Detecting Fraud With Oversampling Techniques and Sparsity Contraints

Prabina Pokharel
p2pokharel@ucsd.edu

Yandong(Dennis) Xiang
yaxiang@ucsd.edu

Jingyu Zhang
jiz036@ucsd.edu

Gal Mishne
gmishne@ucsd.edu

Yusu Wang
yusuwang@ucsd.edu

Abstract

Fraud detection is prevalent now more than ever due to the massive surge in the usage of online platforms. Many techniques exist to combat fraud; however, they often fail to capture the imbalancement in data involving fraudulent activities. It's important to tackle such concern so we can harness its power to correctly predict anomalies. So, the question remains: How can we effectively detect and mitigate fraudulent activities, especially when faced with imbalanced datasets? Our research contributes to the study of such concern with a model that harnesses the benefits of many existing models. We propose a solution that utilizes a combination of oversampling techniques and sparsity constraints to balance and predict fraud data.

Code: https://github.com/yandongxiang/GNN-DSC180A

1 Introduction

In recent years, there has been a huge surge in the usage of online platforms like Reddit, Yelp, and Amazon. While the usage of such platforms has positively influenced our daily lives, fraudulent activities is amid this rapid increase in digital data volume and anonymity of online networks. Fraudulent behaviors are executed and hidden in plain sight of this vast amount of online records. Thus, enhancing fraud detection is crucial to protect us from fraudsters against harmful scams, and even criminal activities.

1.1 Discussion of Prior Work and Our Proposed Solution

With fraud detection in mind, researchers have devised models aimed at enhancing specific aspects of the task. One such paper whose goal is to detect fake news injects data into a social graph refinement component that iteratively updates the edge weights using a learnable degree correction mask Wu and Hooi (2023). This allows for an improvement of edge noise in graph datasets and to join information with a GNN-based detector for better optimization.

Another paper whose task was to detect group frauds in e-commerce platforms used an autoencoder to concatenate weighted structural features with the attributes of customer vertices to improve accuracy Yu et al. (2023).

These are two of many techniques used to detect fraud. As we can see, these are the solutions in place to detect fraudulent activities, however, such models don't tackle the imbalance structure of fraud datasets very well.

So, our proposed solution is the combination of 2 models: GraphSmote (Zhao, Zhang and Wang 2021) and SparseGAD (Gong et al. 2023). GraphSmote, although not commonly used for fraud detection, is an oversampling technique that identifies minority class nodes and creates new nodes that resemble those minority class data points, thus helping it balance. Since datasets involving fraud detection are hugely imbalanced, GraphSmote will help us balance the class distribution in the graph. We will then take the output of this GraphSmote and feed it into SparseGAD, known for anomaly detection by introducing sparsity constraints. Such constraints help highlight significant connections, and anything that substantially deviates from that will be looked into further for fraud. We will discuss our model in more detail in the Methods section of this paper.

By tackling this fraud detection task, we hope to contribute to protecting us against scams.

1.2 Datasets

We'll be implementing our model to three datasets: Amazon, Yelp, and Reddit. These datasets are obtained from the Deep Graph Library (DGL). Below are the links to access these datasets:

- Amazon Dataset

- Yelp Dataset
- Reddit Dataset

1.2.1 Info on the Datasets

Amazon Dataset: This dataset includes product reviews under the Musical Instruments category. This is a binary classification task where users with more than 80% helpful votes are labeled as benign entities and those with less than 20% are labeled as fraudulent. Positive(fraudulent)-Negative(benign) ratio is 1 : 10.5, which shows that it's imbalanced.

Yelp Dataset: This dataset includes hotel and restaurant reviews. This is a binary classification task where it's divided into filtered(spam) and recommended(legitimate). Here, Positive(spam)-Negative(legitimate) ratio is 1 : 5.9, which shows that this dataset is also imbalanced.

Reddit Dataset: This is a graph dataset from Reddit posts made in September 2014. The node refers to the community that a certain post belongs to. This dataset contains 50 large communities. The nodes are connected if the same user comments on both communities. This dataset contains 232,965 posts with an average degree of 492. The first 20 days will be used for training and the remaining 10 days for testing, with 30% used for validation. We are tasked with predicting whether a user is banned. The ground truth states that there is a 3.3% anomaly, which shows that, along with the other two datasets, this is also imbalanced.

All of the datasets we choose for our model come from a reputable library (DGL) and are used in many research papers. They will be useful for our model since it's tasked to fix imbalances while doing anomaly detection.

2 Methods

We are trying to solve the problem by using a graph neural network, or more specifically GAD (Graph Anomaly Detection). Previous works on anomaly detection exist. However, we observed that none of them attempted to create a GraphSMOTE model specifically designed for anomaly detection. This is interesting because while the GraphSMOTE model performs well on imbalanced datasets, it is not designed to capture anomalous users among the common users.

GraphSMOTE is designed to add synthetic nodes to the graph to balance the dataset. In the case of graph anomalous user detection, the anomalous users only represent a small portion of the dataset. To balance out the dataset, we would add synthetic anomalous nodes into the graph to create a balanced, amplified dataset. The GraphSMOTE model typically runs an edge generator to generate edges on the synthetic nodes to find their links to other nodes. To not disturb the connection between the synthetic node and other nodes, we would use a specific method to both preserve node connections and create synthesized nodes.

The GraphSMOTE model alone does not capture the heterophily nature of anomalous nodes. While anomalous users attempt to blend into common users by establishing fake connections with other users, anomalous users often show greater dissimilarity with their connected users. After using the SMOTE method to create a balanced dataset, we would then implement techniques from SparseGAD to add sparsity into the graph and generate a learnable adjacency matrix (homey matrix) to identify whether the connected nodes are similar or dissimilar. In this way, we can allow our model to both address the imbalanced nature of the dataset and observe the nature of heterophily in anomalous nodes.

2.1 Synthetic Node Generation

In particular, we would use a combination of the upsampling method and the SMOTE method to generate synthetic nodes. First, we determine which class label belongs to the minority class. In our case, the minority class is typically the anomalous users, as they generally occur less frequently than the common users. To generate synthetic nodes to amplify the minority class, we use the upsampling method to randomly replicate nodes and reproduce their connections with their neighbors. The reason we aim for identical neighbors for the synthetic nodes is to maintain a similar level of heterogeneity for the links of the minority classes. We then add randomized minor differences in the features of synthetic nodes to create distinctions between synthetic nodes and the original nodes, as in the SMOTE method. Just to clarify, the reason why we avoid randomized differences in link connections is that we do not want to disrupt the model's ability to identify the heterophilic nature of the anomalous nodes.

2.2 Homey Graph Generation

Instead of the original adjacency matrix, we utilized the "homey" adjacency matrix concept from the GraphSMOTE model. In this model, a two-layer MLP (Multi-Layer Perceptron) model is introduced to calculate the cosine similarity of the adjacent node pairs, following this function:

$$H_{uv} = \text{Cosine Similarity} = \frac{z_0^u \cdot z_0^v}{||z_0^u||||z_0^v||} \in [-1, 1] \tag{1}$$

Where each item in the homey matrix H is computed with cosine similarity. H_{uv} ranges from -1 to 1. When $H_{uv} = 1$, it means the node pair is homophilic; when $H_{uv} = -1$, the node pair is heterophilic. However, when the node pair has no relation, $H_{uv} = 0$.

2.3 Sparsification

Although we have calculated cosine similarity for the homey matrix, we still haven't addressed the nature of anomalous users camouflaging themselves among the common users. Thus, we also need the sparsification method from SparseGAD to filter out elements in the adjacency matrix. We will use δ as the threshold to remove unnecessary neighbors.

$$H_{uv} = \begin{cases} 0 & \text{if } H_{uv} \leq \delta \\ H_{uv} & \text{otherwise} \end{cases} \tag{2}$$

After setting δ as a threshold, SparseGAD further utilizes KNN to limit the number of necessary connections. Finally, the GAD-oriented regularization developed in SparseGAD is employed to further sparsify the graph, preventing faulty links from hindering the model's ability to distinguish anomalous users.

2.4 Learning Objective

In the end, we use our model to classify the users into normal users and abnormal users based on the node features and neighborhood similarity. We will use the ROC-AUC score:

$$\text{ROC\_AUC} = \frac{1}{m(m-1)} \sum_{i=1}^{m} \sum_{j=1, j \neq i}^{m} AUC_{ij} \tag{3}$$

We will use the ROC-AUC score as the metric to measure whether our model outperforms the baseline models.

3 Results

4 Discussion

5 Conclusion

References

Gong, Zheng, Guifeng Wang, Ying Sun, Qi Liu, Yuting Ning, Hui Xiong, and Jingyu Peng. 2023. “Beyond Homophily: Robust Graph Anomaly Detection via Neural Sparsification.” In *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence, IJCAI*.

Wu, Jiaying, and Bryan Hooi. 2023. “DECOR: Degree-Corrected Social Graph Refinement for Fake News Detection.” In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*.

Yu, Jianke, Hanchen Wang, Xiaoyang Wang, Zhao Li, Lu Qin, Wenjie Zhang, Jian Liao, and Ying Zhang. 2023. “Group-based Fraud Detection Network on e-Commerce Platforms.” In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. New York, NY, USA Association for Computing Machinery. [Link]

Zhao, Tianxiang, Xiang Zhang, and Suhang Wang. 2021. “GraphSMOTE: Imbalanced Node Classification on Graphs with Graph Neural Networks.” In *Proceedings of the 14th ACM International Conference on Web Search and Data Mining*. ACM. [Link]

Appendices

A.1 Project Proposal

Detecting Fraudulent Activities By Fusing GraphSmote and SparseGAD

Prabina Pokharel
p2pokharel@ucsd.edu

Yandong(Dennis) Xiang
yaxiang@ucsd.edu

Jingyu Zhang
jiz036@ucsd.edu

Gal Mishne
gmishne@ucsd.edu

Yusu Wang
yusuwang@ucsd.edu

1 Proposal

1.1 Problem Statement

In recent years, there has been a huge surge in the usage of online platforms like Reddit, Yelp, and Amazon. While the usage of such platforms has positively influenced our daily lives, fraudulent activities amid this rapid increase in digital data volume and anonymity of online networks. Fraudulent behaviors are executed and hidden in plain sight of this vast amount of online records. Thus, enhancing fraud detection is crucial to protect us against harmful scams, and even criminal activities, by fraudsters.

1.2 Method

This relates to our quarter 1 project because we are working on graph datasets and we are trying to solve the problem by using a graph neural network, or more specifically GAD (Graph Anomaly Detection). Our quarter 1 project relates to a more general graph neural network study, but our quarter 2 project specifically focuses on anomaly detection.

There are previous works on anomaly detection. However, we observed that none of them attempted to create a GraphSmote model specifically designed for anomaly detection. This is interesting because while the GraphSmote model performs well on imbalanced datasets, it is not designed to capture anomalous nodes.

GraphSmote is designed to add synthetic nodes to the graph to balance the dataset. In our case, since anomalous nodes only represent a small portion of the dataset, we would add synthetic anomalous nodes into the graph to create a balanced amplified dataset. Then we would use an edge generator to generate edges and run the GNN model on the new dataset.

However, GraphSmote does not capture the heterophily nature of anomaly nodes. While anomaly nodes attempt to blend in with other nodes, anomaly nodes often show greater dissimilarity with their connected nodes.

Such, we propose modifying the edge generator. We should focus on reserving and generating only the task-related edges. We would then implement techniques from SparseGAD to add sparsity into the graph and generate a learnable adjacency matrix (homey matrix) to identify whether the connected nodes are similar or dissimilar. In that way, we can allow our model to both address the imbalanced nature of the dataset and observe the nature of heterophily in anomalous nodes.

1.3 Primary Output

Our primary output will be a paper about our processes, findings, and improvements. The paper will also link to our GitHub repo with our work in improving fraud detection. Our task is to classify the nodes in our datasets as Normal or Fraudulent and calculate the precision of our detection model with F1-score and ROC-AUC. Our goal is for the two metrics to be significantly higher for our model than the baseline GNNs like GCN, GIN, and GAT, and potentially higher than simply GraphSmote and SparseGAD.

2 Data

2.1 Datasets

We'll be using 3 datasets: Amazon, Yelp, and Reddit. These datasets will be obtained from the Deep Graph Library (DGL). Below are the links to access these datasets:

- Amazon Dataset
- Yelp Dataset
- Reddit Dataset

2.2 Info on the Datasets

Amazon Dataset: This dataset includes product reviews under the Musical Instruments category. This is a binary classification task where users with more than 80% helpful votes are labeled as benign entities and those with less than 20% are labeled as fraudulent. Positive(fraudulent)-Negative(benign) ratio is $1 : 10.5$, which shows that it's imbalanced.

Yelp Dataset: This dataset includes hotel and restaurant reviews. This is a binary classification task where it's divided into filtered(spam) and recommended(legitimate). Here, Positive(spam)-Negative(legitimate) ratio is $1 : 5.9$, which shows that this dataset is also imbalanced.

Reddit Dataset: This is a graph dataset from Reddit posts made in September 2014. The node refers to the community that a certain post belongs to. This dataset contains 50 large communities. The nodes are connected if the same user comments on both communities. This dataset contains 232,965 posts with an average degree of 492. The first 20 days will be used for training and the remaining 10 days for testing, with 30% used for validation. We are tasked with predicting whether a user is banned. The ground truth states that there is a 3.3% anomaly, which shows that, along with the other two datasets, this is also imbalanced.

2.3 Data Quality and Usefulness

All of these datasets come from a reputable library (DGL) and are used in many research papers. They will be useful for our model since it's tasked to fix imbalances while doing anomaly detection.

B Contribution

- Prabina
  - Worked on all sections of the research paper excluding Methods
  - Compiled the accuracy and ROC AUC score for GraphSmote model for the three datasets mentioned (Amazon, Yelp, Reddit) and visualized it for the poster
- Yandong
  - Worked on the Methods section of the research paper
  - Wrote code to obtain accuracy and ROC AUC score for GraphSmote model for the three datasets
  - Worked on the code checkpoint
- Jingyu
  - Wrote code to obtain accuracy and ROC AUC score for GCN, GIN, and GAT models for the three datasets
  - Worked on the code checkpoint