DGA-GNN: Dynamic Grouping Aggregation GNN for Fraud Detection

Mingjiang Duan\textsuperscript{1}, Tongya Zheng\textsuperscript{2,1}, Yang Gao\textsuperscript{1}, Gang Wang\textsuperscript{3,4}, Zunlei Feng\textsuperscript{1,5*}, Xinyu Wang\textsuperscript{1,4}

\textsuperscript{1}Zhejiang University
\textsuperscript{2}Hangzhou City University
\textsuperscript{3}Bangsheng Technology Co., Ltd.
\textsuperscript{4}ZJU-Bangsun Joint Research Center
\textsuperscript{5}Shanghai Institute for Advanced Study of Zhejiang University

\{duanj, tycz, roygao, zunleifeng, wangxinyu\}@zju.edu.cn, wanggang@bsfit.com.cn

Abstract

Fraud detection has increasingly become a prominent research field due to the dramatically increased incidents of fraud. The complex connections involving thousands, or even millions of nodes, present challenges for fraud detection tasks. Many researchers have developed various graph-based methods to detect fraud from these intricate graphs. However, those methods neglect two distinct characteristics of the fraud graph: the non-additivity of certain attributes and the distinguishability of grouped messages from neighbor nodes. This paper introduces the Dynamic Grouping Aggregation Graph Neural Network (DGA-GNN) for fraud detection, which addresses these two characteristics by dynamically grouping attribute value ranges and neighbor nodes. In DGA-GNN, we initially propose the decision tree binning encoding to transform non-additive node attributes into bin vectors. This approach aligns well with the GNN’s aggregation operation and avoids nonsensical feature generation. Furthermore, we devise a feedback dynamic grouping strategy to classify graph nodes into two distinct groups and then employ a hierarchical aggregation. This method extracts more discriminative features for fraud detection tasks. Extensive experiments on five datasets suggest that our proposed method achieves a 3\% - 16\% improvement over existing SOTA methods. Code is available at https://github.com/AtwoodDuan/DGA-GNN.

Introduction

In recent years, the rapid development of the information technology industry has correspondingly led to a dramatic increase in various types of fraud, culminating in substantial annual losses worldwide. As fraud typically manifests within reciprocal links between entities, recent research has chiefly focused on graph-based fraud detection, with particular emphasis on Graph Neural Networks (GNNs). Consequently, numerous researchers have deployed a variety of graph-based techniques to address fraud detection in diverse sectors, including e-Payment (Liu et al. 2021a), social network (Wu et al. 2022), and review management (Li et al. 2019) among others.

To address the aforementioned problem of fraud detection in graph structures, numerous scholars have initiated specific research. Recent methodologies in fraud detection can be categorized into two distinct types: spectral methods and spatial methods. Spectral methods, represented by AMNet, BWGNN, and GHRN (Chai et al. 2022; Tang et al. 2022; Gao et al. 2023), primarily aim to regulate the proportion of low-frequency to high-frequency signals. Among spatial methods, CARE-GNN (Dou et al. 2020), PC-GNN (Liu et al. 2021b), and RioGNN (Peng et al. 2021) adopt an edge pruning strategy, retaining the connections between similar nodes before proceeding with feature aggregation. HFDetector (Shi et al. 2022) differentiates edges into homogeneous and heterogeneous types and performs segregated information aggregation, thereby enabling the conservation of more comprehensive information. However, these methods overlook the non-additivity of certain attributes and the distinguishability of grouped messages from neighbor nodes.

Certain non-additive attributes, such as age and transaction frequency, serve as features of nodes within the fraud graph. Consider the following example: An individual node representing a child or an elderly person tends to have a lower likelihood of being a fraudulent entity. Paradoxically, the arithmetic mean of a child’s age and an elderly person’s age approximates the age of a middle-aged individual, who generally has a higher probability of being a fraudulent entity. Consequently, the existing GNN’s mean aggregation approach encounters conflict due to these non-additive attributes within the fraud graph.

On the other hand, nodes can be categorized as either fraudulent or benign. The integration of messages from these divergent categories can lead to the generation of generalized features. However, such an approach might potentially compromise the distinguishability of grouped messages from neighbor nodes, thereby influencing the decision-making capacity of the model. GAGA (Wang et al. 2023), considering the heterophily of the fraud graph, segregates neighbor-labeled nodes into fraudulent and benign groups and aggregates the data from unlabeled nodes as a single group. However, this method overlooks the heterophily of the unlabeled nodes.

In this paper, to address the issue of non-additivity of attributes and to enhance the distinguishability of grouped messages from neighbor nodes, we propose a Dynamic Grouping Aggregation GNN (DGA-GNN). Firstly, to address the non-additivity of attributes such as age, transaction frequency, and features inclusive of missing values, we have...
designed a tree binning encoding mechanism. This technique segregates the original value domain of each attribute based on their prior probabilities into different groups, subsequently generating a one-hot vector representation for each. This technique sorts each attribute feature into different bins, generating a one-hot bin vector. During the aggregation process of DGA-GNN, the one-hot bin vectors, following feature dimensionality reduction and incorporating original attribute values, are point-wise combined into a new feature vector. This effectively preserves the information related to bin position and prior probabilities. By leveraging dynamic grouping aggregation at both the attribute feature level and the neighbor message level, the proposed method outperforms the current state-of-the-art solutions.

Furthermore, to tackle the distinguishability of grouped messages, we developed a feedback dynamic grouping mechanism. At the end of each epoch, the model estimates the categories of all nodes, including both labeled and unlabeled ones. The model output is recursively transmitted to the subsequent epoch, serving as the grouping information for the next epoch. The task of grouping neighbor nodes is in perfect alignment with the ultimate goal of fraud detection. A more accurate estimation of node categories enhances the distinguishability of grouped messages, thereby further boosting the precision of the task.

Our contribution is therefore the proposed Dynamic Grouping Aggregation approach. At the attribute feature level, we employ value domain grouping, effectively mitigating interference from non-additive attribute features. Also, a dynamic neighbor node grouping mechanism is devised to enhance the distinguishability of neighbor messages. Comprehensive experiments conducted on five real-world datasets demonstrate that the proposed DGA-GNN results in an impressive performance increase of approximately 3% ~ 16%.

Related Work

Graph Neural Networks. GNNs (Defferrard, Bresson, and Vandergheynst 2016; Kipf and Welling 2017; Hamilton, Ying, and Leskovec 2017; Veličković et al. 2018) are firstly motivated by the success of Convolutional Neural Networks (LeCun et al. 1998; LeCun 1998; Krizhevsky, Sutskever, and Hinton 2012) to perform graph convolution in the non-grid data, learning graph representations in a dense-vector paradigm. The learned node and graph representations based on GNNs benefit various downstream tasks like node classification (Kipf and Welling 2017), link prediction (Zhang and Chen 2018), and graph classification (Xu et al. 2019). The universality of GNNs for non-grid graph data has attracted much attention from the industry field and achieved several successful applications (Ying et al. 2018; Wang et al. 2019b). Despite their successful applications, these methods usually assume the homophily neighborhood and behave like a low-pass filter, which is unsuitable for complex fraud detection scenarios.

GNN-based Fraud Detection Method. The mainline research of GNNs focuses on the neighborhood homophily assumption that a node and its neighborhood nodes share similar labels. However, fraud nodes in fraud detection graphs are usually surrounded by nodes that have been cheated, which are typically normal nodes. Previous works have not taken neighborhood heterophily into account when introducing GNNs for fraud detection tasks. Ding et al. (2019) generates anomaly scores of nodes in an AutoEncoder paradigm based on GNNs; Li et al. (2019) and Wang et al. (2019a) introduce the advanced GNNs techniques for spam detection and fraud detection, respectively; Liu et al. (2019) designs a specific multi-hop aggregation mechanism to filter the fraud signal from distant neighbors. Nonetheless, these methods suffer from the homophily assumption of common GNNs, resulting in suboptimal performance in fraud detection. Recently, several methods have been developed for fraud detection, treating graph nodes as two categories: fraudulent and ordinary individuals. These methods, which include the spectral method AMNet (Chai et al. 2022), BWGNN (Tang et al. 2022), and spatial methods H²-FDetector (Shi et al. 2022), and GAGA (Wang et al. 2023), have shown promising results. These methods still fail to solve the distinguishability of grouped messages from neighbor nodes and do not consider the negative impact of non-additive attributes on the fraud detection task.

Methodology

To address the non-additivity of attributes and the distinguishability of grouped messages from neighbor nodes in large-scale fraud graphs effectively, we propose the DGA-GNN, which is composed of three main components: dynamic grouping of the attributes value range, dynamic grouping of neighbor nodes, and hierarchical aggregation. Figure 1 shows the framework of the proposed DGA-GNN.

Dynamic Grouping of Attributes Value Range

In fraud graph datasets, certain non-additive attributes are prevalent, such as age and transaction frequency. When these attributes serve as input, addition-based aggregation can lead to nonsensical feature generation. For instance, calculating the mean age of a child and an elderly individual—both of whom have a minimal likelihood of perpetrating fraud—may yield a result corresponding to middle age, which is associated with a high risk of committing fraud. This can distort the classification in fraud detection.

To address this, we introduce the decision tree binning encoding method, which dynamically groups attribute value ranges. This process effectively transforms non-additive features into additive vectors. In the fraud graph, features of the n-th node are denoted as a d-dimensional feature vector \( x_n \in \mathbb{R}^d \) and the corresponding GT (ground truth) label is denoted as \( y_n \in \{0, 1\} \) (0: benign, 1: fraudulent). Each feature vector comprises a series of attributes \( \mathcal{A} = \{a_1, \ldots, a_d\} \). Additionally, the set of node features and GT label can be denoted as \( \mathcal{X} = \{x_1, x_2, \ldots, x_N\} \) and \( \mathcal{Y} = \{y_1, y_2, \ldots, y_N\} \), respectively.

For every attribute \( a \), it’s pivotal to decide the number of groups and the range of each group for optimal fraud detection. With the GT label as supervision, we adopt a decision tree to sort the features of each attribute into different leaf nodes. The number of leaf nodes corresponds to
Figure 1: The framework of DGA-GNN comprises three parts: dynamic grouping of attributes value range, dynamic grouping of neighbor nodes, and hierarchical aggregation. The former converts on-additive attributes to bin vectors, which are well-matched with GNN’s aggregation operation and avoid nonsensical feature generation. The latter two dynamically group graph nodes into fraudulent and ordinary entities, then aggregate intra-group and inter-group features hierarchically, which will extract more discriminating and independent features for fraud detection.

To achieve optimal grouping outcomes, we have developed a feedback dynamic grouping strategy. This implies that the grouping information for the current epoch is derived from the preceding epoch. Throughout the model training process, as the loss continuously decreases, the overall model’s predictions for grouping become more accurate. The increase in prediction accuracy is fed back into the grouping stage, thereby facilitating dynamic enhancement of the grouping. It is noteworthy that the grouping process is directed not only at the training set but also at the entire dataset, including unlabeled graph nodes. The feedback dynamic grouping can utilize estimated grouping information to achieve improved grouping outcomes.

Hierarchical Aggregation
In a multi-relation graph, to reduce the disturbance of redundant features, we design a hierarchical aggregation strategy, including intra-aggregation and inter-aggregation. For each relation, in the intra-group, different nodes have similar information. Therefore, the same added weight is adopted. On the contrary, inter-group information is quite different, and various central nodes have different weight distributions for different groups. Therefore, we introduce a GAT-like inter-group aggregation strategy to aggregate information from the above groups and relations with dynamic weights.

Within a multi-relation graph, \( r \in \{1, 2, \ldots, R\} \) represents a specific type of edge relation. The neighbor set under each relation \( r \) of the center node \( v \) is defined as \( \mathcal{N}(v) \), which is composed of two groups of nodes as follows:

\[
\mathcal{N}_{r,\bar{r}}(v) = \bigcup_{\bar{g}} \mathcal{N}_{r,\bar{g}}(v), \bar{g} \in \{0, 1\},
\]
Algorithm 1: Decision Tree Binning Encoding

**Input:** Feature matrix and labels \((X, Y)\), attribute set \(A\), the number of bins \(k\) in the decision tree  

**Output:** Decision tree binning encoded feature matrix \(\tilde{X}\)

1. \(\tilde{X} \leftarrow X\)
2. for \(u \in A\) do
3.     Build a Decision Tree (DT) based on \((X^u, Y)\)
4.     Extract all split values as split_list from DT
5.     Sort split_list and build \(k\) bins
6.     Replace origin values of \(X^u\) to the serial_number of bins, \(0 <= \text{serial_number} < k\)
7.     One-hot encode \(X^u\) to get \(X^{\tilde{u}}\)
8. \(\tilde{X} \leftarrow X^{\tilde{u}}|X^{\tilde{u}}\)
9. end for
10. return \(\tilde{X}\)

\(\mathcal{N}_r(v)\) denotes the entire set of neighbors of the central node \(v\) under relation \(r\). \(\mathcal{N}_{r\#}(v)\) represents the set of neighbors associated with the central node under relation \(r\) for the \(#\)-th group. \(u \in \mathcal{N}_{r\#}(v)\) if \(\hat{y}_u = 0\), \(u \in \mathcal{N}_{r\#}(v)\) if \(\hat{y}_u = 1\) where \(u\) is each neighbor of node \(v\). \(\hat{y}_u\) is the estimated prediction of node \(u\) based on the previous epoch. Furthermore, for notational simplicity, we define \(g \in \{0, 1, \ast\}\).

**Intra-group Aggregation.** Under each type of relation, an intra-relation aggregation is performed once, and an intra-group aggregation is carried out two times for two groups. Intra-relation aggregation involves amalgamating all neighbor nodes along with the central node under a specific relation. This can be formalized as follows:

\[
\begin{align*}
\text{h}^{\tilde{l}}_{v,r,s} = \text{Agg}_{\text{mean}}(\text{h}^{\tilde{l}}_u), & \quad \forall u \in \mathcal{N}_{r\#}(v), \quad (2) \\
\text{h}^{\tilde{l}}_{v,r,s} = \text{ReLU}(W^{\text{intra}_{g}}_{\text{intra}_{g}}(\text{h}^{\tilde{l}}_v | \text{h}^{\tilde{l}}_{v,r,s})), & \quad (3)
\end{align*}
\]

where \(\text{Agg}_{\text{mean}}()\) denotes the mean aggregation operation, \(W^{\text{intra}_{g}}_{\text{intra}_{g}}\) is the corresponding weight matrix for intra-relation aggregation, \(||\) denotes the concatenation operation, \(\text{ReLU()}\) denotes the ReLU activation function.

For the certain group neighbors in \(\mathcal{N}_{r\#}(v)\), the features of different nodes are equally aggregated as follows:

\[
\begin{align*}
\text{h}^{\tilde{l}}_{v,r,g} = \text{Agg}_{\text{mean}}(\text{h}_u), & \quad \forall u \in \mathcal{N}_{r\#}(v), \quad (4) \\
\text{h}^{\tilde{l}}_{v,r,g} = \text{ReLU}(W^{\text{intra}_{g}}(\text{h}^{\tilde{l}}_v | \text{h}^{\tilde{l}}_{v,r,g})), & \quad (5)
\end{align*}
\]

where \(W^{\text{intra}_{g}}\) is the corresponding weight matrix for intra-group aggregation.

**Inter-group Aggregation.** To collect information on all groups, the attention mechanism is adopted to obtain the central node \(v\)’s embedding \(\text{h}_v\) of the \(l\)-th layer as follows:

\[
\begin{align*}
\text{h}^l_v = \sum_r \sum_g \alpha^l_{r,g} \text{h}^{\tilde{l}}_{v,r,g}, \quad (6) \\
\alpha^l_{r,g} = \frac{\exp(\omega^l_{r,g})}{\sum_m \exp(\omega^l_{m,g})}, \quad (7) \\
\omega^l_{r,g} = q^T \cdot \tanh(W^{\text{inter}_{1}}_{\text{inter}_{1}} \text{h}^{\tilde{l}-1}_v + W^{\text{inter}_{2}}_{\text{inter}_{2}} \text{h}^{\tilde{l}}_{v,r,g}), \quad (8)
\end{align*}
\]

**Iterative Optimization**

With the aforementioned intra-group and inter-group aggregation operations, we feed the final layer’s embedding \(h^l_v\) into a Multilayer Perceptron (MLP). The MLP function produces a predicted value \(p_v\), representing the probability that node \(v\) is predicted to be fraudulent. Subsequently, to train the DGA-GNN model, we utilize a cross-entropy classification loss for identifying node \(v\), as elaborated below:

\[
L = - \sum_{v \in V} [y_v \log p_v + (1 - y_v) \log (1 - p_v)]. \quad (9)
\]

At the end of each epoch, we evaluate the category estimation for all nodes using the current model, represented as \(\hat{Y} = \{\hat{y}_1, \hat{y}_2, ..., \hat{y}_m\}\). For each node \(v\) with a predicted probability \(p_v\), we apply a decision threshold \(z\). If \(p_v > z\), node \(v\) is classified as fraudulent (\(\hat{y}_v = 1\)); otherwise, it is considered benign (\(\hat{y}_v = 0\)).

\[
\hat{y}_v = \begin{cases} 
1 & \text{if } p_v > z \\
0 & \text{otherwise}
\end{cases} \quad (11)
\]

During the optimization process of the model, the category estimation for all nodes is updated with each iteration. More accurate estimations lead to more precise groupings, which, in turn, yield features with better discriminability, thereby improving the model’s performance.
<table>
<thead>
<tr>
<th>Method</th>
<th>Dataset</th>
<th>Scenario</th>
<th>Metrics</th>
<th>AP</th>
<th>AUC</th>
<th>AP</th>
<th>AUC</th>
<th>AP</th>
<th>AUC</th>
<th>AP</th>
<th>AUC</th>
<th>AP</th>
<th>AUC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Non-GNN Baseline</td>
<td>Finance</td>
<td>Elliptic</td>
<td>T-Finance</td>
<td>61.71</td>
<td>80.08</td>
<td>69.49</td>
<td>85.72</td>
<td>88.89</td>
<td>29.18</td>
<td>68.27</td>
<td>99.49</td>
<td>92.62</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>T-Social</td>
<td>52.8</td>
<td>67.78</td>
<td>29.18</td>
<td>68.27</td>
<td>99.49</td>
<td>92.62</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>YelpChi</td>
<td>29.68</td>
<td>80.89</td>
<td>84.13</td>
<td>28.01</td>
<td>82.72</td>
<td>5.28</td>
<td>67.78</td>
<td>81.16</td>
<td>96.29</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Amazon</td>
<td>94.50</td>
<td>76.58</td>
<td>56.35</td>
<td>49.06</td>
<td>81.75</td>
<td>34.86</td>
<td>66.23</td>
<td>92.62</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Homophilic</td>
<td>GCN</td>
<td>(Kipf and Welling 2017)</td>
<td>38.27</td>
<td>85.28</td>
<td>78.47</td>
<td>92.34</td>
<td>56.35</td>
<td>91.72</td>
<td>24.89</td>
<td>60.99</td>
<td>29.68</td>
<td>80.89</td>
<td></td>
</tr>
<tr>
<td></td>
<td>GAT</td>
<td>(Veličković et al. 2018)</td>
<td>27.15</td>
<td>76.54</td>
<td>78.94</td>
<td>92.89</td>
<td>52.26</td>
<td>93.40</td>
<td>34.86</td>
<td>73.98</td>
<td>85.16</td>
<td>94.50</td>
<td></td>
</tr>
<tr>
<td>Spectral Heterophilic</td>
<td>AMNet (Chai et al. 2022)</td>
<td>69.49</td>
<td>88.89</td>
<td>83.41</td>
<td>95.72</td>
<td>\</td>
<td>\</td>
<td>52.85</td>
<td>83.57</td>
<td>87.87</td>
<td>97.90</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>BWGNN (Tang et al. 2022)</td>
<td>48.39</td>
<td>89.63</td>
<td>85.50</td>
<td>96.29</td>
<td>79.53</td>
<td>97.77</td>
<td>68.05</td>
<td>90.64</td>
<td>88.52</td>
<td>98.08</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>GHRN (Gao et al. 2023)</td>
<td>55.20</td>
<td>89.95</td>
<td>83.78</td>
<td>96.76</td>
<td>87.66</td>
<td>98.26</td>
<td>66.68</td>
<td>91.51</td>
<td>89.84</td>
<td>98.18</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Spatial Heterophilic</td>
<td>CARE-GNN (Liu et al. 2020)</td>
<td>37.19</td>
<td>87.84</td>
<td>61.84</td>
<td>89.95</td>
<td>41.15</td>
<td>78.32</td>
<td>52.96</td>
<td>83.99</td>
<td>85.64</td>
<td>96.96</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>PC-GNN (Liu et al. 2021b)</td>
<td>34.70</td>
<td>88.52</td>
<td>65.11</td>
<td>89.90</td>
<td>51.40</td>
<td>89.29</td>
<td>49.06</td>
<td>81.75</td>
<td>84.86</td>
<td>95.46</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>RioGNN (Peng et al. 2021)</td>
<td>29.06</td>
<td>86.35</td>
<td>62.60</td>
<td>91.32</td>
<td>17.62</td>
<td>81.74</td>
<td>56.45</td>
<td>85.72</td>
<td>87.62</td>
<td>96.91</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>H^2-FDetector (Shi et al. 2022)</td>
<td>10.53</td>
<td>63.18</td>
<td>\</td>
<td>\</td>
<td>\</td>
<td>\</td>
<td>57.48</td>
<td>89.88</td>
<td>84.94</td>
<td>96.05</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>GAGA (Wang et al. 2023)</td>
<td>28.01</td>
<td>82.72</td>
<td>84.13</td>
<td>96.53</td>
<td>78.52</td>
<td>97.80</td>
<td>76.58</td>
<td>93.86</td>
<td>88.34</td>
<td>97.18</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ours</td>
<td>DGA-GNN</td>
<td></td>
<td></td>
<td>81.12</td>
<td>94.22</td>
<td>90.51</td>
<td>97.77</td>
<td>98.19</td>
<td>99.88</td>
<td>92.80</td>
<td>97.95</td>
<td>92.97</td>
<td>98.39</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>+11.63</td>
<td>+4.27</td>
<td>+5.01</td>
<td>+1.01</td>
<td>+10.53</td>
<td>+1.62</td>
<td>+16.22</td>
<td>+4.09</td>
<td>+3.13</td>
<td>+0.21</td>
</tr>
</tbody>
</table>

Table 1: Comparison results(%) with twelve methods (organized into four groups) on five benchmark datasets. Within each group, the leading score is marked with an underscore ‘_’. The notation ‘\’ signifies ‘out of video memory’, while the value following ‘+’ illustrates the enhancement our method attained over the next best score.

Algorithm 2: The Training Algorithm of DGA-GNN

**Input:** The maximum number of iterations $E_{epoch}$, and an attribute graph represented as $G = (V, \mathcal{X}, A, \mathcal{E}, Y)$.

1. Use $(x_{train}, y_{train})$ to fit binning encoder
2. Use binning encoder to get $\tilde{X}$;
3. $H^0 = MLP(\tilde{X})$;
4. **while** $e < E_{epoch}$ **do**
5. Get node grouping from dynamic grouping buffer;
6. **for** $l = 0, 1, ..., L$ **do**
7. **for** $r = 0, 1, ..., R$ **do**
8. Calculate $h^l_{v, r, g}$ by Eq.(2)(3)
9. Calculate $h^l_{v, r, g}$ by Eq.(4)(5) for each group;
10. **end for**
11. Update $h^l$ by Eq.(6)(7)(8);
12. Update the total loss using Eq.(9)(10);
13. Use back-propagation to update model parameters;
14. **end for**
15. Update dynamic grouping buffer with output;
16. **end while**

**Experiments**

**Dataset**

Experiments are conducted on five real-world fraud detection datasets. These datasets comprise Elliptic, designed for illicit Bitcoin transaction detection (Weber et al. 2019); T-Finance, a financial transaction fraud dataset (Tang et al. 2022); T-Social, a social network abnormal account detection dataset (Tang et al. 2022). Additionally, YelpChi and Amazon are included, both widely utilized as fake review datasets in graph fraud detection literature (Rayana and Akoglu 2015; McAuley and Leskovec 2013).

Addressing these datasets presents distinct challenges. For instance, the Elliptic dataset comprises multiple subgraphs sequenced over a timeline, with isolated nodes in the training and validation sets potentially affecting static grouping based solely on the training set. Both T-Finance and T-Social have tens of millions of edges, introducing computational challenges due to algorithmic complexity. Conversely, YelpChi and Amazon are multi-relational graphs, requiring flexible management of multiple relationships and their inter-group effects. A summary of these datasets is provided in Table 2. The detailed descriptions are given in supplementary materials.

**Experimental Setup**

**Baseline and Implementation.** We employ four distinct groups of baseline methodologies for comparison with the proposed method. The first group, encompassing Multi-layer Perceptron (MLP) (Hinton 1990) and Random Forest (RF) (Breiman 2001), serves as a foundational reference to observe outcomes when graph information is absent. We utilize the Scikit-learn toolkit for the implementation of MLP and RF. The second group comprises conventional Graph Neural Network algorithms, namely Graph Convolutional Network (GCN), and Graph Attention Network (GAT) (Kipf and Welling 2017; Veličković et al. 2018). These were implemented utilising the DGL framework. The third and fourth groups represent spectral heterophilic and spatial heterophilic methodologies respectively. The former includes
Table 2: Statistic of five fraud datasets. Fraud(%) denotes the proportion of fraudulent people. #nodes and #edges denote the number of nodes and edges respectively. #features denotes the attribute number in each dataset.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>#nodes</th>
<th>#edges</th>
<th>#features</th>
<th>fraud(%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>T-Finance</td>
<td>39,357</td>
<td>21,222,543</td>
<td>10</td>
<td>4.58</td>
</tr>
<tr>
<td>T-Social</td>
<td>5,781,065</td>
<td>73,105,508</td>
<td>10</td>
<td>3.01</td>
</tr>
<tr>
<td>Elliptic</td>
<td>46,564</td>
<td>73,248</td>
<td>93</td>
<td>9.76</td>
</tr>
<tr>
<td>YelpChi</td>
<td>45,954</td>
<td>3,846,979</td>
<td>32</td>
<td>14.53</td>
</tr>
<tr>
<td>Amazon</td>
<td>11,944</td>
<td>4,398,392</td>
<td>25</td>
<td>6.87</td>
</tr>
</tbody>
</table>

Table 3: The results(%) of an ablation study conducted on two proposed components using the T-Social and YelpChi datasets are shown above. The terms ‘w/o encoding’ and ‘w/o grouping’ represent the removal of the decision tree binning encoding and the grouping strategy, respectively. On the other hand, ‘w/ equality’ and ‘w/ static’ denote the sub-optimal variants of the components.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Metric</th>
<th>T-Social</th>
<th>YelpChi</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>AP</td>
<td>AUC</td>
</tr>
<tr>
<td>w/o encoding</td>
<td>86.15</td>
<td>98.55</td>
<td>78.73</td>
</tr>
<tr>
<td>w/ equality</td>
<td>92.09</td>
<td>98.96</td>
<td>87.02</td>
</tr>
<tr>
<td>w/o grouping</td>
<td>79.97</td>
<td>98.14</td>
<td>80.72</td>
</tr>
<tr>
<td>w/ static</td>
<td>90.16</td>
<td>99.03</td>
<td>88.21</td>
</tr>
<tr>
<td>DGA-GNN</td>
<td>98.19</td>
<td>99.88</td>
<td>92.80</td>
</tr>
</tbody>
</table>

Figure 3: Parameter analysis with emphasis on the number of bins \( k \) and the threshold value \( z \).
**Ablation Study and Parameter Analysis**

To assess the efficacy of the components within DGA-GNN, we performed an ablation study focusing on the decision tree binning encoding and the feedback dynamic grouping strategy. For our proposed modules, we designed two suboptimal variants: equidistant binning encoding and static grouping strategy. The equidistant binning refers to using quantiles to complete value range grouping, while the static grouping strategy involves grouping based solely on labeled node information. The results, as showcased in Table 3, manifest that the fully-equipped DGA-GNN consistently achieves the best performance, thereby demonstrating the effectiveness of each component.

<table>
<thead>
<tr>
<th>Method</th>
<th>non-additivity</th>
<th>additivity</th>
<th>DGA-GNN</th>
</tr>
</thead>
<tbody>
<tr>
<td>AP</td>
<td>90.02</td>
<td>81.86</td>
<td>92.80</td>
</tr>
<tr>
<td>AUC</td>
<td>97.27</td>
<td>96.03</td>
<td>97.95</td>
</tr>
</tbody>
</table>

Table 4: The model’s performance(%) after applying a binning encoder to each subset, respectively.

**Visualization and Discussion**

**Dynamic Neighbor Grouping.** During the initial phases of training, owing to the incomplete training of the model, there are significant fluctuations in the estimations across all node categories. However, as the training deepens, these estimation flips become less frequent, resulting in improved accuracy, as depicted in Figure 5.

**Non-additivity.** The widely recognized YelpChi fraud review dataset was selected for non-additivity analysis. Its thirty-two features were divided evenly into bins, which were then categorized into non-additive and additive subsets based on the degree of monotonicity. Figure 5(a) and Figure 5(b) visualize three representative attributes from each subset. Attributes from the non-additive subset display pronounced non-monotonicity, thereby amplifying the learning complexity of graph network models. Table 4 displays the DGA-GNN’s efficiency when a binning encoder is solely employed on the attributes from both subsets. It was observed that the non-additive subset surpassed the additive subset markedly, affirming the binning encoder’s capability to mitigate the challenges posed by feature non-additivity.

**Conclusion**

In this work, we introduce the DGA-GNN framework for tackling feature non-additivity and enhancing message distinguishability in fraud detection. The framework employs decision tree binning encoding for feature transformation and utilizes feedback dynamic grouping with hierarchical aggregation for improved message distinguishability. This method dynamically classifies adjacent nodes and aggregates their features in an additive manner, hierarchically enhancing their discriminative capabilities. Tests on five fraud datasets confirm the effectiveness of DGA-GNN. Future work will focus on integrating decision trees with GNNs and further exploring heterophily information in fraud detection.
Acknowledgments
This work is funded by National Key Research and Development Project (Grant No: 2022YFB2703100), National Natural Science Foundation of China (U20B2066 and 61972339), and the Starry Night Science Fund of Zhejiang University Shanghai Institute for Advanced Study (Grant No. SN-ZJU-SIAS-001).

References


