Anomaly Detection with Deep Graph Autoencoders on Attributed Networks

Dali Zhu*†, Yuchen Ma*†, Yinlong Liu*†
*Institute of Information Engineering, Chinese Academy of Sciences, Beijing, China
†School of Cyber Security, University of Chinese Academy of Sciences, Beijing, China
{zhudali,mayuchen,liuyinlong}@iie.ac.cn

***Abstract*—Anomaly detection on attributed networks aims to differentiate rare nodes that are significantly different from the majority. It plays an important role in various practical scenarios, such as intrusion detection and fraud detection. However, existing graph-based methods mainly adopt shallow models that cannot capture the highly non-linear interactions between nodes in an attribute network consisting of different information modalities. To tackle the above issues, in this paper, we propose a novel deep model named DeepAE for anomaly detection which (a) can capture the high non-linearity in both topological structure and nodal attributes through graph convolutional autoencoder, (b) fully exploits the intrinsic information of the network with the description of various proximities, (c) and preserve the differences between anomalies and the majority by applying Laplacian sharpening. We perform anomaly detection by measuring the reconstruction errors of nodes. Experimental results on real-world datasets demonstrate that DeepAE outperforms the state-of-art baselines.**

***Index Terms*—Anomaly Detection, Attributed Networks, Autoencoder, Graph Convolutional Network**

I. Introduction

In many real-world applications, information systems are often represented using attributed networks where both the topological structure and nodal attributes are available for the network analysis. Typical examples include social media networks, paper citation networks, communication networks, where links indicate the relationship between users and nodes associated with rich attributes portray users profiles. In recent years, there has been a surge of interest in anomaly detection on attributed networks, and the task is to identify rare instances whose patterns or behaviors deviate significantly from the majority reference nodes [1]. Detecting such anomalies have been applied to a wide range of domains, including spam detection in social networks, fraud detection in financial trading networks, and cyber-attacks detection in computer networks [2]–[4].

Current research on anomaly detection mainly focuses on identifying rare substructures or subgraphs based on the network's structural information to discover anomalies that are different from frequent subgraphs [5], [6]. Other works utilize nodal attributes to quantify the anomaly degree of the node compared with its surrounding neighborhood [7], [8]. Unfortunately, most of them fail to measure the affinity between nodes and attributes, and shallow models are incapable of capturing the highly non-linear interactions in the attribute network.

Recently, graph convolutional network (GCN) [9] has proven to be an effective approach to generalizing convolutional neural network (CNN) to modeling non-Euclidean data such as manifolds or attributed networks. GCN seamlessly integrates the network structure and nodal attributes into a unified low-dimensional representation. It has been shown to be a special form of Laplacian smoothing [10] that mixes the features of a node with its neighbors to make the same cluster similar. VGAE [11] is a successful attempt to exploit deep latent representations of graph-structured data using an autoencoder framework of GCN. However, the pure GCN based model lacks the description of the implicit relationship between different nodes, and the output features may be over-smoothed by multiple convolutional layers, making the differences between anomaly and the majority indistinguishable. Therefore existing deep neural models cannot be directly applied to anomaly detection in attributed networks.

To address the problems as mentioned above, in this paper, we propose a novel unsupervised method, DeepAE, which embeds the topological structure as well as nodal attributes seamlessly into a unified representation through deep graph autoencoder. DeepAE fully exploits the intrinsic information of the network, captures the local and global relational information, and makes the semantic information of nodal attributes in the hidden layer consistent with the original network. Furthermore, Laplacian sharpening is introduced to preserve the differences between normal nodes and anomalies [12]. Then we spot anomalies by leveraging the reconstruction errors of the two information modalities.

In summary, our contributions are listed as follow:

- We propose a novel DeepAE model to detect anomalies through deep graph autoencoder on the attributed network.
- DeepAE can simultaneously capture the high non-linearity in both topological structure and nodal attributes where the implicit relationship of nodes is well exploited. The decoder is customized to amplify the differences between normal nodes and anomalies through Laplacian sharpening.
- Experimental results show that DeepAE outperforms several state-of-art methods on benchmark datasets.

978-1-7281-8086-1/20/$31.00 ©2020 IEEE

II. Problem Statement

In this section, we formally define the anomaly detection problem in attributed networks using high-order, first-order and semantic proximities.

***Definition** 1 (Attributed Network):* An attributed network is defined as $G = (A, X)$ with n nodes, where $A = \{a_{i,j}\}_{i,j=1}^{n}$ represents the adjacency matrix and $X = \{x_{ij}\}_{i=1,\ldots,n}^{j=1,\ldots,m}$ represents the attribute matrix. Each node is associated with a m-dimensional attributes row vector $\mathbf{x}_i \in \mathbb{R}^m (i = 1, \ldots, n)$. If there is a link between the i-th node and the j-th node, $a_{i,j} = 1$. Otherwise, $a_{i,j} = 0$.

***Definition** 2 (Semantic Proximity [13]):* Given two different nodes in the attributed network, the semantic proximity between pairwise vertices is determined by the similarity according to their attributes.

The semantic proximity refers to how similar the attribute of a pair of nodes is. The similarity distribution of nodes in the latent representation should be consistent with that in the original network.

***Definition** 3 (First-order Proximity [14]):* The first-order proximity describes the similarity between a pair of nodes according to their topological connectivity. For each pair of nodes, $a_{i,j} > 0$ indicates the first-order proximity between nodes v_i and v_j. Otherwise, the first-order proximity is 0.

The first-order approximation is an intuitive representation between two connected nodes in a real network. Two related instances are more likely to exhibit similar behavior. The above two proximities can be regarded as local proximity from the perspective of attribute and structure, respectively. Correspondingly, we introduce high-order proximity to complements the first-order proximity and preserve the global network.

***Definition** 4 (High-order Proximity [15]):* The high-order proximity of two nodes describes the similarity between their neighbors. Given an attributed network $G = (A, X)$, let $P = \left(I^1 + I^2 + \ldots, +I^m\right)$ denotes the high-order proximity, where I^m is the m-step relational information through the probability propagation in the network.

***Definition** 5 (Deep Network Embedded Anomaly Detection):* Given an attributed network $G = (A, X)$, the objective of deep network embedded anomaly detection first learns a map $f : \{A, X\} \rightarrow \mathbb{R}^d$, where $d \ll |A|$. Both of the topological relations and nodal attributes should be preserved into a low-dimensional representation. Then, the anomalies which are significantly different from the majority reference nodes can be detected according to their anomaly score.

III. Methodology

Based on the previous analysis, we present a novel deep autoencoder approach DeepAE for anomaly detection, as shown in Fig. 1. The network structure and the node attributes embedded into the same representation space through a joint framework modeled by graph autoencoder. The decoder part of DeepAE is customized to restore the differences between normal nodes and anomalies, which will be used for anomaly detection. In order to preserve the complex interaction between two information modalities, we add constraints to refine the representation. Afterward, we estimate the abnormality of each node by measuring the reconstruction errors to spot anomalies. Details are introduced as follows.

A. Graph Autoencoder

DeepAE consists of multiple layers of graph convolution to capture the highly non-linear information simultaneously from network structure and nodal attributes. The m-th layer takes the adjacency matrix A and the node embedding matrix $H^{(m)}$ as the input, and updates the node embedding matrix to $H^{(m+1)}$ with the weight matrix $\Theta^{(m)}$ as the output. Mathematically, the encoder is defined as:

$$H^{(m+1)} = \sigma\left(\overline{A} H^{(m)} \Theta^{(m)}\right), m = 2, ..., M \tag{1}$$

where $\overline{A}$ denotes the symmetric normalized adjacency of A: $\overline{A} = \widetilde{D}^{-\frac{1}{2}} \widetilde{A} \widetilde{D}^{-\frac{1}{2}}$, $\widetilde{A} = A + I$, $\widetilde{D} = \operatorname{diag}\left(\sum_j \widetilde{A}_{ij}\right)$, and $\sigma(\cdot)$ denotes a non-linear activation function (typically ReLU). $H^0 = X$ is the initial input embedding matrix and I is the identity matrix. After several iterations, the convolution layers aggregate the embeddings of nodes as well as their neighbors, and captures the M-step relational information of both topological structure and nodal attributes. Therefore the last hidden layer $H^{(M)}$ is the target representations of the network nodes transformed from the initial attributes. Accordingly, the decoder also has M layers to obtain the reconstructed attribute matrix $\widehat{X}$, and the reconstructed adjacency matrix $\widehat{A} = sigmoid(HH^T)$ [11].

B. Laplacian sharpening

GCN has been demonstrated to be a special form of Laplacian smoothing [10], which can be written as:

$$H^{(m)} = (I - \gamma \widetilde{D}^{-1}(\widetilde{D} - \widetilde{A}))H^{(m-1)} \tag{2}$$

where $H^{(m)}$ is the new representation of $H^{(m-1)}$, and γ $(0 < \gamma \leq 1)$ is a regularization parameter which controls the weighting between the features of current node and its neighbors. If we set $\gamma = 1$ and replace $\widetilde{D}^{-1}\widetilde{A}$ with $\overline{A}$, Eq. (2) is changed into $H^{(m)} = \widetilde{D}^{-\frac{1}{2}} \widetilde{A} \widetilde{D}^{-\frac{1}{2}} H^{(m-1)}$ which equals to the normal GCN layer in Eq. (1). Laplacian smoothing makes the features of a node and its neighbors become similar which benefit the subsequent classification task. But the smoothed features may be over-mixed with their neighbors in the same cluster and become indistinguishable, so that it is more difficult to identify individual instance especially the anomaly (rare instance) since the uniqueness of features in the original network has been weakened.

In order to distinguish anomalous nodes from the majority within the community, we propose the Laplacian sharpening which amplify the difference between the feature of current node and its neighbors by reducing the impact of smoothing operations. With the replacement of $\overline{A}$, the decoder layer is reformulated as:

$$H_d^{(m+1)} = \sigma(((1-\gamma)I + \gamma \widetilde{D}^{-\frac{1}{2}} \widetilde{A} \widetilde{D}^{-\frac{1}{2}}) H_d^{(m)} \Theta_d^{(m)}) \tag{3}$$

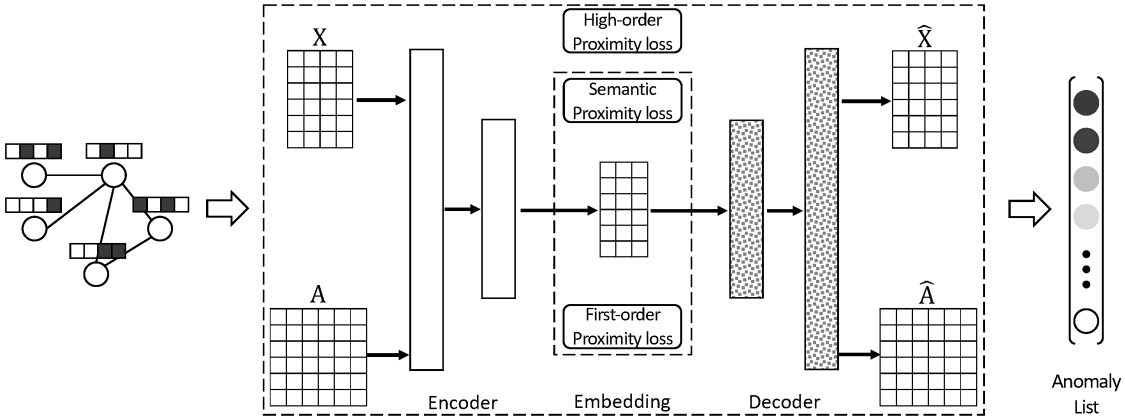


Fig. 1. The overall framework of our proposed DeepAE model

where $H_d^{(0)} = H^{(M)}$ and $\Theta_d^{(m)}$ is the trainable parameters in the decoder layer.

C. Proximity Preservation

To preserve the **high-order proximity**, we minimize the joint reconstruction loss of the topological structure and nodal attributes as shown bellow:

$$\mathcal{L}_h = \beta \sum_{i=1}^{n} \|\mathbf{x}_i - \hat{\mathbf{x}}_i\|_2^2 + \sum_{i=1}^{n} \|\mathbf{a}_i - \hat{\mathbf{a}}_i\|_2^2 \tag{4}$$

where β is a weight parameter which balance the impact of both reconstructions. The constraints can enforce the neural network to capture the data manifold smoothly, thereby preserve the network information among a wider range of samples. As anomalies are much harder to be accurately represented in the hidden layer than normal nodes, the reconstruction error is larger.

It is not enough to just preserve the high-order network information, but also to capture the local proximity. The **first-order proximity** can be regarded as a supervision that constrains the consistent representation of similar nodes in the hidden layer. Inspired by Laplacian eigenmaps (LE) [16], we define the following loss function:

$$\mathcal{L}_f = \sum_{i,j=1}^{n} \hat{a}_{i,j} \left\| \mathbf{h}_i^{(M)} - \mathbf{h}_j^{(M)} \right\|_2^2 = \sum_{i,j=1}^{n} \hat{a}_{i,j} \|\mathbf{h}_i - \mathbf{h}_j\|_2^2 \tag{5}$$

where $\mathbf{h}_i^{(M)}$ is the one-dimensional embedding of node i. Noting that $tr\left(H^T L H\right) = \frac{1}{2}\sum_{i,j=1}^{n} \hat{a}_{i,j} \|\mathbf{h}_i - \mathbf{h}_j\|_2^2$ the loss function can be reformulated as the following term:

$$\mathcal{L}_f = 2tr\left(H^T L H\right) \tag{6}$$

where $L = D - \widehat{A}$, D is the diagonal matrix of $\widehat{A}$ whose entries are the degrees of each node.

As aforementioned analysis, nodes with similar features in the original network should also be same in the embedding space, thus maintaining local proximity in the perspective of node attribute. Inspired by LINE [17], to model the **semantic proximity**, the empirical probability of $P(i,j)$ can be defined as:

$$P(i,j) = \frac{sim_{ij}}{\sum_{a_{ij} \in A} sim_{ij}} \tag{7}$$

where sim_{ij} measures the similarity of two connected nodes which can be calculated by cosine function. We simulate the interaction between two nodes in the form of joint probability $\widehat{P}(i,j)$ by sigmoid function $\widehat{P} = sigmoid(\mathbf{h_i h_j}^T)$ where $\mathbf{h}_i \in R^{nxd}$ is the low-dimensional vector of node v_i. To preserve the semantic proximity, we choose the KL-divergence to minimize the co-occurrence probability between two distributions as the following loss function:

$$\mathcal{L}_s = KL(P \| \hat{P}) \propto - \sum_{a_{ij} \in A} sim_{ij} \log \hat{P}(i,j) \tag{8}$$

As explained in Fig. 1, to preserve proximities and learn the unified complementary representation, we combine the three loss functions mentioned before and jointly minimized the following objective function:

$$\begin{aligned} \mathcal{L} =& \beta \sum_{i=1}^{n} \|\mathbf{x}_i - \hat{\mathbf{x}}_i\|_2^2 + \sum_{i=1}^{n} \|\mathbf{a}_i - \hat{\mathbf{a}}_i\|_2^2 \\ &+ 2tr\left(H^T L H\right) - \sum_{a_{ij} \in A} sim_{ij} \log \hat{P}(i,j) \end{aligned} \tag{9}$$

D. Anomaly Detection

By minimizing the objective function, the latent representation iteratively approximates the original input network until the objective function converges. The weight parameters are optimized by gradient descent and initialized by Xavier method. After several iterations, the reconstruction errors are used to estimate the likelihood of a node being anomalous [18]. Then the anomaly score for each node v_i is computed as follows:

$$socre(\mathbf{v}_i) = \beta \|\hat{\mathbf{x}}_i - \mathbf{x}_i\|_2^2 + \|\hat{\mathbf{a}}_i - \mathbf{a}_i\|_2^2 \tag{10}$$

Specifically, nodes with higher anomaly scores are considered to be anomalies. We can calculate the anomaly ranking based on the corresponding anomaly score.

TABLE I
DESCRIPTION OF BENCHMARK DATASETS

Dataset	#Nodes	#Edges	#Attributes	#Anomalies
Cora	2,780	5,278	1,433	5%
Citeseer	3,327	4,732	3,703	5%
PubMed	19,717	44,338	500	5%

TABLE II
NEURAL NETWORK STRUCTURES

Dataset	# neurons in each layer
Cora	1433-400-100
Citeseer	3703-500-100
PubMed	500-300-100

E. Computational Analysis

The computational complexity of each iteration mainly depends on the computation of graph convolution kernel $\widetilde{D}^{-\frac{1}{2}}\widetilde{A}\widetilde{D}^{-\frac{1}{2}}XW$ and takes $\mathcal{O}(mdh)$ time (given the assumption that A is sparse matrix), where m is the cardinality of edges in the graph, d is the feature dimension on the attributed network, and h is the feature dimension of each convolutional layer. Since the complexity is linear in the number of edges on the network, our proposed model is computationally efficient. Besides, the complexity of Eq.(6) is $\mathcal{O}(md)$, and there is another prediction layer to reconstruct the adjacency matrix, the total time complexity is $\mathcal{O}\left(mdH + n^2\right)$ where H is the sum of all feature dimensions across all convolutional layers.

IV. EXPERIMENTS

A. Datasets

In our experiments, we adopt three real-world datasets[1] to evaluate the effectiveness of DeepAE. The detailed statistics of each dataset are summarized in Tab. I. The edges of each network denote citation relationships, and each node is described by the bag-of-words attribute vector of the corresponding paper. We need to generate the required dataset as there is no ground truth of anomalies in these datasets. We inject a set of anomalies from both the structural perspective and attribute perspective according to the previous research [19], [20]. As small clique is an anomalous structure since it represents a set of nodes that are much more closely related than average [19], we randomly select m nodes to form a fully connected clique, repeat n times, and all the mn nodes are marked as structural anomalies. To guarantee an equal ratio of anomalies from both perspectives, we then randomly select another mn nodes from the network, randomly exchange their attribute value to generate attribute anomalies, while the topological relationship remains unchanged.

B. Baseline Algorithms

We compare DeepAE with five baseline algorithms. The details are as follows:

- LOF [7] is an attribute-based detection algorithm that detects anomalies according to the local density of nodes.
- SCAN [21] is a structure-based detection algorithm that detects anomalies based on the structure and connectivity of nodes at the structural level and only considers structure information.
- CODA [6] detects anomalies based on community detection and only considers attribute information.
- Radar [22] detects anomalies using the residual analysis and the coherence with network information.
- GCN [9] simultaneously project the graph structures and node features of the attributed network into a latent representation. We use vanilla GCN as the convolutional layer of the autoencoder for anomaly detection.

[1]https://github.com/kimiyoung/planetoid/tree/master/data

C. Evaluation Metrics

Evaluation metrics include *AUC*, *precision*@K and *recall*@K. Their analysis detail is as follows:

- AUC: AUC is defined as the area under the ROC curve, which is a widely used metric to measure the ranking quality of a classifier. The higher the AUC, the better the performance of the model at distinguishing between different classes.
- Precision@K: This metric measures the proportion of known anomalies in the top-k position of ranked nodes.
- Recall@K: It measures the proportion of true anomalies selected out of all the ground truth anomalies.

D. Parameter Settings

The architecture of our DeepAE for three datasets is summarized in Tab. II. We use the Relu activation function in both encoder and decoder parts and the dimension of the last encoder layer for different datasets is the same. DeepAE is optimized with Adam [23] optimization technique and trained for 300 epochs. We set the learning rate of the reconstruction loss to 0.025. In addition, we need to fine-tune hyper-parameter γ and β according to different datasets to achieve better performance. The parameters of baselines are set as described in the corresponding papers.

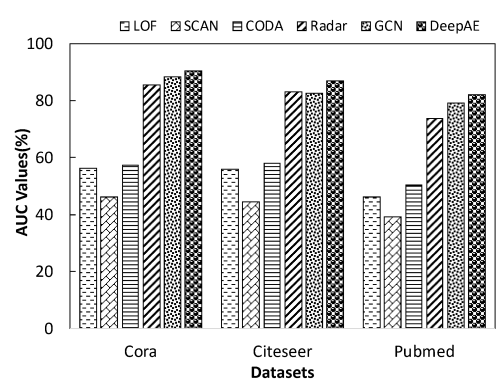
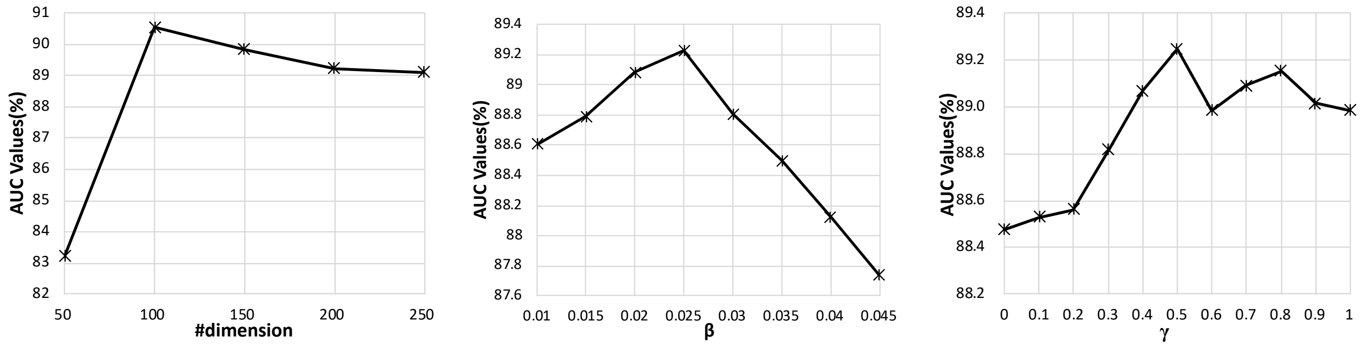


Fig. 2. AUC scores of different methods on three datasets.



(a) #dimension

(b) β

(c) γ

Fig. 3. Sensitivity w.r.t dimension, β and γ

TABLE III
precision@K AND *recall*@K ON DATASETS FOR ANOMALY DETECTION.

Precision@K									
	Cora			Citeseer			PubMed		
K	50	100	200	50	100	200	50	100	200
LOF	0.473	0.369	0.325	0.531	0.442	0.421	0.076	0.073	0.052
Radar	0.793	0.784	**0.762**	0.778	0.756	0.713	0.557	0.531	0.510
GCN	0.815	0.789	0.736	0.806	0.781	0.740	0.617	0.592	0.573
DeepAE	**0.831**	**0.803**	0.745	**0.821**	**0.794**	**0.773**	**0.631**	**0.623**	**0.597**
Recall@K									
	Cora			Citeseer			PubMed		
K	50	100	200	50	100	200	50	100	200
LOF	0.057	0.087	0.113	0.073	0.096	0.121	0.013	0.019	0.023
Radar	0.087	0.213	0.245	0.083	0.167	**0.279**	0.051	0.097	0.183
GCN	0.104	0.217	0.349	0.091	0.193	0.247	0.057	0.103	0.169
DeepAE	**0.124**	**0.243**	**0.401**	**0.096**	**0.213**	0.268	**0.073**	**0.135**	**0.189**

E. Experiment Results

We evaluate the anomaly detection performance by comparing DeepAE with the five baseline algorithms. Fig. 2 illustrates the experimental results in terms of AUC values on three datasets. Precision@K and Recall@K are utilized as the evaluation metric, and the results are reported in Tab. III. There are no results of SCAN and CODA in Tab. III since they are cluster-based algorithms that cannot provide an accurate ranking list for all nodes. From the evaluation results, we draw the following conclusions: (1) The proposed DeepAE outperforms other baseline methods on all the three datasets. We can see that models (Radar, GCN, and DeepAE) based on dual-modal information are superior to the conventional methods (LOF, SCAN, and CODA) merely based either on attribute or structure information, which verifies the importance of performing anomaly detection in attributed networks. (2) Compared with other methods, the deep models (GCN, DeepAE) breaks through the limitation of shallow mechanisms to handle the high non-linearity issue and fully exploits the intrinsic information of the network. (3) Trough the comparison of vanilla GCN and DeepAE, the customized autoencoder achieves better performance in each metric, which demonstrates the importance of preserving the differences between nodes during the embedding.

F. Effects of Parameters

In this part, we investigate the impact of the two key hyper-parameters (β and γ) and the number of the embedding dimension d on the performance of DeepAE. The studies are performed on the Cora dataset with 400 injected anomalies. Fig. 3(a) shows the trend of AUC under different dimensions of the embedding layer. We can observe that the performance of DeepAE is stable, and AUC is higher than 0.89 when $d \geq 100$. However, as the dimension continues to increase, excessive embedding dimension introduces noise, which disturbs the latent representation, resulting in performance degradation. The hyper-parameter β balances the weight between structure reconstruction and attribute reconstruction. Fig. 3(b) shows the trend of AUC under different β, indicating that a suitable balance factor can effectively improve performance. Fig. 3(a) shows how the value of γ influences the performance. In two extreme cases, DeepAE will degenerate into a vanilla GCN based model when γ is set to 0 or degenerate into naive MLP network when γ is set to 1. When γ = 0.5, the model can achieve the best performance.

V. Related Work

A. Graph Based Anomaly Detection

With the extensive application of attribute networks in modeling complex systems, the research of anomaly detection on attributed networks has attracted widespread attention. Some of them mainly focus on the rare substructures or subgraphs in the graph, so the inverse of frequent subgraphs can be regarded as structural anomalies [5] but fail to spot the attribute anomaly. SCAN [21] identifies anomalies by using the structure and the connectivity of the nodes as criteria. OddBall [24] and AMEN [8] extracts features and finds patterns based on the ego-network information for each node to discover anomalous nodes. Other attribute-based methods assume that similar graphs should share similar properties [25]. LOF [7] computes the local density deviation of a given data point with respect to the surrounding neighborhood. CODA [6] simultaneously finds communities as well as spots community anomalies using Markov random fields. Recently, Radar [22] adopt residual analysis and its coherence with network information to perform anomaly detection in a more general way. However, all the aforementioned methods are incapable of handling complex interaction between different information modalities and limited by their shallow mechanisms.

B. Graph Autoencoders

Autoencoders aims to learn the representation of data with minimum reconstruction loss, which preserves structure information and properties of graphs. The implicit assumption is that graphs have an inherent, potentially non-linear low-rank structure [26]. With the rise of deep learning, a large number of deep models for various learning tasks have emerged. DNGR [27] utilizes stacked denoising autoencoders and encodes each vertex to capture the network's non-linearity, and SDNE [14] and LINE [17] further preserves both the local and global network structures. LANE [28] jointly incorporate label, attribute, and structure information into embedding while preserving their correlations. DANE [13] learn consistent and complementary node representations in both topological structure and attributes by preserving various proximities. GAE [9] simultaneously learn the structure and attribute information into latent space through multiple graph convolutional layers. VGAE [11] further combines the graph convolutional network with the variational autoencoder architecture. These models cannot be directly applied to anomaly detection. So we aim to design a deep detection model to bridge the gap.

VI. Conclusion

In this paper, we propose a novel deep autoencoder approach DeepAE for anomaly detection on attributed networks. Specifically, We design a customized graph autoencoder to capture the highly non-linear information in network topological structure and nodal attributes. To further address the complex interaction problem, we jointly preserve the first-order, high-order, and semantic proximity to make two types of information complement each other towards a unified representation. Meanwhile, Laplacian sharpening is leveraged to preserve the differences between normal nodes and anomalies. By jointly optimizing them in the deep model, the reconstruction errors are then employed to spot anomalies. The experimental results demonstrate the effectiveness of our approach to anomaly detection compared with state-of-art methods.

References

[1] C. C. Aggarwal, "Outlier analysis," in *Data mining*. Springer, 2015, pp. 237–263.

[2] P. Rozenshtein, A. Anagnostopoulos, A. Gionis, and N. Tatti, "Event detection in activity networks," in *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2014, pp. 1176–1185.

[3] D. H. Chau, S. Pandit, and C. Faloutsos, "Detecting fraudulent personalities in networks of online auctioneers," in *European Conference on Principles of Data Mining and Knowledge Discovery*. Springer, 2006, pp. 103–114.

[4] Q. Ding, N. Katenka, P. Barford, E. Kolaczyk, and M. Crovella, "Intrusion as (anti) social communication: characterization and detection," in *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2012, pp. 886–894.

[5] C. C. Noble and D. J. Cook, "Graph-based anomaly detection," in *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2003, pp. 631–636.

[6] J. Gao, F. Liang, W. Fan, C. Wang, Y. Sun, and J. Han, "On community outliers and their efficient detection in information networks," in *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2010, pp. 813–822.

[7] M. M. Breunig, H.-P. Kriegel, R. T. Ng, and J. Sander, "Lof: identifying density-based local outliers," in *ACM sigmod record*, vol. 29, no. 2. ACM, 2000, pp. 93–104.

[8] B. Perozzi and L. Akoglu, "Scalable anomaly ranking of attributed neighborhoods," in *Proceedings of the 2016 SIAM International Conference on Data Mining*. SIAM, 2016, pp. 207–215.

[9] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," *arXiv preprint arXiv:1609.02907*, 2016.

[10] Q. Li, Z. Han, and X.-M. Wu, "Deeper insights into graph convolutional networks for semi-supervised learning," in *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[11] T. N. Kipf and M. Welling, "Variational graph auto-encoders," *arXiv preprint arXiv:1611.07308*, 2016.

[12] G. Taubin, "A signal processing approach to fair surface design," in *Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*, 1995, pp. 351–358.

[13] H. Gao and H. Huang, "Deep attributed network embedding." in *IJCAI*, vol. 18, 2018, pp. 3364–3370.

[14] D. Wang, P. Cui, and W. Zhu, "Structural deep network embedding," in *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2016, pp. 1225–1234.

[15] S. Cao, W. Lu, and Q. Xu, "Grarep: Learning graph representations with global structural information," in *Proceedings of the 24th ACM international on conference on information and knowledge management*, 2015, pp. 891–900.

[16] M. Belkin and P. Niyogi, "Laplacian eigenmaps for dimensionality reduction and data representation," *Neural computation*, vol. 15, no. 6, pp. 1373–1396, 2003.

[17] J. Tang, M. Qu, M. Wang, M. Zhang, J. Yan, and Q. Mei, "Line: Large-scale information network embedding," in *Proceedings of the 24th international conference on world wide web*, 2015, pp. 1067–1077.

[18] J. Chen, S. Sathe, C. Aggarwal, and D. Turaga, "Outlier detection with autoencoder ensembles," in *Proceedings of the 2017 SIAM international conference on data mining*. SIAM, 2017, pp. 90–98.

[19] D. B. Skillicorn, "Detecting anomalies in graphs," in *2007 IEEE Intelligence and Security Informatics*. IEEE, 2007, pp. 209–216.

[20] X. Song, M. Wu, C. Jermaine, S. Ranka *et al.*, "Conditional anomaly detection," *IEEE Trans. Knowl. Data Eng.*, vol. 19, no. 5, pp. 631–645, 2007.

[21] X. Xu, N. Yuruk, Z. Feng, and T. A. Schweiger, "Scan: a structural clustering algorithm for networks," in *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2007, pp. 824–833.

[22] J. Li, H. Dani, X. Hu, and H. Liu, "Radar: Residual analysis for anomaly detection in attributed networks." in *IJCAI*, 2017, pp. 2152–2158.

[23] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[24] L. Akoglu, M. McGlohon, and C. Faloutsos, "Oddball: Spotting anomalies in weighted graphs," in *Pacific-Asia Conference on Knowledge Discovery and Data Mining*. Springer, 2010, pp. 410–421.

[25] U. Kang, S. Papadimitriou, J. Sun, and H. Tong, "Centralities in large networks: Algorithms and observations," in *Proceedings of the 2011 SIAM international conference on data mining*. SIAM, 2011, pp. 119–130.

[26] Z. Zhang, P. Cui, and W. Zhu, "Deep learning on graphs: A survey," *IEEE Transactions on Knowledge and Data Engineering*, 2020.

[27] S. Cao, W. Lu, and Q. Xu, "Deep neural networks for learning graph representations," in *Thirtieth AAAI Conference on Artificial Intelligence*, 2016.

[28] X. Huang, J. Li, and X. Hu, "Label informed attributed network embedding," in *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining*. ACM, 2017, pp. 731–739.