ANOMALYDAE: DUAL AUTOENCODER FOR ANOMALY DETECTION ON ATTRIBUTED NETWORKS

Haoyi Fan¹, Fengbin Zhang¹, Zuoyong Li²

¹School of Computer Science and Technology, Harbin University of Science and Technology, Harbin 150080, China.
²Fujian Provincial Key Laboratory of Information Processing and Intelligent Control, Minjiang University, Fuzhou 350121, China.
{isfanhy, zhangfengbin}@hrbust.edu.cn, fzulzytdq@126.com.

ABSTRACT
Anomaly detection on attributed networks aims at finding nodes whose patterns deviate significantly from the majority of reference nodes, which is pervasive in many applications such as network intrusion detection and social spammer detection. However, most existing methods neglect the complex cross-modality interactions between network structure and node attribute. In this paper, we propose a deep joint representation learning framework for anomaly detection through a dual autoencoder (AnomalyDAE), which captures the complex interactions between network structure and node attribute for high-quality embeddings. Specifically, AnomalyDAE consists of a structure autoencoder and an attribute autoencoder to learn both node embedding and attribute embedding jointly in latent space. Moreover, attention mechanism is employed in structure encoder to learn the importance between a node and its neighbors for an effective capturing of structure pattern, which is important to anomaly detection. Besides, by taking both the node embedding and attribute embedding as inputs of attribute decoder, the cross-modality interactions between network structure and node attribute are learned during the reconstruction of node attribute. Finally, anomalies can be detected by measuring the reconstruction errors of nodes from both the structure and attribute perspectives. Extensive experiments on real-world datasets demonstrate the effectiveness of the proposed method.

Index Terms— Anomaly detection, attributed networks, dual autoencoder, graph signal processing

1. INTRODUCTION
Attributed networks [1] are ubiquitous in the real world such as social networks [2], communication networks [3], and product co-purchase networks [4], in which each node is associated with a rich set of attributes or characteristics, in addition to the raw network topology.

Although above mentioned algorithms had their fair share of success, these methods either suffer from severe computational overhead caused by shallow learning mechanisms and subspace selection, or neglect the complex interactions between nodes and attributes by only learning the representations for nodes [14], while interactions between two different modality sources are of great importance for anomaly detection task to capture both structure and attribute induced anomalies. To alleviate the above-mentioned problems, in this paper, we propose a deep joint representation learning framework for anomaly detection through a dual autoencoder (AnomalyDAE), which captures the complex interactions between the network structure and node attribute for high-quality embeddings. Different from [14], which employs one single graph convolutional network (GCN) [15] based encoder for node embedding, AnomalyDAE consists of a structure autoencoder and an attribute autoencoder to learn the latent representation of nodes and attributes jointly by reconstructing the original network topologies and node attributes respectively. Then, anomalies in the network are detected by measuring the reconstruction errors of nodes from both the structure and attribute perspectives.

In sum, the main contributions of this paper are as follows:
• We propose a deep joint representation learning framework for anomaly detection on attributed networks via a dual autoencoder, in which, the complex cross-modality interactions between the network structure and node attribute are captured by learning both node embeddings and attribute embeddings jointly, and then the anomalies can be measured from both the structure and attribute perspectives.

• We conduct extensive experiments on multiple real-world datasets, and the results show that AnomalyDAE consistently outperforms state-of-the-art deep model significantly, with up to 22.32% improvement in terms of the ROC AUC score. The source codes are publicly available.

2. NOTATIONS AND PROBLEM STATEMENT

In this section, we formally define the frequently-used notations and the studied problem. The notations used in this paper are summarized in Table 1.

**Definition 1.** Attributed Network \( \mathcal{G} = \{ \mathcal{V}, \mathcal{E}, \mathcal{X} \} \) is defined as an undirected graph with \( M = |\mathcal{V}| \) nodes and \( |\mathcal{E}| \) edges, each of nodes is associated with a \( N = |\mathcal{X}| \) dimension attribute.

In this paper, we focus on the problem of anomaly detection on attributed networks. The definition of the problem is as follows:

**Problem 1.** Given an attributed network \( \mathcal{G} = \{ \mathcal{V}, \mathcal{E}, \mathcal{X} \} \), our goal is to detect the nodes that are rare and differ significantly from the majority of the reference nodes in terms both the structure and attribute information of the nodes. More formally, we aim to learn a score function \( f : \mathcal{V} \rightarrow \mathbb{R} \), to classify sample \( x_i \), based on the threshold \( \lambda \):

\[
y_i = \begin{cases} 
1, & \text{if } f(V_i) \geq \lambda, \\
0, & \text{otherwise}.
\end{cases}
\]

where \( y_i \) denotes the label of sample \( x_i \), with 0 being the normal class and 1 the anomalous class.

3. METHOD

In this section, we introduce the proposed AnomalyDAE in detail. The proposed AnomalyDAE is an end-to-end joint representation learning framework for anomaly detection on attributed network. As shown in Fig. 1, AnomalyDAE consists of a structure autoencoder for network structure reconstruction, and an attribute autoencoder for node attributes reconstruction. Take the learned node embedding from the structure encoder and the learned attribute embedding from the attribute encoder as inputs, the interactions between the network structure and the node attribute are jointly captured by both structure decoder and attribute decoder during the training of the model. Finally, anomalies in the network can be measured by the reconstruction errors of network structure and node attribute.

### 3.1. Structure Autoencoder

In order to obtain sufficient representative high-level node features, structure encoder firstly transforms the original observed node attribute \( \mathcal{X} \) into the low-dimentional latent representation \( \tilde{\mathcal{V}} \), which is shown as follows:

\[
\tilde{\mathcal{V}} = \sigma(\mathcal{X} \mathcal{W}^{i(1)} + \mathbf{b}^{i(1)})
\]

where \( \sigma(\bullet) \) is the activation function such as ReLU or Tanh, \( \mathcal{W}^{i(1)} \in \mathbb{R}^{D_{in} \times D_{out}} \) and \( \mathbf{b}^{i(1)} \in \mathbb{R}^{D_{out}} \) are the weight and bias learned by encoder, \( D_{in} \) and \( D_{out} \) are the dimensionalities of \( \mathcal{X} \) and \( \tilde{\mathcal{V}} \), respectively.

Given the transformed node embedding \( \tilde{\mathcal{V}} \), a graph attention layer [16] is then employed to aggregate the representation from neighbor nodes, by performing a shared attentional mechanism on the nodes:

\[
e_{i,j} = attn(\tilde{\mathcal{V}}_i, \tilde{\mathcal{V}}_j) = \sigma(\mathbf{a}^T \cdot [\mathcal{W}^{(2)} \tilde{\mathcal{V}}_i \Vert \mathcal{W}^{(2)} \tilde{\mathcal{V}}_j])
\]

where \( e_{i,j} \) is the importance weight of node \( \mathcal{V}_i \) to node \( \mathcal{V}_j \), \( attn(\bullet) \) denotes the neural network parameterized by weights \( \mathbf{a} \in \mathbb{R}^D \) and \( \mathcal{W}^{(2)} \in \mathbb{R}^{D \times D_{out}} \) that shared by all nodes, \( \Vert \) denotes the concatenate operation. Then, the final importance weight \( \gamma_{i,j} \) is normalized through the softmax function:

\[
\gamma_{i,j} = \frac{\exp(e_{i,j})}{\sum_{k \in N_i} \exp(e_{i,k})}
\]

where \( N_i \) denotes the neighbors of node \( \mathcal{V}_i \), which is provided by adjacency matrix \( \mathcal{A} \), and its final embedding \( \tilde{\mathcal{V}}_i \)

---

**Table 1. Notations.**

<table>
<thead>
<tr>
<th>Notation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \mathcal{G} )</td>
<td>Attributed network.</td>
</tr>
<tr>
<td>( \mathcal{V} )</td>
<td>A set of nodes in network.</td>
</tr>
<tr>
<td>( \mathcal{A} )</td>
<td>A set of node attributes in network.</td>
</tr>
<tr>
<td>( \mathcal{E} )</td>
<td>A set of edges in network.</td>
</tr>
<tr>
<td>( M )</td>
<td>The number of nodes.</td>
</tr>
<tr>
<td>( N )</td>
<td>The dimension of attribute.</td>
</tr>
<tr>
<td>( D )</td>
<td>The dimension of embedding.</td>
</tr>
<tr>
<td>( \mathcal{A} \in \mathbb{R}^{M \times M} )</td>
<td>Adjacency matrix of a network.</td>
</tr>
<tr>
<td>( \mathcal{X} \in \mathbb{R}^{M \times N} )</td>
<td>Attribute matrix of all nodes.</td>
</tr>
<tr>
<td>( \tilde{\mathcal{Z}}^V \in \mathbb{R}^{M \times D} )</td>
<td>Latent embedding of nodes.</td>
</tr>
<tr>
<td>( \tilde{\mathcal{Z}}^A \in \mathbb{R}^{N \times D} )</td>
<td>Latent embedding of attributes.</td>
</tr>
</tbody>
</table>
can be obtained by weighted sum based on the learned importance weights as follows:

$$Z_i^\gamma = \sum_{k \in N_i} \gamma_{i,k} \cdot Z_k^\gamma \quad (5)$$

Finally, structure decoder takes the final node embeddings $Z^\gamma$ as inputs to decode them for reconstruction of the original network structure:

$$\hat{A} = \text{Sigmoid}(Z^\gamma (Z^\gamma)^T) \quad (6)$$

where $\text{Sigmoid}()$ is the sigmoid activation function. Here, inner product between two node embeddings is performed to estimate the probability of a link between two nodes:

$$p(\hat{A}_{i,j} = 1|Z_i^\gamma, Z_j^\gamma) = \text{Sigmoid}(Z_i^\gamma (Z_j^\gamma)^T) \quad (7)$$

### 3.2. Attribute Autoencoder

In the attribute encoder, two non-linear feature transform layers are employed to map the observed attribute data to the latent attribute embedding $Z^A$, which can be formulated as follows:

$$\tilde{Z}^A = \sigma((X)^T W^{A(1)} + b^{A(1)}) \quad (8)$$

$$Z^A = \tilde{Z}^A W^{A(2)} + b^{A(2)} \quad (9)$$

where $W^{A(1)} \in \mathbb{R}^{D^{in}_{in} \times D^{out}_{in}}$, $b^{A(1)} \in \mathbb{R}^{D^{out}_{in}}$, $W^{A(2)} \in \mathbb{R}^{D^{in}_{out} \times D}$ and $b^{A(2)} \in \mathbb{R}^{D}$ are the weights and biases learned by two layers, $D^{in}_{in}$, $D^{out}_{in}$, and $D$ are the dimensionalities of $(X)^T$, $\tilde{Z}^A$ and $Z^A$, respectively.

Finally, attribute decoder takes both the node embeddings $Z^\gamma$ learned by structure encoder, and the attribute embeddings $Z^A$ as inputs for decoding of the original node attribute:

$$\hat{X} = Z^\gamma (Z^A)^T \quad (10)$$

in which, the interactions between network structure and node attribute are jointly captured. Different from the structure decoder, in the attribute decoder, no activation function is utilized here for the arbitrary-valued attribute.

In AnomalyDAE, the computational complexity of structure autoencoder and attribute autoencoder are $O(MD + ED + M^2)$ and $O(ND + NM)$ respectively, where $M$ is the number of nodes, $E$ is the number of edge, $N$ is the dimension of attribute, $D$ is the dimension of embedding.

### 3.3. Loss function

The training objective of AnomalyDAE is to minimize the reconstruction errors of both network structure and node attribute:

$$\mathcal{L}_{rec} = \alpha \| (A - \hat{A}) \odot \theta \|_F^2 + (1 - \alpha) \| (X - \hat{X}) \odot \eta \|_F^2 \quad (11)$$

where $\alpha$ is the parameter which control the trade off between structure reconstruction and attribute reconstruction. $\odot$ is the Hadamard product, $\eta$ and $\theta$ are defined as follows:

$$\theta_{i,j} = \begin{cases} 1 & \text{if } A_{i,j} = 0, \\ \theta & \text{otherwise} \end{cases}, \eta_{i,j} = \begin{cases} 1 & \text{if } X_{i,j} = 0, \\ \eta & \text{otherwise} \end{cases} \quad (12)$$

where $\eta > 1$ and $\theta > 1$, which are used to impose more penalty to the reconstruction error of the non-zero elements due to some missing edges or attributes in the real world.

### 3.4. Anomaly Detection

Inspired by the motivation that the pattern of abnormal nodes deviate from the majority of other nodes in either structure or attribute, the anomaly score $S_V$, of node $V_i$ is defined as the reconstruction error from both network structure and node attribute perspective:

$$S_{V_i} = \alpha \| (A_i - \hat{A}_i) \odot \theta_i \|_F^2 + (1 - \alpha) \| (X_i - \hat{X}_i) \odot \eta_i \|_F^2 \quad (13)$$

$$\mathcal{L}_{rec} = \alpha \| (A - \hat{A}) \odot \theta \|_F^2 + (1 - \alpha) \| (X - \hat{X}) \odot \eta \|_F^2 \quad (11)$$

where $\alpha$ is the parameter which control the trade off between structure reconstruction and attribute reconstruction. $\odot$ is the Hadamard product, $\eta$ and $\theta$ are defined as follows:

$$\theta_{i,j} = \begin{cases} 1 & \text{if } A_{i,j} = 0, \\ \theta & \text{otherwise} \end{cases}, \eta_{i,j} = \begin{cases} 1 & \text{if } X_{i,j} = 0, \\ \eta & \text{otherwise} \end{cases} \quad (12)$$

where $\eta > 1$ and $\theta > 1$, which are used to impose more penalty to the reconstruction error of the non-zero elements due to some missing edges or attributes in the real world.

$$S_{V_i} = \alpha \| (A_i - \hat{A}_i) \odot \theta_i \|_F^2 + (1 - \alpha) \| (X_i - \hat{X}_i) \odot \eta_i \|_F^2 \quad (13)$$
4. EXPERIMENTS

4.1. Experimental Setup

Three commonly used real-world datasets [14] are used in this paper to evaluate the proposed method, including BlogCatalog, Flickr, and ACM. The statistics of datasets are shown in Table 2. In the experiment, we train AnomalyDAE with 100, 100, 80 iterations for BlogCatalog, Flickr, and ACM respectively. Adam [17] algorithm is utilized for optimization with learning rate as 0.001. The embedding dimension is set as 128 for all datasets. The parameters \((\alpha, \eta, \theta)\) are set as \((0.7, 5, 40), (0.9, 8, 90), (0.7, 3, 10)\) for BlogCatalog, Flickr, and ACM respectively.

4.2. Result Analysis

4.2.1. Performance Evaluation

We compare AnomalyDAE with the state-of-the-art methods including LOF [18], SCAN [19], AMEN [8], Radar [12], Anomalous [13], and Dominant [14]. The AUC scores (the Area Under a receiver operating characteristic Curve) for anomaly detection are reported in Table 3.

The experimental results show that the proposed AnomalyDAE significantly outperforms all baselines in various datasets. The performance of AnomalyDAE is much higher than traditional anomaly detection methods, i.e., AnomalyDAE outperforms LOF by 48.66\%, SCAN by 70.54\%, and AMEN by 44.44\% respectively on AUC for BlogCatalog dataset, because of that LOF and SCAN consider only network structure or node attribute, and AMEN is designed for anomalous neighborhoods rather than node itself. Besides, for Flickr dataset, AnomalyDAE increases the AUC score by 24.36\% compared with Radar and 25.63\% compared with Anomalous, this is because that residual analysis or CUR decomposition [20] based methods are not only sensitive to network sparsity, but with limited learning ability on large graph. Compared with more recent Dominant, AnomalyDAE achieves gains by 19.68\%, 22.32\%, 15.11\% on BlogCatalog, Flickr, and ACM datasets respectively. Although GCN based encoder in Dominant is capable of learning discriminative node embeddings by aggregating neighbor features, the single graph encoder cannot jointly capture the complex interactions between network structure and node attribute, while AnomalyDAE employs two separate encoders for jointly learning of node embedding and attribute embedding respectively, which considers the complex modality interactions of network structure and node attribute.

4.2.2. Parameter Sensitivity

In this section, we investigate the parameter sensitivity of different numbers of the embedding dimension \(D\) and the trade-off parameter \(\alpha\) for anomaly detection. The experiment results are shown in Fig. 2. We can see that a relative high dimension such as 128 or 256 facilitates high performance because higher dimensional embeddings are capable of encoding more information. However, the dimension with too low value or too high value would degrade the performance because of the weak modeling ability or suffering from overfitting. In terms of \(\alpha\), considering only attribute reconstruction (\(\alpha=0\)) or structure reconstruction (\(\alpha=1\)) would result in poor performance, which demonstrates the importance of the interactions between the network structure and node attribute on attributed network for anomaly detection.

5. CONCLUSION

In this paper, we study the problem of anomaly detection on attributed network by considering the complex modality interaction between network structure and node attribute. To cope with this problem, we propose a deep joint representation learning framework for anomaly detection via a dual autoencoder. By introducing two separate autoencoders for jointly learning of node embedding and attribute embedding, AnomalyDAE performs better than current state-of-the-art methods on multiple real-world datasets.

<table>
<thead>
<tr>
<th>Database</th>
<th># (\mathcal{V})</th>
<th># (\mathcal{E})</th>
<th># (\mathcal{A})</th>
<th># Anomalies</th>
</tr>
</thead>
<tbody>
<tr>
<td>BlogCatalog</td>
<td>5,196</td>
<td>171,743</td>
<td>8,189</td>
<td>300</td>
</tr>
<tr>
<td>Flickr</td>
<td>7,575</td>
<td>239,738</td>
<td>12,047</td>
<td>450</td>
</tr>
<tr>
<td>ACM</td>
<td>16,484</td>
<td>71,980</td>
<td>8,337</td>
<td>600</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Method</th>
<th>BlogCatalog</th>
<th>Flickr</th>
<th>ACM</th>
</tr>
</thead>
<tbody>
<tr>
<td>LOF [18]</td>
<td>49.15</td>
<td>48.81</td>
<td>47.38</td>
</tr>
<tr>
<td>SCAN [19]</td>
<td>27.27</td>
<td>26.86</td>
<td>35.99</td>
</tr>
<tr>
<td>AMEN [8]</td>
<td>53.37</td>
<td>60.47</td>
<td>72.62</td>
</tr>
<tr>
<td>Radar [12]</td>
<td>71.04</td>
<td>72.86</td>
<td>69.36</td>
</tr>
<tr>
<td>Anomalous [13]</td>
<td>72.81</td>
<td>71.59</td>
<td>71.85</td>
</tr>
<tr>
<td>Dominant [14]</td>
<td>78.13</td>
<td>74.9</td>
<td>74.94</td>
</tr>
<tr>
<td>AnomalyDAE</td>
<td>97.81</td>
<td>97.22</td>
<td>90.05</td>
</tr>
</tbody>
</table>

Fig. 2. Parameter sensitivity for anomaly detection.
6. REFERENCES


