applied sciences

MDPI

Article

Detecting Anomalies in Network Communities Based on Structural and Attribute Deviation

Hedia Zardi [1,†], Hanen Karamti [2,*,†], Walid Karamti [1,3,†] and Norah Saleh Alghamdi [2]

[1] Department of Computer Science, College of Computer, Qassim University, Buraydah 51452, Saudi Arabia
[2] Department of Computer Sciences, College of Computer and Information Sciences, Princess Nourah Bint Abdulrahman University, P.O. Box 84428, Riyadh 11671, Saudi Arabia
[3] Data Engineering and Semantics Research Unit, Faculty of Sciences of Sfax, University of Sfax, Sfax 3052, Tunisia
* Correspondence: hmkaramti@pnu.edu.sa
† These authors contributed equally to this work.

Abstract: Anomaly detection in online social networks (OSNs) is an important data mining task that aims to detect unexpected and suspicious users. To enhance anomaly exploration, anomaly ranking is used to assess the degree of user anomaly rather than applying binary detection methods, which depend on identifying users as either anomalous users or normal users. In this paper, we propose a community-based anomaly detection approach called Community ANOMaly detection (CAnom). Our approach aims to detect anomalous users in an OSN community and rank them based on their degree of deviation from normal users. Our approach measures the level of deviation in both the network structure and a subset of the attributes, which is defined by the context selection. The approach consists of two phases. First, we partition the network into communities. Then, we compute the anomaly ranking score, which is composed of a community-structure-based score and an attribute-based score. Experiments on real-world benchmark datasets show that CAnom detects ground-truth groups and outperforms baseline algorithms on accuracy. On synthetic datasets, the results show that CAnom has high AUC and ROC curves even when the attribute number increases; therefore, our model is suitable for today's applications, where the number of attributes is rising.

Keywords: anomaly detection; community-based approach; anomaly rank; social networks; contextual anomaly; attributed network

Citation: Zardi, H.; Karamti, H.; Karamti, W.; Alghamdi, N.S. Detecting Anomalies in Network Communities Based on Structural and Attribute Deviation. *Appl. Sci.* **2022**, *12*, 11791. https://doi.org/10.3390/app122211791

Academic Editors: Antonios Litke and Theodora Varvarigou

Received: 17 October 2022
Accepted: 15 November 2022
Published: 20 November 2022

1. Introduction

Hundreds of millions of people use online social networks (OSNs) daily as communication platforms, as well as a platform to share their information and interests and conduct business, irrespective of geographical distance. Contrary to the benefits offered by social networks, their misuse also increased in terms of malicious users who express a range of different behavior patterns. Such users expose the security and privacy of normal users, and this can lead to substantial losses [1–3]. Such malicious users discredit genuine users' behavior patterns, which carry important information for researchers and analysts. Therefore, the demand for detecting anomalies in OSNs has significantly increased.

A graph is used as a powerful representation of this complex network, where nodes represent the social members and edges represent the interactions among them. This representation of the network is a static graph, which represents a single snapshot of the social network data at a particular time. OSNs are naturally divided into groups called communities, which are groups of people who interact frequently and share several common properties and interests. A community is defined in the graph by dense nodes that have strong connections among them and weak connections with other nodes in the network. An attributed graph (or labeled graph) is a graph where edges and/or nodes provide sufficient information such as individual ages, work status, interaction type,

and duration. 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.

Community-based approaches are graph-based approaches that aim to find densely connected nodes in a graph and spot anomalies among these communities. In the attributed graph, each node having attributes that deviate significantly from those of other nodes existing in its community is marked as an anomaly.

For example, an unemployed man in a community of female high school teachers is an example of a community anomaly. Therefore, the nodes' attributes in the community are analyzed to spot such deviations, and some approaches combine them with an analysis of the community's structure.

This paper's main objective is to propose an approach for detecting community anomalies based on the selected context of relevant attributes in a static and attributed network. The proposed algorithm ranks the nodes by their anomaly degree considering both the graph structure and the node's attributes. The detected anomalous nodes in the network are chosen based on attribute deviation, connectivity, and node importance.

Our approach is conducted in two stages: (1) detecting the communities of the graph and (2) raking the nodes based on their degree of anomaly.

For community detection, we employ the Louvain algorithm to detect disjointed communities in the network, given its sufficient speed and ability to handle large networks. In the second stage, we define an anomaly score that combines the structure score and the node attribute score to gain an accurate score for the nodes. Our approach selects the context of relevant attributes to improve the detection of community anomalies and prevent the irrelevant attributes from masking the existence of any anomalies.

The remainder of the paper is organized as follows. Section 2 is dedicated to presenting some known approaches. Section 3 outlines the proposed model. In Section 4, a series of experimental results are given. Section 5 presents a summary and our suggestions for future work.

2. Related Work

A variety of approaches has been introduced to detect anomalies in graph data. Most of these approaches are specified for a particular problem, as some address unattributed graphs, while others are applied for the attributed graphs only, or even for a graph with a specific type of attributes. By categorizing the approaches based on their technique of spotting anomalies in the network graph, they can be grouped into structure-based approaches and community-based approaches.

Structure-based approaches [4–8] exploit the structure of the graph to define patterns that normal nodes obey and detect the anomalous nodes that do not. There is much information that could be extracted from the structure, such as the centric features and the closeness features. Approaches based on the extraction of structured features can be categorized into graph-centric features that are associated with the global graph structures and node-level features that are associated with the local graph properties. The node-level features measure the (in/out) degree and centrality, such as closeness and betweenness.

The community-based approaches [9–15] are based on the notion of finding densely connected groups of nodes in the network, such as communities. Then, identifying the anomalous nodes in such communities. In fact, the community defines the context of the node, as it should share common properties with other nodes in its community. However, when the node deviates from these common properties, it is considered anomalous. In contrast, other approaches, such as structure-based approaches, can detect only global anomalies, which is less useful in real applications.

Most community-based approaches detect anomalies as a second step after partitioning the network and identifying communities [9]. Some approaches integrate community detection and anomaly detection in a single step [16,17].

An unattributed graph consists of only nodes and edges between these nodes. Therefore, the anomaly detection approaches in unattributed graphs are based only on the structural information of the network. They analyze the interaction between nodes and employ the network structure to extract meaningful features. There is much information extracted from the structure, such as the centric features and the closeness features, which could be node-level features or egonet-level features.

In attributed graphs, where nodes' attributes provide additional information, some algorithms are based only on these attributes, while others rely on both the structural information and the nodes' attributes. However, these attributes raise a challenge for some approaches that consider all the node attributes [10] due to the increasing number of attributes in today's applications. In addition, the node attributes include some irrelevant attributes that scatter the full attribute space and make any deviation invisible due to the curse of dimensionality [11]. It implies that more attributes hinder the detection of an anomaly, as well as presenting other challenges. One of the most important challenges is the concentration of scores, which drives the values to become numerically similar as dimensionality increases. Therefore, the task of finding relevant attributes, also called context selection, plays an important role in the accurate computing of anomaly scores [12–15,18].

3. A Community Anomaly Detection Approach Based on Structural/Attribute Information

3.1. Basic Idea

Our approach is a community-based approach that aims to detect anomalous nodes in network communities. These community anomalies are embedded within specific graph communities and have deviating attribute values, which cannot be detected by global approaches that detect deviation from the entire range of graph nodes. Thus, we focus on anomalous nodes that deviate from their community with respect to both the graph structure and node attributes. We consider these two aspects because some nodes can be well-connected to their community, but they deviate strongly from the attribute values. In contrast to traditional approaches that consider all the attributes, we focus on a subset of relevant attributes by introducing context selection. Context selection reduces the problem of irrelevant attributes scattering the full-attribute space and masking the anomalous nodes. Additionally, context selection reduces the time complexity of the algorithm. In this paper, we use single-view context selection that considers a single subset of attributes to select, and the rest of the attributes, which are not relevant, are neglected. The single-view method reduces time complexity, which is the drawback of multi-view approaches where all possible subsets are tested to determine the most optimal one.

Our approach consists of two steps; in the first step, we partition the network into communities. This step helps to identify the anomalous nodes that invalidate the community definition. Indeed, the nodes that do not share the same properties with other nodes in the same community should be considered anomalous. To partition the network, we use a modularity-based approach for its sufficient speed and ability to handle large networks.

In the second step, we calculate the anomaly score for each node. This score is used to rank the nodes according to their degree of anomaly, considering both structure information and attribute information about the node. As a result, the anomalous nodes are ranked more highly compared with the normal nodes, which are assigned low scores. Our score is composed of two parts: (i) the community-structure-based score that considers the structure information and (ii) the attribute-based score that considers the attributes' information.

In structure-based scoring, we investigate the importance of the node based on a centrality measure. As indicated by [18], most central nodes represent the core of regularity in their communities. Furthermore, we measure the degree of node connectivity to its community. Thus, anomalous nodes within a given community tend to be less central, less important, and less connected.

In attribute-based scoring, we select the context globally by defining the most relevant attributes in the graph. We identify the relevant attributes based on their power to preserve

the local structure of the nodes. After selecting the relevant attributes, we measure the node's deviation by computing its distance to other nodes in its community.

Finally, we integrate the community-structure-based score and the attribute-based score into an aggregated anomaly score to rank the nodes in each community and detect anomalous nodes; namely, those with a higher anomaly score.

3.2. Proposed Community Anomaly Detection Approach

Our model proceeds in two phases, where the graph communities are obtained in the first phase, and the anomaly scores are associated with nodes in the second phase. The second phase combines structure-based scoring and attribute-based scoring. In this section, we present our method phases in detail with an illustrated example.

3.2.1. First Phase: Community Detection

In the first phase, we partition the graph into communities. In our model, we use a very well-known modularity-based approach to detect the communities in the graph, called the Louvain Algorithm [19]. We use the Louvain algorithm for its fast performance. The algorithm initially assigns each node to a unique community and iteratively moves each node to its neighbor to maximize modularity gain. It stops when further modularity gain is impossible. The computation of gains in modularity is defined as:

$$Gain - modularity = [\frac{S_{in} + L_{i,C}}{2w} - (\frac{S_t + L_i}{2w})^2] - [\frac{S_{in}}{2w} - (\frac{S_t}{2w})^2 - (\frac{L_i}{2w})^2] \quad (1)$$

where S_{in} is the sum of edges weights inside the community C, S_t is the sum of weights of edges connecting the different nodes in community C, L_i is the sum of edges weights of the node i, $L_{i,C}$ is the sum of weights of edges connecting the node i by C, and w denotes the sum of the weights of all the edges in the network.

3.2.2. Second Phase: Anomaly Scoring

In this phase, we consider the structure information and attribute information to characterize the communities' anomalous nodes. Therefore, the node's anomaly score has to integrate its deviation within the relevant attributes and its connections to its community. This score raises new challenges, as one has to unify the information from both components defined in this section: the deviation in the relevant attribute values and the connections within the community. Our anomaly score has two parts: the community-structure-based score and the attribute-based score, which are described in the next subsections.

- Community-structure-based score

The community-structure-based score investigates the node's importance and the node's attachment to its community. Therefore, we use the eigenvector centrality and node connectivity to calculate the community-structure-based score, which is described as:

$$StrAnomaly = 1 - Ec(v) \times con(v) \quad (2)$$

where $Ec(v)$ is its eigenvector centrality and $Con(v)$ is the connectivity of node v. The node connectivity is a measure that we introduce to assess the node's connection strength to its community compared with other nodes in the community. A high value is produced when the node has a connection to its community greater than the average connection in the community. The node connectivity is described as:

$$con(v) = \frac{nb(v)}{\frac{\sum_{j=1}^{n} nb(v_j)}{|c|}} \quad (3)$$

where nb is the node neighbors in the community, and $|c|$ is the number of nodes in the community. A high connectivity value indicates normal nodes, and a low connectivity value indicates anomalous nodes.

The eigenvector centrality [20] measures the node's importance by calculating the number and the importance of its neighbors. We apply eigenvector centrality, since this precisely corresponds to our intuition for estimating the notion of connections to important nodes and/or parts of the network, which is relevant for anomaly detection. The eigenvector centrality of a node is the sum of eigenvector centralities of its neighbors, which is defined as the following: for a given graph $G := (V, E)$ with $|V|$ nodes, let $A = (av, t)$ be the adjacency matrix, where $(av, t) = 1$ if the node v is linked to the node t , and $(av, t) = 0$ otherwise. The relative centrality score Ec of node v can be defined as:

$$E_c = \frac{1}{\lambda} \sum_{t \in m(v)} E_c(t) = \frac{1}{\lambda} \sum_{t \in G} a_{v,t} E_c(t) \tag{4}$$

where $m(v)$ is a set of the neighbors of v, $Ec(t)$ the eigenvector centrality of $t \in m(v)$, and λ is a constant. With a small rearrangement, it can be written in vector notation as the eigenvector equation:

$$AE_c = \lambda E_c \tag{5}$$

While there will be many different eigenvalues λ, only the largest positive value is considered. A high eigenvector centrality means the node has many important neighbors, and it is a normal node. The algorithm assigns an initial eigenvector centrality equal to one for all nodes. It then iterates to recalculate the eigenvector centrality (Equation (4)) until the change in the computed values between two iterations is smaller than an error tolerance equal to 1.0×10^{-6}. For example, in Figure 1, the graph has eight nodes, and is partitioned into two communities by the Louvain algorithm. In the first community, the most important node is node B, as it has the greatest number of neighbors, and it is connected to the most important nodes. In contrast, the less important node is E, as it has the least number of neighbors. Figure 2 shows the basic principle of the eigenvector centrality in simplified form, in which a node's centrality is the sum of its neighbors' centralities.

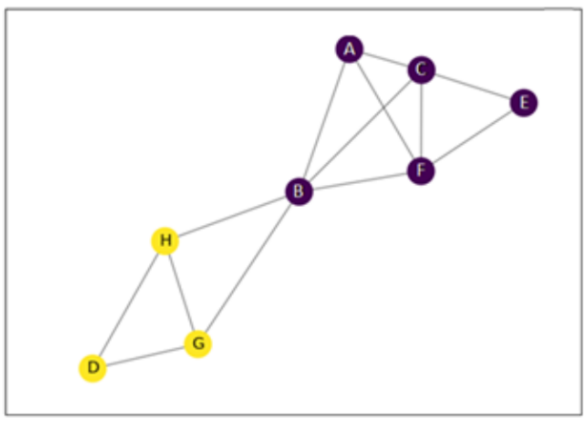

Figure 1. Example of a graph with communities.

The mathematical calculation of the eigenvector centrality of the graph's nodes is computed as:

- We present the graph with an adjacency matrix, and for eigenvector centrality,we set an initial value to all nodes equal to 1.
- We multiply the two vectors and set λ as the normalized value of the resulted vector.
- By applying Equation (4), we obtain the eigenvector centrality of the first iteration.
- We repeat the same calculation until the eigenvector values converge. The following calculation shows two iterations of the eigenvector centrality:

- The first iteration: the normalized value is (9.59).

$$\begin{bmatrix} 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 \end{bmatrix} \times \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 3 \\ 5 \\ 4 \\ 2 \\ 2 \\ 4 \\ 3 \\ 3 \end{bmatrix} /9.59 = \begin{bmatrix} 0.31 \\ 0.52 \\ 0.41 \\ 0.20 \\ 0.20 \\ 0.41 \\ 0.31 \\ 0.31 \end{bmatrix}$$

- The second iteration: the normalized value is (3.49).

$$\begin{bmatrix} 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 \end{bmatrix} \times \begin{bmatrix} 0.31 \\ 0.52 \\ 0.41 \\ 0.20 \\ 0.20 \\ 0.41 \\ 0.31 \\ 0.31 \end{bmatrix} = \begin{bmatrix} 1.34 \\ 1.75 \\ 1.44 \\ 0.62 \\ 0.82 \\ 1.44 \\ 1.03 \\ 1.03 \end{bmatrix} /3.49$$

$$= \begin{bmatrix} 0.38 \\ 0.5 \\ 0.41 \\ 0.18 \\ 0.23 \\ 0.41 \\ 0.3 \\ 0.3 \end{bmatrix} \begin{bmatrix} A \\ B \\ C \\ D \\ E \\ F \\ G \\ H \end{bmatrix}$$

As the result shows, node B has the highest value, and node D has the lowest value. When we compare nodes E and D's values, which both have two neighbors, we find that *E* has a higher value, as its neighbors have higher values (more important).

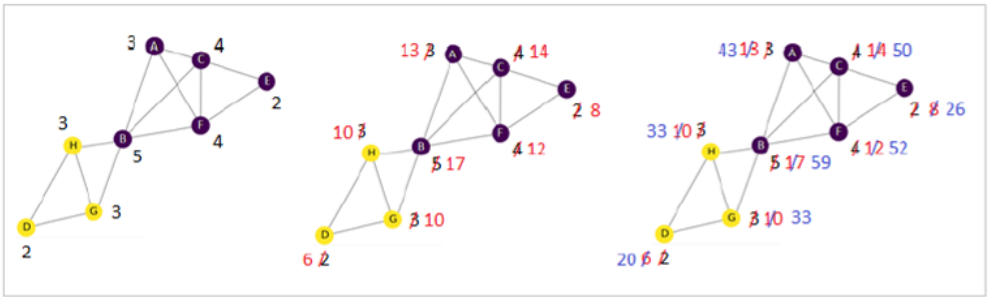

Figure 2. Iterations of the eigenvector centrality.

- Attribute-based score

The attribute-based score aims to measure the deviation of the node attributes from the other nodes in the same community in terms of a subset of relevant attributes defined by the context selection. For the selection of relevant attributes, we use the Laplacian score [21] to filter the full attributes space by ranking each attribute with a Laplacian score as follows:

$$L_r = \frac{\sum_{ij}(f_{ri} - f_{rj})^2 S_{ij}}{\sum_i (f_{ri} - U_r)^2 D_{ii}} \quad (6)$$

where $(f_{ri} - f_{rj})$ is the r^{th} attribute values for the nodes i and j, and $U_r = \sum_{i=1}^{v} \frac{f_{ri}}{v}$ denotes the mean of the r^{th} attribute. D denotes a diagonal matrix with $D_{ii} = \sum_j S_{ij}$, and S is the similarity matrix, whose elements are defined as follows:

$$S_{ij} = \begin{cases} e^{-\frac{\|(x_i - x_j)^2\|}{2\theta^2}} & \text{if } x_i \text{ and } x_j \text{ are neighbors} \\ 0 & \text{Otherwise.} \end{cases}$$

where θ is a constant, and x_i and x_j are considered as neighbors if x_i is among the k nearest neighbors of x_j in terms of Euclidean distance or x_j is among the k nearest neighbors of x_i.

As the smallest Laplacian score is preferable, the best attribute is the attribute that minimizes Equation (6) by smaller $(f_{ri} - f_{rj})$. Therefore, the best attribute is the one in which the nodes are close to each other, if, and only if, they are connected. As a result, we choose a subset with size N of the most important attributes, where N is an input parameter for our algorithm. Since the context (relevant attributes) is defined, we use these relevant attributes to estimate the attribute-based score that we define as the mean of all the attributes' scores, which is described as:

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

where n is the number of attributes in the relevant attributes set A, and $S(v, a_r)$ is the attribute score of node v for the attribute a_r, $\forall a_r \in A$, which is defined 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 denotes the other nodes in the community, $|C|$ is the number of nodes in the community, and $d(v, v_j)$ is the distance between the nodes v and v_j , which is set at 0 if the distance between the two nodes is equal and less than the mean distance (Md) or 1 otherwise.

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

where $a_r(v_i)$ is the value of the attribute a_r in the attribute vector of node v, and $Md(C_{(v)}, a_r)$ denotes the mean distance of attribute a_r in the community of node v, and it is described as follows:

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

where $(a_{r(vi)} - a_{r(vj)})$ is the distance between the node v_i and the node v_j , and p is the number of node pairs in the community.

The anomaly score $AnomalyS$ of a node v is defined as:

$$AnomalyS(v) = \alpha * StrAnomaly(v) + (1 - \alpha) * AttrAnomaly(v) \tag{10}$$

where α is a weight used for controlling the relative importance of structure-based anomalies and attribute-based anomalies. The value of α depends on the type of network.

3.2.3. Algorithm

Our algorithm, called the CAnom algorithm, is described as follows. It computes the anomaly score of each node in four steps: (1) determine the node community, (2) calculate its community-structure-based score, (3) select its relevant attributes, and (4) compute its attribute-based score. Finally, all nodes are ranked based on their anomaly score. As parameters, we require the number of attributes to select (N), which determines how many relevant attributes should be selected according to their Laplacian score. In addition,

α in Equation (10) refers to the weight of the structure-based anomaly and the attribute-based anomaly in the anomaly score.

Firstly, we partition the graph into communities using the Louvain algorithm (line 2). Then, we calculate the eigenvector centrality of each node (line 3). We iterate over each node to compute its connectivity before the community-structure-based score is assigned (line 6). Thereafter, we calculate the Laplacian score for each attribute (line 8), then we select N attributes with the lowest Laplacian score to be the relevant attributes (line 9). Next, we iterate over each community and use the relevant attributes to compare the distance between the node and its community nodes with the mean distance for the community (lines 10–16). Finally, we unify the community-structure-based score and attributes-based score (lines 17–19) and return the anomaly score of the nodes from G (line 20). Our community-based detection algorithm is called CAnom (Algorithm 1), an amalgamation of the words community and anomaly.

Algorithm 1 CAnom

Require: $G : (V, E)$, A: Attributes, N : number of select attributes
Ensure: Ranking of all $v \in V$
1: Initialize empty vectors *str_anomaly*, *atr_anomaly*, *total_anomaly* for all $v \in V$
2: C $\leftarrow$ Partition(G)
3: EC $\leftarrow$ eigenvector centrality for all $v \in G$
4: For each $v \in G$ do
5: Compute connectivity (v) (Equation (6))
6: Compute StrAnomaly score (v) (Equation (2))
7: End For
8: For all $a_r \in A$ calculate Laplacian score (Equation (6))
9: A' $\leftarrow$ subset of N attributes with the lowest Laplacian score
10: For each community C_k in C do
11: Md $\leftarrow$ mean values of attributes from A' in C_k (Equation (9))
12: For each v_i in C_k do
13: $s_{vi} \leftarrow$ a dictionary containing for each attribute a_r of v_i its anomaly score
14: Compute attrAnomaly[v] (Equation (7))
15: End For
16: End For
17: For each $v \in G$ do
18: AnomalyS[v] = α strAnomaly[v] + $(1 - \alpha)$ attrAnomaly[v] (Equation (10))
19: End For
20: Return AnomalyScore

3.2.4. Complexity Analysis

First, we partition the graph into communities using the Louvain algorithm, which costs $O(vlogv)$, where v is the number of nodes in the graph. After that, we calculate the eigenvector centrality that costs $O(v+e)$, where e is the number of edges in the graph. Then, we compute the StrAnomaly score with the linear cost with the node number. Next, we define the context using the Laplacian score, which costs $O(mv^2)$, where m is the number of attributes. After that, we compute the AtrrAnomaly score that costs $O(v^2)$. Overall, the CAnom algorithm (Algorithm 1) has a computational complexity of $O(max(mv^2, v+e))$. The worst case occurs when all the graph nodes are assigned to one community, as a quadric analysis is performed for each community. Therefore, in a real network where the number of communities is high, the algorithm's performance increases.

4. Experimental Evaluation of Performances

To extensively research the performance of CAnom (Algorithm 1), we compared it with two state-of-the-art anomaly detection algorithms. Before giving the results of this comparison, we briefly introduce the competing methods.

- CODA [10] is one of the most famous models used to detect anomalies in social network communities. The model detects communities and identifies anomalous nodes in one step. It uses the full attributes set of the nodes.
- ConSub [22] is based on a statistical selection of a subset of nodes' attributes that show dependencies with the graph structure. The selection of a subset of attributes combines with a distance-based outlier model called DistOut to detect anomalous nodes in the communities.

4.1. Evaluation Measures

To evaluate the quality of the community anomaly detection model and check its validity on real and synthetic datasets, we perform a comparison between the obtained nodes' ranks of the model and the ground truth. The closer the rank is to the true label, the more efficient the community-based anomaly detection model is. The first measure used is Area Under The Curve (AUC). This is one of the most important performance measurements for classification models and anomaly ranking. It tests the model's capability to distinguish between two classes, and a higher AUC means a better model. In a machine learning classification problem, the model predicts classes compared with the actual classes, and the results fall into four areas. These are either true positive (predicted anomaly for an actual anomaly), false positive (predicted anomaly for an actual normal), true negative (predicted normal for an actual normal), or false negative (predicted normal for an actual anomaly). The proportion of positives that are correctly identified is called sensitivity, and the proportion of negatives that are correctly identified is called specificity. In the classification model, we define various thresholds and plot the Sensitivity (also called true positive rate) with the false positive rate, which is (1-Specificity), to plot the ROC curve . Therefore, the best model is the one that can reach the maximum value of sensitivity and specificity at a certain threshold value, which means all positives are correctly identified and all negatives correctly identified. The maximum value is the one located in the top-left corner of the ROC plot. As a result, the area under the curve measures the ROC curve's ability to reach the maximum value of sensitivity and specificity (top-left corner). The second measure used in this evaluation is the model's runtime, as it is important to check the model's capability to detect the community anomaly within a reasonable time. A lower runtime is preferable when it is combined with good-quality results.

4.2. Community Anomaly Detection in Real Datasets

In this section, we test our model on two real networks: the Disney network and the Book network and compare our model's performance with those of the CODA and ConSub models. Our proposed model has only two parameters:

- The α in the Equation (10) that we set at 0.5 to give the same weight to the structure-based anomaly and the attribute-based anomaly.
- The number of attributes to include. We set the second parameter to be the half-number of attributes and a maximum of 10 attributes, as we observed that a higher number of attributes increase the runtime without significant improvement in quality.

For CODA and ConSub parameter settings, we tested many values. The ones that give the best results are described in Table 1.

Table 1. Models Parameter setting.

CODA	Communities number (K) = 8	Percentage of anomaly (r) = 0.05	link importance (λ) = 0.01
ConSub	Interval size (K) = 10	The number of Monte Carlo iterations (M) = 150	The significance level (α) = 0.05

4.2.1. Disney Network

The Disney network was extracted from the Amazon co-purchase network considering only Disney DVDs. The network consists of 124 nodes and 334 edges, and each product in

the graph is described by 30 attributes, such as product prices, review ratings, and so on. Even though the network is small, it is an interesting dataset due to its complex graph and attribute structure, and it is used to evaluate most of the anomaly detection models. For this real-world dataset, the ground truth of objects as anomalies or normal is not available. A user experiment was carried out to provide the ground truth of the dataset where high school students manually labeled each object as either an anomaly or normal.

Figure 3 illustrates the results of our model on the Disney network compared with CODA and ConSub. The results show that the CAnom model achieves good-quality results, with an AUC equal to 0.92 (see Figure 4). In comparison, the other models obtain lower-quality results, with AUCs equal to 0.82 and 0.50 for ConSub and CODA, respectively. The ConSub model achieves good-quality results compared with the CODA model, which achieves the worst-quality results. The runtime evaluation is presented in Figure 3, which shows the runtime for the three models. The CAnom model shows the best runtime with 0.05 s, followed by the CODA model with 6.05 s. The ConSub model is the slowest model, at 8.93 s. From this experiment on the Disney network, we can conclude that, among the three models, the CAnom model achieves the highest-quality results for detecting community anomalies. In addition, the CAnom model is the fastest of the three. While the ConSub model achieves good-quality results, it suffers from a slow runtime, and the CODA model is considered the worst-performing model for identifying community anomalies in a real-world network.

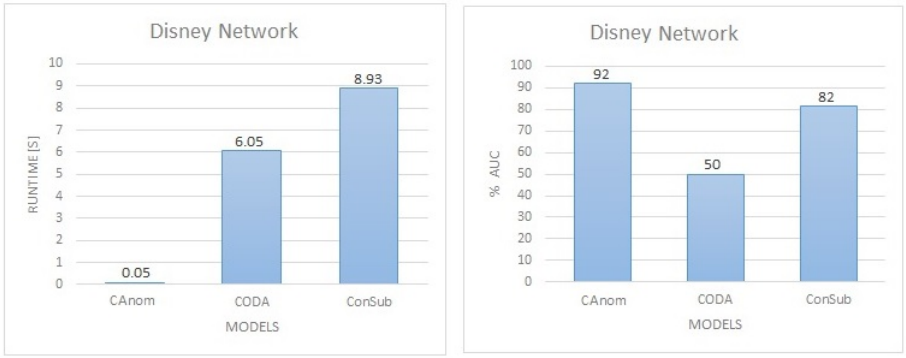

Figure 3. Quality and runtime evaluation for the Disney network.

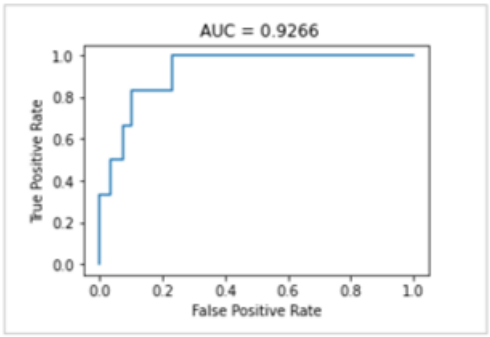

Figure 4. AUC and ROC curve of the Disney network.

4.2.2. Book Network

The Book network was extracted from the Amazon co-purchase network and included books labeled with the tag 'amazon fail' by the users [23]. Amazon allowed customers to tag products to categorize them, and several tags were used to indicate suspicious products, such as the 'amazon fail' tag. The network consists of 1468 nodes and 3695 edges, and each object is described with 28 attributes. This dataset's ground truth was created by considering a particular book an anomaly when it was labeled as 'amazon fail' by at least 20 users.

Figure 5 illustrates the quality of the CAnom, CODA, and ConSub models by calculating the AUC. The CAnom model achieves the best-quality results compared with the other models, with an AUC equal to 0.68 (see Figure 6), while CODA achieves the worst quality, with AUC equal to 0.53 . The runtime evaluation in the Book network shows that the CAnom model achieves the best results, with an execution time equal to 1.79 s, which outperforms the other models. The CODA model is the slowest, at 36 s (see Figure 5).

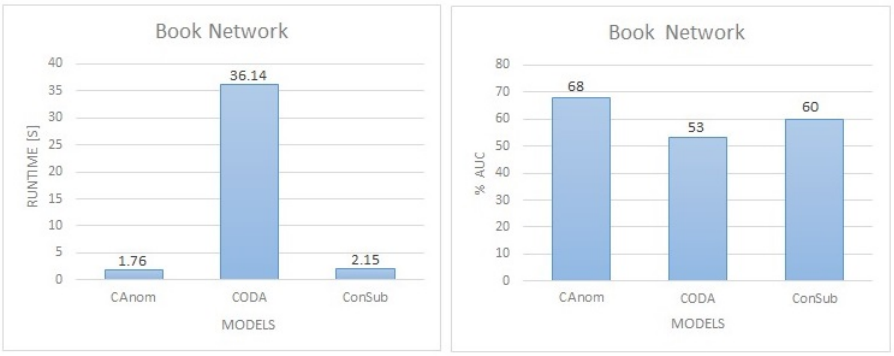

Figure 5. Quality and runtime evaluation for the Book network.

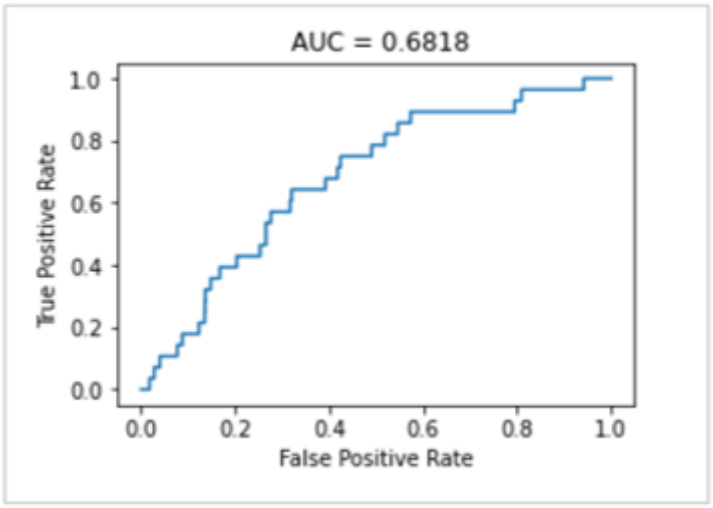

Figure 6. AUC and ROC curve of the Book network.

4.3. Community Anomaly Detection in Synthesis Data

The evaluation of the anomaly detection models is not an easy task due to the rarity of suitable datasets that include anomalies and the lack of ground truth that informs us which data points are true anomalies. Therefore, synthetic datasets are generally made for performance evaluation. These datasets are used to compare a model's behavior on real data against the generated data. We used generated synthetic datasets of different sizes and attribute numbers based on [22]. For the reproduction of the properties observed in real networks, the graph is generated by following a power-law distribution. The node attributes are divided into relevant and irrelevant attributes by 50%. For relevant attributes, nodes were assigned values from a Gaussian distribution, while for the irrelevant attributes, nodes were assigned values from a uniform distribution. To ensure no deviating values were present in the relevant attributes, a hyper-ellipsoid was used to cut the tails of each Gaussian distribution. For anomalous nodes, the values of their attributes were manipulated to be random values outside the range defined by the hyper-ellipsoids of their communities. The anomaly ratio is 10%, which determines the number of anomalous nodes in the communities. The anomalies are only detected by considering combinations of at

least two relevant attributes. Each synthetic dataset contains two files: the graphml file contains the graph with its nodes attributes, and the true file contains the ground truth of nodes, where 1 indicates anomalous nodes and 0 indicates normal nodes. For the evaluation of the model with respect to the increasing numbers of attributes, we use synthetic datasets with 1000 nodes and different attributes 2, 10, 20, 40, 60, and 80. For the evaluation of the model with respect to increasing the network's size, we use the synthetic datasets with ten attributes and different numbers of nodes. For the parameter setting, we set our model parameter to be the half-number of attributes to measure the effect of attribute increase, while we used the same setting in the real network experiments for the other models.

4.3.1. First Experiment

We evaluate the quality of our model with an increasing number of attributes by using the AUC with datasets of 1000 nodes. Figure 7 shows the variation of the AUC of the tested models when the number of attributes increases. We can conclude that our model has the best AUC compared with ConSub and CODA, which is between 0.87 and 0.97. The runtime evaluation is illustrated in Figure 8, which shows the runtimes with an increasing number of attributes. Both CAnom and ConSub show the best scalability with respect to the attributes' increase, while the CODA runtime roughly increases with the number of attributes. It is important to note that matrix operations are costly, and in CODA, these operations are performed for each attribute, which increases its runtime.

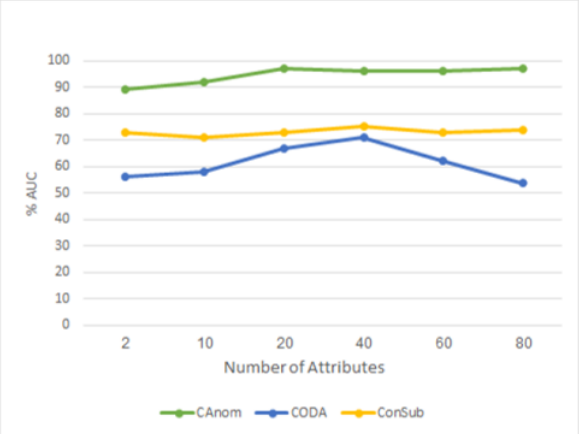

Figure 7. Variations in AUC according to the different number of attributes for each of the tested models.

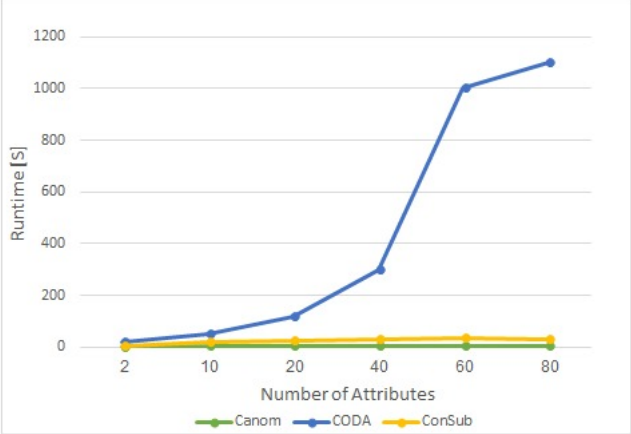

Figure 8. Variations in runtime according to the different number of attributes for each of the tested models.

4.3.2. Second Experiment

The model evaluation with the increased size network is conducted with networks of 500, 1000, 2000, 3500, 6000, and 10,000 nodes. The AUC of the CAnom model starts from 0.89 in the smallest network to 0.96 in the largest network. In fact, CAnom has the highest AUC compared with the other models (see Figure 9). The evaluation of the runtime with the increasing size of the network is shown in Figure 10. While ConSub shows faster runtimes overall in comparison with the other models, the CAnom has a better runtime than CODA.

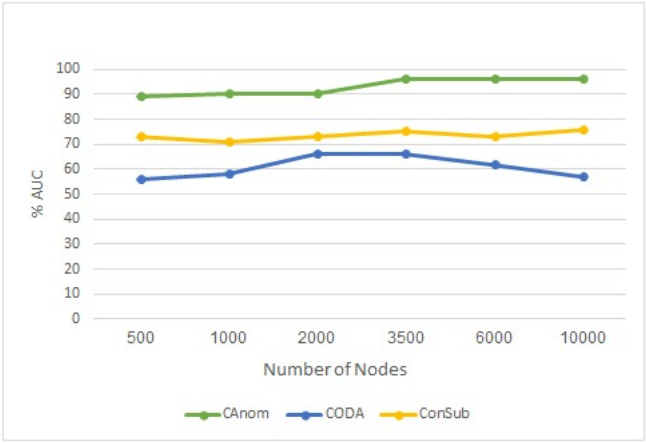

Figure 9. Variations in AUC according to the different number of nodes for each of the tested models.

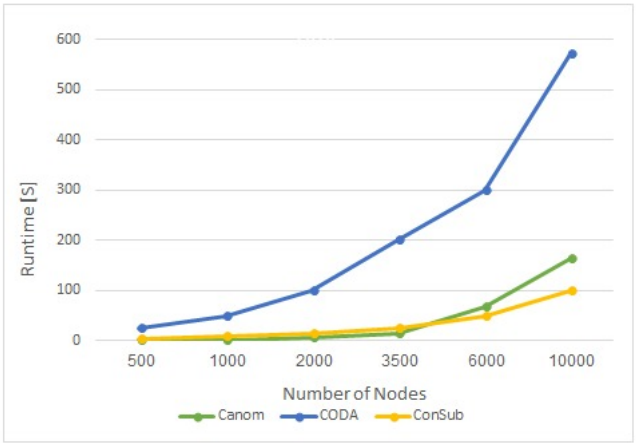

Figure 10. Variations in runtime according to the different number of nodes for each of the tested models.

4.3.3. Dissection of Results

From the analysis of these results, and after applying the CAnom model on synthesized data and comparing it with CODA and ConSub, it can be concluded that the proposed model gives the best result. The model achieves good-quality results and high scalability with the attribute number increasing, as it defines the relevant attributes of the network instead of considering the full attribute space. Therefore, the model is suitable for today's applications, where the number of attributes is rising. The ConSub model achieves better results than CODA, as it also defines the context of the network, while CODA considers all of the network attributes.

5. Conclusions

In this paper, we are interested in detecting anomalies that deviate from their communities in contrast to normal users that commonly share many properties with their fellow

community members. Therefore, in this model, we defined an anomaly ranking score that integrates the deviation in the network structure and the attributes' values. Instead of considering all the attributes, we use the most relevant attributes. Our model has two steps: detecting communities in the network and ranking the network's nodes based on their anomalies. The anomaly score combines the community-structure-based score that measures node importance and connectivity and the attribute-based score that compares the node distance to other nodes in its community with the community mean distance. The final anomaly score represents the node anomaly degree, where anomalous nodes have the highest scores.

Due to the integration of both structure information and attribute information in our ranking score, our model easily identified complex anomalies, which deviated in either their structure, the values of their attributes, or both. The integration of these properties of the anomaly presents a new challenge of unifying the information from both parts to ensure complex anomaly detection. Thus, we balance quality with efficiency by joining the graph structure with the attribute information. In our approach, we did not include all the network attributes, as the existence of irrelevant attributes is inevitable in the networks. Instead, we selected the context which is a subset of the relevant attributes to brighten up any deviation in such values. Experiments on both real-world and synthetic datasets illustrate that our model achieves good performance, as it outperforms the existing models and can detect anomalies that other models miss. An extension of our proposed approach could be carried out in many different ways. In our approach to anomaly detection, we focus on numerical node attributes; however, node attributes in real-world networks consist of different attribute types. Therefore, a mixture of attribute types such as binary, categorical, and continuous values could be considered for inclusion into future models.

Author Contributions: Conceptualization, H.Z. and W.K.; Funding acquisition, H.K.; Investigation, H.Z. and R.M.; Methodology, H.Z. and N.S.A.; Project administration, W.K.; Supervision, H.K. and N.S.A.; Writing—original draft, H.Z. and W.K.; Writing— review & editing, H.Z. and H.K. All authors have read and agreed to the published version of the manuscript.

Funding: This research project was funded by the Deanship of Scientific Research, Princess Nourah bint Abdulrahman University, through the Program of Research Project Funding After Publication, grant No. (43-PRFA-P-9).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The datasets generated during and/or analysed during the current study are publicly available. The dataset and code is available from the corresponding author on reasonable request.

Acknowledgments: Deanship of Scientific Research, Princess Nourah bint Abdulrahman University, through the Program of Research Project Funding After Publication, grant No. (43-PRFA-P-9).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Fire, M.; Goldschmidt, R.; Elovici, Y. Online Social Networks: Threats and Solutions. *IEEE Commun. Surv. Tutor.* **2014**, *16*, 2019–2036. [CrossRef]
2. Abdul, O. Linkcalculator—An Efficient Link-Based Phishing Detection Tool. *Acta Inform. Malays.* **2020**, *4*, 37–44. [CrossRef]
3. Lin, Z.; Wang, H.; Li, S. Pavement anomaly detection based on transformer and self-supervised learning. *Autom. Constr.* **2022**, *143*, 104544. [CrossRef]
4. Hassanzadeh, R.; Nayak, R.; Stebila, D. Analyzing the Effectiveness of Graph Metrics for Anomaly Detection in Online Social Networks. In Proceedings of the 13th International Conference on Web Information Systems Engineering, WISE'12, Paphos, Cyprus, 28–30 November 2012; Springer: Berlin/Heidelberg, Germany, 2012; pp. 624–630. [CrossRef]
5. Kaur, R.; Singh, S. A Comparative Analysis of Structural Graph Metrics to Identify Anomalies in Online Social Networks. *Comput. Electr. Eng.* **2017**, *57*, 294–310. [CrossRef]

6. Chaudhary, A.; Mittal, H.; Arora, A. Anomaly Detection using Graph Neural Networks. In Proceedings of the 2019 International Conference on Machine Learning, Big Data, Cloud and Parallel Computing (COMITCon), Faridabad, India, 14–16 February 2019; pp. 346–350. [CrossRef]
7. Rezaei, A.; Kasirun, Z.M.; Rohani, V.A.; Khodadadi, T. Anomaly detection in Online Social Networks using structure-based technique. In Proceedings of the 8th International Conference for Internet Technology and Secured Transactions (ICITST-2013), London, UK, 9–12 December 2013; pp. 619–622.
8. Wang, H.; Gao, Q.; Li, H.; Wang, H.; Yan, L.; Liu, G. A Structural Evolution-Based Anomaly Detection Method for Generalized Evolving Social Networks. *Comput. J.* **2020**, *65*, 1189–1199. [CrossRef]
9. Chaparro, C.; Eberle, W. Detecting Anomalies in Mobile Telecommunication Networks Using a Graph Based Approach. In Proceedings of the FLAIRS Conference, Hollywood, FL, USA, 18–20 May 2015.
10. Gao, J.; Liang, F.; Fan, W.; Wang, C.; Sun, Y.; Han, J. 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, KDD '10, Washington, DC, USA, 24–28 July 2010; Association for Computing Machinery: New York, NY, USA, 2010; pp. 813–822. [CrossRef]
11. Zimek, A.; Schubert, E.; Kriegel, H.P. A survey on unsupervised outlier detection in high-dimensional numerical data. *Stat. Anal. Data Min. ASA Data Sci. J.* **2012**, *5*, 363–387. [CrossRef]
12. Moser, F.; Colak, R.; Rafiey, A.; Ester, M. Mining Cohesive Patterns from Graphs with Feature Vectors. In Proceedings of the 2009 SIAM International Conference on Data Mining, Sparks, NV, USA, 30 April–2 May 2009; Volume 593–604. pp. 593–604. [CrossRef]
13. Sánchez, P.I.; Müller, E.; Irmler, O.; Böhm, K. 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, SSDBM '14, Aalborg, Denmark, 30 June–2 July 2014; Association for Computing Machinery: New York, NY, USA, 2014. [CrossRef]
14. Ji, T.; Gao, J.; Yang, D. A Scalable Algorithm for Detecting Community Outliers in Social Networks. In Proceedings of the 13th International Conference on Web-Age Information Management, Habin, China, 18–20 August 2012; pp. 434–445. [CrossRef]
15. Sun, C.; Li, Q.; Li, H.; Zhang, S.; Zheng, Y. Community Outlier Based Fraudster Detection. In Proceedings of the Knowledge Science, Engineering and Management, Melbourne, VIC, Australia, 19–20 August 2017; Li, G., Ge, Y., Zhang, Z., Jin, Z., Blumenstein, M., Eds.; Springer International Publishing: Cham, Switzerland, 2017; pp. 410–421.
16. Win, H.; Lynn, K. Community Detection in Social Network with Outlier Recognition. *Adv. Sci. Technol. Eng. Syst. J.* **2018**, *3*, 21–27. [CrossRef]
17. Win, H.; Lynn, K. *Community and Outliers Detection in Social Network*; Springer: Singapore, 2019; pp. 58–67. [CrossRef]
18. Müller, E.; Sánchez, P.I.; Mülle, Y.; Böhm, K. Ranking outlier nodes in subspaces of attributed graphs. In Proceedings of the Workshops Proceedings of the 29th IEEE International Conference on Data Engineering, ICDE 2013, Brisbane, QLD, Australia, 8–12 April 2013; Chan, C.Y., Lu, J., Nørvåg, K., Tanin, E., Eds.; IEEE Computer Society: Washington, DC, USA, 2013; pp. 216–222. [CrossRef]
19. Blondel, V.D.; Guillaume, J.L.; Lambiotte, R.; Lefebvre, E. Fast unfolding of communities in large networks. *J. Stat. Mech. Theory Exp.* **2008**, *2008*, P10008. [CrossRef]
20. Newman, M. *Networks*; OUP: Oxford, UK, 2018.
21. He, X.; Cai, D.; Niyogi, P. Laplacian Score for Feature Selection. In Proceedings of the NIPS, Vancouver, BC, Canada, 5–8 December 2005; pp. 507–514.
22. Sánchez, P.I.; Müller, E.; Laforet, F.; Keller, F.; Böhm, K. Statistical Selection of Congruent Subspaces for Mining Attributed Graphs. In Proceedings of the 2013 IEEE 13th International Conference on Data Mining, Dallas, TX, USA, 7–10 December 2013; Xiong, H., Karypis, G., Thuraisingham, B., Cook, D.J., Wu, X., Eds.; IEEE Computer Society: Washington, DC, USA, 2013; pp. 647–656. [CrossRef]
23. Iglesias Sánchez, P. Context Selection on Attributed Graphs for Outlier and Community Detection. Ph.D. Thesis, Karlsruher Institut fur Technologie, Karlsruhe, Germany, 2015. [CrossRef]