Proceedings of the 4th ITB Graduate School Conference
Innovation and Discovery for Sustainability
July 6, 2023

Tackling Class Imbalance Problem with Adversarial Attention-based Variational Graph Autoencoder: Study in Fraud Detection

Nur Alibasyah Wiriaatmadja[1,*] & Finny Oktariani[1,2]

[1]Department of Computational Science, Institut Teknologi Bandung, Jalan Ganesa 10, Bandung 40132, Indonesia
[2]Combinatorial Mathematics Research Group, FMIPA, Institut Teknologi Bandung, Jalan Ganesha 10, Bandung 40132, Indonesia
[*]Email: nuralibasyah@gmail.com

Abstract. Class Imbalance is often encountered in many classification problems for machine learning, resulting in bias towards the majority class. Various techniques have been developed to address this issue, focusing on oversampling the minority class or undersampling the majority class. This research aims to tackle the class imbalance problem with a different approach by using Adversarial Attention-based Variational Graph Autoencoder (AAVGA) introduced by Weng et al. in 2020. This approach is studied for fraud detection, which graph model of classification task are first constructed by mapping each entity as nodes and transaction between them as edges. The experiment is conducted by varying class distributions to analyze how imbalance class influences the prediction score. We obtained that our approach produced better precision and recall score, even for extremely imbalanced dataset.

Keywords: *imbalance classification; graph embedding; graph autoencoder; graph neural network; fraud detection.*

1 Introduction

In the context of classification using machine learning, class imbalance occurs when the number of data points in one class is significantly larger than the number of data points in the other classes. This issue can result in the model being biased towards the majority class and having poor accuracy for the minority class [1].

The class imbalance problem is commonly encountered and naturally occurs in real-world problems, such as fraud detection. The number of fraudulent transactions is much smaller compared to the number of genuine transactions [2]. In this case, the class imbalance problem needs to be addressed first because it is important to identify every case of fraudulent transactions to prevent losses for the company and service users.

Various techniques have been developed to address the class imbalance problem. In general, we can preprocess the data to balance the number of each class and we can employ Cost-Sensitive (CS) or One Class Classification (OCC) within the algorithm [3]. Such preprocessing is often done by oversampling the minority class or undersampling the majority class.

Uprising approach in fraud detection with machine learning includes graph modeling, such as Graph Attentive Network for Financial Fraud Detection [4]. High learning capacity of deep graph representation learning opens improvement possibilities for dealing with fraud detection and class imbalance problem in general.

In this paper, we wish to investigate how the graph representation learning approach using Adversarial Attention-based Variational Graph Autoencoder (AAVGA) introduced by Weng et al. [5] tackles the class imbalance problem. We used the generated representation of transaction graph as inputs for machine learning model and compared the result with direct machine learning application for transaction log data.

To use graph modeling approaches, we transform the transaction log data into graph by identifying each entity as nodes and transaction between them as edges. To simplify the problem into node classification task, we use line graph transformation to create new graph which represent each transaction as nodes. This graph will be used as input for AAVGA to generate latent representation of each transaction in low dimensional space. This representation can be used as input for various machine learning algorithms.

2 Class Imbalance in Transaction Graph

The Transaction Graph used in this paper is transformed from synthetic transaction data created by Lopez-Rojas et al. [6] called BankSim. The dataset contains a total of 594,643 records with 7,200 fraudulent transactions. Each transaction log contains information about the customer, merchant, and transaction amount.

The graph transformation process is similar to graph transformation process in [7]. Each unique customer and merchant are identified as a node and each log is considered as an edge connecting them. By this operation, we obtain an adjacency matrix $A \in \mathbb{R}^{n \times n}$ of the graph Γ. To generate features of the graph, we consider the age and gender of customers as node features $X \in \mathbb{R}^{n \times d_1}$ and transaction details as edge features $X^e \in \mathbb{R}^{m \times d_2}$. We labeled each node as customer or merchant, and each edge as fraudulent or genuine transaction.

The line graph $L(\Gamma)$ is obtained by associating a node with each edge of the graph and connecting two nodes with an edge if and only if the corresponding edges of the graph have a node in common. This transformation is done to represent each transaction as a node, therefore simplifying the problem into node classification. The edge features and node features of Γ are merged into node features of $L(\Gamma)$. Line graph transformation process are illustrated in Figure 1.

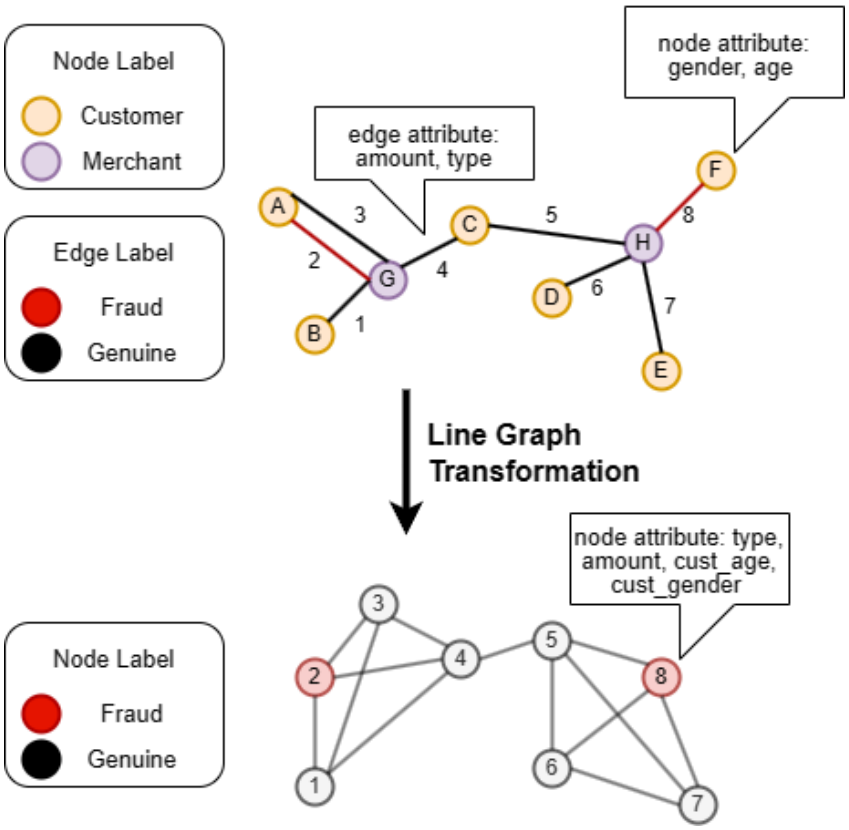

Figure 1. Illustration of Line Graph Transformation process.

Due to computational limitations, we sampled only 2000 logs of transactions from the original dataset. We varied the class distribution to analyze the influence of class imbalance problem in detecting fraudulent transactions. In total, we generated four different datasets with 1:1, 3:1, 9:1, and 98.75:1.25 ratio of genuine and fraudulent transactions. The last ratio, 98.75:1.25 are similar to the ratio in original dataset. Summary of each data set are listed in Table 1.

Table 1. Summary of Sampled Dataset

Statistics	Genuine (0)	Fraudulent (1)
Ratio: 1:1		
n	1000	1000
Amount Mean	31.54	508.51
Amount std	28.1	783.32
Ratio: 3:1		
n	1500	500
Amount Mean	33.35	566.84
Amount std	31.82	848.65
Ratio: 9:1		
n	1800	200
Amount Mean	32.78	45.38
Amount std	45.38	716.36
Ratio: 98.75:1.25		
n	1975	25
Amount Mean	31.27	438.8
Amount std	27.25	667.23

3 Adversarial Attention-Based Variational Graph Autoencoder

The AAVGA model proposed by Weng et al. [5] use autoencoder framework and adversarial mechanism to improve the generated latent representation from encoder, which consists of attention mechanisms layers. The autoencoder and adversarial mechanisms are trained simultaneously by optimizing each loss functions.

3.1 Attention Autoencoder

The graph attention layer computes a weight coefficient of each node with its neighborhood. If $\mathbf{x}_i \in \mathbb{R}^{d^{(l)}}$ are feature of node v_i and $\mathbf{W}$ is weight parameter matrices, weight coefficient e_{ij} of neighbor node v_j are

$$e_{ij} = LeakyReLU\left(a^T\left[\mathbf{W}\mathbf{x}_i || \mathbf{W}\mathbf{x}_j\right]\right), \quad (1)$$

Where $a(\cdot)$ is a function that calculates the correlation between two nodes, which we select a single fully connected layer. The calculated weight then normalized with all neighbors $N(v_i)$ via softmax normalization.

Multi-head attention mechanism computes the normalized weight α_{ij} k times and then averaged to compute new feature vector of node v_i,

$$\mathbf{x}_i' = \sigma\left(\frac{1}{k}\sum_{\mathrm{k}=1}^{\mathrm{k}}\sum_{v_j\in N(v_i)}\alpha_{ij}^{(k)}\mathbf{W}^{(k)}\mathbf{x}_j\right). \tag{2}$$

The attention encoder is used to fit mean matrix $\boldsymbol{\mu}$ and covariant matrix $\boldsymbol{\sigma}$**:**

$$\boldsymbol{\mu} = GAT_{\mu}(\mathbf{X}, \mathbf{A}) \tag{3}$$

$$\boldsymbol{\sigma} = GAT_{\sigma}(\mathbf{X}, \mathbf{A}). \tag{4}$$

Therefore, a variational graph encoder is defined by an inference model

$$\boldsymbol{P}(\mathbf{Z}|\mathbf{X}, \mathbf{A}) = \prod_{i=1}^{N} q(z_i|\mathbf{X}, \mathbf{A}), \tag{5}$$

$$q(z_i|\mathbf{X}, \mathbf{A}) = \mathcal{N}\left(z_i\middle|\mu_i, diag\left(\sigma_i^2\right)\right). \tag{6}$$

The decoder task is reconstructing the adjacency matrix, which are possible using the generated latent representation $\mathbf{Z}$. Reconstructed adjacency matrix $\widehat{\mathbf{A}}$ are obtained by

$$\widehat{\mathbf{A}} = \sigma(\mathbf{Z}\mathbf{Z}^T). \tag{7}$$

The standard normal distribution is chosen as prior distribution for latent variable $p(\mathbf{Z})$. The autoencoder are trained by minimizing the loss function

$$\mathcal{L} = \mathcal{L}_{recon} + \mathcal{L}_{KL} = -\mathbb{E}_{q(\mathbf{Z}|\mathbf{X}, \mathbf{A})} + KL[q(z_i|\mathbf{X}, \mathbf{A})||p(\mathbf{Z})] \tag{8}$$

The KL divergence is added to optimize the variational autoencoder by forcing each normal distribution to approach the standard normal distribution.

3.2 Adversarial Strategy

Adversarial strategies are used to force the latent representation to follow the prior distribution, which imposed to reduce the deviation of the data in the training process. The encoder of autoencoder acts as a generator, and the discriminator D are single fully connected layer. The discriminator objective is to classify the inputs are generated or real, which in this case are sampled from prior distribution p_z.

Optimization of adversarial strategies are done by minimizing the loss function:

$$\mathcal{L} = -\frac{1}{2}\mathbb{E}_{p_z}\log D(Z) - \frac{1}{2}\mathbb{E}_{\mathbf{X}}\log\Big(1 - D\big(G(\mathbf{X},\mathbf{A})\big)\Big) \quad (9)$$

4 Results and Discussion

The experiment is conducted using PyTorch Geometric packages in Python for AAVGA and scikit-learn and XGB packages for machine learning algorithm. We use XGBoost with default parameters as classifier. The data are split into train set and test with 85:15 ratio. The evaluation metrics are calculated from concatenation of training and testing prediction. We recorded the accuracy (ACC), average precision (AP), and Area under ROC Curve (AUC), for each dataset and compared our approach result with direct machine learning application to transaction log dataset. ACC metrics often tend to bias on imbalance classification task, therefore AUC and AP metrics are added. The evaluation metrics are shown in Table 2. We also recorded the confusion matrix (Genuine transaction is Negative and Fraudulent transaction is Positive) shown in Table 3.

Table 2 and Table 3 shows that our approach gives better evaluation metrics results on three of four datasets used. On extremely imbalance data distribution similar to original dataset, our approach obtains perfect accuracy and predicts all of fraudulent transaction correctly. Whereas direct machine learning applications obtained almost perfect results by only three out of two thousand are wrongly predicted.

Direct machine learning applications obtained better results on the 3:1 dataset, with slightly better evaluations metrics than ours. On the dataset, our model obtained higher False Negative (FN) which is vital indicator for fraud detection cases. False negative means the model predicted the transaction is genuine but is actually fraudulent.

By assuming the real transaction graph follows the distribution similar to our original data, we conclude that our approach is promising to be applied in real data, especially on graph-designed dataset. For further improvement, we may combine our approach with common methods to handle imbalanced data such as undersampling and oversampling.

Table 2. Evaluation metrics result.

	ACC	AUC	AP
Ratio: 1:1			
AAVGA	**0.993**	**0.993**	**0.986**
Direct ML	0.992	0.992	0.985
Ratio: 3:1			
AAVGA	0.993	0.986	0.977
Direct ML	**0.995**	**0.994**	**0.982**
Ratio: 9:1			
AAVGA	**0.999**	**0.995**	**0.991**
Direct ML	0.996	0.984	0.963
Ratio: 98.75:1.25			
AAVGA	**1.000**	**1.000**	**1.000**
Direct ML	0.999	0.98	0.887

Table 3. Confusion matrix result.

	TN	FP	FN	TP
Ratio: 1:1				
AAVGA	**986**	**14**	**0**	**1000**
Direct ML	983	13	3	997
Ratio: 3:1				
AAVGA	1498	2	13	487
Direct ML	**1494**	**6**	**4**	**496**
Ratio: 9:1				
AAVGA	**1800**	**0**	**2**	**198**
Direct ML	1978	2	6	194
Ratio: 98.75:1.25				
AAVGA	**1975**	**0**	**0**	**25**
Direct ML	1973	2	1	24

5 Conclusions

In this research, we construct a set of transaction graphs based on transaction logs dataset to apply a graph representation learning with AAVGA model. The transaction graph is made by identifying unique customers and merchants as nodes and a transaction between them is considered as an edge. To use this graph for fraud detection, we used line graph transformation which mapped an edge into a node in new graph, which represent each transaction as nodes. The representation learned by AAVGA are used as inputs XGBoost classifier.

We varied the number of genuine and fraudulent class to follow 1:1, 3:1, 9:1, and 98.75:1.25 ratio to examine the influence of imbalance classification and analyze the graph representation learning potential to handle imbalance classification

without preprocessing. The results show that our approach obtained slightly better results on 1:1, 9:1, and 98.75:1.25 dataset and obtained slightly worse result on 3:1 dataset. On 98.75:1.25 dataset, our approach gained perfect accuracy with all predictions on fraudulent and genuine transactions are correct.

We conclude that our approach is promising to be applied on larger and extremely imbalanced datasets, which naturally occurs on fraud detection tasks. Common methods to handle imbalanced data such as oversampling and undersampling are highly possible to improve our approach.

References

[1] He, Haibo, and Edwardo A. Garcia. "Learning from imbalanced data." IEEE Transactions on knowledge and data engineering 21.9 (2009): 1263-1284.

[2] Wei, Wei, et al. "Effective detection of sophisticated online banking fraud on extremely imbalanced data." World Wide Web 16 (2013): 449-475.

[3] Makki, Sara, et al. "An experimental study with imbalanced classification approaches for credit card fraud detection." IEEE Access 7 (2019): 93010-93022.

[4] Wang, Daixin, et al. "A semi-supervised graph attentive network for financial fraud detection." 2019 IEEE International Conference on Data Mining (ICDM). IEEE, 2019.

[5] Weng, Ziqiang, Weiyu Zhang, dan Wei Dou. "Adversarial attention-based variational graph autoencoder." IEEE Access 8 (2020): 152637-152645.

[6] Lopez-Rojas, Edgar Alonso, & Stefan Axelsson. "Banksim: A bank payments simulator for fraud detection research." 26th European Modeling and Simulation Symposium, EMSS. 2014.

[7] Wiriaatmadja, N. A., & Oktariani, F. (2023). “Adversarial Attention-Based Variational Graph Autoencoder for Fraud Detection in Online Financial Transaction”. [Manuscript in preparation].