A Deep Multi-View Framework for Anomaly Detection on Attributed Networks

Zhen Peng, Minnan Luo, Jundong Li, Luguo Xue, and Qinghua Zheng

Abstract—The explosion of modeling complex systems using attributed networks boosts the research on anomaly detection in such networks, which can be applied in various high-impact domains. Many existing attempts, however, do not seriously tackle the inherent multi-view property in attribute space but concatenate multiple views into a single feature vector, which inevitably ignores the incompatibility between heterogeneous views caused by their own statistical properties. Actually, the distinct but complementary information brought by multi-view data promises the potential for more effective anomaly detection than the efforts only based on single-view data. Furthermore, the abnormal patterns naturally behave diversely in different views, which coincides with people's desire to discover specific abnormality according to their preferences for views (attributes). Most existing methods cannot adapt to people's requirements as they fail to consider the idiosyncrasy of user preferences. Therefore, we propose a multi-view framework ALARM to incorporate user preferences into anomaly detection and simultaneously tackle heterogeneous attribute characteristics through multiple graph encoders and a well-designed aggregator that supports self-learning and user-guided learning. Experiments on synthetic and real-world datasets, e.g., Disney, Books, and Enron, corroborate the improvement of ALARM in detection accuracy evaluated by the AUC metric and its effectiveness in supporting user-oriented anomaly detection.

Index Terms—Anomaly detection, attributed networks, graph convolutional networks, unsupervised learning

1 INTRODUCTION

ANOMALY detection concerns with identifying rare, unexpected, and suspicious instances among a swarm of normal instances [1]. It has significant implications in helping practitioners and decision-makers to discover, manage, and circumvent abnormal patterns from data, and has a wide spectrum of applications such as spam detection, financial fraud detection, and intrusion detection in cybersecurity [2]. In the past few decades, many techniques have been developed for spotting anomalies in multi-dimensional data points or plain networks. As attributed networks become a prevalent tool in modeling real-world complex systems (e.g., social media networks, protein-protein interaction networks, and collaboration networks), anomaly detection on attributed networks has attracted a surge of research attention. Compared to plain networks, the additional node attributes enable attributed networks to hold richer semantic information. Thus, how to further improve the performance of anomaly detection and how to steer the detection process with the help of rich semantics become a vital research issue.

- *Zhen Peng, Minnan Luo, Luguo Xue, and Qinghua Zheng are with the Ministry of Education Key Lab of Intelligent Networks and Network Security, National Engineering Lab for Big Data Analytics, School of Computer Science and Technology, Xi'an Jiaotong University, Shaanxi 710049, China. E-mail: zhenpeng27@outlook.com, {minnluo, qhzheng}@xjtu.edu.cn, luguoxuecx@gmail.com.*
- *Jundong Li is with the Department of Electrical and Computer Engineering, Department of Computer Science, School of Data Science, University of Virginia, Charlottesville, VA 22904 USA. E-mail: jundong@virginia.edu.*

Manuscript received 28 June 2019; revised 3 July 2020; accepted 31 July 2020. Date of publication 7 Aug. 2020; date of current version 29 Apr. 2022. (Corresponding author: Minnan Luo.)
Recommended for acceptance by Muhammad Aamir Cheema.
Digital Object Identifier no. 10.1109/TKDE.2020.3015098

On one hand, it should be recognized that most data for analytical tasks are collected from different sources or gathered from multiple feature extractors [3]. In other words, data instances are usually depicted by heterogeneous feature spaces in the form of multiple views. For instance, in Fig. 1, several types of data (views) can be utilized to describe users in social media such as Twitter: basic personal information like email address, location, and phone number, following/follower information, texts they post, images they upload and so on. As another example, a video clip is the combination of audio and image signals which can be seen as two different views of the video. Since each view suffices for mining knowledge and multiple views provide distinct but complementary information, how to cope with the multi-view data skillfully and avoid the incompatibility issue naturally becomes a challenge for anomaly detection. Nonetheless, most of the existing efforts [4], [5], [6], [7] neglect the inherent multi-view property of data, while treating all attributes equivalently by concatenating different views into a single vector. This treatment is not physically meaningful and may adversely affect the anomaly detection performance as different views may correspond to different patterns (texts and images) and have their own unique statistical properties.

On the other hand, there are different types of anomalies in the dataset, such as structural, contextual, and community anomaly [6]. As both of the latter two are determined using attribute values within a specific context (view), an instance might be considered as an anomaly in a given context, while remaining normal in a different context. Fig. 1 gives a toy example to illustrate it. In the third view, u_3 and u_6 (marked with red) are spotted as anomalies due to their published malicious comments, while their following/follower information is normal in the second view and u_5 is seen as abnormal because of too many followers. Interestingly, the above property of

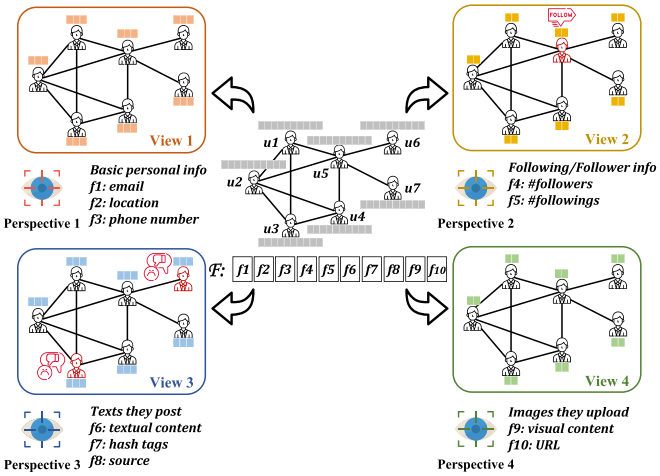


Fig. 1. A toy example to illustrate diverse types of data (views) for social media users.

anomalies provides a realistic basis for user-oriented anomaly detection. Since different users may focus on different views based on their own needs, the anomalies shown in different focused views naturally correspond to their own interests. For example, when regulators intend to locate malicious users who disseminate misinformation, they are more likely to focus on the third and fourth views which include large amounts of user-published text and image information. In contrast, the remaining two views seem to be unrelated to their purpose and can be excluded by regulators. In this case, u_3 and u_6 will be recognized as anomalies with a higher probability than u_5. Obviously, most of the existing methods do not allow users to steer anomaly detection since they either use all attributes [4], [6], [8] or perform unsupervised feature selection [7], [9], [10], which implies they fail to consider user needs. In order to satisfy the practical demands, incorporating user preferences into anomaly detection which enables users to guide the detection process becomes another challenge that ought to be overcome.

To tackle the aforementioned challenges, in this paper, we propose a deep multi-view framework for anomaly detection on attributed networks, named ALARM, which includes a multi-view attributed network encoder, an aggregator and a structure & attribute decoder. For the encoder, we resort to GNNs to model attributed networks because of their good capability of encoding both network topology and node attributes for graph mining [11]. And the aggregator is designed to support two modes: self-learning and user-guided learning. In the latter mode, user-provided preference vectors would be incorporated into the learning task so that users can steer anomaly detection based on their interests. Specifically, ALARM encodes the given attributed network with multiple GNNs to tackle multi-view attributes, the extracted view-based hidden embeddings are passed through a well-designed aggregator to generate unified representations and then are decoded from two aspects of structure and attributes. The reconstruction errors post the encoder and decoder phases are used for spotting anomalies. The main contributions of our work are as follows:

- *Problem Formulations*: we formally define the problem of anomaly detection on multi-view attributed networks supporting self-learning and user-guided learning. The key idea is to encode the given attributed network under different views separately, and then aggregate the view-based embeddings into a new unified latent space using self-learning parameters or user preferences.
- *Algorithms*: we propose a deep multi-view framework ALARM for anomaly detection on attributed networks, which provides two mechanisms: self-learning and user-guided learning. We also give the time complexity analysis of the proposed framework.
- *Experimental Findings*: we perform experiments on both synthetic and real-world datasets, and the experimental results show that the proposed framework ALARM achieves the best performance in most cases. Also, the case study demonstrates the effectiveness of ALARM in incorporating user preferences into anomaly detection.

2 Related Work

In line with the focus of our work, we briefly review the related work in the following areas: (1) multi-view representation learning, (2) deep learning methods for graphs, (3) anomaly detection on networks. A brief summary for related works is shown in Table 1.

2.1 Multi-View Representation Learning

This area aims to investigate the problem of embedding inputs from the multi-view data to a new shared latent space for extracting useful representations. Representative examples in the early study include canonical correlation analysis (CCA) [12] and its kernel extensions [29], [30], which have limitations on capturing high-level interactions between multi-view data in spite of empirical success. With the explosion of deep learning, various deep architecture-based models such as multi-modal deep Boltzmann machines [16], multi-modal deep autoencoders [17], and multi-modal recurrent neural networks [31] are proposed to ease the above restrictions. In general, the key point of multi-view representation learning lies in learning a good interaction mechanism between multi-view data. Typically, most of the existing methods capture the interaction among multiple views through feature alignment which seeks to conduct alignment between representations learned from different views or representation fusion that fuses separate view-based features into a single compact representation [32].

2.2 Deep Learning Methods for Graphs

The model of graph neural networks (GNNs) is first outlined in [21]. However, this method is costly due to the adoption of expensive neural "message-passing" algorithms. Inspired by the work of [33] which provides a version of graph convolutions based on spectral graph theory, in recent years, a number of approaches concentrate on the improvements, extensions, and approximations of spectral convolutions [22], [34], [35], [36]. Among them, one of the most prominent work is the graph convolutional networks (GCNs) [22]. The core idea of GCNs lies in aggregating features from local graph neighborhoods to generate a new feature representation for a given node. Subsequently, graph

TABLE 1
A Brief Summary of Related Works on Different Topics

Topics	Algorithms	Comments/Limitations
Multi-View Representation Learning	Multi-View Representation Alignment Methods [12], [13], [13], [14], [15]	Performing feature alignment between representations learned from different views
	Multi-View Representation Fusion Methods [16], [17], [18], [19], [20]	Fusing separate features learned from distinct views into a single compact representation
Deep Learning Methods for Graphs	GNN [21]	Computationally expensive
	GCN [22]	Full-batch learning, transductive setting
	GAT [23]	Full-batch learning, computationally expensive
	GraphSAGE [24]	Inductive setting, relatively slow training speed
	DIFFPOOL [25]	Graph-level representation, slow convergence
Anomaly Detection on Networks	SCAN [26]	Spotting structural anomalies, neglecting attributes
	OddBall [27]	Spotting structural anomalies, neglecting attributes
	CODA [4]	Using all attributes, no feature selection
	Radar [6]	Using all attributes, no feature selection
	ConOut [9]	With subspace selection, computationally expensive
	ANOMALOUS [7]	With attribute and instance selection, high memory cost
	FOCUSCO [28]	Incorporating user preferences, time-consuming
	DOMINANT [8]	GCN based, using all attributes, no feature selection

attention networks (GATs) [23] achieves the goal by specifying different weights to different nodes in a neighborhood. Recently, the number of various deep algorithms on graphs, from semi-supervised or unsupervised graph representation learning methods [24], [37] to graph pooling algorithms [25], [38], has been growing rapidly. In this work, we would transform GNN-like models into a deep multi-view architecture that integrates data from diverse views to explore better anomaly detection performance.

2.3 Anomaly Detection on Networks

Generally, the existing techniques can be broadly divided into two categories according to the type of networks they work on. The first is anomaly detection on plain networks. Utilizing structure information such as node degree and subgraph centrality to spot anomalies is a major characteristic of algorithms oriented plain networks since the network topology is the only available information in such networks. Among them, SCAN [26] clusters vertices based on a structural similarity measure while detecting hubs and outliers. OddBall [27] spots anomalous nodes based on egonet patterns. As attributed networks are widely used in modeling various real information systems, many detection algorithms [4], [5], [39], [40] designed for such networks are put forward, forming the second class of methods. Among them, CODA [4] identifies communities and detects community anomalies simultaneously via a unified probabilistic model. Radar [6] adopts residual analysis to detect anomalies in an unsupervised manner. And DOMINANT [8] can be regarded as its deep learning vesion. These methods, nonetheless, enforce the Homophily assumption [41] in all attributes, which is usually untenable in real scenarios. To overcome this issue, some approaches conduct subspace selection [7], [10], [42] to find relevant attributes for anomaly detection. Among them, ConOut [9] decides a subgraph and statistically relevant subset of attributes for each node and then detects anomalies in selected local context. ANOMALOUS [7] achieves anomaly detection and attribute selection simultaneously via CUR factorization and residual analysis. Different from the above methods, FOCUSCO [28] incorporates user preferences into graph mining, then performs clustering and personalized anomaly detection based on the preference vector. Despite the empirical success, it requires users to provide a set of exemplar nodes, which means that users need to have some prior knowledge of the datasets and spend much time labeling them manually. Obviously, this method is laborious and time-consuming in implementation. By contrast, our proposed framework is more flexible, supporting self-learning and user-guided learning, users can decide whether to introduce preferences based on their own needs.

3 PROBLEM FORMULATION

Following the commonly used notations, we adopt bold uppercase characters (e.g., $\mathbf{A}$) to denote matrices, bold lowercase characters (e.g., $\mathbf{b}$) to indicate vectors and normal lowercase characters (e.g., c) as scalars. Sets are represented by uppercase calligraphic letters (e.g., $\mathcal{V}$) and $|\mathcal{V}|$ for the size of a set $\mathcal{V}$. Given a matrix $\mathbf{A} \in \mathbb{R}^{n\times d}$, we use $\mathbf{a}_i$ and $\mathbf{A}_{ij}$ to denote its ith row and (i, j)th entry, respectively. As for the vector and matrix norms, the ℓ_2 norm of a vector is denoted by $\|\cdot\|_2$ and the Frobenius norm of matrix $\mathbf{A}$ is written as $\|\mathbf{A}\|_F = \sqrt{\sum_{i=1}^{n}\sum_{j=1}^{d}\mathbf{A}_{ij}^2}$. And then, we define the multi-view attributed network as follows:

Definition 1 (Multi-View Attributed Networks). *A multi-view attributed network $\mathcal{G}(\mathcal{V}, \mathcal{E}, \mathcal{F})$ consists of $|\mathcal{V}| = n$ nodes and $|\mathcal{E}| = m$ edges. Meanwhile, each node $v_i \in \mathcal{V}$ is affiliated with a set of d-dimensional attributes (features) $\mathcal{F} = \{f_1, f_2, \ldots, f_d\}$ which can be represented by k distinct feature spaces with k views. For example, in Disney co-purchase network [43], each film possesses a large number of descriptive attributes which are roughly extracted from four viewpoints and form four different views including* price information, rating information, review information *and* other product information, *details are shown in Fig. 2.*

According to the definition, we adopt an adjacency matrix $\mathbf{A} \in \mathbb{R}^{n\times n}$ to record the structure topology of $\mathcal{G}$,

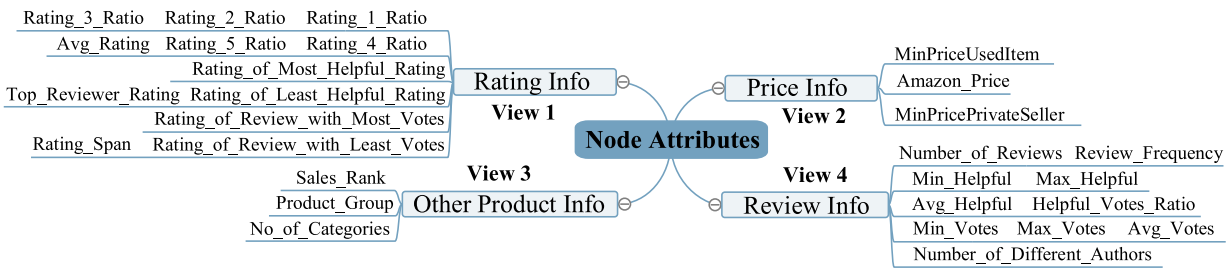


Fig. 2. A brief description of multi-view attributes collected from four perspectives on Disney dataset.

where $\mathbf{A}_{ij} = 1$ indicates node v_i and v_j are connected with each other. In this work, we mainly focus on undirected networks where an edge is an unordered pair of two nodes $(v_i, v_j) \in \mathcal{E}$, so $\mathbf{A}$ is symmetric. For directed networks, $\mathbf{A}$ is asymmetric, we use $\mathbf{A} = \max(\mathbf{A}, \mathbf{A}^T)$ to deal with it [44]. In addition, let $\mathbf{X} \in \mathbb{R}^{n \times d}$ denotes the node attributes of all n nodes, where $\mathbf{x}_i \in \mathbb{R}^d$ is the attribute information for the ith node v_i. Specifically, each node has representations in k different views, i.e., $\mathbf{x}_i = (\mathbf{x}_i^{(1)}, \ldots, \mathbf{x}_i^{(k)})$, where $\mathbf{x}_i^{(p)} \in \mathbb{R}^{D_p}$, D_p is the dimension of the pth view. Obviously, $d = \sum_{p=1}^{k} D_p$, so $\mathbf{x}_i \in \mathbb{R}^d$. With the above notations, we now formally define the studied problem as follows.[1]

Problem 1 (Anomaly Detection on Multi-View Attributed Networks Supporting Self-Learning and User-Guided Learning). *Given a multi-view attributed network $\mathcal{G}$ represented by $\mathbf{A}$ and $\mathbf{X} = (\mathbf{X}^{(1)}, \ldots, \mathbf{X}^{(k)})$, in self-learning mode, the problem aims to spot several nodes which are rare and differ significantly from the majority reference nodes; in user-guided mode, the user is additionally required to provide a preference vector $\mathbf{a}$ where views (attributes) of interest to the user are assigned with larger weights, then the task is to detect anomalies in views that the user focuses on (is interested in).*

4 Methodology

In this section, we elaborate on the proposed framework ALARM and illustrate its architecture in Fig. 3. The entire framework is an unsupervised deep learning model similar to the autoencoder [45], which consists of three essential components: (1) *multi-view attributed network encoder* - leveraging multiple graph neural networks [11] to embed the given multi-view attributed network; (2) *aggregator* - aggregating view-based embeddings learned from per GNN to obtain unified representations across all views; (3) *structure & attribute decoder* - attempting to reconstruct the original network topology and node attributes with unified representations. After certain rounds of iterations, we can rank anomalies according to reconstruction errors with the fact that the larger the residuals in the reconstruction process, the more likely the instances are to be anomalous [46].

4.1 Preliminary

The powerful performance of CNNs [47] in solving image-related issues has successfully led researchers to explore sophisticated deep architectures for graph-structured data. One of the most recent prominent developments in GNNs is the deep architecture known as GCNs, the core of which lies in updating the feature vector $\mathbf{h}_i$ of each node v_i via recursively aggregating features of its neighboring nodes v_j

$$\mathbf{h}_i^{(l)} = f_{relu}\left(\sum_{v_j \in \{v_i\} \cup \mathcal{N}(v_i)} a_{ij} \mathbf{W}^{(l)} \mathbf{h}_j^{(l-1)}\right), \tag{1}$$

where $\mathbf{h}_i^{(l)}$ is the representation of v_i at the lth layer/iteration, $\mathbf{W}^{(l)}$ is a layer-specific trainable weight matrix at the lth layer, $f_{relu}(x) = max(0, x)$ is a non-linear activation function, a_{ij} is the (i, j)th entry of $\hat{\mathbf{A}}$ ($\hat{\mathbf{A}} = \mathbf{D}^{-\frac{1}{2}} \bar{\mathbf{A}} \mathbf{D}^{-\frac{1}{2}}$, $\bar{\mathbf{A}} = \mathbf{A} + \mathbf{I}_n$, $\mathbf{D}_{ii} = \sum_j \bar{\mathbf{A}}_{ij}$), and $\mathbf{h}_i^{(0)}$ can be gained from its original attributes $\mathbf{x}_i$ directly. Note that both attribute information $\mathbf{X}$ and the graph structure $\mathbf{A}$ are taken into consideration. Meanwhile, the nonlinear transformation f_{relu} can capture the nonlinearity of data. The above two strongpoints enable GCNs to effectively cope with the complex interactions of two kinds of information in attributed networks, thus enables us to learn high-quality embedding representations.

Obviously, an additional decoder can be used together with GNNs to form a simple autoencoder which completes the process of compression and reconstruction, i.e., compressing the input data $\mathbf{X}$ into low-dimensional codings and then generating $\tilde{\mathbf{X}}$ - a recovered version of the original input, from the condensed representations. Formally, the training process of an autoencoder can be described as minimizing following reconstruction errors (e.g., squared errors), where $Encode(\cdot)$ refers to the output of the last layer of the encoder

$$\mathcal{L}(\mathbf{X}, \tilde{\mathbf{X}}) = \|\mathbf{X} - Decode(Encode(\mathbf{X}))\|_F^2. \tag{2}$$

Moreover, many previous studies [6], [46], [48] have shown that the magnitude of reconstruction errors can be regarded as a strong indicator of abnormality since anomalies do not conform to the patterns of the majority and can not be accurately reconstructed. Therefore, we can label abnormal nodes based on this indicator.

4.2 Multi-View Attributed Network Encoder

Given $\mathcal{G}(\mathcal{V}, \mathcal{E}, \mathcal{F})$ with k views, we have corresponding k-view attributes $(\mathbf{X}^{(1)}, \ldots, \mathbf{X}^{(k)})$, where $\mathbf{X}^{(i)} \in \mathbb{R}^{n \times D_i}$, D_i is the dimension of the ith view. Considering these views exhibit heterogeneous properties, it is indispensable to model the multi-view information more synergistically. Otherwise, these views may even degrade learning performance. Thus, we design a deep multi-view architecture consisting of multiple independent GNNs and an aggregator to effectively

1. Our model is generally applicable to the attributed network with multiple network topologies in which nodes can be related in multiple ways, i.e., the correlation between nodes can form multiple graphs from different views, resulting in multiple adjacency matrices $\mathcal{A} = \{\mathbf{A}^{(1)}, \ldots, \mathbf{A}^{(k)}\}$.

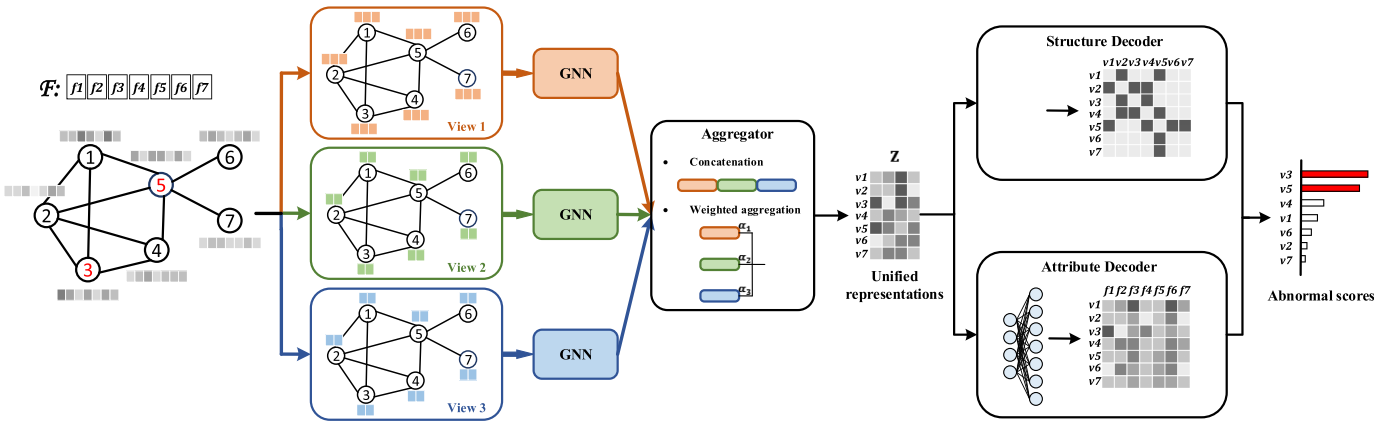


Fig. 3. The proposed deep multi-view framework ALARM for anomaly detection on attributed networks. Assuming that the given multi-view attributed network $\mathcal{G}$ contains 3 views, i.e., $k = 3$, which are fed into 3 distinct GNNs to extract view-based features. These hidden embeddings are then aggregated across views and passed through structure & attribute encoder to obtain abnormal scores for spotting anomalies.

learn and fuse specific knowledge contained in each view of data. With the goal of obtaining the view-based representation, we feed each view to a GNN separately. In this paper, we consider a two-layer GCN for encoding the given attributed under each view (we explore the influence of model depth later). After two layers of convolution operations, each view is compressed into a h_2-dimensional latent representation $\mathbf{U}_i \in \mathbb{R}^{n \times h_2}$. To avoid excessive model parameters, we can share the weight matrices under each view. In specific, we share the weight matrix $\mathbf{W}^{(2)} \in \mathbb{R}^{h_1 \times h_2}$ at the second layer, while retaining the specific weight matrices $\mathbf{W}_i^{(1)} \in \mathbb{R}^{D_i \times h_1}$ untouched to carefully capture the unique characteristics of each view. Whether to use the weight sharing strategy depends on the specific situation.

4.3 Aggregator

For low-dimensional representations extracted from different views, it is significant to be aware of the complementarity between this view-based information. Specifically, each feature space gives a description of the data at one certain perspective, taking advantage of the complementary information between them to aggregate multiple views seems more sensible. In this paper, we apply two kinds of view aggregation strategies.

Concatenation - concatenating vectors across all views into a new vector. It is one simple and traditional solution. Finally, we will obtain a combined representation $\mathbf{Z} \in \mathbb{R}^{n \times (k \times h_2)} = \|_{i=1}^{k} \mathbf{U}_i$, where $\|$ indicates concatenation.

Weighted Aggregation - giving each view-based vector a different weight and then summing them up, which means that each view has a distinct proportion in the final representation. To be more specific, the combined results $\mathbf{Z} \in \mathbb{R}^{n \times h_2}$ is selected from the following set:

$$\mathcal{Z} = \left\{ \mathbf{Z} \,|\, \mathbf{Z} = \sum_{i=1}^{k} \alpha_i \mathbf{U}_i, \alpha_i \geq 0, \sum_{i=1}^{k} \alpha_i = 1 \right\}, \tag{3}$$

when $\alpha_i = 0$, the ith view will be ignored. The aggregation weights $\{\alpha_i\}_{i=1}^{k}$ play a role similar to the attention mechanism, here we show three ways to obtain this set of weights:

- Randomly generate a vector $\mathbf{e} \in \mathbb{R}^k$ and then create the aggregation weights using the softmax function

$$\alpha_i = softmax(\mathbf{e})_i = \frac{exp(e_i)}{\sum_{j=1}^{k} exp(e_j)}. \tag{4}$$

  Utilizing the error backpropagation algorithm, α_i can be continuously optimized and updated. The resulting weights represent the optimal view aggregation ratio considered by the proposed model.
- Users manually assign a vector $\mathbf{a} \in \mathbb{R}^k$ reflecting their own preferences which are captured by those views (attributes) with large weights. Similarly, use (4) to calculate the aggregation weights, but be aware these weights are constant and can not be updated.
- Conduct user modeling and infer their concerns based on vector representations. This method requires some additional auxiliary information such as a social network, or even more. To give a simple example, we can use LINE [49] or node2vec [50] algorithm to learn the representation $\mathbf{v_a} \in \mathbb{R}^v$ for a user. The aggregation weights are then computed using a softmax layer

$$\alpha_i = \frac{exp\left(\beta_i^T \mathbf{v_a} + b_i\right)}{\sum_{j=1}^{k} exp\left(\beta_j^T \mathbf{v_a} + b_j\right)}, \tag{5}$$

  where $\{\beta_i, b_i\}_{i=1}^{k}$ are parameters to be learned.

In experiments, we employ the first two methods to produce weights, the third one is left for future work. Also, it should be emphasized that the user-guided mode refers to aggregating views using the user-given weights in weighted aggregation strategy, while the self-learning mode includes concatenating views and aggregating views using self-learning weights.

4.4 Structure & Attribute Decoder

After obtaining the encoded representation $\mathbf{Z}$, the output of the aforementioned aggregator, we intend to focus on the reconstruction of attributed network $\mathcal{G}$ as reconstruction errors can be regarded as indicators of the anomaly.

For one thing, the structure decoder takes the latent representations as input and then calculates inner product between them to achieve the rebuilt adjacency matrix $\tilde{\mathbf{A}}$

$$\tilde{\mathbf{A}} = sigmoid\left(\mathbf{Z}\mathbf{Z}^T\right). \tag{6}$$

For another, the attribute decoder aims at approximating the original node attributes from the encoded embeddings. To be more specific, we leverage a simple fully-connected layer to reconstruct the attribute information as follows:

$$\tilde{\mathbf{X}} = f_{relu}\Big(\mathbf{Z}\mathbf{W}^{(2)} + \mathbf{B}\Big), \tag{7}$$

where $\mathbf{W}^{(2)} \in \mathbb{R}^{(k\times h_2)\times d}$ or $\mathbf{W}^{(2)} \in \mathbb{R}^{h_2\times d}$ according to the aggregation strategy adopted, $\mathbf{B} \in \mathbb{R}^{n\times d}$ is the corresponding bias term. It should be noted that when employing the user-given weights, if $\alpha_i = 0$, the attributes contained in the ith view need not be considered for reconstruction since the user does not care about this view at all.

4.5 Optimization and Detection

At this point, the modules of the framework ALARM have been introduced in detail. In view of the fact that attributed networks exhibit unique characteristics in both the topology and attribute, the loss of two above aspects ought to be taken into consideration.

Regarding topology recurrence, it can be achieved via maximizing the following likelihood estimation:[2]

$$\prod_{i,j}^{n} \tilde{\mathbf{A}}_{ij}^{\mathbf{A}_{ij}} (1 - \tilde{\mathbf{A}}_{ij})^{(1-\mathbf{A}_{ij})}. \tag{8}$$

As can be observed, when $\mathbf{A}_{ij} = 1$, we want to rebuild this link $\tilde{\mathbf{A}}_{ij}$ with the highest probability. Otherwise, when $\mathbf{A}_{ij} = 0$, $\tilde{\mathbf{A}}_{ij}$ should tend to be 0. Mathematically, maximizing (8) is equivalent to minimize the negative log-likelihood

$$\mathcal{L}_s = \sum_{i=1}^{n}\sum_{j=1}^{n} -\big[\gamma \mathbf{A}_{ij} log\tilde{\mathbf{A}}_{ij} + \big(1 - \mathbf{A}_{ij}\big) log\big(1 - \tilde{\mathbf{A}}_{ij}\big)\big], \tag{9}$$

where $\gamma = 1$. But in this work, it is set with a specific value, which takes the form of weighted cross entropy. More specifically, γ is the positive sample coefficient which allows one to trade off recall and precision by up- or down-weighting the cost of a positive error relative to a negative error. We have $\gamma = (n \times n - m)/m$ which can handle the sparsity of the network flexibly. When the network is sparse, $\gamma > 1$ increases the recall; and when the network is relatively dense, $\gamma < 1$ increases the precision.

Besides, $\mathcal{L}_a$ is formulated as a simple Frobenius norm aiming to measure the quality of reconstructed attributes

$$\mathcal{L}_a = \|\mathbf{X} - \tilde{\mathbf{X}}\|_F^2. \tag{10}$$

By minimizing the objective function including the structure loss $\mathcal{L}_s$ and the attribute loss $\mathcal{L}_a$

$$\mathcal{L} = \mathcal{L}_s + \mathcal{L}_a, \tag{11}$$

the proposed deep framework can gradually approximate the original attributed network $\mathcal{G}$ with the convergence of the objective function $\mathcal{L}$. After the iterative optimization processs, the abnormal score for the ith node can be computed by

$$s(v_i) = \lambda\|\mathbf{a}_i - \tilde{\mathbf{a}}_i\|_2^2 + (1-\lambda)\|\mathbf{x}_i - \tilde{\mathbf{x}}_i\|_2^2, \tag{12}$$

where the first term and the second term report the degree of deviation in structure and attribute, respectively, and λ is a trade-off parameter. Since nodes with high scores are more likely to be anomalous, we can rank anomalies according to abnormal scores.

4.6 Complexity Analysis

In the encoding phase, the computational complexity of two-layer convolution is $O(mh_1\hat{h}_i)$ for the ith GCN encoder, where $\hat{h}_i = \max(D_i, h_2)$. Hence, modeling k views with k two-layer GCNs costs $O(mh_1H)$, where $H = \max\{\hat{h}_i\}_{i=1}^{k}$. The aggregation operation has complexity $O(nkh_2)$, and the structure and attribute reconstruction need $O(n^2h_2)$ and $O(ndh_2)$, respectively. Note that k and d is usually smaller than n, the overall time complexity is $num\_iters * O(mh_1H + n^2h_2)$.

5 EXPERIMENTS

In this section, we empirically evaluate the effectiveness of ALARM on both synthetic and real-world datasets. A Case study is also implemented to intuitively show how ALARM integrates the user preference to achieve user-oriented anomaly detection. In specific, we attempt to answer the following three research questions:

- *Q1* How effective is the proposed framework on synthetic datasets with only a single view?
- *Q2* How impactful is the proposed framework on real-world networks with inherent multi-view attributes?
- *Q3* Whether the proposed framework supports user-oriented settings and provides anomaly detection results satisfying the user preference?

5.1 Compared Methods

Our main goal is to provide higher quality anomaly detection services on attributed networks, so we compare ALARM with five state-of-the-art methods which are popular in the field of anomaly detection in attributed networks. Although there exist some multi-view methods [51], [52], [53], they are set up for different problems (e.g., clustering, outlier detection on vector data). To the best of our knowledge, there is no multi-view anomaly detection algorithm for attributed networks. All the methods included in experiments are listed as follows:

- *ConOut* [9]: ConOut selects a subgraph and a statistically relevant subset of attributes for each node, and then detects anomalies in selected local context.
- *AMEN* [54]: AMEN spots anomalous neighborhoods by considering both attribute and topology information. Due to the fact that it is designed for abnormal cluster detection rather than point anomaly detection, for comparison, we treat all nodes in anomalous clusters as anomalies.
- *Radar* [6]: Radar is an unsupervised anomaly detection model which detects anomalies via learning and analyzing the residual errors of attribute information and its coherence with network topology.
- *ANOMALOUS* [7]: ANOMALOUS achieves joint optimization of anomaly detection and attribute selection based on CUR factorization and residual analysis. It

2. For unweighted networks, i.e., $\mathbf{A} \in \{0,1\}^{n\times n}$, cross-entropy loss is used to measure reconstruction error, while for weighted networks, mean squared error is more suitable.

spots anomalies with representative instances and attributes selected in an unsupervised manner.

- DOMINANT [8]: DOMINANT is a deep model which utilizes a principled graph convolutional autoencoder to seamlessly model the attributed network and conduct anomaly detection in a joint framework. DOMINANT lacks careful consideration and processing of multi-view data, making it significantly different from ALARM.
- ALARM-concat: ALARM-concat processes the hidden embeddings of all views by concatenating the view-based vectors into a new single vector.
- ALARM-*weighted*: ALARM-weighted processes the hidden embeddings of all views via calculating the weighted sum of the view-based vectors.

5.2 Parameter Settings and Metrics

For all baseline methods, we take their source codes to conduct plenty of experiments and choose the parameter settings which can get the best experimental results. As for our framework, it is implemented by TensorFlow [55] and then conducted on a Tesla V100-PCIE-32GB GPU. For SynGraph dataset with more than 50,000 nodes, we use the subsampling skill introduced in [24] to make it fit into GPU memory (this strategy has been successfully used to learn large graphs [56]). In detail, we first randomly select a minibatch of 128 nodes, and then for each selected node, we sample 8 and 5 neighbors at its first and second-level neighborhoods, respectively. In this way, each minibatch contains up to $128 + 128 \times 8 + 128 \times 8 \times 5 = 6272$ nodes (there may be duplicate nodes). We initialize weights using the initialization introduced in Glorot & Bengio [57], and train the proposed model for a maximum of 200 epochs with Adam optimizer [58]. We choose the hidden layer size of the GCN encoder from $\{2^6, 2^5, 2^4, 2^3, 2^2, 2^1\}$. The learning rate is tuned in the range of $\{0.01, 0.005, 0.001\}$. Furthermore, the number of split views for synthetic datasets is selected from $\{3, 5, 10, 15, 20\}$, while it is consistent with the number of inherent views in real-world datasets (more details later). The parameter λ is tuned in the range of [0,1].

For a fair comparison, we adopt the criteria of AUC that has been widely used as an evaluation metric in previous anomaly detection methods [6], [7], [8] to measure the detection performance. In specific, the AUC score is the area under the ROC curve, which is a plot of true positive rate (an abnormal node is identified as an anomaly) against false positive rate (a normal node is identified as an anomaly). From the statistical perspective, the AUC score indicates the probability that a randomly selected anomaly is ranked higher than a normal node. If the score approaches 1, the method has excellent anomaly detection performance.

5.3 Results on Synthetic Datasets

This part is mainly for answering the first question *Q1* from two aspects: (1) the quality of anomaly detection, and (2) training time. Here ALARM runs in self-learning mode.

5.3.1 Data Conformation

To assess the quality of our algorithm compared to other approaches with ground truth anomalies, we adopt four synthetic attributed graphs with the different number of attributes and varying size ranges. The statistics of these datasets are listed in Table 2. Note that whether the synthetic datasets have multi-view attributes or not, we regard them as single-view data.

TABLE 2
Statistics of Synthetic Datasets

	Flickr	Aminer	ConGraph	SynGraph
#nodes	7,000	11,085	10,000	52,570
#edges	397,224	51,866	37,848	65,542
#attributes	12,047	27	10	28
anomaly ratio	3.2%	4.98%	10%	4.52%

Flickr[3] is an image sharing website, users interact with others and form a social network. Tags attached to uploaded photos reflect users' interests, which provide attribute information. Besides, photos are classified into nine predefined categories. To build anomalies, we first determine a class with the fewest nodes by counting the proportion of each category. Then, by randomly eliminating some nodes belonging to this category, the class eventually accounts for only 3.2 percent of the entire dataset. In this way, it becomes a rare class in which nodes can be treated as anomalies since they are the minority in the dataset.

Aminer is an academic dataset in the field of computer science collected from ArnetMiner.[4] Authors who have co-author relationships form link information. The statistics of their publications such as the number of papers are considered as node attributes. Additionally, the publications are mainly from four areas: Artificial Intelligence (AI), Computer Vision (CV), Data Mining (DM) and DataBase (DB). Similarly, we treat the research area with occur in less than 5 percent of the authors as a rare category, authors belonging to this class are labeled as anomalies.

ConGraph[5] is a synthetic graph containing the ground truth labels of anomalies. Node degrees follow a power law distribution aiming at reproducing the properties observed in real-world networks. Meanwhile, 10-dimensional node attributes are divided into network structure related and network structure irrelevant following uniform random distribution, each accounting for 50 percent. For relevant attributes, a one-dimensional attribute could be merged with another subset of attributes to form a higher-dimensional relevant attributes with a probability of 20 percent decided by the parameter*probability for a subspace*. And for irrelevant attributes, nodes are assigned values from a uniform random distribution. See [59] and the website[5] for detailed explanation.

SynGraph is a subgraph extracted from the biggest component of the Amazon co-purchase network[5]. Each item has rich attributes such as ratings, prices and the number of reviews. However, no ground truth of anomalies is provided in this dataset, thus we need to inject some into it for evaluation. Considering that there are many types of anomalies in the real world, we follow two commonly used methods [60], [61] to build anomalies from both the structure

3. http://dmml.asu.edu/users/xufei/datasets.html
4. https://www.aminer.cn/
5. https://www.ipd.kit.edu/~muellere/consub/

TABLE 3
AUC Scores and Training Time of Different Algorithms on Synthetic Datasets

Algorithm	Flickr		Aminer		ConGraph		SynGraph	
	AUC	Time(s)	AUC	Time(s)	AUC	Time(s)	AUC	Time(s)
ConOut	0.4707	$>$3h	0.5942	363.08	0.6955	77.32	0.6016	1742.24
AMEN	0.4999	$>$3h	0.5009	46.13	0.4273	13.57	0.4765	40.32
Radar	0.6383	2744.49	0.9055	1825.92	0.7421	773.01	-	-
ANOMALOUS	0.6503	$>$3h	0.9084	9237.87	**0.7669**	3300.19	-	-
DOMINANT	0.6473	158.87	0.9235	184.04	0.7205	313.67	0.7714	2243.12
ALARM-concat (5 views)	**0.6918**	144.18	**0.9417**	104.10	0.7314	451.16	**0.7814**	2414.09
ALARM-weighted (5 views)	**0.6922**	135.85	**0.9538**	96.50	0.7554	469.92	**0.7820**	2416.17

perspective and the attribute perspective. Specifically, we randomly select some nodes and make them become a small fully connected clique via adding links between them, and then all the nodes in the clique are regarded as structural anomalies. Besides, we apply the attribute perturbation schema introduced by [61] to generate the other type of anomalies. Ultimately, we inject about 1050 structural anomalies and about 2100 contextual anomalies into this graph at a ratio of 1:2, leading to an anomaly ratio of 4.52 percent on this dataset (these two types of anomalies may overlap during construction).

For data with only one view, there exist various practical methods to convert it to multi-view data. In this work, we employ a simple way to cope with all synthetic datasets, i.e., partitioning the whole attribute set into diverse disjoint views randomly. As a matter of fact, numbers of previous experiments in multi-view learning apply this strategy [51], [62] and show desirable performance.

5.3.2 Performance Evaluation

Table 3 reports the anomaly detection performance including AUC scores and training time of various approaches on four synthetic datasets. We have the following observations.

In terms of performance, the proposed ALARM framework (-concat and -weighted) achieves the best performance on Flickr, Aminer and SynGraph datasets, which demonstrates the superiority of neural networks in the anomaly detection task. Further comparison with DOMINANT which treats data in a single-view pattern substantiates a noteworthy fact that when natural multi-view property does not exist in data, performance improvements can still be observed using manufactured splits such as random partitioning [51], [63]. Due to the particularity of ConGraph dataset in construction, ANOMALOUS seems to be more effective, although ALARM exhibits comparable results. Specifically, half of the attributes in ConGraph are noises and contain many congruent subspaces [59], which are carefully considered and treated in ANOMALOUS, but this is a challenge for traditional GCNs used in our framework. But interestingly, ALARM requires much less training time (451s and 470s) to obtain competitive results with ANOMALOUS, while ANOMALOUS takes about seven times (3,300s). Besides, ALARM-weighted gives better results than ALARM-concat as it is able to model the interactions between multiple distinct views.

In terms of training time, ConOut and AMEN are vulnerable to the size of attribute dimension as ConOut has to examine each attribute to filter out local context, and AMEN involves a large number of calculations between attribute vectors. ANOMALOUS is relatively more time-consuming due to complex matrix operations, while ALARM using deep learning method is more advantageous. Additionally, in the experiments, when running Radar, and ANOMALOUS on SynGraph dataset, the process was killed by the operating system because of memory exhaustion, so no results were obtained. On the contrary, in this case, ALARM (-concat and -weighted) still works well and becomes the best choice.

5.3.3 Effects of the Number of Split Views

Recall that the random multi-view attributes construction strategy has been applied to synthetic datasets, and how the number of split views affects detection performance naturally becomes an attractive issue that worths deep investigation. Table 4 presents some experimental results of ALARM-weighted $w.r.t.$ different numbers of views. It is interesting to find that the AUC scores of ALARM-weighted keep rather stable while varies the number of split views, and the training time increases slightly with the growing number of views since more convolution parameters in the GCN layer $\{\mathbf{W}_i\}_{i=1}^{k}$ are to be trained. Accordingly, in practice, it is inclined to split a relatively small number of views which are usually sufficient to achieve good performance. Since instances in ConGraph dataset have only ten attributes, here we omit the results of 15 and 20 views.

5.4 Results on Real-World Datasets

This part will address the second question $Q2$ in terms of anomaly detection performance. Here ALARM runs in self-learning mode.

5.4.1 Data Description

We adopt three real-world attributed networks[5] which have been widely used in the previous research [6], [7], [10] to compare the anomaly detection performance of different methods. These datasets are data with natural multi-view property. Detailed descriptions are illustrated in Table 5.

Disney is a co-purchase network of movies. Each movie has 28 available attributes such as ratings, number of reviews, number of different authors and prices. These attributes are multi-view in nature with inherent four different views. Fig. 2 gives an intuitive understanding of the dataset, and details can be found in [43]. The ground truth anomalies (e.g., products with too low star ratings, products with particularly few reviews) are manually marked by high

TABLE 4
AUC Scores and Training Time of ALARM-Weighted Under Varying Split Views

Dataset	3 Views		5 Views		10 Views		15 Views		20 Views	
	AUC	Time(s)	AUC	Time(s)	AUC	Time(s)	AUC	Time(s)	AUC	Time(s)
Flickr	0.6922	134.96	0.6922	135.85	0.6923	136.70	0.6920	142.60	0.6917	147.99
Aminer	0.9446	92.52	0.9538	96.50	0.9466	98.39	0.9487	98.43	0.9462	100.94
ConGraph	0.7502	467.65	0.7554	469.92	0.7346	472.98	-	-	-	-
SynGraph	0.7765	2406.49	0.7820	2416.17	0.7824	2477.64	0.7824	2516.39	0.7823	2565.08

school students who assign labels following specific criteria such as low ratings and few reviews.

Books is also a co-purchase network including various books as nodes and has similar attributes as Disney. We still adopt its inherent four-view attributes for experiments. As for the ground truth, we find that many Amazon users take the*amazonfail* tag to express their disagreement with sales ranks. They think the ratings and sales ranks from some books were misplaced by Amazon, thus labeling these books with such tags. Based on this, we regard a book that is labeled*amazonfail* at least by 20 users as anomalies.

Enron is an email communication network where email addresses represent nodes, email transmission between them forms edges. Attributes include average content length, different encodings count, etc. For a detailed explanation of the attribute information, please refer to [43]. In the experiments, the following three views, i.e., *content information*, *recipient and sender information*, and *subject and frequency information* are adopted. Besides, we regard spam email addresses as anomalies. According to the list of spam addresses taken from the benchmark in [64], we label the ground truth anomalies.

5.4.2 Performance Comparison

The performance of different approaches on three real-world datasets is shown in Fig. 4. As can be clearly observed, ALARM (-concat and -weighted) outperforms all other methods on Books and Enron datasets, meanwhile ALARM-weighted is substantially the same as ANOMALOUS on Disney dataset in terms of AUC scores. It is necessary to recognize that the superiority of the neural network is reflected in its ability of learning optimal parameters from big datasets, and its performance may be impaired since small datasets cannot provide sufficient variants in learning samples. Here Disney dataset is small in size (only 124 nodes), which may affect ALARM and make traditional machine learning methods (e.g., ANOMALOUS) better. Contrarily, on large dataset Enron, ALARM exhibits significantly stronger performance than traditional methods. Besides, our ALARM method achieves higher AUC scores than DOMINANT which neglects the inherent multi-view property of data. Hence, it is safe to conclude that for data with inherent multiple views, being treated in a multi-view style is feasible and reasonable, and indeed improves performance since the unique statistical characteristics of each view are considered and processed carefully. Similarly, ALARM-weighted performs better than ALARM-concat, which can be explained by the fact that ALARM-weighted achieves a view-level attribute selection by aggregating views with different weights, while ALARM-concat only simply concatenates view-based embeddings implying each view has the same weight. In addition, the better performance of the latter five methods again reflects the effectiveness of residual analysis in anomaly detection.

TABLE 5
Details of Real-World Datasets

	Disney	Books	Enron
#nodes	124	1,418	13,533
#edges	334	3,695	176,987
#attributes	28	28	20
#views	4	4	3
anomaly ratio	4.8%	2%	0.04%

5.5 A Case Study

In this subsection, we use ALARM-weighted to conduct a case study on Disney network for illustrating the last question *Q3*. Here, ALARM runs in the user-guided mode.

First, we consider a situation where users want to understand the performance of movies on star ratings with the aim of satisfying their curiosity about discovering those with extreme ratings. Accordingly, users only focus on the view that contains the rating information with the neglect of other views, i.e., $\alpha_{rating} = 1$, $\alpha_{others} = 0$. Several abnormal movies detected by ALARM-weighted for this task are shown on the left in Fig. 5. Specifically, node N_1 corresponds to the film *The Nightmare Before Christmas/James and the Giant Peach*, it only has the 5-star rating and 1-star rating, accounting for 86 and 14 percent, respectively. Such ratings are unevenly distributed and somewhat extreme. Besides, it is a typical structural anomaly since it is an isolated co-purchase product. Node N_2 corresponds to the film *Honey, We Shrunk Ourselves*,

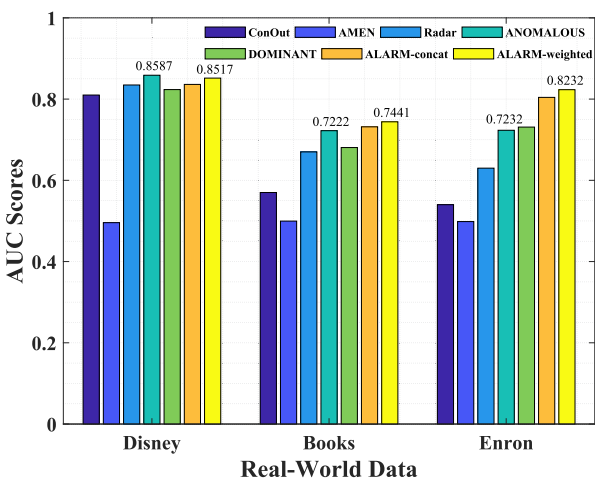


Fig. 4. Performance comparison (AUC scores) of different methods for anomaly detection.

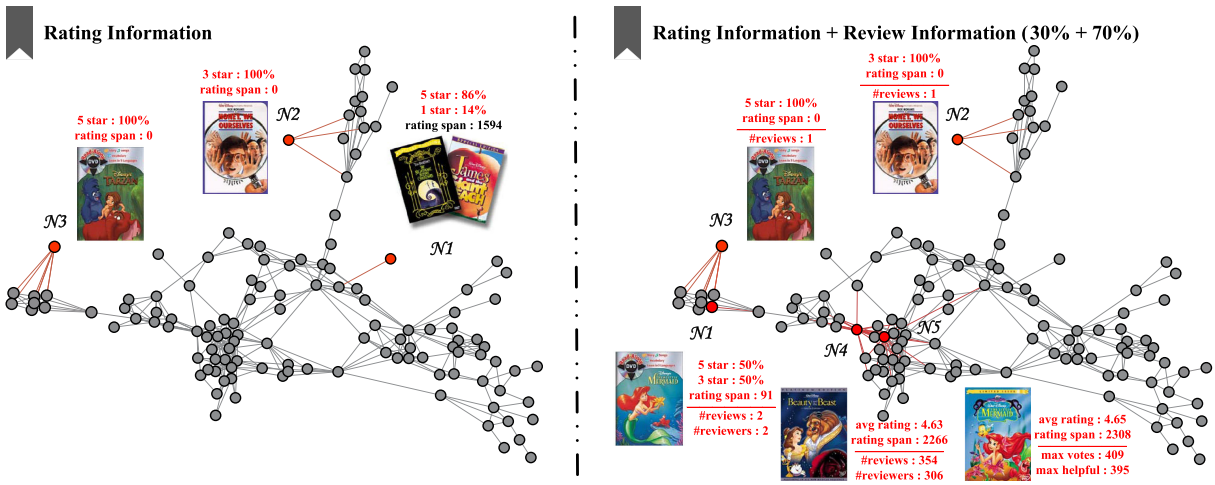


Fig. 5. Abnormal films detected by ALARM-weighted in the user-guided mode from two views (*left*: rating information, *right*: rating and review information accounted for 30 and 70 percent respectively). Items marked with red are anomalous.

it only has the 3-star rating and its *rating span*[6] is 0, which is rather different from others. Node N_3 that corresponds to the film *Tarzan* is similar to N_2, it only has 5-star rating and its *rating span* equals to 0 as well, which makes it anomalous.

In the next situation, users intend to screen out outstanding works that are prevalent in the public while avoiding those unattractive movies. Consequently, users only focus on two views involving the rating information and review information at a ratio of 3:7, i.e., $\alpha_{rating} = 0.3$, $\alpha_{review} = 0.7$, $\alpha_{others} = 0$, since in their cognition, whether a movie is popular is reflected by the ratings and the number of reviews. The results are given on the right side of Fig. 5. As can be found, N_2 and N_3 are detected again as they have only one review, the number of reviews is too little compared with that of other films. Node N_1 corresponds to the film *The Little Mermaid 2002* which has very few reviews and reviewers, its ratings are unevenly distributed with only the 3-star rating and 5-star rating, each accounting for 50 percent, and its *rating span* is 91. As a result, it may be an unpopular movie. Node N_4 corresponds to the film *Beauty and the Beast* that has large numbers of reviews and reviewers, node N_5 corresponds to the film *The Little Mermaid 1989* that has a great many review votes, and both of them have extremely high average star ratings, which clearly highlights their popularity and attractiveness in the public.

5.6 Parameter Analysis

In this part, we mainly investigate the impact of the following two parameters on model performance: (1) the parameter λ, (2) the hidden layer size.

5.6.1 Trade-Off Between Structure and Attribute

Note that the parameter λ controls the proportion of structure and attribute reconstruction errors in the abnormal score $s(v_i)$. When $\lambda = 1$, only the degree of deviation in structure is considered in $s(v_i)$, while $s(v_i)$ only reflects the deviation in terms of attributes when $\lambda = 0$. In both two extreme cases, ALARM downgrades to a mediocre model that can merely spot structural or contextual anomalies. In this section, we investigate the impact of structure and attribute reconstruction errors on the final identification of anomalies by varying the value of λ. Fig. 6 illustrates the performance of ALARM-weighted on Books, Enron (real-world datasets), and SynGraph (synthetic dataset) under different magnitudes of λ. As can be seen, the changing pattern of AUC scores on Books and Enron datasets is basically consistent, and the best AUC is obtained on the order of $\lambda = 0.001$, which means more emphasis on considering attribute errors contributes to improving the performance of anomaly detection. While the change rule is quite distinct on SynGraph dataset, and the best AUC is achieved when the magnitude of λ becomes 0.1. After conducting further analysis to examine the characteristics of datasets, we may observe that the anomalies existed in the real-world datasets including Books and Enron mostly belong to contextual and community anomalies which exhibit huge deviations in attribute values, while the number of structural anomalies is relatively small. Naturally, on real-world datasets, the magnitude of λ is set to 0.001, forcing the computed abnormal score to reflect much more deviations in attribute values. On the contrary, recall that we inject structural and contextual anomalies into the synthetic dataset SynGraph at a ratio of 2:4. Thus, there is little difference in magnitude between these two types of anomalies, which reflects in the optimal value of λ. Overall, the results indicate that considering abnormal scores from only structure or attribute aspect is insufficient to dig out

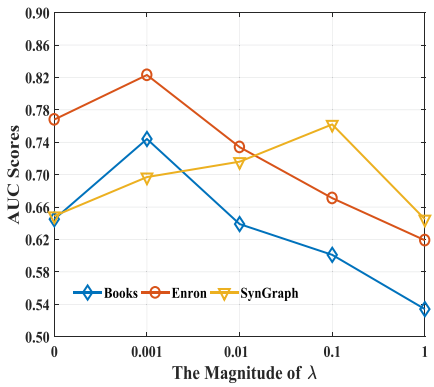


Fig. 6. Impact of different magnitudes of λ on detection performance.

6. It means the time between the first rating and the last rating.

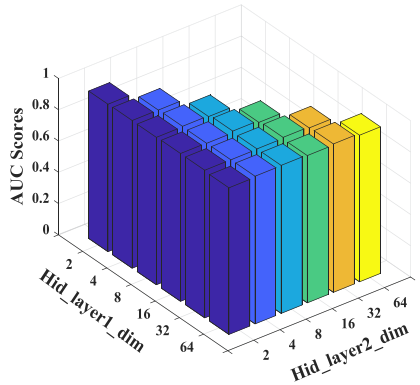


Fig. 7. AUC scores of different hidden layer sizes on Aminer dataset.

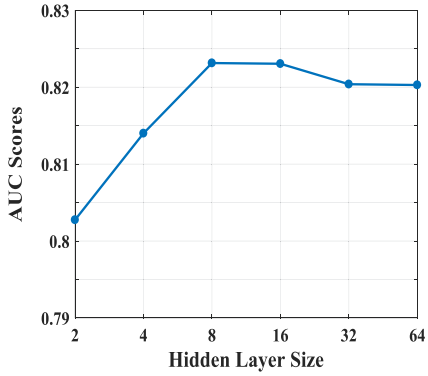


Fig. 8. AUC scores of different hidden layer sizes on Enron dataset.

various types of anomalies, it is necessary to find a proper trade-off parameter to balance the structure and attribute reconstruction errors for the best detection performance.

5.6.2 Hidden Layer Size

Next we explore the impact of different hidden layer sizes (i.e., dimensionality) on detection performance. Here we select Aminer and Enron datasets as representatives. Figs. 7 and 8 show how the AUC scores vary with different dimensionality on these two datasets, respectively. First, on Aminer, the encoder we employed is a 2-layer GCN. Typically in a deep encoding process, the size of the latter layer is less than or equal to that of the previous layer. Specifically, if the dimension of the first hidden layer is 16, then the dimension of the next hidden layer generally does not exceed 16. Thus in Fig. 7, only half of the parameter settings are valuable. We observe that the proposed ALARM can almost achieve the best performance under various settings. Second, the GCN encoder on Enron has only one layer. Fig. 8 indicates that the increase of dimensions does help the performance improvement, but only a little. Based on the above results, it is safe to conclude that our proposed ALARM is not overly sensitive to hidden layer size.

5.7 Influence of Model Depth

Now we explore the influence of model depth on detection performance via adjusting the number of GCN layers in the encoder. In addition to the standard ALARM model (here we select ALARM-weighted as the representative), we report the results of its variant that uses an only single GCN encoder, which means this variant ignores the inherent multi-view property of data. Moreover, due to the difficulty of training deep neural networks, suggested by [65], we also experiment with a counterpart residual version of standard ALARM model, which adds residual connections between hidden layers to facilitate the training of deep networks. In experiments, we start introducing residual shortcuts from the second layer to prevent the difference in the input and output dimension of the first layer. The detailed network architecture is illustrated in Fig. 9. We train these three models for fixed epochs (100 on Enron, 200 on ConGraph) in an unsupervised pattern and the involved hyperparameters remain untouched.

Fig. 10 summarizes the AUC scores of these models on Enron and ConGraph datasets. It can be seen that the best results for the datasets considered here are obtained with a 1- or 2-layer model. As the number of layers increases, training the single encoder model can become difficult and its performance decreases explicitly. Compared to the above variant, the performance degradation of the standard ALARM model is relatively less severe. We attribute it to the adoption of an aggregator in ALARM, which aggregates the output of multiple encoders through learnable parameters such that provides additional adjustment space to fit in with the deep model. Furthermore, the increase of model depth

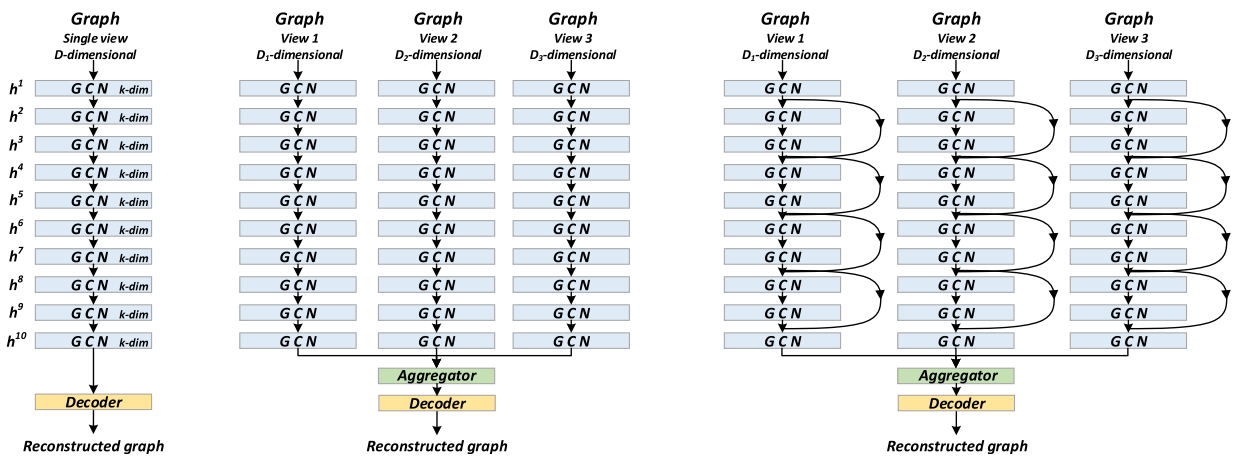


Fig. 9. Network architecture for a variant that treat data in a single-view pattern (*left*), standard ALARM model (*middle*), and a residual variant (*right*), taking the given multi-view attributed network $\mathcal{G}$ contains 3 views for example.

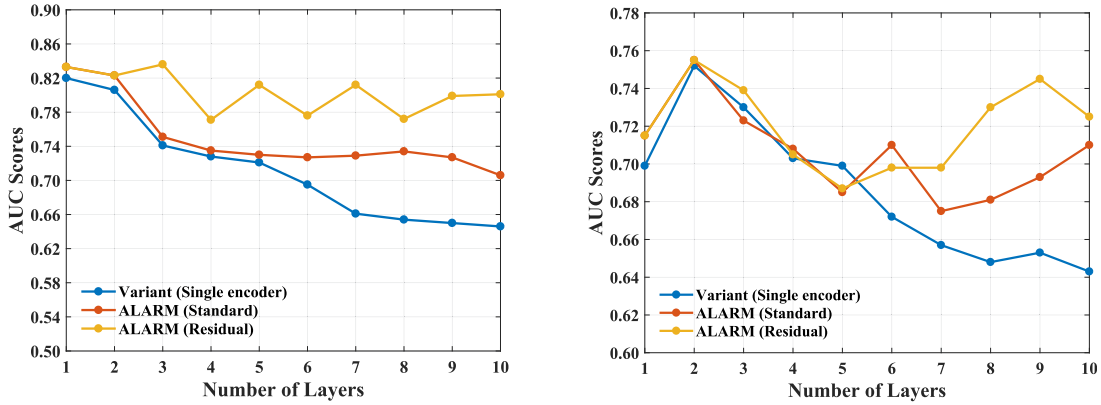


Fig. 10. On datasets Enron (*left*) and ConGraph (*right*), the influence of model depth on detection performance.

highlights the performance gap between models with and without residual connections. This observation validates the merit of enabling the model to transmit information from the previous layer's input to subsequent layers via identity shortcut connections during training deep neural networks.

6 Conclusion

In this work, we introduce a deep multi-view framework ALARM for detecting anomalies on attributed networks, which supports self-learning and user-guided learning. ALARM appropriately tackles heterogeneous attribute characteristics brought by multi-view data and skillfully implements user-oriented anomaly detection via extracting view-based features with multiple independent GCNs and aggregating them based on user-oriented settings reflecting user preferences. Experiments on synthetic datasets show that even if the data does not contain natural multi-view attributes, ourALARM still achieves the best results by randomly splitting the attribute set. And it has an advantage in training speed. In addition, experiments on real-world datasets demonstrate the effectiveness ofALARM in detecting anomalies on multi-view attributed networks. Besides, a case study on the Disney dataset indicates that our approach can well incorporate user preferences into anomaly detection, which provides personalized anomaly detection services for users. Overall, we revisit the problem of anomaly detection from a multi-view perspective and propose a deep multi-view framework to better deal with the differentiation and association between multiple views. Meanwhile, we introduce a user-oriented setting into the model so that users can steer the detection process, while few existing techniques take the user's interest into consideration. Besides, our ALARM framework can be intuitively extended to deal with time series anomaly detection problems. A simple method is to run theALARM framework on each timestamp to detect anomalies at each moment. Since our algorithm requires relatively little training time, it will be feasible in practice. Especially, when users focus on different views at different moments (ALARM runs in the user-guided mode), the above method seems to be a good choice as it provides users with a way to steer the detection process by inputting preferences at any time. Another promising method is to explore a more sophisticated GNN architecture to capture the temporal characteristics of graphs. For example, we can use the spatio-temporal graph convolutional network (STGCN) [66] to replace the original GCN to encode the time-evolving attributed networks.

Future work will concentrate on (1) inferring user preferences with additional user modeling information rather than the manual assignment by users themselves; (2) introducing feature selection into traditional GCNs or using other advanced GNNs to improve detection accuracy; (3) exploring the impact of other aggregation strategies, e.g., element-wise max operation, mean operation or more sophisticated aggregators on performance.

Acknowledgments

This work was supported by National Key Research and Development Program of China (No. 2018AAA0101400), National Nature Science Foundation of China (No. 61872287 and No. 61532015), Innovative Research Group of the National Natural Science Foundation of China (No. 61721002), Innovation Research Team of Ministry of Education (IRT_17R86), and Project of China Knowledge Center for Engineering Science and Technology.

References

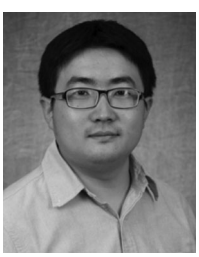
[1] C. C. Aggarwal, "Outlier analysis," in *Data Mining*. Berlin, Germany: Springer, 2015, pp. 237–263.
[2] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: A survey," *ACM Comput. Surv.*, vol. 41, no. 3, 2009, Art. no. 15.
[3] C. Xu, D. Tao, and C. Xu, "A survey on multi-view learning," *Neural Comput. Appl.*, vol. 23, no. 7–8, pp. 2031–2038, 2013.
[4] J. Gao, F. Liang, W. Fan, C. Wang, Y. Sun, and J. Han, "On community outliers and their efficient detection in information networks," in *Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2010, pp. 813–822.
[5] L. Akoglu, H. Tong, B. Meeder, and C. Faloutsos, "PICS: Parameter-free identification of cohesive subgroups in large attributed graphs," in *Proc. SIAM Int. Conf. Data Mining*, 2012, pp. 439–450.
[6] J. Li, H. Dani, X. Hu, and H. Liu, "Radar: Residual analysis for anomaly detection in attributed networks," in *Proc. Int. Joint Conf. Artif. Intell.*, 2017, pp. 2152–2158.
[7] Z. Peng, M. Luo, J. Li, H. Liu, and Q. Zheng, "ANOMALOUS: A joint modeling approach for anomaly detection on attributed networks," in *Proc. Int. Joint Conf. Artif. Intell.*, 2018, pp. 3513–3519.
[8] K. Ding, J. Li, R. Bhanushali, and H. Liu, "Deep anomaly detection on attributed networks," in *Proc. SIAM Int. Conf. Data Mining*, 2019, pp. 594–602.
[9] P. I. Sánchez, E. Müller, O. Irmler, and K. Böhm, "Local context selection for outlier ranking in graphs with multiple numeric node attributes," in *Proc. Int. Conf. Sci. Stat. Database Manage.*, 2014, Art. no. 16.

[10] P. I. Sánchez *et al.*, "Efficient algorithms for a robust modularity-driven clustering of attributed graphs," in *Proc. SIAM Int. Conf. Data Mining*, 2015, pp. 100–108.
[11] J. Zhou, G. Cui, Z. Zhang, C. Yang, Z. Liu, and M. Sun, "Graph neural networks: A review of methods and applications," 2018, [Online] Available: https://arXiv:1812.08434
[12] H. Hotelling, "Relations between two sets of variates," *Biometrika*, vol. 28, no. 34, pp. 321–372, 1936.
[13] P. L. Lai and C. Fyfe, "Kernel and nonlinear canonical correlation analysis," *Int. J. Neural Syst.*, vol. 10, no. 05, pp. 365–377, 2000.
[14] W. W. Hsieh, "Nonlinear canonical correlation analysis by neural networks," *Neural Netw.*, vol. 13, no. 10, pp. 1095–1105, 2000.
[15] A. Frome *et al.*, "DeViSE: A deep visual-semantic embedding model," in *Proc. 26th Int. Conf. Neural Inf. Process. Syst.*, 2013, pp. 2121–2129.
[16] N. Srivastava and R. R. Salakhutdinov, "Multimodal learning with deep Boltzmann machines," in *Proc. 25th Int. Conf. Neural Inf. Process. Syst.*, 2012, pp. 2222–2230.
[17] J. Ngiam, A. Khosla, M. Kim, J. Nam, H. Lee, and A. Y. Ng, "Multimodal deep learning," in *Proc. Int. Conf. Mach. Learn.*, 2011, pp. 689–696.
[18] H. Su, S. Maji, E. Kalogerakis, and E. G. Learned-Miller, "Multi-view convolutional neural networks for 3D shape recognition," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2015, pp. 945–953.
[19] S. Rastegar, M. S. Baghshah, H. R. Rabiee, and S. M. Shojaee, "MDL-CW: A multimodal deep learning framework with cross weights," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 2601–2609.
[20] F. Wang, W. Zuo, L. Lin, D. Zhang, and L. Zhang, "Joint learning of single-image and cross-image representations for person re-identification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 1288–1296.
[21] M. Gori, G. Monfardini, and F. Scarselli, "A new model for learning in graph domains," in *Proc. Int. Joint Conf. Neural Netw.*, 2005, vol. 2, pp. 729–734.
[22] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *Proc. Int. Conf. Learn. Representations*, 2017.
[23] P. Velickovic, G. Cucurull, A. Casanova, A. Romero, P. Lio, and Y. Bengio, "Graph attention networks," in *Proc. Int. Conf. Learn. Representations*, 2018.
[24] W. Hamilton, Z. Ying, and J. Leskovec, "Inductive representation learning on large graphs," in *Proc. 31st Int. Conf. Neural Inf. Process. Syst.*, 2017, pp. 1024–1034.
[25] Z. Ying, J. You, C. Morris, X. Ren, W. Hamilton, and J. Leskovec, "Hierarchical graph representation learning with differentiable pooling," in *Proc. 32nd Int. Conf. Neural Inf. Process. Syst.*, 2018, pp. 4800–4810.
[26] X. Xu, N. Yuruk, Z. Feng, and T. A. Schweiger, "SCAN: A structural clustering algorithm for networks," in *Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2007, pp. 824–833.
[27] L. Akoglu, M. McGlohon, and C. Faloutsos, "Oddball: Spotting anomalies in weighted graphs," in *Proc. Pacific-Asia Conf. Knowl. Discov. Data Mining*, 2010, pp. 410–421.
[28] B. Perozzi, L. Akoglu, P. Iglesias Sánchez, and E. Müller, "Focused clustering and outlier detection in large attributed graphs," in *Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2014, pp. 1346–1355.
[29] F. R. Bach and M. I. Jordan, "Kernel independent component analysis," *J. Mach. Learn. Res.*, vol. 3, no. Jul, pp. 1–48, 2002.
[30] D. R. Hardoon, S. Szedmak, and J. Shawe-Taylor, "Canonical correlation analysis: An overview with application to learning methods," *Neural Comput.*, vol. 16, no. 12, pp. 2639–2664, 2004.
[31] J. Mao, W. Xu, Y. Yang, J. Wang, Z. Huang, and A. Yuille, "Deep captioning with multimodal recurrent neural networks (m-RNN)," 2014. [Online] Available: https://arxiv.org/abs/1412.6632
[32] Y. Li, M. Yang, and Z. M. Zhang, "A survey of multi-view representation learning," *IEEE Trans. Knowl. Data Eng.*, vol. 31, no. 10, pp. 1863–1883, Oct. 2019.
[33] J. Bruna, W. Zaremba, A. Szlam, and Y. LeCun, "Spectral networks and locally connected networks on graphs," in *Proc. Int. Conf. Learn. Representations*, 2013.
[34] M. Defferrard, X. Bresson, and P. Vandergheynst, "Convolutional neural networks on graphs with fast localized spectral filtering," in *Proc. 30th Int. Conf. Neural Inf. Process. Syst.*, 2016, pp. 3844–3852.
[35] M. M. Bronstein, J. Bruna, Y. LeCun, A. Szlam, and P. Vandergheynst, "Geometric deep learning: Going beyond euclidean data," *IEEE Signal Process. Mag.*, vol. 34, no. 4, pp. 18–42, Jul. 2017.
[36] R. Li, S. Wang, F. Zhu, and J. Huang, "Adaptive graph convolutional neural networks," in *Proc. AAAI Conf. Artif. Intell.*, 2018, pp. 3546–3553.
[37] J. Chen, T. Ma, and C. Xiao, "FastGCN: Fast learning with graph convolutional networks via importance sampling," in *Proc. Int. Conf. Learn. Representations*, 2018.
[38] J. Lee, I. Lee, and J. Kang, "Self-attention graph pooling," in *Proc. Int. Conf. Mach. Learn.*, 2019, pp. 6661–6670.
[39] B. Long, Z. M. Zhang, X. Wu, and P. S. Yu, "Spectral clustering for multi-type relational data," in *Proc. Int. Conf. Mach. Learn.*, 2006, pp. 585–592.
[40] Y. Zhou, H. Cheng, and J. X. Yu, "Graph clustering based on structural/attribute similarities," *Proc. VLDB Endowment*, vol. 2, no. 1, pp. 718–729, 2009.
[41] M. McPherson, L. Smith-Lovin, and J. M. Cook, "Birds of a feather: Homophily in social networks," *Annu. Rev. Sociol.*, vol. 27, no. 1, pp. 415–444, 2001.
[42] S. Gunnemann, I. Farber, B. Boden, and T. Seidl, "Subspace clustering meets dense subgraph mining: A synthesis of two paradigms," in *Proc. IEEE Int. Conf. Data Mining*, 2010, pp. 845–850.
[43] P. I. Sánchez, "Context selection on attributed graphs for outlier and community detection," Ph.D. dissertation, Dept. Inform., Karlsruhe Institute of Technology, Karlsruhe, Germany, 2015.
[44] Q. Gu and J. Han, "Towards feature selection in network," in *Proc. ACM Int. Conf. Inf. Knowl. Manage.*, 2011, pp. 1175–1184.
[45] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, 2015, Art. no. 436.
[46] H. Tong and C.-Y. Lin, "Non-negative residual matrix factorization with application to graph anomaly detection," in *Proc. SIAM Int. Conf. Data Mining*, 2011, pp. 143–153.
[47] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998.
[48] C. Zhou and R. C. Paffenroth, "Anomaly detection with robust deep autoencoders," in *Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2017, pp. 665–674.
[49] J. Tang, M. Qu, M. Wang, M. Zhang, J. Yan, and Q. Mei, "LINE: Large-scale information network embedding," in *Proc. Int. Conf. World Wide Web*, 2015, pp. 1067–1077.
[50] A. Grover and J. Leskovec, "node2vec: Scalable feature learning for networks," in *Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2016, pp. 855–864.
[51] S. Bickel and T. Scheffer, "Multi-view clustering," in *Proc. IEEE Int. Conf. Data Mining*, 2004, pp. 19–26.
[52] A. Alvarez, M. Yamada, A. Kimura, and T. Iwata, "Clustering-based anomaly detection in multi-view data," in *Proc. ACM Int. Conf. Inf. Knowl. Manage.*, 2013, pp. 1545–1548.
[53] Y.-X. Ji *et al.*, "Multi-view outlier detection in deep intact space," in *Proc. IEEE Int. Conf. Data Mining*, 2019, pp. 1132–1137.
[54] B. Perozzi and L. Akoglu, "Scalable anomaly ranking of attributed neighborhoods," in *Proc. SIAM Int. Conf. Data Mining*, 2016, pp. 207–215.
[55] M. Abadi *et al.*, "TensorFlow: A system for large-scale machine learning," in *Proc. USENIX Symp. Operating Syst. Des. Implementation*, 2016, pp. 265–283.
[56] P. Velickovic, W. Fedus, W. L. Hamilton, P. Lio, Y. Bengio, and R. D. Hjelm, "Deep graph infomax," in *Proc. Int. Conf. Learn. Representations*, 2019.
[57] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proc. Int. Conf. Artif. Intell. Statist.*, 2010, pp. 249–256.
[58] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. Int. Conf. Learn. Representations*, 2015.
[59] P. I. Sánchez, E. Müller, F. Laforet, F. Keller, and K. Böhm, "Statistical selection of congruent subspaces for mining attributed graphs," in *Proc. IEEE Int. Conf. Data Mining*, 2013, pp. 647–656.
[60] D. B. Skillicorn, "Detecting anomalies in graphs," in *Proc. IEEE Int. Conf. Intell. Secur. Informat.*, 2007, pp. 209–216.
[61] X. Song, M. Wu, C. Jermaine, and S. Ranka, "Conditional anomaly detection," *IEEE Trans. Knowl. Data Eng.*, vol. 19, no. 5, pp. 631–645, May 2007.
[62] U. Brefeld, C. Büscher, and T. Scheffer, "Multi-view discriminative sequential learning," in *Proc. Eur. Conf. Mach. Learn.*, 2005, pp. 60–71.
[63] S. Sun, "A survey of multi-view machine learning," *Neural Comput. Appl.*, vol. 23, no. 7/8, pp. 2031–2038, 2013.
[64] V. Metsis, I. Androutsopoulos, and G. Paliouras, "Spam filtering with naive bayes-which naive bayes?" in *Proc. Conf. Email Anti-Spam*, 2006, pp. 28–69.

[65] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 770–778.
[66] B. Yu, H. Yin, and Z. Zhu, "Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting," in *Proc. Int. Joint Conf. Artif. Intell.*, 2018, pp. 3634–3640.

Zhen Peng received the BEng degree from the School of Software Engineering, Xi'an Jiaotong University, China, in 2017. She is currently working toward the PhD degree in the School of Computer Science and Technology, Xi'an Jiaotong University, China. Her research interests include machine learning and its applications, such as social network analysis and graph mining.

Minnan Luo received the PhD degree from the Department of Computer Science and Technology, Tsinghua University, China, in 2014. She was a postdoctoral research with the School of Computer Science, Carnegie Mellon University, Pittsburgh, Pennsylvania. She is currently an associate professor with the School of Computer Science and Technology, Xi'an Jiaotong University, China. Her research interests include machine learning and optimization, data mining, image processing, and cross-media retrieval.



Jundong Li received the BEng degree from the College of Computer Science and Technology, Zhejiang University, China, in 2012, the MSc degree from the Department of Computing Science, University of Alberta, Canada, in 2014, and the PhD degree of computer science from Arizona State University, Tempe, Arizona, in 2019. He is an assistant professor of Electrical and Computer Engineering Department and Computer Science Department at the University of Virginia, Charlottesville, Virginia. His research interests include data mining, machine learning and their applications in social media.

Luguo Xue received the BEng degree from the School of Electronic and Information Engineering, and the MSc degree from the School of Computer Science and Technology from Xi'an Jiaotong University, China, in 2017 and 2020, respectively. He currently works at ByteDance. His research interests include graph mining and its applications.

Qinghua Zheng received the BEng degree of computer software, the MSc degree of computer organization and architecture, and the PhD degree of system engineering from Xi'an Jiaotong University, China, in 1990, 1993, and 1997, respectively. He was a postdoctoral researcher with Harvard University, Cambridge, Massachusetts, in 2002. He is currently a professor with the School of Computer Science and Technology, Xi'an Jiaotong University, China. His research interests include computer network security, intelligent E-learning theory and algorithm, multimedia e-learning, and trustworthy software.