caliber.

B. Real Benchmark Dataset

The Book network and Disney network serve as our testbeds in this part, and the performance of our model is compared to that of the CODA and ConSub models. The only variable in our suggested model is how many attributes to include. The number of characteristics was set to half-number and 10 attributes maximum; adding more characteristics increased the run-time without notably enhancing the quality.

TABLE I. Models Parameter Setting

CODA	Number of communities= 8	Anomaly percentage= 0.05	Link importance= 0.01
ConSub	Size of interval= 10	The number of Monte Carlo iterations= 150	The significance level=0.05

Table I provides information about the other models' parameter settings.

We compare our findings with those of ConSub and CODA in order to assess our methodology using the following real networks:

- Disney Network: the Amazon co-purchase network was divided up into a Disney network that exclusively considered Disney DVDs. In the graph, each product is characterized by 30 properties, including review ratings, product prices, and other information. The network has 124 nodes and 334 edges. The network, although being a tiny dataset, is used to test the majority of anomaly detection models because of its intricate graph and attribute structure. The ground truth of whether an object is an abnormality or normal is not available for this real-world dataset. As part of a user experiment to determine the dataset's ground truth, high school students personally classified each object as normal or an abnormality.[18] presents a thorough explanation of the dataset and the user experiment.
- Book Network: this network which was based on Amazon Co-Purchase Network, includes books that users have tagged as "amazon fails" [19]. On Amazon, customers could use tags to describe items, and different tags like the "amazon fail" tag was used to indicate dubious products. The network has 1468 nodes, 3695 edges, and 28 attributes that are used to describe each object. The basis for this dataset was established by classifying a book as an anomaly when at least 20 users had labeled it as an "amazon fail".
- Enron Network: we employ email transmission as edges between email addresses on the Enron communication network. Spam dataset outliers were defined as addresses that have sent spam. This network contains 13 533 nodes, 176 967 edges. There are 20 attributes present in each node that provide aggregate information about the average number of recipients, the average content length, or the time interval between two mails [19].

C. Synthesis Benchmark Dataset

Evaluating anomaly detection methods is not a straightforward process due to the lack of suitable datasets containing anomalies and the lack of ground truth that defines which data points are actual abnormalities. As a result, performance evaluation is typically the purpose of synthetic datasets. These datasets are utilized to compare a model's performance on synthetic versus real data. Based on [7], synthetic datasets of various attribute counts and sizes are created. To replicate the characteristics of real networks, the graph is created by following a power-law distribution. Relevant attributes obtained values from a Gaussian distribution, while irrelevant attributes obtained values from a uniform distribution. To ensure there were no abnormal values in the relevant attributes, the tails of each Gaussian distribution were truncated using a hyper ellipsoid (see Fig. 3). Anomaly nodes' characteristic values were modified to be random numbers beyond the boundaries of their communities' hyper ellipsoids. The anomalous nodes number within the communities is determined by the anomaly ratio, which is 10%. Only when at least two pertinent attribute combinations are taken into account can the anomalies be found.

The graphml file and the true file are the two files that make up any synthetic dataset. The graph and each node's properties can be found in the graphml file. The true file includes the actual nodes, with a ground truth value of 0 for normal nodes and 1 for anomalous nodes. We use synthetic datasets that contain 1000 number of nodes and various characteristics 2, 10, 20, 40, 60, and 80 to assess the performance of the model's as the number of attributes increases. To evaluate how well the model performs when the network size is increased, we use synthetic datasets with varying numbers of nodes and ten attributes. We configured our model parameter to be the half-number of attributes to test the impact of increase in attributes a while utilising the same configuration in the real network trials for the other models.

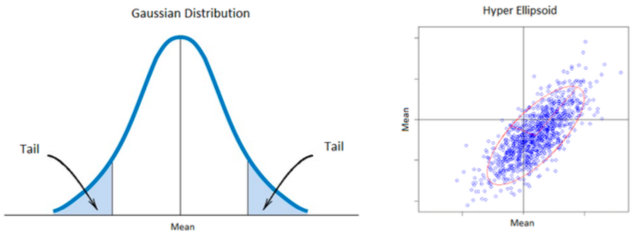


Fig. 3. Gaussian distribution with cutting tails.

V. Results for Real and Synthesis Datasets

A. Results for Real Datasets

1) Disney network: Fig. 4 and 5 illustrate our model's results applied on the Disney network in contrast to CODA and ConSub. The findings show that the "Anomaly Discover" model produces good-quality outcomes with an AUC of .83 (see Fig. 4). In contrast, the other models produce results of poorer quality, with an AUC of.82 for ConSub and.50 for CODA, respectively. The ConSub model performs better than the CODA model, which produces the lowest-quality outputs. In Fig. 5, the runtime evaluation is displayed. The CODA model comes in second with a runtime of 6.05 seconds, just 0.04 seconds behind the "Anomaly Discover" model.

At 8.93 seconds, the ConSub model is the slowest model. The "Anomaly Discover" model yields the best outcomes for identifying community abnormalities, according to this experiment on the Disney Network, out of the three methods. Among the three, the "Anomaly Discover" model is the fastest. While the ConSub model delivers high-quality findings but operates slowly, the CODA model is recognised as the least effective model for identifying community-anomalies in a real-world network.

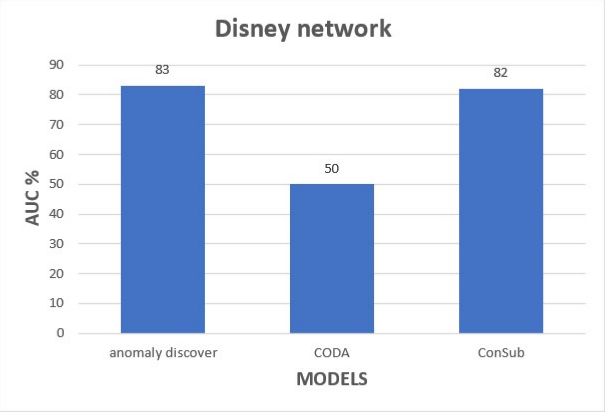


Fig. 4. AUC and ROC curve for the disney network.

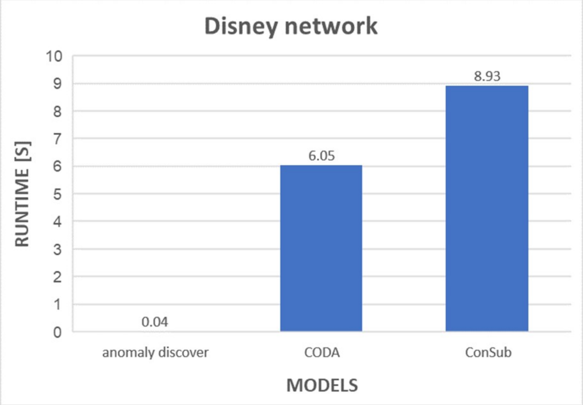


Fig. 5. Evaluation of runtime for disney network.

2) *Book Network:* By computing the AUC, Fig. 6 shows how well the "Anomaly Discover", CODA, and ConSub models perform. In comparison to the other models, the "Anomaly Discover" model produces least findings (see Fig. 6), whereas ConSub produces the highest-quality results (AUC = 0.60). The "Anomaly Discover" model outperforms the CODA models in the Book network's runtime evaluation, which shows that it executes in 12.48 seconds and yields the best results. The CODA model is the slowest at 36 seconds (see Fig. 7).

3) *Enron network:* The outcomes of our model on the Enron network in comparison to CODA and ConSub are shown in Fig. 8 and 9. The figures demonstrate that our model produces good-quality outcomes with an AUC of .78 (see Fig. 8), but it was the slowest (see Fig. 9). In contrast, the CODA produces a result that has lower quality with an AUC .46 but it was the fastest. ConSub produces a result inferior to the "Anomaly Discover" model, but it was faster than our model.

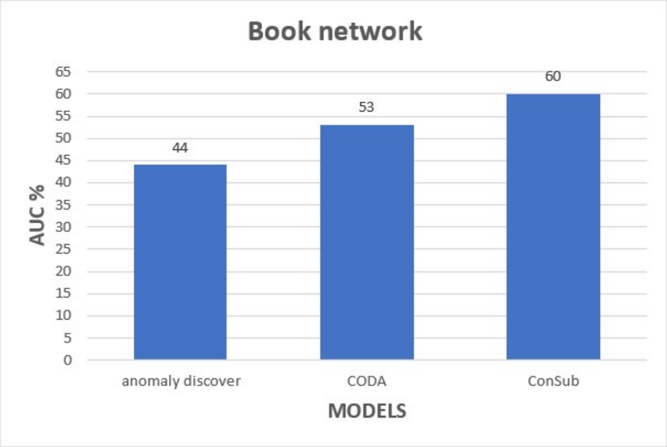


Fig. 6. AUC and ROC curve for the book network.

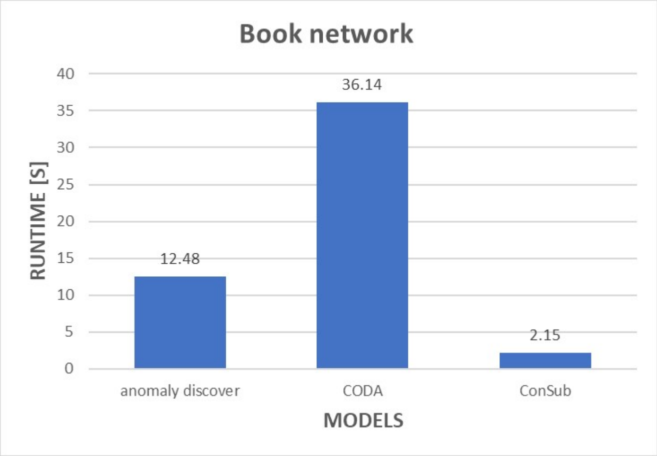


Fig. 7. Evaluation of runtime for book network.

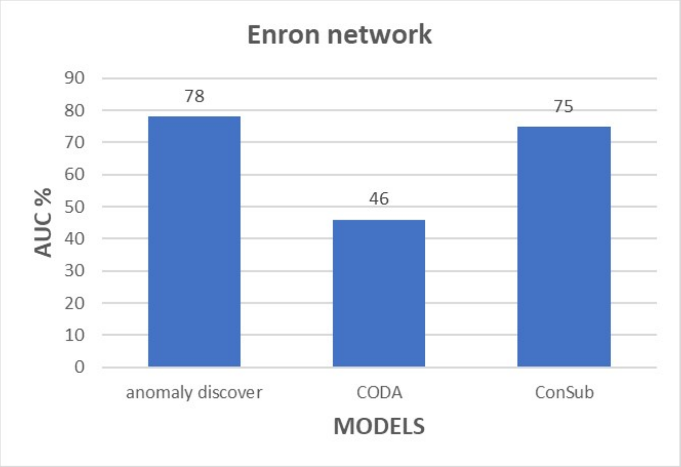


Fig. 8. AUC and ROC curve for the enron network.

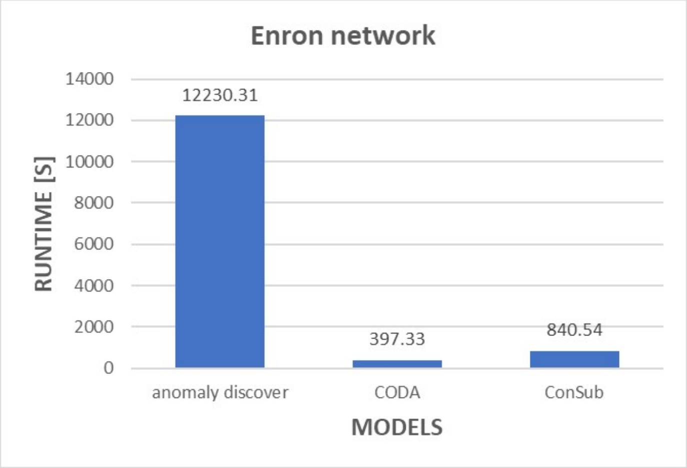


Fig. 9. Evaluation of runtime for enron network.

B. Results for Synthesis Datasets

1) First experiment: We evaluate our model performance as we add more attributes using the AUC curve with datasets of 1000 numbers of nodes. The variance in AUC of the tested models as the number of features rises is depicted in Fig. 10. In comparison to ConSub and CODA, our model results in the best AUC, which ranges from 0.88 to 0.98. Fig. 11 illustrates the runtime evaluation by showing the runtimes with increasing numbers of attributes. In spite of the fact that CODA runtime often grows as the number of characteristics does as well, "Anomaly Discover" and ConSub both provide the best scalability in this regard. It's important to keep in mind that matrix operations are costly, and with CODA, they are performed for each attribute, increasing the runtime.

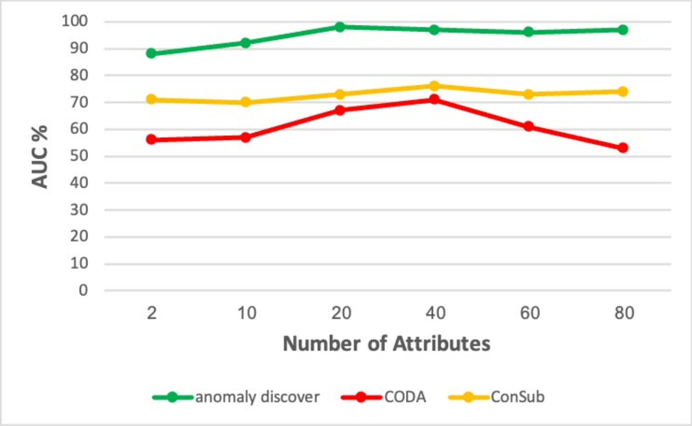


Fig. 10. Variations in AUC using different number of attributes for each tested models.

2) Second experiment: Networks with 500, 1000, 2000, 3500, 6000, and 10000 nodes are utilised to evaluate the model with a larger network. In the smallest network, the AUC of the "Anomaly Discover" model is 0.93, and in the largest network, it is 0.90. In fact, when compared to the other models, "Anomaly Discover" has the greatest AUC (see Fig. 12). Fig. 13 displays the evaluation of the runtime as the network size increases. ConSub has overall faster runtimes than the other models, although the "Anomaly Discover" outperforms CODA in networks with 500, 1000, 2000, and 3500 nodes, while CODA outperforms "Anomaly Discover" in networks with 6000 and 10000 nodes.

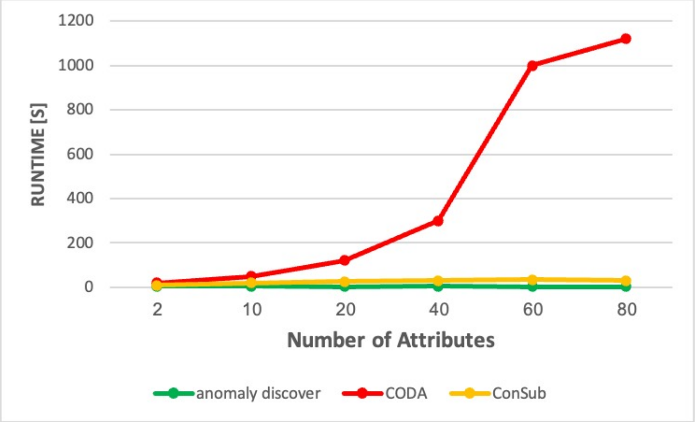


Fig. 11. Variations in runtime using different number of attributes for each tested models.

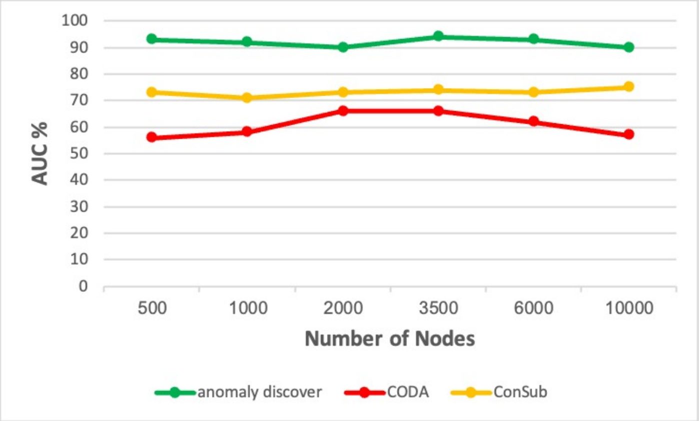


Fig. 12. Variations in AUC using the different number of nodes for each tested models.

C. Discussion

The outcomes of these tests show how well "Anomaly Discover" works to identify community anomalies in both real and synthetic networks. In fact, we've achieved things that are extremely intriguing, which earlier approaches like ConSub [7] couldn't. As the model defines the pertinent network properties rather than taking into account the entire attribute space, it increases the number of attributes while still producing high-quality results and scalability. As a result, the model is appropriate for applications used today, when the number of attributes is increasing. Since the ConSub model also describes the network context, whereas CODA [17] simply considers the network attributes, it performs better than CODA in terms of performance.

VI. Conclusion

In this study, we focused on finding anomalous users in online networks. In particular, we are seeking to identify

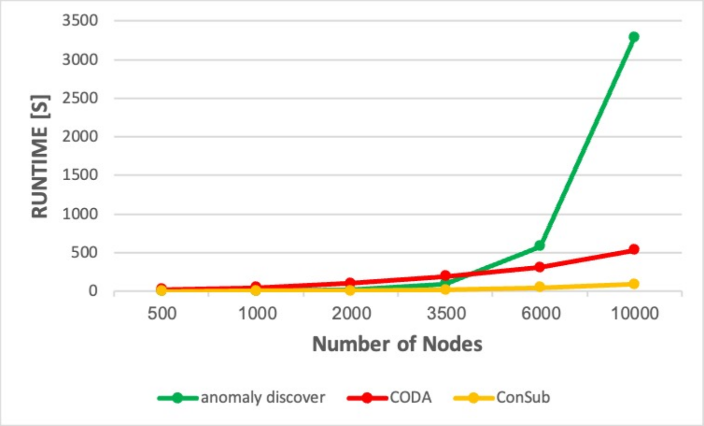


Fig. 13. Variations in runtime using different number of nodes for each tested models.

anomalies that diverge from their communities in comparison to normal users who frequently share numerous attributes with their members of the community. In fact, by using both structure information and attributes information, we were able to create an anomalous ranking score to efficiently find complicated anomalies that differed in either their characteristics values or their structure, or both. To highlight any deviation in these values, the context is selected, which is a subset of the relevant attributes. Given that many real-world networks are heterogeneous, this approach enables the combining of the attributes of mixed types.

We next go over different ways that an extension of our suggested approach could be implemented. Other features of online social networks, such as user communication through comments or message exchange, could be evaluated to find anomalies, though. While they might have common attribute values and structural characteristics with their community, these data might signify an unexpected communication pattern, making them a useful indicator of anomalous nodes.

References

[1] S. A. Moosavi, M. Jalali, N. Misaghian, S. Shamshirband, and M. H. Anisi, "Community detection in social networks using user frequent pattern mining," *Knowl. Inf. Syst.*, vol. 51, no. 1, p. 159–186, apr 2017. [Online]. Available: https://doi.org/10.1007/s10115-016-0970-8

[2] M. Bouguessa, *A Model-Based Approach for Mining Anomalous Nodes in Networks*, 01 2020, pp. 213–237.

[3] L. Akoglu, H. Tong, and D. Koutra, "Graph-based anomaly detection and description: A survey," *CoRR*, vol. abs/1404.4679, 2014. [Online]. Available: http://arxiv.org/abs/1404.4679

[4] H. N. Win and K. T. Lynn, "Community detection in facebook with outlier recognition," in *2017 18th IEEE/ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD)*, 2017, pp. 155–159.

[5] M. Bouguessa, *A Model-Based Approach for Mining Anomalous Nodes in Networks*, 01 2020, pp. 213–237.

[6] E. Müller, P. I. Sánchez, Y. Mülle, and K. Böhm, "Ranking outlier nodes in subspaces of attributed graphs," in *2013 IEEE 29th International Conference on Data Engineering Workshops (ICDEW)*, 2013, pp. 216–222.

[7] P. I. Sánchez, E. Müller, F. Laforet, F. Keller, and K. Böhm, "Statistical selection of congruent subspaces for mining attributed graphs," in *2013 IEEE 13th International Conference on Data Mining*, 2013, pp. 647–656.

[8] P. I. Sánchez, E. Müller, O. Irmler, and K. Böhm, "Local context selection for outlier ranking in graphs with multiple numeric node attributes," in *Proceedings of the 26th International Conference on Scientific and Statistical Database Management*, ser. SSDBM '14. New York, NY, USA: Association for Computing Machinery, 2014. [Online]. Available: https://doi.org/10.1145/2618243.2618266

[9] S. Mekouar, N. Zrira, and E. H. Bouyakhf, "Community outlier detection in social networks based on graph matching," *International Journal of Autonomous and Adaptive Communications Systems*, vol. 11, p. 209, 01 2018.

[10] G. V. Daniel and M. Venkatesan, "Robust graph based deep anomaly detection on attributed networks," in *2021 11th International Conference on Cloud Computing, Data Science & Engineering (Confluence)*, 2021, pp. 1029–1033.

[11] Y. Wang and Y. Li, "Outlier detection based on weighted neighbourhood information network for mixed-valued datasets," *Information Sciences*, vol. 564, pp. 396–415, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0020025521001870

[12] L. Akoglu, M. McGlohon, and C. Faloutsos, "oddball: Spotting anomalies in weighted graphs," in *Advances in Knowledge Discovery and Data Mining*, M. J. Zaki, J. X. Yu, B. Ravindran, and V. Pudi, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 410–421.

[13] R. Kaur and S. Singh, "A comparative analysis of structural graph metrics to identify anomalies in online social networks," *Computers & Electrical Engineering*, vol. 57, pp. 294–310, 2017. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0045790616307959

[14] A. Chaudhary, H. Mittal, and A. Arora, "Anomaly detection using graph neural networks," in *2019 International Conference on Machine Learning, Big Data, Cloud and Parallel Computing (COMITCon)*, 2019, pp. 346–350.

[15] V. Blondel, J.-L. Guillaume, R. Lambiotte, and E. Lefebvre, "Fast unfolding of communities in large networks," *Journal of Statistical Mechanics Theory and Experiment*, vol. 2008, 04 2008.

[16] Y. Yang, H. T. Shen, Z. Ma, Z. Huang, and X. Zhou, "L2,1-norm regularized discriminative feature selection for unsupervised learning," in *Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence - Volume Volume Two*, ser. IJCAI'11. AAAI Press, 2011, p. 1589–1594.

[17] J. Gao, F. Liang, W. Fan, C. Wang, Y. Sun, and J. Han, "On community outliers and their efficient detection in information networks," in *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '10. New York, NY, USA: Association for Computing Machinery, 2010, p. 813–822. [Online]. Available: https://doi.org/10.1145/1835804.1835907

[18] J. Z. Huang, "Extensions to the k-means algorithm for clustering large data sets with categorical values," *Data Mining and Knowledge Discovery*, vol. 2, pp. 283–304, 1998.

[19] P. Iglesias Sánchez, "Context selection on attributed graphs for outlier and community detection," Ph.D. dissertation, 2015.