Detecting Fraud With Oversampling Techniques and Sparsity Contraints

Prabina Pokharel, Yandong (Dennis) Xiang, Jingyu Zhang, Gal Mishne, Yusu Wang

University of California San Diego

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.

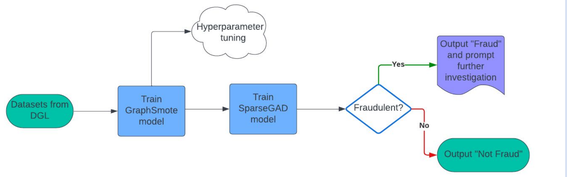
Dataset and Background

Our model uses the following three datasets: Amazon, Yelp, and Reddit. These datasets are obtained from the Deep Graph Library (DGL). These datasets are in a graph format.

All of the datasets has one structure in common: imbalance class. Amazon dataset, for example, ncludes 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. Here, p=Positive (fraudulent) - Negative (benign) ratio is 1:10.5, which shows that it's imbalanced.

Datasets such as this will be useful for our model since it's tasked to fix imbalances while doing anomaly detection.

Data Workflow



In this workflow, we have the datasets from DGL as inputs. We feed these into Graph-Smote model. There, we finetune some hyperparameters to achieve a better ROC-AUC score. We take the output of Graph-Smote and feed it as input to SparseGAD model. From its output, we predict whether a data is fraud or not. If it's fraud, we notify authorities in-charge to delve deeper to conclude whether it was truly a fraud or not.

We will expand on some topic or discuss a section in more detail to utilize this space.

Our Model

We utilize the combination of 2 models: GraphSmote and SparseGAD. 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.

How over-sampling words in GraphSmote is that the model would generate synthetic nodes. First, we determine which class label belongs to the minority class, which in our case, is 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, as seen in the image below. 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.

Overcoming sparsity task arose from the concern about the nature of anomalous users camouflaging themselves among the common users. Thus, we added sparsification method from SparseGAD as seen on the figure below to filter out elements in the adjacency matrix. We will use a delta as the threshold to remove unnecessary neighbors.

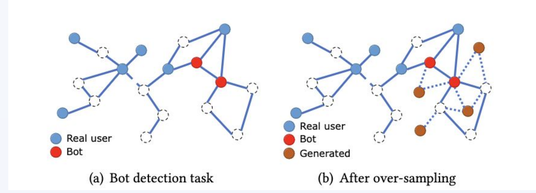


Figure 1: Showcasing over-samping in latent space

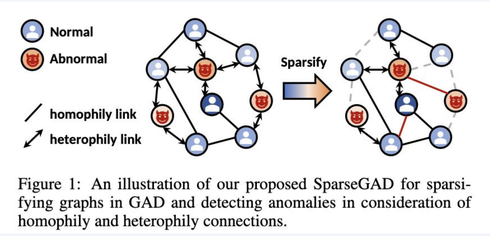


Figure 1: An illustration of our proposed SparseGAD for sparsifying graphs in GAD and detecting anomalies in consideration of homophily and heterophily connections.

Figure 2: Illustration of SparseGAD's sparsification process

Results

Results yet to be finalized.

Future Work

We'd like to expand our model to other datasets. We'd also like to finetune our model further. We are interested in obtaining a partnership to deploy this model and bridge cybersecurity fraud.

References

Bibliography

GitHub