Bi-Level Selection via Meta Gradient for Graph-based Fraud Detection

Linfeng Dong[1,2], Yang Liu[1,2], Xiang Ao*[1,2,3], Jianfeng Chi[4], Jinghua Feng[4], Hao Yang[4], and Qing He[1,2,5]

[1] Key Lab of Intelligent Information Processing of Chinese Academy of Sciences (CAS), Institute of Computing Technology, CAS, Beijing 100190, China
[2] University of Chinese Academy of Sciences, Beijing 100049, China
[3] Institute of Intelligent Computing Technology, Suzhou, CAS, Suzhou, China
[4] Alibaba Group, Hangzhou, China
[5] Henan Institutes of Advanced Technology, Zhengzhou University, Zhengzhou 450052, China
{donglinfeng19s,liuyang17z,aoxiang,heqing}@ict.ac.cn
{bianfu.cjf,jinghua.fengjh,youhiroshi.yangh}@alibaba-inc.com

Abstract. Graph Neural Networks (GNNs) have achieved remarkable successes by utilizing rich interactions in network data. When applied to fraud detection tasks, the scarcity and concealment of fraudsters bring two challenges: class imbalance and label noise. In addition to overfitting problem, they will compromise model performance through the message-passing mechanism of GNNs. For a fraudster in a neighborhood dominated by benign users, its learned representation will be distorted in the aggregation process. Noises will propagate through the topology structure as well. In this paper, we propose a **B**i-**L**evel **S**election (BLS) algorithm to enhance GNNs under imbalanced and noisy scenarios observed from fraud detection. BLS learns to select instance-level and neighborhood-level valuable nodes via meta gradient of the loss on an unbiased clean validation set. By emphasizing BLS-selected nodes in the model training process, bias towards majority class (benign) and label noises will be suppressed. BLS can be applied on most GNNs with slight modifications. Experimental results on two real-world datasets demonstrate that BLS can significantly improve GNNs performance on graph-based fraud detection.

Keywords: Graph Neural Network; Fraud Detection; Imbalanced Learning

1 Introduction

Recently, Graph Neural Networks (GNNs) are widely used in fraud detection tasks [2,6]. These approaches build an end-to-end learning paradigm. First, each node is encoded into a representation by aggregating and transforming the information of its neighbors, namely the *representation learning phase*. Then, the

* corresponding author.

learned representation is passed to a classifier to identify the fraudsters from the benign users, namely the *classification phase*.

Despite the remarkable success existing GNN-based methods achieved, the severe class imbalance and noisy label are still vital problems in fraud detection. Due to the contingency of fraudulent activities, the number of positive (fraud) samples is far less than the number of negative (benign) samples in fraud detection tasks. Meanwhile, the concealment of fraudulent activity leads to noisy label problem. Real-world users labeled as benign could either be benign or potentially fraudulent. As a result, the negative instances in the training set may consist of noisy labels.

Based on these observations, we emphasize two key challenges of GNN-based fraud detection as follows:

Neighborhood-level imbalance and noise: In the representation learning phase, due to the propagation mechanism on topology, excess benign neighbors will dominate the network structure and dilute the feature of fraudsters, resulting in inaccurate embeddings of fraudulent nodes.

Instance-level imbalance and noise: In the classification phase, the majority class will dominate the training loss during the gradient descent step, leading to a biased decision boundary. Undiscovered fraudsters (noise) will contribute wrong gradient direction, thus polluting the learned classification boundary.

To tackle the bi-level imbalanced and noisy problems, we propose **B**i-**L**evel **S**election (BLS), a lightweight algorithm for GNN-based fraud detection that learns to select valuable nodes on instance level and neighborhood level through a meta-learning paradigm. BLS first forms an small unbiased and clean meta validation set by picking nodes from training set with high assortativity (the ratio of 1-hop neighbors that share the same label as itself). Then BLS uses the meta set to to guide the training process. It selects valuable nodes according to their potential impact on the meta gradient of validation loss. It follows such assumptions: a better selection of valuable training nodes will improve the model performance and reduce the validation loss.

We integrate BLS with three GNN frameworks: GCN, GAT, and GraphSAGE. Experiments on two real-world fraud detection datasets demonstrate that our algorithm can effectively improve the performance of GNN under imbalanced and noisy settings. BLS enhanced GNNs also outperform state-of-the-art.

Our contributions can be summarized as follows:

- We propose BLS, a meta gradient based algorithm to address the imbalanced and noisy label problem in graph-based fraud detection. In both representation learning and classification phase, BLS adopts a unified meta-learning paradigm to select instance-level and neighborhood-level valuable nodes.
- Compared to existing methods, BLS is the first work that considers the impact of class imbalance and noisy label on the message-passing mechanism of GNNs.
- The proposed BLS algorithm has high portability that can be applied on any GNN framework. By applying BLS on widely-used GNNs, we achieved

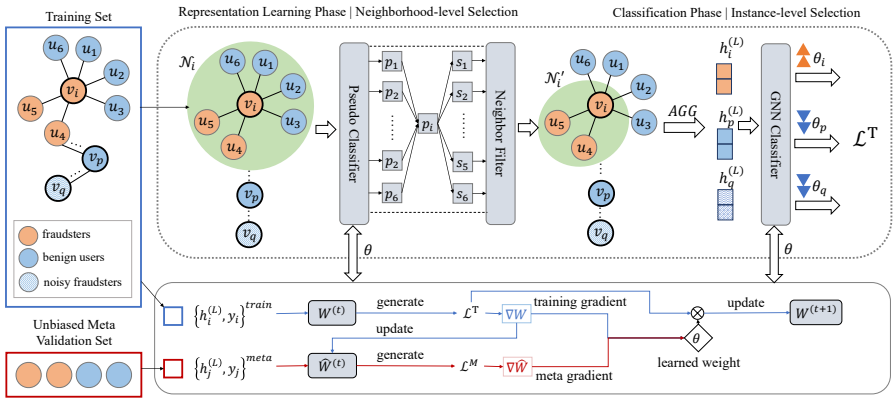

Fig. 1. Illustration of the BLS's workflow.

significant improvement compared to base models and state-of-the-art on two real-world datasets.

2 Methodology

In this section, we propose **Bi**-**L**evel **S**election (BLS), a lightweight meta-gradient-based method that can fit into general GNN structures. BLS addresses the class imbalance and noisy label problems in fraud detection with the following two key strategies: (1) select instance-level valuable nodes in classification phase by assigning weights θ_i, detailed in Section 2.1; (2) select neighborhood-level valuable nodes in representation learning phase by a filtered neighborhood $\mathcal{N}_i^{'}$, detailed in Section 2.2.

2.1 Instance-level Node Selection

To select valuable nodes on instance level, we learn a weight θ_i for each node v_i in the training set by a meta-learning mechanism as soft selection. We denote the training set as $\{v_i, y_i\}_{i=1}^{N}$, the unbiased meta validation set as $\{v_j, y_j\}_{j=1}^{M}$, where $M \ll N$. The prediction of GNN is denoted as $F(h_i^{(L)}, W_f)$, where W_f is the parameters of GNN classifier. The cross-entropy loss function is denoted as $l(\cdot, \cdot)$. Searching for optimal GNN parameters W_f^* and optimal weights θ^* is a nested loops of optimization. To reduce computation cost, following the analysis of [9], we compute θ by one-step gradient approximation. At each iteration step t, the optimizer updates W_f from current parameter $W_f^{(t)}$ with step size α and uniform weights $\theta_i = \frac{1}{n}$ according to training loss on a mini-batch $\{v_i, y_i\}_{i=1}^{n}$:

$$\hat{W_f}^{(t)}(\theta) = W_f^{(t)} - \alpha \cdot \sum_{i=1}^{n} \theta_i \cdot \frac{\partial}{\partial W_f} l(y_i, F(h_i^{(L)}, W_f^{(t)})) \tag{1}$$

Then we slightly perturb the weights θ to evaluate the impact of each training node on the model performance on the meta validation set. We search for the optimal weight θ^* to minimize the meta loss by taking a single gradient step on the meta-validation set:

$$\theta_i \propto -\beta \cdot \frac{1}{M} \sum_{j=1}^{M} \left(\frac{\partial\, l_j(\hat{W}_f)}{\partial\, \hat{W}_f} \Big|_{\hat{W}_f = \hat{W}_f{}^{(t)}} \right)^{\top} \left(\frac{\partial\, l_i(W_f)}{\partial\, W_f} \Big|_{W_f = W_f^{(t)}} \right) \tag{2}$$

where $l_i(W_f)$ is $l(y_i, F(h_i^{(L)}, W_f))$, and $l_j(\hat{W}_f)$ is $l(y_j, F(h_j^{(L)}, \hat{W}_f))$. We take $\theta_i = \max(\theta_i, \frac{\theta_{th}}{n})$ and then batch-normalize θ_i, where $\theta_{th} \in [0, 1]$ is a hyperparameter representing the minimal weight threshold. Then we can compute the weighted cross-entropy (CE) loss $\mathcal{L}_{GNN}$:

$$\mathcal{L}_{GNN} = -\sum_{i=1}^{N} \theta_i \cdot (y_i \log p_i + (1 - y_i) \log(1 - p_i)) \tag{3}$$

2.2 Neighborhood-level Node Selection

To select valuable nodes on neighborhood-level, BLS forms a subset $\mathcal{N}_i^{'} \subseteq \mathcal{N}_i$ for each center node, where $\mathcal{N}_i$ is the original neighborhood. We append a pseudo classifier before the aggregation each GNN layer to inference pseudo labels. Then we filter the neighborhood according to the pseudo label affinity scores. The ℓ-th layer pseudo classifier $G^{(\ell)}$ parameterized by $W_g^{(\ell)}$ takes the representation $h_i^{(\ell-1)}$ of node v_i from the previous layer as input and generates a pseudo label. Then, the pseudo label affinity score between center node v_i and its neighbor $v_j \in \mathcal{N}_i$ is computed by the L-1 distance function:

$$\hat{p}_i^{(\ell)} = G^{(\ell)}(h_i^{(\ell-1)}, W_g^{(\ell)}) \tag{4}$$

$$S_{ij}^{(\ell)} = 1 - \|\hat{p}_i^{(\ell)} - \hat{p}_j^{(\ell)}\|_1 \tag{5}$$

We sort the neighbors by $S_{ij}^{(\ell)}$ in descending order and select top-k neighbors to form the filtered neighborhood $\mathcal{N}_i^{(\ell)}$, $k = \lceil \rho \cdot |\mathcal{N}_i| \rceil$. With the filtered neighborhood $\mathcal{N}_i^{'(\ell)}$ of center node v_i, we apply neighbor aggregation on v_i:

$$h_i^{(\ell)} = \sigma(W^{(\ell)}(h_i^{(\ell-1)} \oplus AGG(\{h_j^{(\ell-1)} | v_j \in \mathcal{N}_i^{'}\}))) \tag{6}$$

The quality of filtered neighborhood highly depends on the accuracy of predicted pseudo labels. Therefore, we adopt a layer-wise direct supervised weighted loss $\mathcal{L}_{PSE}^{(\ell)}$ similar to Eq. (3). The overall loss function can be formulated as the combination of layer-wise pseudo classifier loss and GNN loss:

$$\mathcal{L} = \mathcal{L}_{GNN} + \sum_{\ell=1}^{L-1} \mathcal{L}_{PSE}^{(\ell)} \tag{7}$$

3 Experiments

In this section, we evaluate BLS-enhanced GNNs on two graph-based fraud detection datasets. Specifically, we aim to answer the following research questions: (RQ1.) How much improvement does BLS bring to the base models under imbalanced and noisy circumstances? (RQ2.) Does BLS outperform other imbalanced learning methods on fraud detection tasks? (RQ3.) How do the key components of BLS contribute to the overall fraud detection performance?

3.1 Experimental Setup

Datasets. We adopt two real-world graph-based fraud detection datasets YelpChi [8] and Amazon [7], collected from online platforms Yelp.com and Amazon.com. Reviews (node) with less than 20% helpful vote are considered as fraudulent nodes. The statistics of the two datasets are shown in Table 1, where $|N|$, $|E|$, $|R|$ stand for number of nodes, edges and edge types. PR is the ratio of positive nodes (fraudsters).

Table 1. Statistics of two graph-based fraud detection datasets.

Dataset	$\|N\|$	$\|E\|$	$\|R\|$	**PR**
YelpChi	45,954	3,846,979	3	14.5%
Amazon	11,944	4,398,392	3	6.9%

Baselines and Evaluation Metrics. BLS is a lightweight method that can be applied to various existing GNN architectures. We select three widely-used GNNs (GCN, GraphSAGE and GAT) and their multi-relational extensions (GCN_M, GAT_M and GraphSAGE_M) as base models. We also compare BLS-enhanced GNNs with the state-of-the-art graph-based fraud detection methods: GraphConsis [6], CARE-GNN [2] and PC-GNN [5]. We adopt two widely used metrics AUC score and G-Mean [5] for evaluation.

Experimental Settings. We set node embedding dimension d and hidden layer dimension as 64, L as 2, learning rate of Adam optimizer lr as 0.01, training epochs as 1000, batch size as 1024 for YelpChi dataset and 256 for Amazon dataset. For BLS, we set the preserving proportion ρ to 0.5, the minimal weight threshold ω_{th} to 0.01. The train/valid/test ratio are 40%, 20%, 40%. We use 266 (5%) nodes in the YelpChi training set and 107 (9%) nodes in the Amazon training set as meta validation. We conduct 10 runs on two datasets with all the compared models and report the average value with standard deviation of the performance metrics.

3.2 Overall Evaluation (RQ1)

We evaluate the performance of all compared methods on the graph-based fraud detection task with two datasets. According to the main results shown in Table

2, we have the following observations: (1) By incorporating BLS, all three base models gain significant improvements in terms of AUC and G-Mean. (2) Compared to the baselines, BLS-enhanced GAT_M achieves the best performance on both datasets. By selection under meta guidance, BLS can generate more credible layer-wise pseudo label affinity scores. Thus, the filtering process based on affinity scores is able to reduce the noise in the neighborhood of fraudulent nodes. Then we evaluate the performance of BLS-enhanced GNNs on against

Table 2. Performance comparison on two graph-based fraud detection datasets.

Dataset	YelpChi		Amazon	
Methods	AUC	G-Mean	AUC	G-Mean
GCN	59.02±1.08	55.61±2.96	79.83±1.38	73.38±4.29
GraphSAGE	58.46±3.03	46.90±5.82	81.13±2.93	75.17±5.28
GAT	64.18±1.84	59.53±4.37	88.48±1.36	85.69±4.72
GraphConsis	69.83±3.42	58.57±3.85	87.41±3.34	76.77±4.86
CARE-GNN	78.44±0.69	70.13±2.17	93.14±0.74	85.63±0.71
PC-GNN	79.87±0.14	71.60±1.30	95.86±0.14	90.30±0.44
GCN_M	74.62±1.38	68.72±1.92	92.93±2.04	83.22±3.85
GraphSAGE_M	77.12±2.56	69.15±3.96	93.63±3.17	85.92±4.20
GAT_M	81.73±1.48	75.33±3.52	93.71±1.06	85.82±3.65
BLS+GCN_M	83.28±0.86	76.31±2.74	94.42±1.55	87.41±0.78
BLS+GraphSAGE_M	86.50±0.78	80.02±2.93	94.71±1.33	87.44±2.02
BLS+GAT_M	**89.26±1.04**	**81.82±3.02**	**95.93±0.73**	**90.72±1.64**

noisy-label circumstances. We randomly choose 0% to 50% fraudulent nodes and flip their labels. By label flipping, we simulate unidentified fraudsters which are labeled as benign. Fig. 2 shows the change on AUC scores respecting noisy label ratio. We can observe that BLS-enhanced GNNs always achieve a higher AUC score than the corresponding base models, proving the robustness of BLS toward label noises.

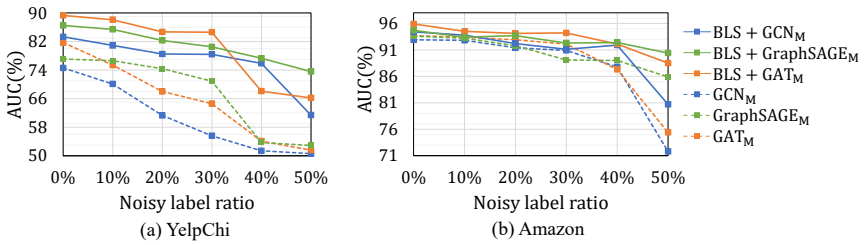

Fig. 2. Performance of BLS-enhanced GNNs w.r.t noisy label.

3.3 Comparison with Imbalanced Learning methods (RQ2)

To further observe the effectiveness of meta selection strategy of BLS, we use two imbalanced learning methods Focal Loss [4] and CB Loss [1] to replace the node-level selection on GAT. We can observe that BLS achieves highest scores on both datasets, demonstrating that BLS is able to filter unidentified fraudsters with meta knowledge and prevent over-fitting on the majority class.

Table 3. Performance comparison of BLS with other imbalanced learning methods

Dataset	YelpChi		Amazon	
Strategy	AUC	G-Mean	AUC	G-Mean
Focal Loss	84.21	77.62	93.84	87.79
CB Loss	86.28	79.09	93.96	88.15
BLS	**89.26**	**81.82**	**95.93**	**90.72**

3.4 Ablation Study (RQ3)

In this subsection, we explore how the two key components in BLS, i.e., instance-level selection and neighborhood-level selection, improve GNN models. We take BLS-enhanced GAT for demonstration, $\text{BLS}_{\setminus N}$ removes neighborhood-level selection, $\text{BLS}_{\setminus I}$ removes instance-level selection, $\text{BLS}_{\setminus NI}$ removes both strategies. As Fig. 3 illustrated, the complete model achieves the best performance on all metrics, removing each component will cause performance dropping except G-mean on Amazon, proving that both components are effective for graph-based fraud detection tasks.

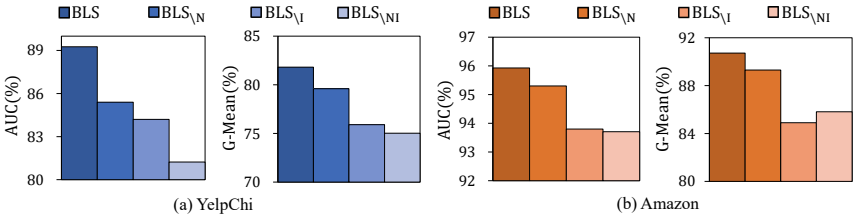

Fig. 3. Ablation study of two key components of BLS on YelpChi and Amazon.

4 Related Work

Graph-based fraud detection focus on analyzing the interactions and connectivity patterns to identify fraudulent activities. GraphConsis [6] and CARE-GNN [2] are GNN-based anti-spam model, tackle the inconsistency problems and camouflages in fraud detection. PC-GNN [5] is designed for imbalanced supervised

learning on graphs. It incorporates a two-step resampling method to reconstruct label-balanced sub-graphs. Meta-GDN [3] uses deviation loss and cross-network meta-learning algorithm for network anomaly detection tasks. Unlike our problem setting, Meta-GDN treats negative nodes as unlabeled nodes and requires extra auxiliary networks.

5 Conclusion

In this work, we propose a lightweight algorithm named Bi-Level Selection (BLS) that can be incorporated into general GNN architectures to handle the class imbalance and noisy label problems usually appeared in graph-based fraud detection tasks. BLS selects valuable nodes on two levels guided by meta gradient of validation loss. Experiments on two benchmark graph-based fraud detection datasets demonstrate the effectiveness of our algorithm.

Acknowledgement

The research work is supported by the National Natural Science Foundation of China under Grant (No.61976204, 92046003, U1811461). Xiang Ao is also supported by the Project of Youth Innovation Promotion Association CAS, Beijing Nova Program Z201100006820062.

References

1. Cui, Y., Jia, M., Lin, T.Y., Song, Y., Belongie, S.: Class-balanced loss based on effective number of samples. In: CVPR. pp. 9268–9277 (2019)
2. Dou, Y., Liu, Z., Sun, L., Deng, Y., Peng, H., Yu, P.S.: Enhancing graph neural network-based fraud detectors against camouflaged fraudsters. In: CIKM. pp. 315–324 (2020)
3. Kaize, D., Qinghai, Z., Hanghang, T., Huan, L.: Few-shot network anomaly detection via cross-network meta-learning. In: WWW. pp. 2448–2456 (2021)
4. Lin, T.Y., Goyal, P., Girshick, R., He, K., Dollár, P.: Focal loss for dense object detection. In: ICCV. pp. 2980–2988 (2017)
5. Liu, Y., Ao, X., Qin, Z., Chi, J., Feng, J., Yang, H., He, Q.: Pick and choose: A gnn-based imbalanced learning approach for fraud detection. In: WWW. pp. 3168–3177 (2021)
6. Liu, Z., Dou, Y., Yu, P.S., Deng, Y., Peng, H.: Alleviating the inconsistency problem of applying graph neural network to fraud detection. In: SIGIR. pp. 1569–1572 (2020)
7. McAuley, J.J., Leskovec, J.: From amateurs to connoisseurs: modeling the evolution of user expertise through online reviews. In: WWW. pp. 897–908 (2013)
8. Rayana, S., Akoglu, L.: Collective opinion spam detection: Bridging review networks and metadata. In: KDD. pp. 985–994 (2015)
9. Ren, M., Zeng, W., Yang, B., Urtasun, R.: Learning to reweight examples for robust deep learning. In: ICML. pp. 4334–4343 (2018)