IDGL: An Imbalanced Disassortative Graph Learning Framework for Fraud Detection

Junhang Wu[1,2], Ruimin Hu[1,2(✉)], Dengshi Li[1,3], Lingfei Ren[1,2], Wenyi Hu[1,2], and Yilong Zang[1,2]

[1] National Engineering Research Center for Multimedia Software, School of Computer Science, Wuhan University, Wuhan 430072, China
hurm1964@gmail.com
[2] Hubei Key Laboratory of Multimedia and Network Communication Engineering, Wuhan University, Wuhan 430072, China
[3] School of Artificial Intelligence, Jianghan University, Wuhan 430056, China

Abstract. The thriving growth of Internet service not only facilitates our daily lives but also incubates various fraudulent activities with concealment. The traceable interactive behaviors forming the graph-like data provide a great opportunity for graph-based fraud detection. Owing to the stellar performance of assortative graph learning, GNN-based fraud detection methods escalate much attention. However, the fraud graph is not always assortative but more likely disassortative as the fraudsters usually camouflage themselves via building numerous connections with normal users. Additionally, the GNN-based fraud detection methods also suffer from graph imbalance issues as the number of fraudsters is far less than that of the normal users. To address these problems, an imbalanced disassortative graph learning framework (IDGL) is proposed with two key components. First, an adaptive dual-channel convolution filter is developed to adaptively combine the advantage of low- and high-frequency signals from its neighbors so as to assimilate the nodes with assortative edges and discriminate the nodes with disassortative edges. Second, a label-aware nodes and edges sampler is designed with the consideration of nodes' popularity and corresponding label class frequency, which helps the model simultaneously eliminate the bias towards the major classes and pay more attention to the valuable connections (fraud-fraud, fraud-benign). Extensive experiments on two public fraud datasets demonstrate the effectiveness of our method.

Keywords: Fraud detection · Graph disassortativity · Graph imbalance · Adaptive frequency filter · Label-aware sampler

J. Troya et al. (Eds.): ICSOC 2022, LNCS 13740, pp. 616–631, 2022.
https://doi.org/10.1007/978-3-031-20984-0_44

1 Introduction

With the thriving growth of Internet services facilitating our daily life, there also brings various kinds of fraudulent behaviors. The fraudsters or attackers disguise as the benign users to do some malicious activities and conceal themselves within the mass of data, which has caused great damage to finance security [10,16,20], cyber security [4] and comment management [3,14]. Fortunately, our online behavior is always traceable no matter whether we are benign or fraudulent, and we can transform these interactive behaviors as graph-like data where the users and their interactions are treated as the nodes and the edges, respectively. Recently, the emerging graph neural network (GNN) has shown its great representation power of graph data, which makes GNN-based fraud detection methods escalate extensive attention.

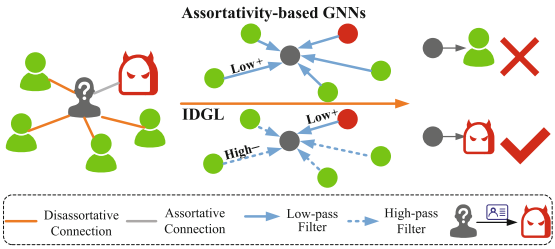


Fig. 1. Illustration of graph disassortativity and imbalance

Although these GNN-based fraud detection methods have made much progress, there still exist the following two main challenges.

Graph Disassortativity. Generally, GNNs update the representation of nodes by aggregating the signals from their neighbors, which can be treated as a low-pass filter to retain the commonality between the connected nodes [9,18]. Benefiting from the smoothness of the low-frequency filter, it works well for assortative graphs, i.e., similar nodes tend to make the connections [1], which makes GNN-based fraud or anomaly detection effective as it assumes that the fraudulent or abnormal nodes with the same malicious goals tend to make the connections with each other. However, some studies [5,7,14,19] have shown that the fraud graphs are not always assortative but more likely disassortative as the fraudsters often camouflage themselves by making many connections with the benign users to make them look normal with less suspiciousness, which makes the fraud graph flood with numerous disassortative connections, i.e., the entities from different classes tend to make the connections. Consequently, the low-pass smoothing aggregation mechanisms of GNNs are insufficient to support the inference for these disassortative graphs as it enables the fraudsters to achieve their intentions, i.e., the fraudulent features are concealed within the myriads of benign ones. As shown in Fig. 1, given a query user who has far more connections with the

normal user than the fraudster, the assortativity-based GNNs tend to classify him/her into the normal user as its behavior features have been concealed by his numerous benign neighbors within the low-pass smoothing aggregation. Under this circumstance, the high-frequency signals (i.e., the difference between entity nodes) are more suitable for these disassortative graphs.

Graph Imbalance. First, the number of fraudsters is generally far less than that of the benign users, which causes the graph node imbalance issues. According to Amazon and YelpChi datasets, only 9.50% and 14.53% of labeled entities are the fraudulent ones respectively, as introduced in Table 1. The graph node imbalance may make the training bias towards the majority class (i.e., the benign users) with the reduction of model generalization ability. Expect for node imbalance, the edge imbalance is more urgent as it directly guides the node aggregation process. There are three types of edges in the fraud graph: edges between normal entity nodes (denoted as N-N), edges between the fraudulent ones (denoted as F-F), and the edges between the fraudulent and normal ones (denoted as F-N). As introduced in Table 2, the number of N-N edges is far more than that of others, which helps the normal user detection by retaining commonalities between them. But for F-F and F-N edges, they are rare but valuable, and we should pay more attention to them with exploring the commonalities between fraudsters and the difference between fraudsters and normal users. However, most current GNN-based fraud detection studies pay little attention to both nodes and edges imbalance and haven't made full use of labeled F-F and F-N edges to detect the new fraudsters. Unfortunately, the graph imbalance issues further exacerbate the disassortativity of the graph with more difficulties for discrimination.

To address the above challenges, we propose an Imbalanced Disassortative Graph Learning framework (IDGL) to simultaneously adaptively aggregate low- and high-frequency signals from assortative and disassortative connections on the imbalanced fraud graph. Specifically, IDGL is composed of four module layers: 1) a re-embedding layer. Some recent studies [2] have emphasized that the performance and robustness of the model may be hurt by the entanglement of graph filters and parameter matrices, and the fraudsters usually camouflage themselves with the similar raw features to the normal users. Therefore, a non-linear re-embedding layer is applied to relearn the representations of nodes; 2) an adaptive dual-channel convolution layer, which is used to adaptively combine the advantage of dual-channel (i,e., the low- and high-frequency) signals from its neighbors to assimilate the nodes with assortative edges and discriminate the nodes with disassortative edges; 3) a representation fusion layer, which combines the intermediate embeddings to be the final representation of nodes; 4) an imbalanced-oriented classification layer. To alleviate the effects of graph imbalance, a label-aware nodes and edges sampler is designed with the consideration of nodes' popularity (i.e., degree) and corresponding label class frequency. Sampled nodes are used for classification training to eliminate the bias towards the major classes, and the sampled edges are treated as the supervision information to facilitate the training of adaptive filters and make the model pay more

attention to the valuable edges (i.e., F-F and F-N layers). The contributions of the paper can be listed as follows:

- We formulate the graph-based fraud detection problem as an imbalanced disassortative node classification task and propose an imbalanced disassortative graph learning framework to deal with the disassortativity and graph imbalance issues on the graph.
- An adaptive dual-channel convolution filter is further developed for fraud detection to assimilate the nodes with assortative edges and discriminate the nodes with disassortative edges. A label-aware node and edge sampler is proposed to relieve graph imbalance issues with more attention to the valuable edge information.
- Experiments on two public real-world datasets demonstrate the effectiveness of our proposed IDGL for fraud detection.

2 Preliminaries

2.1 Definition

Definition 1 (Assortativity and disassortativity). Given a graph, if two nodes (e.g., v_i and v_j), which make the connection as an edge, belong to different classes, then we treat the connection as a disassortative edge, denoted as ε_{ij}^{-}, and if they belong to the same class, the connection is an assortative edge, denoted as ε_{ij}^{+}. The larger disassortativity of the graph, the more nodes from different classes tend to connect with each other, and vice versa. For our task, the fraud graph is of both assortativity and disassortativity at the same time.

Definition 2 (Graph). Consider a graph $G = \{V, X, \{\varepsilon^+, \varepsilon^-\}, A, Y\}$, $V = \{v_1, v_2, ..., v_N\}$ is the set of nodes, N is the number of nodes; $X \in \mathbb{R}^{N \times d}$ is the original d-dimension feature vector of all of N nodes; For $\{\varepsilon^+, \varepsilon^-\}$, ε^+ and ε^- represent the assortative and disassortative edge sets respectively where $\varepsilon^+ \cup \varepsilon^- = \varepsilon$ and $\varepsilon^+ \cap \varepsilon^- = \emptyset$; A is the corresponding adjacency matrice of the graph where $A_{ij} = 1$, if $e_{ij} \in \varepsilon$; Y is the set of labels for the nodes, and each one has a label $y_i \in \{0, 1\}$ where 1 represents the *fraudster* and 0 represents the *benign*.

Definition 3 (Multi-relation Graph). There are different relations among the nodes, and $G = \left\{V, X, \{\varepsilon_r^+, \varepsilon_r^-\}\big|_{r=1}^{R}, \{A_1, A_2, ..., A_R\}, Y\right\}$ is defined as a multi-relation graph, where $e_{ij}^r \in \{\varepsilon_r^+, \varepsilon_r^-\}$ represents the edge between the node v_i and v_j under the relation $r \in \{1, 2, ..., R\}$ and A_r is the corresponding adjacency matrix.

2.2 Problem Formulation

Definition 4 (Graph-based Fraud Detection). Considering a multi-relation graph G, which has been defined in definition 2, the task is to detect the fraud nodes from the benign ones in the given graph. Specifically, given

the structural information of the graph $\{A_1, A_2, ..., A_R\}$ and the original feature information X, we need to learn the function f to map the nodes into a d-dimension feature vector $z_i \in \mathbb{R}^d$ across the multi-relation graph where $d \ll N$. With the learned embedding and the labeled nodes, a classifier is trained to detect whether a given unlabeled node is a fraudster.

3 Overview Framework of Method

In this section, we present the proposed IDGL framework, as shown in Fig. 2. IDGL includes four module layers: the re-embedding embedding layer, adaptive dual-channel convolution layer, representation fusion layer, and the imbalance-oriented classification layer. For the first module layer, it is used to add some uncertainty by the dense and dropout layer to relieve the effect of feature camouflage. Furthermore, the second module layer is introduced to make full use of low- and high-pass signals to deal with the graph disassortativity issue. The third module layer makes the fusion of the intermediate representation to be the final features of nodes. The final module layer is used to deal with the graph imbalance and detect the fraudsters.

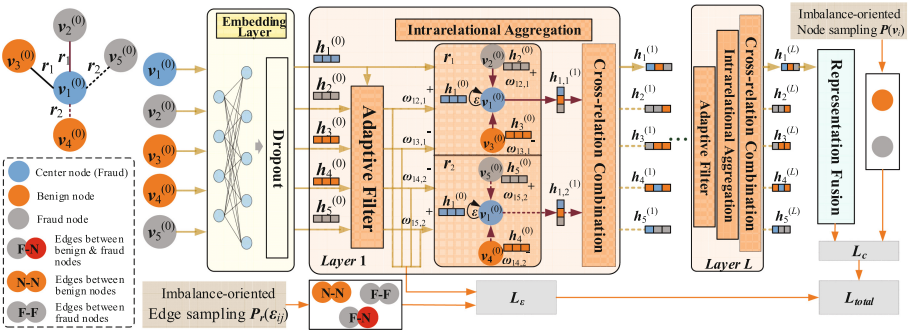


Fig. 2. An illustration of proposed framework of IDGL

3.1 Re-embedding Layer

Fraudsters usually learn the normal users to camouflage themselves, so the original attribute features between the fraudsters and the normal users are of great similarity. Therefore, it is highly desirable to add some uncertainty into the original features to re-learn the feature similarity between nodes to cater downstream fraudster detection tasks. In the paper, a dense-based dropout layer is introduced to encode the embeddings of the nodes without depending on the network topology, and it is denoted as follows:

$$h_i^{(0)} = \sigma\left(dropout\left(x_i, \eta\right) \cdot \mathbf{W}_0\right), \tag{1}$$

where x_i is the original feature vector of node v_i, $dropout(\cdot, \eta)$ is the function which drops the neurons from the network with a certain probability η during network training, $\mathbf{W}_0$ is a learnable weight matrix, $h_i^{(0)}$ represents the embedding of node v_i, and $\sigma(\cdot)$ is the non-linear **ReLU** activation function. Based on the above components, the difference between the fraudsters and the normal users will be further amplified via network topology based on the improved convolution and aggregation strategies.

3.2 Adaptive Dual-channel Convolution Layer

Dual-channel Graph Convolution. Following GCN [8], from the perspective of graph signal processing, the graph convolution $*_G$ between the given signal x and filter f are denoted as:

$$f *_G x \approx \theta \left(I_N + D^{-\frac{1}{2}} A D^{-\frac{1}{2}} \right) x, \tag{2}$$

where $I_N \in \mathbb{R}^{N \times N}$ denotes the identity matrix and $D = diag\{d_1, d_2, ..., d_N\}$ is the diagonal degree matrix ($d_i = \sum_j A_{ij}$). From Eq. 2, we can observe that $I_N + D^{-\frac{1}{2}} A D^{-\frac{1}{2}}$ can be considered to smooth the central node by its adjacent nodes' information via summation between signals, and it can be treated as a low-pass filter to capture the commonalities between the central node and its adjacent nodes [1]. Naturally, for a high-pass graph filter, it should be used to capture the difference between them. Heuristically, a low-pass filter f_L and a high-pass filter f_H are designed as follows:

$$\begin{aligned} f_L *_G x &\approx \theta \left(\varepsilon I_N + \hat{D}^{-\frac{1}{2}} A \hat{D}^{-\frac{1}{2}} \right) x \\ f_H *_G x &\approx \theta \left(\varepsilon I_N - \hat{D}^{-\frac{1}{2}} A \hat{D}^{-\frac{1}{2}} \right) x, \end{aligned} \tag{3}$$

where $\hat{D} = I + D$, and $\varepsilon \in [0, 1]$ is a scaling factor. We can generalize Eq. 3 to the signal $X \in \mathbb{R}^{N \times d}$ (i.e., a d-dimension representation for each node) as follows:

$$\begin{aligned} Z_L &= (\varepsilon I_N + \hat{D}^{-\frac{1}{2}} A \hat{D}^{-\frac{1}{2}}) X \Theta \\ Z_H &= (\varepsilon I_N - \hat{D}^{-\frac{1}{2}} A \hat{D}^{-\frac{1}{2}}) X \Theta, \end{aligned} \tag{4}$$

where $\Theta \in \mathbb{R}^{d \times M}$ is the matrix of learnable filter parameter, and $Z_L \in \mathbb{R}^{N \times M}$ and $Z_H \in \mathbb{R}^{N \times M}$ are the signal matrix convolved by low-pass and high-pass filters, respectively. Then the low-pass and high-pass convolution of the node i can be denoted as:

$$\begin{aligned} (F_L X)_i &= \left(\varepsilon X_i + \sum_{j \in N(i)} \frac{1}{\sqrt{\hat{D}_{ii} \hat{D}_{jj}}} X_j \right) \Theta \\ (F_H X)_i &= \left(\varepsilon X_i - \sum_{j \in N(i)} \frac{1}{\sqrt{\hat{D}_{ii} \hat{D}_{jj}}} X_j \right) \Theta, \end{aligned} \tag{5}$$

where $N(i)$ is the set of one-hop neighborhoods of node i.

Learnable Channel Fusion Aggregation. Given the above low- and high-pass graph filters, the next step is how to aggregate both low- and high-frequency information from the node's neighbors, respectively. Naturally, a basic idea is to add the weight parameter $\alpha_{ij,r}$ to balance the importance between such two filters under the relation r:

$$\tilde{h}_{i,r} = \alpha_{ij,r}(F_L H)_{i,r} + (1 - \alpha_{ij,r})\,(F_H H)_{i,r}, \tag{6}$$

where $\alpha_{ij,r} \in [0, 1]$, and Eq. 6 can be further expanded as follows:

$$\tilde{h}_{i,r} = \left(\varepsilon h_{i,r} + \sum_{j \in N_r(i)} \frac{2\alpha_{ij,r} - 1}{\sqrt{\hat{D}_{ii,r}\hat{D}_{jj,r}}} h_{j,r} \right) \Theta_r, \tag{7}$$

where $\tilde{h}_{i,r}$ is the aggregated embedding of node v_i under the relation r. Here, we set a learnable coefficient $\omega_{ij,r} = 2\alpha_{ij,r} - 1$, where $\omega_{ij,r} \in [-1, 1]$. For $\omega_{ij,r}$, it decides whether a low or high-frequency signal should be extracted between the node v_i and v_j, and thus the features of both the node itself v_i and its neighbor v_j should be considered together. Naturally, a shared self-gating mechanism is used to learn $\omega_{ij,r}$ as follows:

$$\omega_{ij,r}^{(l)} = \tanh\left(\left(\mathbf{W}_r^{(l)} \right)^T \left[h_i^{(l-1)} || h_j^{(l-1)} \right] \right), \tag{8}$$

where $h_i^{(l-1)} \in \mathbb{R}^{d_{v_i} \times 1}$ and $h_j^{(l-1)} \in \mathbb{R}^{d_{v_j} \times 1}$ are the representations of the nodes v_i and v_j at l-th layer, $\mathbf{W}_r^{(l)} \in \mathbb{R}^{\left(d_{v_i} + d_{v_j}\right) \times 1}$ is a trainable matrix, $||$ is the concatenation operation, and $\tanh(\cdot)$ is the *hyperbolic tangent* function, which makes the value ω_{ij} in the range of $(0, 1)$. Finally, the aggregation of node i can be denoted as follows:

$$h_{i,r}^{(l)} = \sigma\left(\left(\varepsilon h_i^{(l-1)} + \sum_{j \in N_r(i)} \frac{\omega_{ij,r}^{(l)}}{\sqrt{\hat{D}_{ii,r}\hat{D}_{jj,r}}} h_j^{(l-1)} \right) \Theta_r^{(l-1)} \right). \tag{9}$$

Layer Architecture and Cross-relation Combination. In the previous sections, we have introduced the message passing paradigm of our method. Here, we formally define the convolution layer of our method under the r-th relation, and the mathematical formulation is denoted as follows:

$$\begin{aligned}
&h_i^{(0)} = \sigma\left(dropout\left(x_i, \eta\right) \cdot \mathbf{W}_0\right) \\
&... \\
&\omega_{ij,r}^{(l)} = \tanh\left(\left(\mathbf{W}_r^{(l)} \right)^T \left[h_i^{(l-1)} || h_j^{(l-1)} \right] \right) \\
&h_{i,r}^{(l)} = \sigma\left(\left(\varepsilon h_i^{(l-1)} + \sum_{j \in N_r(i)} \frac{\omega_{ij,r}^{(l)}}{\sqrt{\hat{D}_{ii,r}\hat{D}_{jj,r}}} h_j^{(l-1)} \right) \Theta_r^{(l-1)} \right) \\
&h_i^{(l)} \leftarrow [h_{i,1}^{(l)}, h_{i,2}^{(l)}, ..., h_{i,R}^{(l)}] \\
&...
\end{aligned} \tag{10}$$

In Eq. 10, we can observe that $h_{i,r}^{(l)}, r \in [1, 2, ..., R]$ (i.e., the signal of node i learned under the different relation at the l-th layer) is concatenated as a new signal $h_i^{(l)}$, and it will be the input of node feature for the next layer.

3.3 Representation Fusion Layer

In our model, the node embedding outputted by different neural network layers has different smoothness and sharpness. In the model, the designed low- and high-pass filters make the node embedding outputted by different convolution layers have different smoothness and sharpness, and they can help facilitate the downstream classification task. Thus, we combine the intermediate embeddings outputted by the different layers as the final representation of the node:

$$z_i = [x_i, h_i^{(0)}, h_i^{(1)}, ..., h_i^{(L)}], \tag{11}$$

where L is the number of convolution layer.

3.4 Imbalance-Oriented Classification Layer

As introduced in Eq. 9, for each graph neural layer l under the relation r, $\alpha_{ij,r}^{(l)}$ can be directly calculated by the learnable weight $\omega_{ij,r}^{(l)}$ as: $\alpha_{ij,r}^{(l)} = 0.5\left(\omega_{ij,r}^{(l)} + 1\right) \in [0, 1]$, and it can be used to measure the assortativity and disassortativity of the edge e_{ij}, so as to be the supervised information to balance the weight between low- and high-pass filters. Actually, the supervision signal from the known label nodes can be treated as the ground truth to make an auxiliary loss:

$$\mathcal{L}_r^{(l)} = -\sum_{e_{ij,r} \in \varepsilon_{t,r}} \left[y_{ij,r} \log\left(\alpha_{ij,r}^{(l)}\right) + (1 - y_{ij,r}) \log\left(1 - \alpha_{ij,r}^{(l)}\right) \right], \tag{12}$$

where $y_{ij,r} \in \{0, 1\}$ is the label of the assortative edge (i.e., $y_{ij,r} = 1$) or the disassortative edge (i.e., $y_{ij,r} = 0$) under the relation r, and $\varepsilon_{t,r}$ is the edge set whose source nodes and target nodes have been labeled under the relation r. For each layer and each relation, the final loss for assortative and disassortative edges can be formulated as follows:

$$\mathcal{L}_\varepsilon = \frac{1}{L \times R} \sum \mathcal{L}_{\varepsilon_{t,r}}^{(l)}, \tag{13}$$

where L and R are the number of layers and relations, respectively.

Additionally, given the final embedding of nodes z_i, the fraud detection problem can be treated as a binary node classification problem, and we use cross-entropy loss function to model it:

$$\begin{aligned} \mathcal{L}_c &= -\sum_{i \in V} \left[y_i \log\left(p_i\right) + (1 - y_i) \log\left(1 - p_i\right) \right] \\ p_i &= softmax\left(MLP\left(z_i\right)\right). \end{aligned} \tag{14}$$

To sum up, we define the overall loss of our method as follows:

$$\mathcal{L}_{total} = \gamma_c \mathcal{L}_c + \gamma_\varepsilon \mathcal{L}_\varepsilon + \gamma \|\Theta\|^2, \tag{15}$$

where γ_c, and γ_ε ($\gamma_c + \gamma_\varepsilon = 1$) are the weights to balance the importance of different losses, $\|\Theta\|^2$ is the regularization term to avoid over-fitting and γ is the control coefficient. Note that, to alleviate the influence of sample imbalance problem (i.e., the number of normal users is significantly larger than that of the fraudsters), a label-aware sampler is proposed to take the nodes' label frequency and degree information into consideration, which make the minority class of relatively high sampling probability. First, as to the node sampling for classification, the sampling probability is denoted as follows:

$$P(v_i) \propto \frac{\sqrt{d_i}}{Z(\mathcal{C}(v_i))}, \tag{16}$$

where $d_i = \sum_{r=1}^{R} \sum_j A_{ij,r}$ is the degree of node v_i under all relations, and $Z(\mathcal{C}(v_i))$ represents the label frequency of class $\mathcal{C}(v_i)$. Note that, $\sqrt{d_i}$ means that more "popular" nodes are more likely to selected, and $Z(C(v_i))$ means the more "rare" nodes are more likely to be selected.

Table 1. Datasets statistic information

YelpChi					Amazon				
#nodes (Fraudster%)	Relation type	Relations	Class	#Class	#nodes (Fraudster%)	Relation type	Relations	Class	#Class
45954 (14.53%)	R-U-R	49315	1	6677	11944 (10.5%)	U-P-U	175608	1	821
	R-T-R	573616	0	39277		U-S-U	3566479	0	7818
	R-S-R	3402743	–	0		U-V-U	1036737	–	3305
	ALL	3846979				ALL	4398392		

[1] For Class: 1: spam or fraudulent; 0: legitimate or benign; -: unlabeled.

The set of the sampled nodes is denoted as V_s. Next, for the edge sampling under the relation r, the sampling probability is defined as follows:

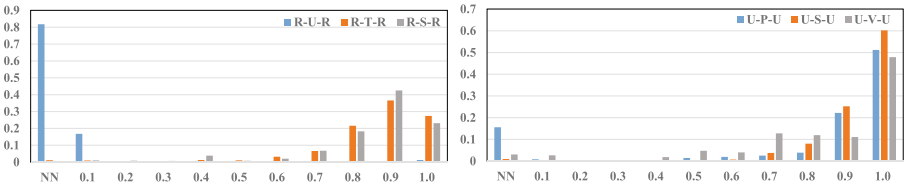
$$P(\varepsilon_{ij,r}) \propto \frac{\sqrt{d_{i,r} d_{j,r}}}{Z(\mathcal{C}(\varepsilon_{ij,r}))}, \tag{17}$$

where $d_{i,r} = \sum_k A_{ik,r}$ is the degree of node v_i under the relation r, and $Z(\mathcal{C}(\varepsilon_{ij,r}))$ is the edge label (i.e., the assortative or disassortative edge) frequency of class $\mathcal{C}(\varepsilon_{ij,r})$. The sets of the sampled edge under all relations are marked as: $\{\varepsilon_{s,r}\}\big|_{r=1}^{R}$. Similarly, $\sqrt{d_{i,r} d_{j,r}}$ and $Z(C(\varepsilon_{ij,r}))$ represents the popularity and rareness of the edge e_{ij} under the relation r. For the edges between fraudsters (F-F) and the edges between the fraudsters and the normal users (F-N), they are rare but valuable. Thus, $P(\varepsilon_{ij,r})$ can make F-N and F-F edges be selected at a higher probability.

4 Experiments

4.1 Experiment Setup

Datasets. Two public real-world fraud detection datasets (i.e., Yelp review dataset and Amazon dataset[13]) are used to validate the performance of IDGL. **YelpChi** dataset collects the reviews of hotels and restaurants on the Yelp platform, and the reviews are treated as the node with three relations: 1) R-U-R represents the reviews, which are provided by the same user, are linked; 2) R-T-R represents the reviews, which are given to the same product within the same month; 3)R-S-R represents the reviews, which are given to the same product with the same star-rating, are linked. The nodes are labeled by Yelp's filter (spam) and recommendation (legitimate). For **Amazon**, it is composed of users with their comments on the musical instruments. Here, users are treated as the node with three different types of relations: 1) U-P-U represents the users, who make the comments on at least one same product, are linked; 2) U-S-U represents the users, who give at least one same star-rating within a same week, are linked; 3) U-V-U represents the users, who have top-5% mutual review TF-IDF similarities, are linked. Note that the user is labeled the normal user or the fraudster according to more than 80% or less than 20% helpful votes. The statistics of such two datasets are shown in Tab. 1.



(a) YelpChi (b) Amazon

Fig. 3. Disassortativity Evidence. X-axis is the proportion of disassortative edges in the neighborhood of fraud nodes. Y-axis is the proportion of the corresponding fraudulent nodes among all fraudsters. NN is the fraudulent node without any neighbor.

Baselines. In this section, several state-of-the-art GNN-based methods and their variants are compared to verify the effectiveness of our proposed method. The source code of our model is available.[1]

GCN [8] is a general GNN model which aggregates the embedding information of node's first-order neighours. GAT [15] is an attention-based graph neural network which takes the attention mechanism into the process of aggregation. GraphSAGE [6] is an inductive graph neural network which takes the multi-order node sampling strategy into node aggregation. GEM [12] is an improved graph neural network for malicious accounts detection which constructs the heterogeneous account-device graphs by summarizing the weakness of the attackers.

[1] https://github.com/Shzuwu/IDGL.

FdGars [17] is a GCN-based method for fraud detection which reconstruct a relational graph between the fraudsters and the normal users based on multi-context information. GraphConsis [11] is a heterogeneous GNN-based method that aims to address the inconsistency problems of context, feature, and relation. CARE-GNN [3] is a GNN-based method that investigates the camouflage behavior of fraudsters and their negative influence on GNN-based fraudster detectors, and proposes a label-aware similarity measure and a similarity-aware selector. FRAUDRE [19] is an improved GNN method that focuses on the graph inconsistency and imbalance issues of the camouflaged fraudsters.

Note that we perform GCN, GAT, GraphSAGE, and FdGars on the graphs characterized by Definition 2 (i.e., ALL in Table 1), and perform the other methods on multi-relation graphs characterized by Definition 3.

Evaluation Metrics. Since YelpChi and Amazon datasets have imbalanced classes, following previous work, we use AUC, Macro-Recall and Macro-F1 for performance evaluation. As a widely used binary classification metric, AUC is computed based on the relative ranking of prediction probabilities of all samples, and it could eliminate the influence of class imbalance.

4.2 Evidence of Graph Disassortativity and Imbalance

First, we investigate the existence of graph disassortativity. Specifically, we calculate the proportion of disassortative edges to all one-hop neighbors of each fraudulent node under different relation subgraphs, and further count the proportion changes of these fraudulent nodes among all of them with the growth of disassortativity proportion, as shown in Fig. 3. We can observe that there are numerous fraudulent nodes with high disassortativity in such subgraphs, and more than half-past of fraudsters are with larger than 80% disassortativity proportion. Consequently, the fraud graph is of disassortativity naturally, and we need to take the separation of assortativity and disassortativity into consideration.

Table 2. Edge type statistic information

Edge	YelpChi			Amazon		
	R-U-R	R-T-R	R-S-R	U-P-U	U-S-U	U-V-U
N-N	48,261	420,783	2,539,220	112,330	2,670,843	665,149
F-F	878	14,781	88,406	3,397	4,474	925
F-N	176	138,052	775,117	31,655	124,232	26,970

[1] N-N: edges between normal users; F-F: edges between fraudsters; F-N: edges between fraudsters and normal users.

Next, we further study the imbalance of the fraud graph, and we find that the imbalance of nodes and edges is widespread. Specifically, for node imbalance, we have counted it in Table 1, and we can observe that only 14.53% and 9.50% of labeled nodes are fraudsters, which makes the model dominated by the majority

class (i.e., the benign users) with the reduction of model generalization ability. Furthermore, we also investigate the edge imbalance of fraud graphs. Specifically, we first classify the edge type into three classes, namely N-N(edges between normal users), F-F(edges between fraudsters), and F-N(edges between the normal users and fraudsters), and then count their numbers under the different relational subgraphs, as shown in Table 2. We can observe that the number of N-N edges is of maximum quantity, which may make the node aggregation dominated by N-N edges and weaken the ability to model differences (between fraudsters and normal users) and commonalities (between the fraudsters) while they are the keys to fraud detection. Consequently, the imbalance of nodes and edges is widespread in fraud detection, and we need to design an imbalanced-oriented nodes and edges sampling strategy to eliminate bias toward the majority classes.

4.3 Performance Comparison

In this section, we compare our proposed method IDGL with the state-of-the-art methods on both Amazon and YelpChi datasets. Note that we also perform the metrics (i.e., AUC, Macro-Recall, and Macro-F1) under different ratios (from 10% to 40%) of both datasets, as shown in Table 3, and we have the following observations.

First, compare the methods which are performed on the single graph (i.e., GCN, GAT, GraphSage, and FdGars) with the ones which are performed on the multi-graph (i.e., GME, GraphConsis, CARE-GNN, FRAUDRE, and IDGL), the latter is mostly better than formers, expect for GME, which brings two aspects of inspiration. On the one hand, the multi-relation graph contains richer

Table 3. Performance under various ratios of Amazon and YelpChi training sets. Recall and F1 are the abbreviation of Macro-Recall and Macro-F1.

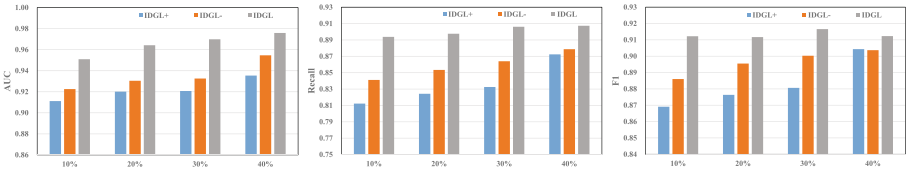
Data	Method	10%			20%			30%			40%		
		AUC	Recall	F1	AUC	Recall	F1	AUC	Recall	F1	AUC	Recall	F1
Amazon	GCN	77.26	50.00	47.51	77.42	50.00	47.51	76.99	50.00	47.51	77.94	50.00	47.51
	GAT	76.96	50.00	47.50	76.99	50.00	47.51	76.61	50.00	47.50	77.35	50.00	47.50
	GraphSage	69.87	50.00	47.50	71.75	50.00	47.50	72.51	50.00	47.51	71.49	50.00	47.50
	GME	70.24	69.56	75.52	72.05	71.55	75.53	73.99	72.12	68.48	74.44	70.66	74.42
	FdGars	81.10	73.41	55.32	81.19	73.47	55.31	80.91	72.90	55.01	80.82	72.82	55.14
	GraphConsis	82.67	82.63	75.97	84.22	84.21	81.93	84.46	84.37	79.06	85.15	85.10	77.98
	CARE-GNN	88.16	88.19	88.21	88.25	87.95	85.80	87.41	84.89	75.70	87.36	83.90	88.36
	FRAUDER	90.37	89.12	91.02	88.99	88.71	90.67	91.51	88.01	91.11	88.18	88.61	91.10
	IDGL	**95.09**	**89.37**	**91.22**	**96.42**	**89.76**	**91.17**	**96.98**	**90.61**	**91.65**	**97.58**	**90.73**	**91.23**
YelpChi	GCN	52.12	50.00	46.08	53.88	50.00	46.08	52.62	50.00	46.08	53.12	50.00	46.08
	GAT	50.14	50.00	46.08	49.94	50.00	46.08	49.97	50.00	46.08	49.67	50.00	46.08
	GraphSage	52.94	50.00	46.08	55.39	50.00	46.08	56.08	50.00	46.10	56.45	50.00	46.10
	GME	64.35	50.00	46.08	64.28	51.33	48.89	69.63	51.04	48.24	70.88	50.38	46.87
	FdGars	47.36	49.19	48.76	47.54	49.40	48.93	47.71	49.52	49.02	47.91	49.42	48.93
	GraphConsis	64.12	64.72	61.3	63.89	64.46	62.93	60.94	61.44	62.73	61.02	61.67	63.03
	CARE-GNN	69.73	65.68	52.86	70.47	66.94	57.55	72.42	67.32	57.39	70.99	66.80	56.47
	FRAUDER	72.21	66.44	55.34	72.51	67.30	58.22	73.72	67.91	59.24	72.22	66.98	59.26
	IDGL	**85.38**	**74.50**	**70.23**	**88.65**	**78.65**	**72.84**	**90.04**	**80.06**	**74.36**	**91.14**	**82.36**	**76.37**

information than the single one, which may provide the chance to make performance improvements. On the other hand, the richer data means a more complex relationship, which means that it is unworkable to directly apply GNNs to fraud detection under the multi-relation graph, and we need to deal with the relationship between the node and its neighbors more carefully. GraphConsis, CARE-GNN, and FRAUDRE have achieved the promising performance by introducing similarity measure and fraud-aware module into the node aggregation process, and IDGL outperforms all other SOTA methods via the learnable high- and low-filter to adaptively learn the difference and similarity commonalities between the node and its neighbors to facilitate the target task.

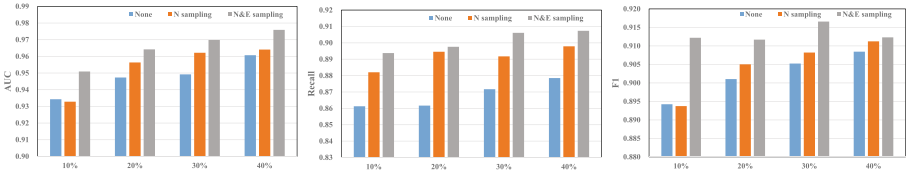
Second, it has been introduced in Table 1 that node imbalance is widespread in fraud detection. For GNN-based fraud detection methods, CARE-GNN and FRAUDRE take the influence of node imbalance into consideration to eliminate the training bias towards the majority class (i.e., the normal users), and achieve better performance than other methods. However, they haven't taken edge imbalance into consideration. As we discussed in Sect. 4.2, we categorize the edge type into three classes: N-N, F-F, and F-N. For F-N, we can treat it as a guide to learn the difference between the fraudsters and the normal users, and F-F is rare but valuable for us to learn the commonalities between the fraudsters, which helps better fraud detection. Thus, an edge sampling method is proposed to make the model pay more attention to the edge of F-N and F-F. Consequently, IDGL achieves better performance than CARE-GNN and FRAUDRE by taking both node and edge sampling into consideration.

4.4 Ablation Analysis

High- and Low-Filters. To demonstrate the effectiveness of the adaptive filter, we conduct the ablation study on the Amazon dataset by ranging the percentage of the training dataset from 10% to 40%, as shown in Fig. 4 (a), and a similar



(a) The Effect of Learnable Adaptive Filter



(b) The Effect of Nodes and Edges Sampling

Fig. 4. Ablation Analysis of Learnable Adaptive Filter and Nodes/Edges Sampling on Amazon dataset with AUC, Recall and F1.

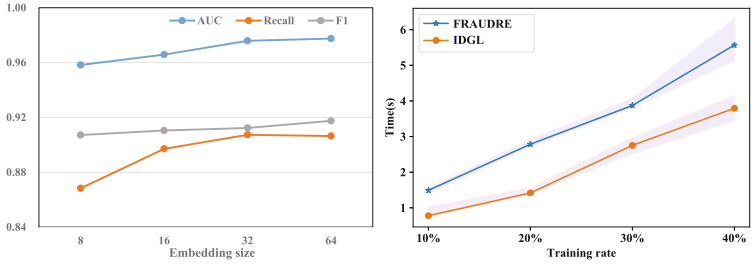
result can also be observed on YelpChi dataset. Specifically, we construct two ablation models by replacing the learnable adaptive high- and low-filer with the only low-filer (i.e., $\alpha_{ij,r} = 1$, denoted as IDGL+) and the only high-filter (i.e., $\alpha_{ij,r} = 0$, denoted as IDGL−), respectively. We can observe that both IDGL+ and IDGL− can be applied to fraud detection with competitive performance. Additionally, IDGL− have better performance than IDGL+, because the fraud graph is of great disassortativity as proved above, and IDGL+ only aggregates the low-frequency signals from its neighbors without considering the difference between classes which makes itself submerged with lack of discrimination. IDGL outperforms such two methods by combining the advantages of low- and high-filters adaptively.

Node and Edge Sampling. To demonstrate the effectiveness of node and edge sampling, we construct two ablation models by removing either edge sampling or both of them on the Amazon dataset with ranging the percentage of training dataset from 10% to 40%, as shown in Fig. 4 (b), and the similar result can also be observed on YelpChi dataset. We can observe that compared with the ablation model without any imbalance-oriented sampling, the one with node sampling performs better as it can mitigate the imbalance issue to some extent. By further introducing edge sampling to make the model pay more attention to the valuable edge types (i.e., F-F and F-N), the performance has been further improved, which shows the effectiveness of node and edge sampling.

4.5 Parameter Sensitivity and Running Efficiency

In this section, we investigate the sensitivity and running efficiency.

First, with 40% of the Amazon dataset as the training set, we vary the value of embedding dimensionality in the range of [8,64], and the result is depicted in Fig. 5(a). We can observe that it first makes a slight improvement with embedding size increasing, and it becomes stable after 32. Considering a larger embedding dimensionality requires higher computational complexity, we finally set d as 32 to make the balance between performance and complexity.



(a) Impact of Embedding Size (b) Running Efficiency

Fig. 5. Performance of IDGL with varing embedding size and running efficiency on Amazon dataset.

Second, to investigate the running efficiency of IDGL, with different percentages of the training set on the Amazon dataset, we compare the average training time of each epoch with FRAUDRE, which has the most competitive performance among all the baseline methods. To be fair, we set the number of convolution layers to 2, the embedding size to 32, and the batch size to 256 for both of them, and the result is depicted in Fig. 5 (b). We can observe that IDGL runs faster than FRAUDER with more time efficiency.

5 Conclusion

In the paper, we propose an imbalanced disassortative graph learning framework called IDGL to solve the graph disassortativity and imbalance issues. To tackle the graph disassortativity, an adaptive dual-channel convolution filter is further developed to adaptively combine the advantage of dual-channel (i,e., the low- and high-frequency) signals from its neighbors, which helps assimilate the nodes with assortative edges and discriminate the nodes with disassortative edges. For graph imbalance issues, a label-aware nodes sampler and edges sampler are designed with the consideration of nodes' popularity and corresponding label class frequency, which helps the model simultaneously eliminate the bias towards the major classes and pay more attention to the valuable edges (i.e., F-F and F-N). Extensive experiments on two public fraud datasets demonstrate the effectiveness of our method.

Acknowledgments. We first gratefully acknowledge anonymous reviewers who read this draft and make any helpful suggestions. The work is supported by the National Nature Science Foundation of China (No. U22A201181, U1803262, U1736206), National Social Science Fund of China (No. 19ZDA113), and the Application Foundation Frontier Project of Wuhan Science and Technology Bureau (No. 2020010601012288).

References

1. Bo, D., Wang, X., Shi, C., Shen, H.: Beyond low-frequency information in graph convolutional networks. arXiv preprint arXiv:2101.00797 (2021)
2. Cui, G., Zhou, J., Yang, C., Liu, Z.: Adaptive graph encoder for attributed graph embedding. In: KDD, pp. 976–985 (2020)
3. 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)
4. Dou, Y., Ma, G., Yu, P.S., Xie, S.: Robust spammer detection by nash reinforcement learning. In: KDD, pp. 924–933 (2020)
5. Ge, S., Ma, G., Xie, S., Philip, S.Y.: Securing behavior-based opinion spam detection. In: 2018 IEEE BigData, pp. 112–117. IEEE (2018)
6. Hamilton, W., Ying, Z., Leskovec, J.: Inductive representation learning on large graphs. In: NeurIPS, vol. 30 (2017)
7. Kaghazgaran, P., Alfifi, M., Caverlee, J.: Wide-ranging review manipulation attacks: model, empirical study, and countermeasures. In: CIKM, pp. 981–990 (2019)

8. Kipf, T.N., Welling, M.: Semi-supervised classification with graph convolutional networks. arXiv preprint arXiv:1609.02907 (2016)
9. Li, Q., Wu, X.M., Liu, H., Zhang, X., Guan, Z.: Label efficient semi-supervised learning via graph filtering. In: CVPR, pp. 9582–9591 (2019)
10. Liu, Y., et al.: Pick and choose: a GNN-based imbalanced learning approach for fraud detection. In: WWW, pp. 3168–3177 (2021)
11. 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)
12. Liu, Z., Chen, C., Yang, X., Zhou, J., Li, X., Song, L.: Heterogeneous graph neural networks for malicious account detection. In: CIKM, pp. 2077–2085 (2018)
13. McAuley, J.J., Leskovec, J.: From amateurs to connoisseurs: modeling the evolution of user expertise through online reviews. In: WWW, pp. 897–908 (2013)
14. Shi, F., Cao, Y., Shang, Y., Zhou, Y., Zhou, C., Wu, J.: H2-FDetector: a GNN-based fraud detector with homophilic and heterophilic connections. In: WWW, pp. 1486–1494 (2022)
15. Veličković, P., Cucurull, G., Casanova, A., Romero, A., Lio, P., Bengio, Y.: Graph attention networks. arXiv preprint arXiv:1710.10903 (2017)
16. Wang, D., et al.: A semi-supervised graph attentive network for financial fraud detection. In: ICDM, pp. 598–607. IEEE (2019)
17. Wang, J., Wen, R., Wu, C., Huang, Y., Xion, J.: FdGars: fraudster detection via graph convolutional networks in online app review system. In: WWW, pp. 310–316 (2019)
18. Wu, F., Souza, A., Zhang, T., Fifty, C., Yu, T., Weinberger, K.: Simplifying graph convolutional networks. In: ICML, pp. 6861–6871. PMLR (2019)
19. Zhang, G., et al.: FRAUDRE: fraud detection dual-resistant to graph inconsistency and imbalance. In: 2021 ICDM, pp. 867–876. IEEE (2021)
20. Zhong, Q., et al.: Financial defaulter detection on online credit payment via multi-view attributed heterogeneous information network. In: WWW, pp. 785–795 (2020)