Anomaly Detection on Attributed Networks Using Graph Neural Networks

Luka Kurešević
Departmant of mathematics and informatics
Faculty of Science
University of Novi Sad
luka.kuresevic@gmail.com

November 4, 2025

Abstract

Anomaly detection in attributed networks is a critical task with applications in fraud detection, cybersecurity, and social network analysis. Graph Neural Networks (GNNs) have shown strong potential in capturing complex dependencies in graph-structured data. In this work, we build upon the *DOMINANT* method proposed by Ding et al., which uses Graph Convolutional Networks (GCNs) for deep anomaly detection on attributed networks. We propose an alternative model that replaces GCN layers with Graph Attention Networks (GATs), hypothesizing that attention mechanisms can enhance the detection of anomalies by assigning different importances to neighboring nodes. We evaluate both models on the benchmark datasets Cora and CiteSeer to compare their anomaly detection performance.

1 Introduction

Anomaly detection aims to identify instances that deviate significantly from the majority of data. In the context of attributed networks, where nodes have associated feature vectors and edges represent relationships, detecting anomalies is particularly challenging due to the additional layer of complexity that structural information presents.

Graph Neural Networks (GNNs) have emerged as powerful tools to model relational data by leveraging both node attributes and network topology. The *DOMINANT* method [1] represents one of the first deep anomaly detection frameworks for attributed networks. It employs Graph Convolutional Networks (GCNs) to learn joint representations of node features and graph structure in an unsupervised fashion.

In this work, we propose a variant of DOMINANT that replaces GCNs with Graph Attention Networks (GATs) [3]. Unlike GCNs, GATs introduce attention coefficients to dynamically weigh the importance of neighboring nodes during feature aggregation. It is our hypothesis that this can improve anomaly detection performance by providing more expressive local neighborhood modeling, particularly when some nodes have noisy or irrelevant neighbors. We evaluate this hypothesis on two widely used citation network datasets: Cora and CiteSeer.

2 Preliminaries

2.1 Graph Convolutional Networks

Graph Convolutional Networks (GCNs), introduced by Kipf and Welling [2], extend convolutional operations to graph-structured data. Let a graph be defined as $G = (V, E)$ with N nodes, adjacency matrix A, and feature matrix $X \in \mathbb{R}^{N\times F}$. A GCN layer performs propagation as:

$$H^{(l+1)} = \sigma(\tilde{D}^{-1/2}\tilde{A}\tilde{D}^{-1/2}H^{(l)}W^{(l)}), \tag{1}$$

where $\tilde{A} = A+I$ includes self-loops, $\tilde{D}$ is its degree matrix, $W^{(l)}$ is a trainable weight matrix, and $\sigma(\cdot)$ is an activation function. This operation aggregates normalized information from each node's neighbors, capturing both local graph structure and node attributes.

2.2 Graph Attention Networks

Graph Attention Networks (GATs) [3] introduce an attention mechanism to the message-passing process. Instead of treating all neighbors equally, GATs compute *attention coefficients*:

$$\alpha_{ij} = \frac{\exp(\mathrm{LeakyReLU}(a^\top[Wh_i \,\|\, Wh_j]))}{\sum_{k\in\mathcal{N}(i)} \exp(\mathrm{LeakyReLU}(a^\top[Wh_i \,\|\, Wh_k]))}, \tag{2}$$

where $\|$ denotes concatenation, W is a learnable weight matrix, and a is an attention vector. These coefficients play a role in weighing the contributions of neighbor features, giving greater importance to more relevant messages during aggregation. In turn, robustness to noise and outliers is improved. The aggregated features are subsequently mapped through a non-linear activation function (σ):

$$h_i' = \sigma\left(\sum_{j\in\mathcal{N}(i)} \alpha_{ij}Wh_j\right). \tag{3}$$

3 Data Preparation

3.1 Anomaly Injection

To test how well anomaly detection models work, we need a known list of anomalies (ground truth). Because standard datasets like Cora and CiteSeer don't have pre-labeled anomalies, we have to add fake ones to the networks for our tests.

Following methods from [1], we create a mix of anomalies by changing both the graph structure and the node features:

- **Structural Anomalies:**
  We mess with the network structure by making small, highly-connected groups called cliques. This is a good way to represent a structural anomaly, where a few nodes are connected much more tightly to each other than to the rest of the network.
  We randomly pick m nodes and connect all of them to each other, creating one clique. All m nodes are marked as anomalies. We do this n times, making a total of $m \times n$ structural anomalies.

- **Attribute Anomalies:**
  We create anomalies by changing node features, making sure we add the same number of attribute anomalies as structural ones ($m \times n$).
  For each node chosen to be an anomaly, we replace its feature vector x_i with the features x_j of another node j. We pick j to be the node whose features are most different (biggest Euclidean distance, $\|x_i - x_j\|_2$) from node i among k sampled nodes. This makes the node's attributes look very different from its neighbors.

Adding these controlled anomalies gives us the known ground truth needed to measure model success using metrics like AUC.

3.2 Datasets

We conduct experiments on two citation network datasets:

- **Cora:** Contains 2,708 nodes (papers), 5,429 edges (citations), and 1,433-dimensional binary feature vectors representing word occurrences. Each node belongs to one of seven classes.
- **CiteSeer:** Consists of 3,327 nodes, 4,732 edges, and 3,703-dimensional feature vectors, also with six classes.

4 DOMINANT

DOMINANT [1] is an unsupervised deep anomaly detection framework for attributed networks. It jointly reconstructs the attribute matrix X and the adjacency matrix A using a GCN-based autoencoder. The model minimizes a reconstruction loss combining structural and attribute reconstruction errors:

$$\mathcal{L} = \|A - \hat{A}\|_F^2 + \|X - \hat{X}\|_F^2, \tag{4}$$

where $\hat{A}$ and $\hat{X}$ are the reconstructed adjacency and attribute matrices.

Nodes with high reconstruction errors are more likely to be anomalous. The underlying intuition is that GCNs are better at reconstructing nodes consistent with the overall graph structure and attributes, while anomalies yield higher residuals.

4.1 Motivation for Using GATs

GCNs aggregate information uniformly from neighbors, which can dilute the signal from relevant neighbors and amplify noise from irrelevant ones, especially in sparse or noisy graphs. Introducing attention-based weighting to anomaly detection could make reconstruction errors more discriminative, potentially enhancing precision.

5 Experiments

5.1 Setup

Experiments were conducted on a Linux Mint 21.04 machine with 8GB of RAM and an Intel i5 9th Gen CPU.

Hyperparameters used during data preparation and anomaly injection were:

- Clique size for structural anomalies: $m = 15$
- Number of injected cliques/attribute candidates: $n = 7$
- Number of sampled nodes for attribute perturbation: $k = 50$

while training hyperparameters include:

- Hidden dimension size: $hidden\_dim = 64$
- Number of training epochs: $epoch = 101$
- Learning rate: $lr = 5 \times 10^{-3}$
- Dropout rate: $dropout = 0.3$
- Trade-off parameter for loss function: $\alpha = 0.8$

5.2 Metrics

Model performance is evaluated using the **Area Under the Receiver Operating Characteristic Curve**(ROC-AUC). **ROC-AUC** is a standard metric in binary classification, and its relevance here lies in its ability to measure the model's capacity to rank anomalies (positive class) higher than normal instances (negative class), irrespective of a specific classification threshold. A higher AUC value (closer to 1.0) indicates an overall better anomaly detection capability.

5.3 Results

Based on the three provided plots, the experimental evaluation comparing the GCN-based and GAT-based anomaly detection models on Cora and CiteSeer reveals distinct performance characteristics and loss behaviors.

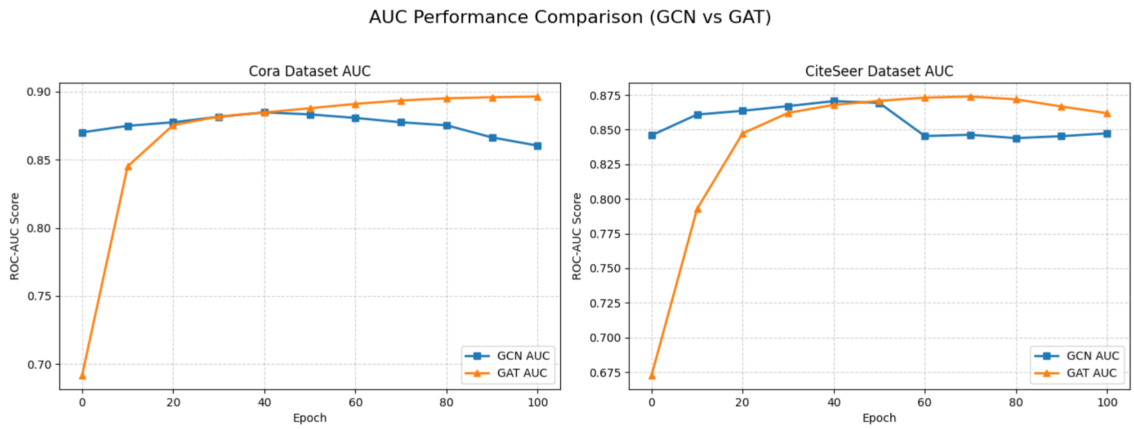

Figure 1: AUC Performance Comparison (GCN vs GAT) on Cora and CiteSeer.

5.3.1 AUC Performance Comparison (GCN vs GAT)

The AUC comparison plot (Figure 1) indicates that the GAT-based model generally achieves better overall anomaly detection performance than the GCN-based model on both datasets.

- **Cora Dataset:** GAT (AUC peaking around 0.896) consistently outperforms GCN (AUC peaking near 0.885). GCN's AUC shows a slight decline in later epochs, while GAT's performance remains stable near its peak.
- **CiteSeer Dataset:** GAT demonstrates a stronger, more rapid improvement in the first 40 epochs. It stabilizes at a slightly higher peak AUC (around 0.874) compared to GCN (AUC around 0.870).

The attention mechanism utilized by GAT likely contributes to its superior performance by adaptively weighting neighbor information, improving the model's ability to reconstruct the attributes and structure of non-anomalous nodes more accurately.

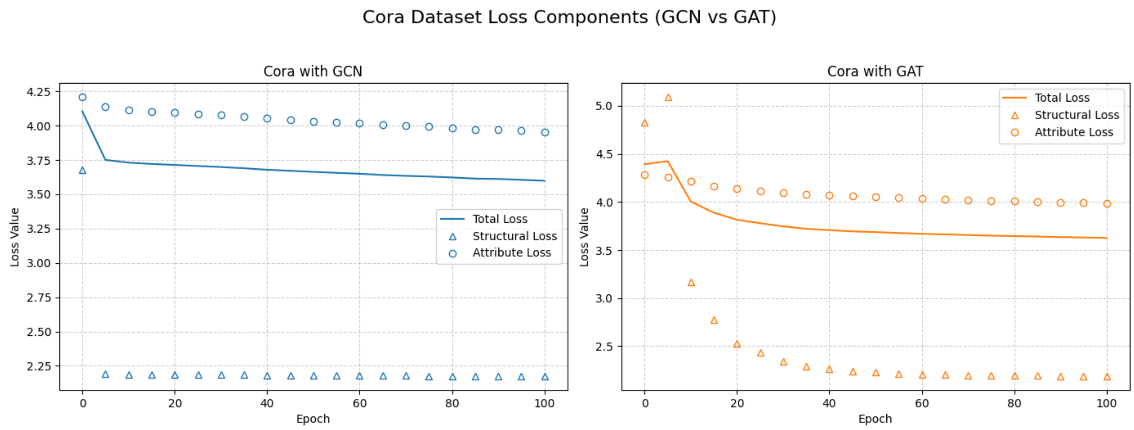

Figure 2: Cora Dataset Loss Components (GCN vs GAT).

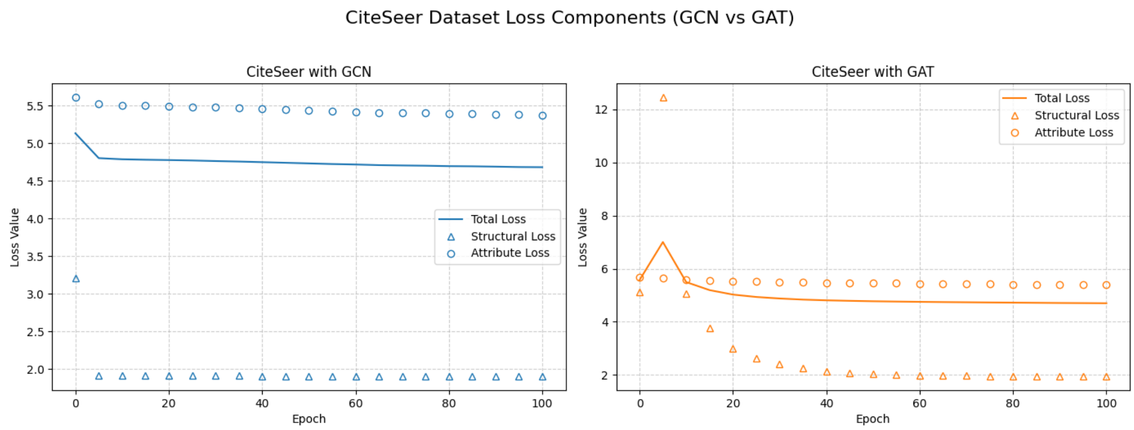

Figure 3: CiteSeer Dataset Loss Components (GCN vs GAT).

5.3.2 Analysis of Loss Components

The loss component plots (Figures 2 and 3) illustrate the model's learning dynamics by separating the Total Loss ($\mathcal{L}$), the Structural Loss ($\mathcal{L}_{\text{struct}}$), and the Attribute Loss ($\mathcal{L}_{\text{feat}}$). The analysis confirms that $\mathcal{L}_{\text{struct}}$ is minimized quickly, making $\mathcal{L}_{\text{feat}}$ the dominant factor in the training challenge ($\alpha = 0.8$).

Cora Dataset Loss

- For both GCN and GAT (Figure 2), the Structural Loss component is extremely low (≈ 2.2) and nearly flat throughout training, indicating rapid and effective reconstruction of the graph structure.
- The Total Loss is primarily driven by the Attribute Loss, which steadily decreases from ≈ 4.2 to ≈ 4.0. The difficulty in the Cora anomaly detection task lies mainly in accurately reconstructing the attribute information.

CiteSeer Dataset Loss

- Similar to Cora, the Structural Loss for both models (Figure 3) is minimized early and remains very low (≈ 2.0).
- The CiteSeer loss values are generally higher than those on Cora, reflecting a harder reconstruction task. The attribute loss remains high (≈ 5.5) for both models.

5.3.3 Summary

The results highlight two main takeaways: (1) The GAT-based model's attention mechanism appears to offer a slight advantage in anomaly detection performance on both datasets (Figure 1). (2) For both models, minimizing structural loss was fairly easy (Figures 2 and 3). This suggests that improving the attribute information reconstruction accuracy is the key bottleneck for future performance gains.

6 Conclusions

This work extended the DOMINANT framework by incorporating Graph Attention Networks (GATs) for anomaly detection in attributed networks. Our experiments demonstrated that the GAT-based model generally yielded *higher AUC scores* than the GCN-based model on Cora and CiteSeer. This suggests that adaptive neighbor weighting contributes positively to anomaly sensitivity. We observed that minimizing the *structural loss* was simpler compared to reconstructing the *node attribute information.* Future work will involve testing on larger, more diverse datasets such as BlogCatalog and Flickr, and developing strategies to enhance the accuracy of attribute reconstruction.

References

[1] Kaize Ding, Jundong Li, Rohit Bhanushali, and Huan Liu. Deep Anomaly Detection on Attributed Networks. In *Proceedings of the 2019 SIAM International Conference on Data Mining (SDM)*, pages 594–602. SIAM, 2019.

[2] Thomas N. Kipf and Max Welling. Semi-Supervised Classification with Graph Convolutional Networks. In *International Conference on Learning Representations (ICLR)*, 2017.

[3] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph Attention Networks. In *International Conference on Learning Representations (ICLR)*, 2018.