ACTIVE GRAPH ANOMALY DETECTION USING MIXUP

Ishika Agarwal, Qinghai Zhou, Hanghang Tong
Department of Computer Science
University of Illinois, Urbana-Champaign
{ishikaa2, qinghai2, htong}@illinois.edu

ABSTRACT

Graph anomaly detection (GAD) aims to learn a function that can detect anomalous entities in a graph. In this tiny paper, we explore node-level anomaly detection. Although graph data is prevalent, ground truth labels are hard to acquire. Hence, active learning can be used to obtain soft labels for efficient supervised learning. Furthermore, we can make better use of the scarce labeled data by applying a data augmentation strategy such as mixup. In this paper, we propose AMUGraph, a method for Active anomaly detection that uses MixUp for data augmentation using soft label on Graphs. Our code can be found on GitHub.[1].

1 INTRODUCTION

Graph data are ubiquitous and can represent complex systems of relationships in diverse domains. Unfortunately, these networks can contain malicious components whose behavior deviates from that of the general population, i.e., anomalies. Moreover, learning requires ground truth labels which are difficult and expensive to obtain. To alleviate this issue, we can use active learning, which assumes no access to labeled data, but access to an oracle (a human expert) that can provide ground truth data. In order to avoid abuse on the oracle, we impose a small query budget.

Because of the data complexity, it can be infeasible to expect human labels to be fully confident in their predictions. Therefore, we assume the oracle will provide soft labels - intuitively, a soft label of $(0.3, 0.7)$ means the oracle is 70% confident the node is anomalous. We employ pool-based active learning where in each of the R rounds, we query r points. To further the informativeness of the labeled data, we use mixup as a data augmentation strategy to simulate more labeled data.

Using mixup with soft labels is a common technique in a variety of domains such as image anomaly detection Zhu et al. (2023), to intent classification in dialogue systems Cheng et al. (2022). Soft labels not only express uncertainty, but are useful to teach the model to be uncertain for wrong predictions and reduce the chance of overconfident predictions. Mixup also helps to refine the decision boundaries, which further refines the predictions.

Problem Formulation. As input, we are given (1) a graph $G = (V, E, X)$ with nodes V with attributes X and edges E, (2) an oracle $\mathcal{O}$ that provides soft labels. Our goal is to learn a function $f(x; \theta)$ that outputs a label $\{$benign, anomalous$\}$ for a given node x.

To determine on which nodes to query, we use the normalized reconstruction loss, L2 norm, between the original and reconstructed node embeddings. We assume that nodes with a reconstruction loss near 0.5 are nodes with high uncertainty. Therefore, the model randomly chooses r points with reconstruction losses in the range of $[0.4, 0.8]$ for training. This range was treated as a hyperparameter (see details in Appendix C).

2 METHODOLOGY

We explain the key components of the proposed framework below. For reference, Appendix A contains a diagram of the model training architecture for AMUGraph.

[1] https://github.com/agarwalishika/AMUGraph

	Pubmed	Yelp
DOMINANT	60.6	57.4
GDN	61.7	67.8
AMUGraph-SVAE	50.7	51.9
AMUGraph-NoAugment	52.3	51.9
AMUGraph-Random	54.7	65.7
AMUGraph	**74.1**	**72.6**

Table 1: ROC-AUC (%) on the selected datasets and baselines. Higher is better.

SemiVAE. We adopt the concept of the SemiVAE Huang et al. (2022). The training set is split into benign points and anomalies. Benign points are trained using the original VAE loss function. Anomalies are trained to maximize reconstruction loss. This principle allows us to create a scoring function based on the reconstruction loss. If the reconstruction loss is low, the point is classified as benign; if the reconstruction loss is high, the point is classified as anomalous.

Mixup. Due to the nature of soft labels, each node has a *benign component* and an *anomalous component*. Consider two node embeddings x_p and x_q with soft labels (p_b, p_a) and (q_b, q_a), respectively[2]. To mix two node embeddings, we calculate a weighted sum of the anomalous component of one node and the benign component of another node. Using mathematical notation, the new embedding is $x_{new} = p_b \cdot x_p + q_a \cdot x_q$ and the soft label is $y_{new} = (p_b, q_a)$[3]. Overall, we employ this strategy to augmented every pair of queried nodes to obtain a larger training set.

Classification. For final classification, we use the latent representation (treat this as a hyperparameter). First, we use the soft labels to split the points into anomalous and benign nodes. A node x_p is anomalous if $p_a > p_b$ and benign otherwise. Next, we treat the values in each dimension as a distribution (see Appendix B for illustration). For a new testing node w, the latent representation is extracted. Then, for each dimension, we calculate the probability density function (PDF) of the value in that dimension according to the both distributions (anomalous and benign). Finally, e average the probabilities across the dimensions and retrieve a new soft label (w_b, w_a). If $w_a > w_b$, the node w is classified as anomalous and benign otherwise.

3 EXPERIMENTS

We conduct experiments on two datasets: Pubmed Sen et al. (2008) and Yelp Rayana & Akoglu (2015). We run our training algorithm for 5 rounds with 0.69% labels for each round (total of 3.5% of the dataset). Because we do not have access to a human labeler, we train a GCN on the training set of each dataset. We train the GCN for 40 epochs with a learning rate of 0.001. Our code is made available here. To gauge performance, we use the ROC-AUC score.

We use a few baselines. Firstly, we use the Graph Deviation Network (GDN) from Ding et al. (2021), which is another active learning method. Secondly, we compare our method with Ding et al. (2019)'s DOMINANT method that works on the latent representations from an autoencoder. Finally, we report the performance of our method without the mixup augmentation strategy (called AMUGraph-NoAugment), without the active learning strategy, where we randomly query points (called AMUGraph-Random), and without either (called AMUGraph-SVAE). Table 3 shows the results for our experiments. As shown, our method is competitive against both baselines and shows improvement in ROC-AUC.

Discussion. The objective of the SemiVAE allows the model to learn to reconstruct benign points, guiding this towards a one-class classification problem, which is an easier task in representation learning. Also, using active learning querying strategies helps to identify points that provide the most information, which improves the learning process. Furthermore, mixup works well with active learning, because mixup will create points that are close to the decision boundaries, ultimately enriching the information provided by the querying.

[2]The benign component is denoted with the subscript b and the anomalous component with subscript a.

[3]Empirically, there is no difference between $p_b \cdot x_p + q_a \cdot x_q$ and $p_a \cdot x_p + q_b \cdot x_q$, which is expected.

4 CONCLUSION

In this tiny paper, we introduce AMUGraph, an algorithm for training a GAD model using active learning with mixup data augmentation, which allows us to use soft labels instead of hard labels. We provide some experimentation to show that mixup can be a useful tool paired with active learning for graph anomaly detection.

URM STATEMENT

The authors acknowledge that at least one key author of this work meets the URM criteria of ICLR 2024 Tiny Papers Track.

REFERENCES

Zifeng Cheng, Zhiwei Jiang, Yafeng Yin, Cong Wang, and Qing Gu. Learning to classify open intent via soft labeling and manifold mixup. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 30:635–645, 2022.

Kaize Ding, Jundong Li, Rohit Bhanushali, and Huan Liu. Deep anomaly detection on attributed networks. In *SIAM International Conference on Data Mining (SDM)*, 2019.

Kaize Ding, Qinghai Zhou, Hanghang Tong, and Huan Liu. Few-shot network anomaly detection via cross-network meta-learning. In *Proceedings of the Web Conference 2021*, pp. 2448–2456, 2021.

Fengbin Zhang Haoyi Fan and Zuoyong Li. Anomalydae: Dual autoencoder for anomaly detection on attributed networks. In *45th International Conference on Acoustics, Speech, and Signal Processing*. IEEE, 2020.

Tao Huang, Pengfei Chen, and Ruipeng Li. A semi-supervised vae based active anomaly detection framework in multivariate time series for online systems. In *Proceedings of the ACM Web Conference 2022*, pp. 1797–1806, 2022.

Shebuti Rayana and Leman Akoglu. Collective opinion spam detection: Bridging review networks and metadata. In *Proceedings of the 21th acm sigkdd international conference on knowledge discovery and data mining*, pp. 985–994, 2015.

Prithviraj Sen, Galileo Namata, Mustafa Bilgic, Lise Getoor, Brian Galligher, and Tina Eliassi-Rad. Collective classification in network data. *AI magazine*, 29(3):93–93, 2008.

Fei Zhu, Zhen Cheng, Xu-Yao Zhang, and Cheng-Lin Liu. OpenMix: Exploring Outlier Samples for Misclassification Detection. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 12074–12083, Vancouver, BC, Canada, June 2023. IEEE. ISBN 9798350301298. doi: 10.1109/CVPR52729.2023.01162. URL https://ieeexplore.ieee.org/document/10205216/.

A AMUGRAPH METHODOLOGY DIAGRAM

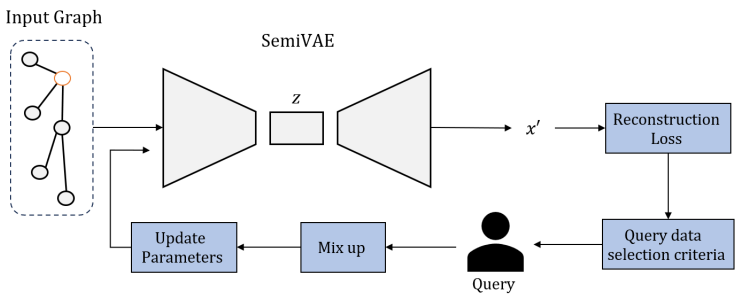

Figure 1: The model training architecture for AMUGraph. The orange node is anomalous. Here, z represents the latent space, and x' represents the reconstructed node x.

The mathematical formulation for the SemiVAE is as follows:

$$L(x) = \begin{cases} \mathbb{E}_{q_\phi}[\log(p_\theta(x|z))] - KL(q_\phi(z|x)||p_\theta(z)) & \text{if x is benign} \\ -\mathbb{E}_{q_\phi}[\log(p_\theta(x|z))] & \text{if x is anomalous} \end{cases} \tag{1}$$

B CLASSIFICATION

Here, we provide a helper figure to illustrate the classification method of AMUGraph

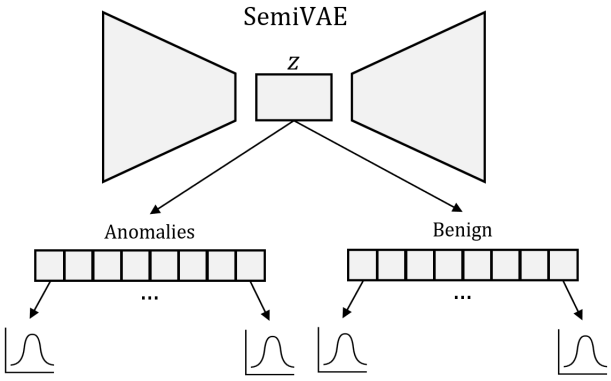

Figure 2: Creating distributions for the latent representations for classification.

C HYPERPARAMETER TUNING

Here, we provide a few details on the hyperparameter tuning.

As mentioned before, we simulate a human labeler by training a GCN on the data. We use the exact same training and testing split across the GCN and SemiVAE to ensure there is no label leakage.

We tried a variety of hyperparameters for each model in our architecture.

For the simulated labelerd GCN, we performed a grid search across $\{20, 40, 60, 80, 100\}$ for epochs and $\{0.001, 0.002, 0.005, 0.01, 0.02\}$ for the learning rate.

For AMUGraph, we tested with various query selection ranges. As a reminder, the selection range is the range of (normalized) reconstruction losses that indicate the model is uncertain about a node reconstruction and, therefore, needs to be queried. The range has a lower threshold and upper threshold. We run a grid search over the lower threshold values of $\{0.2, 0.3, 0.4, 0.45, 0.5\}$ and upper threshold values of $\{0.5, 0.55, 0.6, 0.7, 0.8\}$. As mentioned before, the most optimal range was [0.4, 0.8]. Also, we tried the values $\{8, 16, 32, 64, 128$ and $256\}$ for the latent representation dimension. 8 was the optimal dimension size for the Yelp dataset while 32 was most optimal for the Pubmed dataset.

For the baselines DOMINANT and GDN, we use the hyperparameters provided in the source code found on Github. We find that the performance of both algorithm match to that in the original papers.

Query Budget. We chose 3.5% as our query budget due to the results of our initial experimentation. We tried looking for the smallest number of labeled points that performed the best and we found 3.5% to be such a number.

C.1 EMPIRICAL RESULTS

In our main text, we mention that we run our algorithm for 5 rounds with a query budget 0.69% and that our querying criteria was a reconstruction loss in the range of $[0.4, 0.8]$. Additionally, we use a latent dimension size of 32. Here, we provide empirical results from the hyperparameter tuning we performed.

Table 2 demonstrates the performance with varying round-query budget values for the Yelp dataset. It seems that with fewer rounds and more query budget in each round, the performance improves. This could be due to the larger training set in the settings with fewer rounds, which allows for more robust learning.

Rounds	Query Budget (%)	ROC-AUC(%)
5	103 (0.69)	72.6
10	51 (0.35)	70.5
16	32 (0.21)	68.3
20	25 (0.17)	67.1

Table 2: ROC-AUC wrt varying round-query budget values. Note that each round-query budget pair adds up to 3.5% of the entire dataset.

Table 3 demonstrates the performance with varying lower and upper threshold values for the Yelp dataset. AMUGraph seems to suffer with larger thresholds (i.e., $[0.2, 0.8]$) and smaller thresholds (i.e., $[0.2, 0.4]$, $[0.4, 0.6]$, and $[0.6, 0.8]$). Larger thresholds might include points with high certainty, therefore not gain much information during querying. Next, while it is our assumption that uncertainty results in reconstruction losses near 0.5, that might not be true for all cases. Therefore, smaller thresholds, even if they are near 0.5, would not be flexible enough to fully capture uncertainty. For these reasons, $[0.4, 0.8]$ seems to be a good compromise.

Table 4 demonstrates the performance with varying latent dimension sizes on both datasets. There does not seem to be a clear reason why a latent dimension size of 32 works the best with both datasets. However, one can notice a stark performance improvement with a latent dimension size of 32 compared to that of 16 or 64.

	0.2	0.4	0.6
0.4	67.3	x	x
0.6	67.1	65.1	x
0.8	66.4	72.6	64.0

Table 3: ROC-AUC wrt varying lower and upper threshold values. The column labels indicate the lower threshold while the rows indicate the upper threshold.

	Pubmed	Yelp
8	52.2	65.4
16	66.6	57.9
32	74.1	72.6
64	51.6	67.5

Table 4: ROC-ARC wrt varying latent dimension sizes.

D DATASETS

In this paper, we use two datasets: Pubmed Sen et al. (2008) and Yelp Rayana & Akoglu (2015). The Pubmed graph dataset is a citation network dataset where nodes are publications and edges indicate citation relationships. This is a node classification dataset where the goal is to predict the category of a certain publication. The Yelp dataset (or Fraud Yelp Dataset) contains restaurant and hotel reviews where nodes are reviews and edges connect reviews to users, reviews to products, and reviews to reviews. This is another node classification dataset where the goal is to identify fraudulent reviews. Table 5 show the statistics of these datasets.

	Pubmed	Yelp
# Nodes	19,717	45,954
# Edges	88,651	8,051,348
% Training	74.9	74.9
% Anomalous	20.81	14.53

Table 5: Statistics of the Pubmed and Yelp graph datasets.

E UNSUPERVISED METHODS

In the below table, we compare our method with the unsupervised method AnomalyDAE Haoyi Fan & Li (2020). We find that AnomalyDAE performs better than our method. Still, the settings of the two problems are different in which our work aims to generate good soft labels for anomaly detection.

	Pubmed	Yelp
DOMINANT	60.6	57.4
GDN	61.7	67.8
AMUGraph-SVAE	50.7	51.9
AMUGraph-NoAugment	52.3	51.9
AMUGraph-Random	54.7	65.7
AnomalyDAE	**74.3**	**74.0**
AMUGraph	74.1	72.6

Table 6: ROC-AUC (%) on the selected datasets and baselines. Higher is better.