Inductive Anomaly Detection on Attributed Networks

Kaize Ding[1] , **Jundong Li**[2,3] , **Nitin Agarwal**[4] and **Huan Liu**[1]

[1]Computer Science and Engineering, Arizona State University, USA
[2]Electrical and Computer Engineering, University of Virginia, USA
[3]Computer Science & School of Data Science, University of Virginia, USA
[4]Information Science, University of Arkansas Little Rock, USA
kaize.ding@asu.edu, jundong@virginia.edu, nxagarwal@ualr.edu, huan.liu@asu.edu

Abstract

Anomaly detection on attributed networks has attracted a surge of research attention due to its broad applications in various high-impact domains, such as security, finance, and healthcare. Nonetheless, most of the existing efforts do not naturally generalize to unseen nodes, leading to the fact that people have to retrain the detection model from scratch when dealing with newly observed data. In this study, we propose to tackle the problem of inductive anomaly detection on attributed networks with a novel unsupervised framework: AEGIS (adversarial graph differentiation networks). Specifically, we design a new graph neural layer to learn anomaly-aware node representations and further employ generative adversarial learning to detect anomalies among new data. Extensive experiments on various attributed networks demonstrate the efficacy of the proposed approach.

1 Introduction

In a variety of real-world applications (e.g., social spam detection, financial fraud detection, and network intrusion detection), detecting anomalies from networked data plays a vital role in keeping malicious behaviors or attacks at bay. With the increasing usage of attributed networks for modeling various information systems, anomaly detection on attributed networks has become a fundamental learning task, which aims to accurately characterize and detect anomalies (i.e., abnormal nodes) whose patterns (w.r.t., structure and attributes) deviate significantly from the majority reference nodes.

As it is costly and labor-intensive to obtain the label information of anomalies, anomaly detection on attributed networks is predominately carried out in an unsupervised manner [Li *et al.*, 2017; Ding *et al.*, 2019b]. Due to the fact that real-world attributed networks are rapidly growing, the problem of anomaly detection on attributed networks can be further divided into two settings based on the way how new data is handled: (1) *transductive setting* and (2) *inductive setting*. The former performs anomaly detection on a single, fixed attributed network that includes new nodes and the latter an-

ticipates to handle newly observed nodes or (sub)networks with a previously learned model. Though extensive research has been conducted on the first setting and achieved immense success [Gao *et al.*, 2010; Perozzi and Akoglu, 2016; Li *et al.*, 2017], inductive anomaly detection on attributed networks has heretofore received little attention. Restricted by their upfront access to global network structure (e.g., methods based on matrix factorization and spectral convolution), transductive anomaly detection methods need to retrain the model when new data arrives, which tends to be computationally expensive. Hence, we are motivated to make the initial investigation on the problem of inductive anomaly detection on attributed networks and develop a novel inductive anomaly detection algorithm.

Given its capability of learning representations on newly observed nodes without retraining the whole model from scratch, a series of graph neural networks [Hamilton *et al.*, 2017; Bojchevski and Günnemann, 2017; Velickovic *et al.*, 2017] have drawn great interests from researchers lately. Instead of training a distinct embedding vector for each node, those methods learn a set of aggregator functions to aggregate features from a node's local neighborhood. Inspired by their success, we propose to tackle the studied problem by virtue of inductive representation learning. However, building a principled inductive anomaly detection model for attributed networks remains a daunting task due to the following two challenges: (1) *The existing graph neural networks are ineffective to characterize the node abnormality since they are not tailored for anomaly detection problems*. On the one hand, as malicious users might build spurious connections with normal nodes to camouflage their noxious intentions, directly aggregating features from neighboring nodes may cause the learned representations of anomalies to be inexpressive for detection; On the other hand, due to the fact that network structures of many real-world attributed networks are highly sparse, solely relying on the context information aggregated from the local neighborhood could be less informative and noisy [Cao *et al.*, 2015; Chen *et al.*, 2019]. The above issues necessitate a new design of graph neural network, which allows the model to learn anomaly-aware node representations from arbitrary-order neighbors. (2) *Unseen anomalies that emerged in the newly added data could incur the infeasi-*

bility of previously learned detection models. For an inductive anomaly detection model, its training network is only partially observed. Though normal data tends to be stable, anomalies in the observed and unseen data could be from very different manifolds [Pang *et al.*, 2019]. Thus the previously learned anomaly detection model might lose its discriminability on newly observed nodes [Lawrence *et al.*, 1997; Caruana *et al.*, 2001]. As such, how to improve the generalization ability of inductive models for detecting those unseen anomalies is imperative to solve.

To address the challenges above, in this paper, we propose an unsupervised framework: AEGIS (<u>a</u>dversarial <u>g</u>raph differentiation autoencoders) for inductive anomaly detection on attributed networks. Built upon our graph differentiative layers, AEGIS first learns anomaly-aware node representations through an autoencoder network GDN-AE. Afterwards, AEGIS trains a generative adversarial network (Ano-GAN) to improve the model generalization ability on newly added data. Specifically, the generator aims to generate informative potential anomalies, while the discriminator tries to learn a decision boundary that separates the potential anomalies from the normal data. As such, the proposed framework eliminates the restriction of transductive models and acquires strong capability in detecting anomalies among newly added nodes. In summary, our main contributions are three-fold:

- To the best of our knowledge, we are the first to study the problem of inductive anomaly detection on attributed networks, which addresses the limitation of existing anomaly detection methods.

- We propose a novel graph differentiative layer and further develop a principled framework AEGIS that is applicable to perform anomaly detection in both inductive and transductive settings.

- We evaluate our proposed approach on various benchmark datasets. Extensive experimental results demonstrate its superior performance.

2 Related Work

Anomaly Detection on Attributed Networks. As attributed network has been widely used to model a wide range of complex systems, the studies of anomaly detection on attributed networks have attracted increasing attention in the research community. For instance, AMEN [Perozzi and Akoglu, 2016] considers the ego-network information for each node and discovers anomalous neighborhoods on attributed networks. Radar [Li *et al.*, 2017] characterizes the residuals of attribute information and its coherence with network information for anomaly detection. With the rocketing growth of deep neural networks, researchers also propose to solve the problem of anomaly detection on attributed networks based on deep learning techniques. Dominant [Ding *et al.*, 2019a] achieves superior performance over other shallow methods by building a deep autoencoder architecture on top of the graph convolutional networks. Despite their great progress, all the aforementioned methods can only perform anomaly detection on a single attributed network of fixed size, making the detection model limited when dealing with newly observed nodes. As a

necessary supplement in the research field of anomaly detection on attributed networks, AEGIS is capable of performing anomaly detection on newly observed nodes without retraining and achieves superior empirical results.

Graph Neural Networks. Driven by the momentous success of deep learning, a mass of efforts have been devoted to developing deep neural networks for graph-structured data lately [Kipf and Welling, 2016; Hamilton *et al.*, 2017; Velickovic *et al.*, 2017]. Among them, graph convolutional networks (GCNs) [Kipf and Welling, 2016], which extends the operation of convolution on graph-structured data in the spectral domain for network representation learning, has achieved enormous success in various research fields [Zhou *et al.*, 2019; Liu *et al.*, 2019; Guo *et al.*, 2020; Wang *et al.*, 2020]. More recently, the study of inductive representation learning on graphs has attracted a surge of research interests. Instead of training individual embeddings for each node, GraphSAGE [Hamilton *et al.*, 2017] learns a function that generates embeddings by sampling and aggregating features from a node's local neighborhood. Similarly, graph attention networks (GAT) [Velickovic *et al.*, 2017] is an attention-based model that allows specifying different weights to nodes in the local neighborhood, which greatly enhances the model capacity compared to other graph neural networks. Nevertheless, all these methods focus on network representation learning, but it remains unclear how this power can be shifted to the anomaly detection problem.

3 Problem Formulation

Throughout this paper, we use calligraphic fonts, bold lowercase letters, and bold uppercase letters to denote sets (e.g., $\mathcal{V}$), vectors (e.g., $\mathbf{x}$), and matrices (e.g., $\mathbf{X}$), respectively. Generally, an attributed network can be represented by $\mathcal{G} = (\mathbf{A}, \mathbf{X})$, where $\mathbf{A}$ denotes the adjacency matrix and $\mathbf{X}$ denotes the attribute matrix. The task of anomaly detection on attributed networks can be classified into the transductive setting and inductive setting. To make the results more interpretable, we formulate them as two ranking problems:

Problem 1 *Inductive Anomaly Detection on Attributed Networks: Given a partially observed attributed network $\mathcal{G} = (\mathbf{A}, \mathbf{X})$ for training and a newly observed (sub)network $\mathcal{G}' = (\mathbf{A}', \mathbf{X}')$ for testing, the task is to rank all the nodes in $\mathcal{G}'$ according to the degree of abnormality, such that abnormal nodes should be ranked on higher positions.*

It is worth mentioning that, although we aim at developing an inductive anomaly detection method, the proposed method is able to handle the transductive setting as well.

4 Proposed Approach

In this section, we will first present the building block layer used to construct the proposed framework AEGIS. We then describe the architecture of AEGIS and its learning process for inductive anomaly detection on attributed networks.

4.1 Graph Differentiative Layer

We will start by describing a single *graph differentiative layer* (Figure 1(a)) used to construct any graph differentiation networks (GDNs) for inductive anomaly detection. Apart from
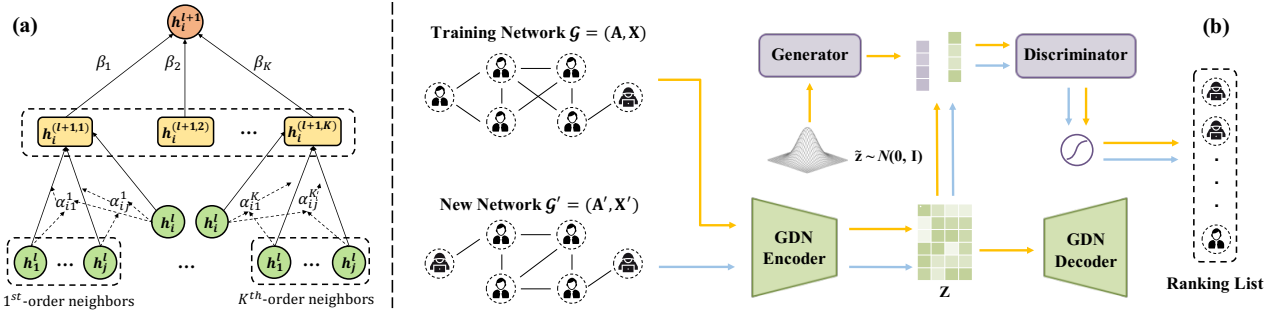
Figure 1: (a) The graph differentiative layer. (b) The proposed inductive anomaly detection framework AEGIS. Note that AEGIS is trained with the partially observed network $\mathcal{G}$, and can directly detect anomalies on the new network $\mathcal{G}'$ in a feed-forward way. The yellow arrows denote the training flow and the blue arrows denote the inference flow. Figure best viewed in color.

the existing GNNs, GDN is capable of learning anomaly-aware node representations from arbitrary-order neighborhoods. Specifically, a GDN layer has an attention-based hierarchical structure described as follows:

Node-level Attention. According to the principle of Homophily, instances with similar patterns are more likely to be linked together in attributed networks [Li *et al.*, 2017], and the measuring of homophily has become an effective way to detect anomalies [Perozzi and Akoglu, 2016]. Thus for each node, we adopt attention mechanism to capture the feature difference between it and its neighbors. In this way, it enables the learned representation to differentiate a node if its features deviate significantly from its neighbors'. Specifically, for any graph differentiative layer l, it learns the representation of node i by:

$$\mathbf{h}_i^{(l)} = \sigma\left(\mathbf{W}_1\mathbf{h}_i^{(l-1)} + \sum_{j \in \mathcal{N}_i} \alpha_{ij}\mathbf{W}_2\Delta_{i,j}^{(l-1)}\right), \quad (1)$$

where $\mathbf{h}^{(l-1)} \in \mathbb{R}^F, \mathbf{h}^{(l)} \in \mathbb{R}^{\tilde{F}}$ denotes the input and output representation of node i, respectively. $\Delta_{i,j}^{(l-1)} = \mathbf{h}_i^{(l-1)} - \mathbf{h}_j^{(l-1)}$ is the feature difference between node i and j. $\mathbf{W}_1, \mathbf{W}_2 \in \mathbb{R}^{F \times \tilde{F}}$ are two trainable weight matrices and σ is a nonlinear activation function. $\mathcal{N}_i$ denotes the neighboring nodes of node i. Here α_{ij} is the attention coefficient between node i and node j, which can be expressed as:

$$\alpha_{ij} = \frac{\exp\left(\sigma(\mathbf{a}^{\mathrm{T}}\mathbf{W}_2\Delta_{i,j}^{(l-1)})\right)}{\sum_{k \in \mathcal{N}_i} \exp\left(\sigma(\mathbf{a}^{\mathrm{T}}\mathbf{W}_2\Delta_{i,k}^{(l-1)})\right)}, \quad (2)$$

where $\mathbf{a} \in \mathbb{R}^{\tilde{F}}$ is the attention vector that assigns importance to different neighbors of node i. Apart from the graph attention networks (GAT) [Velickovic *et al.*, 2017], we generate the attentional weights based on the feature differences rather than the concatenation of two neighboring features, in order to explicitly measure network homophily and characterize the abnormality of each node.

Similarly, by extracting k^{th}-order neighbors of node i from $\mathbf{A}^k = \underbrace{\mathbf{A} \cdot \mathbf{A} \dots \mathbf{A}}_{k}$, we can compute its kth-order node representation $\mathbf{h}_i^{(l,k)}$. As different neighborhoods encode differ-

ent context information, those neighborhood-specific representations could be used for addressing the sparsity issue and learning more powerful anomaly detector.

Neighborhood-level Attention. Then we propose to aggregate K neighborhood-specific representations to a unified representation. As neighbors from different distances contribute differently to characterize a node, we propose to apply location-based attention [Luong *et al.*, 2015] on those neighborhood-specific representations, in order to capture the significance of different neighborhoods. Formally, at each layer l, the final embedding of node i can be integrated by:

$$\mathbf{h}_i^{(l)} = \sum_{k=1}^K \beta_i^k \mathbf{h}_i^{(l,k)}, \quad (3)$$

where β_i^k denotes the attention coefficient on k^{th}-order representation $\mathbf{h}_i^{(l,k)}$, which can be formulated as:

$$\beta_i^k = \frac{\exp\left(\sigma(\hat{\mathbf{a}}^{\mathrm{T}}\mathbf{h}_i^{(l,k)})\right)}{\sum_{k'=1}^K \exp\left(\sigma(\hat{\mathbf{a}}^{\mathrm{T}}\mathbf{h}_i^{(l,k')})\right)}. \quad (4)$$

Note that $\hat{\mathbf{a}} \in \mathbb{R}^{\tilde{F}}$ is the attention vector that allows our model to specify different significance to different intermediate representations for learning the unified representation of each node. In this way, our GDN layer is able to aggregate expressive context information for characterizing node abnormality from neighbors with various numbers of hops away.

4.2 Adversarial Graph Differentiation Networks

Following previous research [Ding *et al.*, 2019a; Chalapathy *et al.*, 2017], we can directly build an unsupervised model (e.g., autoencoders) using GDN layers for detecting anomalies. However, it may not work as expected in our studied problem: under the inductive setting, a previously learned anomaly detection model may encounter unseen anomalies among the newly added data, yielding poor performance in practice. To counter this issue, we propose a joint framework based on generative adversarial learning [Goodfellow *et al.*, 2014] to improve model robustness by generating informative potential anomalies.

As depicted in Figure 1(b), the proposed framework AEGIS consists of two learning phases. The first phase aims to
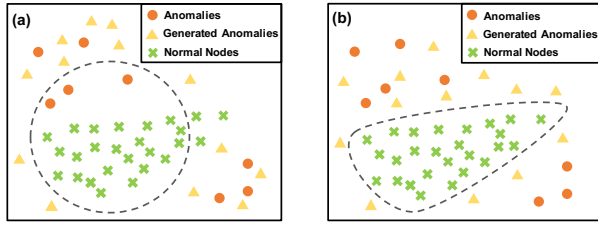
Figure 2: Illustration of the learning mechanism behind Ano-GAN. (a) At its early training stage, Ano-GAN cannot generate informative anomalies; (b) After training, Ano-GAN is able to generate anomalies that generally lie close to normal data.

learn node representations from the input attributed network through an autoecoder network (w.r.t., GDN-AE), which is built with the graph differentiative layers. Specifically, the encoder Enc compresses the input attributed network to low-dimensional node representations $\mathbf{Z}$, and the decoder Dec reconstructs the input data afterwards. The encoder of GDN-AE learns anomaly-aware node representations, and is expected to map the normal and abnormal nodes to different regions in the latent feature space.

With the learned anomaly-aware node representations, the second phase aims to train a generative adversarial network (w.r.t., Ano-GAN) that can accurately model the distribution of normal data. Specifically, the generator G takes noises sampled from a prior distribution $p(\tilde{\mathbf{z}})$ as input, and attempts to generate informative potential anomalies. Meanwhile, the discriminator D tries to distinguish whether an input is the representation of a normal node or a generated anomaly. This mini-max game can be formally defined as follows:

$$\min_{G} \max_{D} \mathbb{E}_{\mathbf{z}\sim\mathbf{Z}}[\log D(\mathbf{z})] + \mathbb{E}_{\tilde{\mathbf{z}}\sim p(\tilde{\mathbf{z}})}[\log(1 - D(G(\tilde{\mathbf{z}})))], \quad (5)$$

where $p(\tilde{\mathbf{z}})$ is the prior distribution. The previous research [Makhzani et al., 2015] and our preliminary experiments show that Gaussian prior is a robust option for different datasets.

During the learning process, the generator G gradually learns the generating mechanism and synthesizes an increasing number of potential anomalies that may arise in the unseen data. As shown in Figure 2, the discriminator D can accurately learn the real data distribution and describe the decision boundary that encloses the concentrated normal nodes. In other words, the generator G effectively improves the capability of the discriminator D to identify normal data by generating informative potential anomalies. To avoid the generated anomalies being mixed with normal data, we follow the idea in [Dai et al., 2017] to train the Ano-GAN.

4.3 Model Learning

In order to learn the proposed model, different components of AEGIS are jointly trained in two phases and each phase requires dedicated training objective functions. Specifically, The reconstruction loss of GDN-AE can be formulated as:

$$\mathcal{L}_{\text{AE}} = \frac{1}{n} \sum_{i=1}^{n} ||Dec(Enc(\mathbf{x}_i)) - \mathbf{x}_i||^2. \quad (6)$$

Algorithm 1: The training process of AEGIS

Input: Attributed network $\mathcal{G} = (\mathbf{A}, \mathbf{X})$, Training epochs $Epoch_{AE}$, and $Epoch_{GAN}$
Output: Well-trained GDN-AE and Ano-GAN

1   $i \leftarrow 0$;
2   **while** $i < Epoch_{AE}$ **do**
3   Compute the reconstructed node attributes;
4   Update GDN-AE with the loss function Eq. (6);
5   $i \leftarrow 0$;
6   **while** $i < Epoch_{GAN}$ **do**
7   Sample P instances from the node representations $\mathbf{Z}$;
8   Generate P instances from the prior distribution $p(\tilde{\mathbf{z}})$;
9   Update the generator G with loss function Eq. (7);
10   Update the discriminator D with loss function Eq. (8);
11   **return**

	# nodes	# edges	# attributes	# anomalies
BlogCatalog	5,196	171,743	8,189	300
Flickr	7,575	239,738	12,047	450
ACM	16,484	71,980	8,337	600

Table 1: Summary of the attributed network datasets

The loss function of Ano-GAN can be represented by the conventional cross-entropy loss for training a binary classifier. In practice, the generator and the discriminator of Ano-GAN are trained separately. For the generator G, the loss function can be defined as:

$$\mathcal{L}_{\text{G}} = \mathbb{E}_{\tilde{\mathbf{z}}\sim p(\tilde{\mathbf{z}})}[\log(1 - D(G(\tilde{\mathbf{z}})))], \quad (7)$$

and the loss function of learning discriminator D is:

$$\mathcal{L}_{\text{D}} = -\mathbb{E}_{\mathbf{z}\sim\mathbf{Z}}[\log D(\mathbf{z})] - \mathbb{E}_{\tilde{\mathbf{z}}\sim p(\tilde{\mathbf{z}})}[\log(1 - D(G(\tilde{\mathbf{z}})))]. \quad (8)$$

The detailed model training process is illustrated in Algorithm 1. After the model converges on the training network, the discriminator D learns the distribution of normal nodes, and can be directly used to detect anomalies on any newly observed nodes or (sub)networks.

4.4 Inductive Anomaly Detection

As our objective is to solve the problem of inductive anomaly detection on attributed networks, now we will elaborate on how to utilize the previously learned model to perform anomaly detection on newly observed (sub)networks. Note that after the training phase, AEGIS is capable of handling newly added data without retraining the model. To compute the anomaly scores of unseen nodes, we can retain the parameters of previously learned model and directly feed the new (sub)network $\mathcal{G}' = (\mathbf{A}', \mathbf{X}')$ into it. AEGIS will learn the layer-wise representation of each unseen node in a feed-forward way. Finally, we can compute the anomaly score of node i according to the output of the discriminator D:

$$score(\mathbf{x}_i') = p(y_i' = 0|\mathbf{z}_i') = 1 - D(\mathbf{z}_i'). \quad (9)$$

5 Experiments

In this section, we perform empirical evaluations on various real-world datasets to verify the effectiveness of the proposed model AEGIS in both inductive and transductive settings.
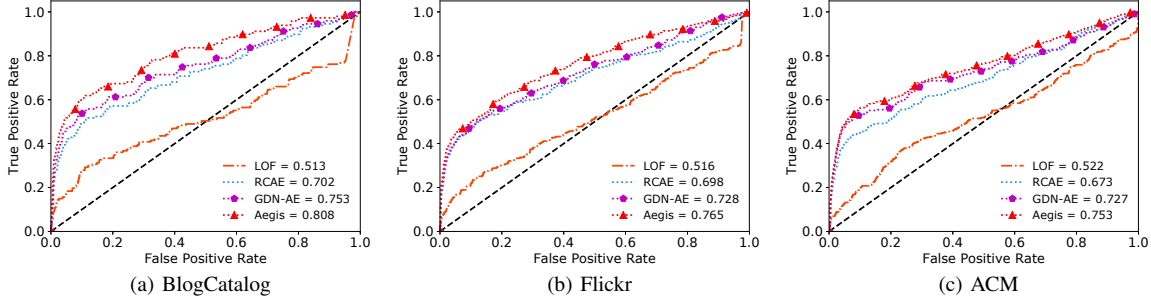
Figure 3: Inductive anomaly detection results on three datasets w.r.t. ROC curve and AUC value.

Methods	BlogCatalog			Flickr			ACM		
	Pre@50	Pre@100	Pre@200	Pre@50	Pre@100	Pre@200	Pre@50	Pre@100	Pre@200
LOF	0.324	0.212	0.145	0.366	0.255	0.190	0.156	0.128	0.087
RCAE	0.558	0.450	0.307	0.580	0.566	0.423	0.486	0.435	0.360
GDN-AE (ours)	0.622	0.505	0.345	0.640	0.594	0.452	0.542	0.467	0.405
AEGIS (ours)	**0.704**	**0.568**	**0.382**	**0.722**	**0.661**	**0.485**	**0.626**	**0.533**	**0.432**

Table 2: Inductive anomaly detection results on three datasets w.r.t. precision@K.

5.1 Experiment Setup

Compared Methods. In the experiments, we compare the proposed model AEGIS with different baseline methods. Specifically, LOF [Breunig *et al.*, 2000] detects anomalies at the contextual level by considering attributes. ConOut [Sánchez *et al.*, 2014] detects anomalies by determining its subgraph and its relevant subset of attributes. RCAE [Chalapathy *et al.*, 2017] and GCN-AE [Ding *et al.*, 2019a] are two autoencoder-based methods for detecting anomalies on *i.i.d.* data and attributed networks, respectively. Additionally, we include the component GDN-AE of the proposed framework as another baseline. To summarize, LOF, RCAE and GDN-AE are inductive models that support both transductive and inductive settings, while ConOut and GCN-AE are two state-of-the-art transductive methods.

Evaluation Datasets. In the experiments, we employ three public real-world attributed network datasets (BlogCatalog [Wang *et al.*, 2010], Flickr [Tang and Liu, 2009], and ACM [Tang *et al.*, 2008]) for performance comparison. Due to the shortage of ground truth anomalies, we follow the perturbation scheme introduced in [Ding *et al.*, 2019a] to inject a combined set of anomalies (i.e., *structural anomalies* and *contextual anomalies*) for each dataset. The statistics of our evaluation datasets are shown in Table 1. For performance evaluation, two standard evaluation metrics (ROC-AUC and Precision@K) are used to measure the performance of different anomaly detection algorithms.

Implementation Details. In AEGIS, the GDN-AE is built with one 64-dimension hidden layer with ELU [Clevert *et al.*, 2015] nonlinearity. Its output layer has a linear activation function. For Ano-GAN, the generator has one hidden layer (32-neuron) and the dimension of its output layer is 64. The discriminator has one hidden layer (32-neuron) with ReLU [Nair and Hinton, 2010] activation function, and we

employ sigmoid activation function in its last layer. AEGIS is optimized with the Adam [Kingma and Ba, 2014] optimizer. We set the learning rate of the reconstruction loss to 0.005. The training epoch of GDN-AE is 200, while the training epoch of Ano-GAN is 50. We set the parameter K to 3 (BlogCaltalog), 2 (Flickr), 3 (ACM). In addition, the number of samples P is set to $0.05 \times n$ for each dataset.

5.2 Experimental Results

Inductive Setting. In order to verify the effectiveness of the proposed framework, we first conduct the empirical evaluation under the inductive setting. Three inductive models are included. Specifically, for each dataset, we first randomly sample 50% nodes from the whole network and extract the link relations among these nodes to construct a partially observed attributed network $\mathcal{G}$. Similarly, we sample another 40% data to construct the newly observed attributed (sub)network $\mathcal{G}'$ for testing and the remaining 10% data is for the validation purpose. After AEGIS is trained on the partially observed attributed network $\mathcal{G}$, we directly apply the learned model to $\mathcal{G}'$. We repeat this process 10 times and report the average results in Figure 3 and Table 2. To summarize, we make the following observations:

- Under the inductive setting, our model AEGIS achieves superior anomaly detection performance over other baseline methods, which demonstrates its capability for detecting anomalies on newly added data without retraining from scratch.

- The performance of LOF and RCAE largely fall behind in our experiments since they merely consider the nodal attributes for measuring node abnormality. Meanwhile, GDN-AE cannot achieve competitive results with AEGIS, which verifies that AEGIS is able to improve its generalization ability by generating potential anomalies.
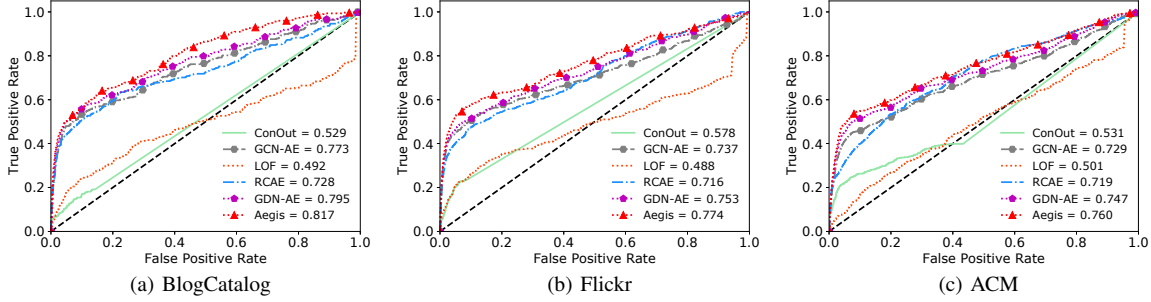
Figure 4: Transductive anomaly detection results on three datasets w.r.t. ROC curve and AUC value.

| Methods | BlogCatalog | | | Flickr | | | ACM | | |
	Pre@50	Pre@100	Pre@200	Pre@50	Pre@100	Pre@200	Pre@50	Pre@100	Pre@200
ConOut	0.380	0.200	0.130	0.440	0.280	0.255	0.540	0.470	0.310
GCN-AE	0.758	0.712	0.593	0.756	0.727	0.685	0.620	0.589	0.534
LOF	0.300	0.220	0.180	0.420	0.380	0.270	0.180	0.130	0.115
RCAE	0.624	0.610	0.526	0.666	0.685	0.653	0.460	0.460	0.450
GDN-AE (ours)	0.772	0.723	0.622	0.776	0.742	0.699	0.632	0.601	0.542
AEGIS (ours)	**0.778**	**0.730**	**0.638**	**0.784**	**0.757**	**0.705**	**0.640**	**0.606**	**0.545**

Table 3: Transductive anomaly detection results on three datasets w.r.t. precision@K.

Transductive Setting. Next, we evaluate the effectiveness of AEGIS under the transductive setting. Specifically, each dataset is used as a single fixed network, and each method directly performs anomaly detection on it. The results are presented in Figure 4 and Table 3 (averaged over 10 runs). Based on the results, we have the following observations:

- The proposed model AEGIS outperforms all the baseline methods on all the three attributed networks. It implies that even though our approach AEGIS is mainly developed for inductive anomaly detection on attributed networks, it can also achieve competitive performance in the transductive setting.

- Our basic model GDN-AE obtains better performance than the state-of-the-art baseline GCN-AE, which demonstrates the effectiveness of the proposed graph differentiative layer. It verifies the advantage of GDN-AE for learning anomaly-aware node representations from arbitrary-order neighbors.
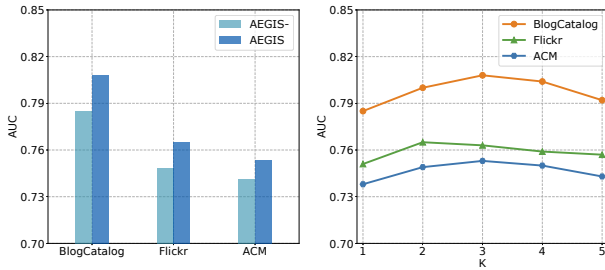
Figure 5: (left) Effect of node-level attention. (right) Effect of parameter K in neighborhood-level attention.

5.3 Further Analysis

In this subsection, we make further analysis on the proposed graph differentiative layer. Note that all the results are reported under the inductive setting since we have similar observations in the transductive setting.

Effect of Node-level Attention. We first study the effect of node-level attention by replacing it with the vanilla graph attention mechanism used in [Velickovic *et al.*, 2017]. As shown in Figure 5 (left), AEGIS outperforms this variant by a noticeable margin on three datasets. It verifies that the node-level attention enables the model to learn anomaly-aware node representations by highlighting the feature difference between a node and its neighbors'.

Effect of Neighborhood-level Attention. We further analyze the significance of neighborhood-level attention, which is controlled by the parameter K. We report the AUC scores over different choices of K in Figure 5 (right). For the datasets considered here, the best results are obtained when K is set to 2 or 3. This confirms that using high-order neighborhoods is able to provide richer context information for learning anomaly-aware node representations. However, overfitting could become an issue if K is too large.

6 Conclusion

In this paper, we make the initial investigation on the research problem of inductive anomaly detection by developing a principled framework AEGIS. The proposed framework not only eliminates the retraining restriction of transductive models, but also acquires strong capability in detecting anomalies among newly added nodes. By conducting extensive experiments, the results demonstrate the superiority of the proposed model over the state-of-the-art methods.

References

[Bojchevski and Günnemann, 2017] Aleksandar Bojchevski and Stephan Günnemann. Deep gaussian embedding of graphs: Unsupervised inductive learning via ranking. *arXiv preprint arXiv:1707.03815*, 2017.

[Breunig *et al.*, 2000] Markus M Breunig, Hans-Peter Kriegel, Raymond T Ng, and Jörg Sander. Lof: identifying density-based local outliers. In *SIGMOD Record*, 2000.

[Cao *et al.*, 2015] Shaosheng Cao, Wei Lu, and Qiongkai Xu. Grarep: Learning graph representations with global structural information. In *CIKM*, 2015.

[Caruana *et al.*, 2001] Rich Caruana, Steve Lawrence, and C Lee Giles. Overfitting in neural nets: Backpropagation, conjugate gradient, and early stopping. In *NIPS*, 2001.

[Chalapathy *et al.*, 2017] Raghavendra Chalapathy, Aditya Krishna Menon, and Sanjay Chawla. Robust, deep and inductive anomaly detection. In *ECML-PKDD*, 2017.

[Chen *et al.*, 2019] Fengwen Chen, Shirui Pan, Jing Jiang, Huan Huo, and Guodong Long. Dagcn: Dual attention graph convolutional networks. *arXiv preprint arXiv:1904.02278*, 2019.

[Clevert *et al.*, 2015] Djork-Arné Clevert, Thomas Unterthiner, and Sepp Hochreiter. Fast and accurate deep network learning by exponential linear units (elus). *arXiv preprint arXiv:1511.07289*, 2015.

[Dai *et al.*, 2017] Zihang Dai, Zhilin Yang, Fan Yang, William W Cohen, and Ruslan R Salakhutdinov. Good semi-supervised learning that requires a bad gan. In *NIPS*, 2017.

[Ding *et al.*, 2019a] Kaize Ding, Jundong Li, Rohit Bhanushali, and Huan Liu. Deep anomaly detection on attributed networks. In *SDM*, 2019.

[Ding *et al.*, 2019b] Kaize Ding, Jundong Li, and Huan Liu. Interactive anomaly detection on attributed networks. In *WSDM*, 2019.

[Gao *et al.*, 2010] Jing Gao, Feng Liang, Wei Fan, Chi Wang, Yizhou Sun, and Jiawei Han. On community outliers and their efficient detection in information networks. In *KDD*, 2010.

[Goodfellow *et al.*, 2014] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, and Yoshua Bengio. Generative adversarial nets. In *NIPS*, 2014.

[Guo *et al.*, 2020] Ruocheng Guo, Jundong Li, and Huan Liu. Learning individual causal effects from networked observational data. In *WSDM*, 2020.

[Hamilton *et al.*, 2017] Will Hamilton, Zhitao Ying, and Jure Leskovec. Inductive representation learning on large graphs. In *NIPS*, 2017.

[Kingma and Ba, 2014] Diederik P Kingma and Jimmy Lei Ba. Adam: Amethod for stochastic optimization. In *ICLR*, 2014.

[Kipf and Welling, 2016] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *ICLR*, 2016.

[Lawrence *et al.*, 1997] Steve Lawrence, C Lee Giles, and Ah Chung Tsoi. Lessons in neural network training: Overfitting may be harder than expected. In *AAAI/IAAI*, 1997.

[Li *et al.*, 2017] Jundong Li, Harsh Dani, Xia Hu, and Huan Liu. Radar: Residual analysis for anomaly detection in attributed networks. In *IJCAI*, 2017.

[Liu *et al.*, 2019] Xu Liu, Jingrui He, Sam Duddy, and Liz O'Sullivan. Convolution-consistent collective matrix completion. In *CIKM*, 2019.

[Luong *et al.*, 2015] Minh-Thang Luong, Hieu Pham, and Christopher D Manning. Effective approaches to attention-based neural machine translation. *arXiv preprint arXiv:1508.04025*, 2015.

[Makhzani *et al.*, 2015] Alireza Makhzani, Jonathon Shlens, Navdeep Jaitly, Ian Goodfellow, and Brendan Frey. Adversarial autoencoders. *arXiv preprint arXiv:1511.05644*, 2015.

[Nair and Hinton, 2010] Vinod Nair and Geoffrey E Hinton. Rectified linear units improve restricted boltzmann machines. In *ICML*, 2010.

[Pang *et al.*, 2019] Guansong Pang, Chunhua Shen, and Anton van den Hengel. Deep anomaly detection with deviation networks. In *KDD*, 2019.

[Perozzi and Akoglu, 2016] Bryan Perozzi and Leman Akoglu. Scalable anomaly ranking of attributed neighborhoods. In *SDM*, 2016.

[Sánchez *et al.*, 2014] Patricia Iglesias Sánchez, Emmanuel Müller, Oretta Irmler, and Klemens Böhm. Local context selection for outlier ranking in graphs with multiple numeric node attributes. In *SSDBM*, 2014.

[Tang and Liu, 2009] Lei Tang and Huan Liu. Relational learning via latent social dimensions. In *KDD*, 2009.

[Tang *et al.*, 2008] Jie Tang, Jing Zhang, Limin Yao, Juanzi Li, Li Zhang, and Zhong Su. Arnetminer: extraction and mining of academic social networks. In *KDD*, 2008.

[Velickovic *et al.*, 2017] Petar Velickovic, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. Graph attention networks. *arXiv preprint arXiv:1710.10903*, 2017.

[Wang *et al.*, 2010] Xufei Wang, Lei Tang, Huiji Gao, and Huan Liu. Discovering overlapping groups in social media. In *ICDM*, 2010.

[Wang *et al.*, 2020] Jianling Wang, Kaize Ding, Ziwei Zhu, Yin Zhang, and James Caverlee. Key opinion leaders in recommendation systems: Opinion elicitation and diffusion. In *WSDM*, 2020.

[Zhou *et al.*, 2019] Qinghai Zhou, Liangyue Li, and Hanghang Tong. Towards real time team optimization. In *Big Data*, 2019.