DGA-GNN: Dynamic Grouping Aggregation GNN for Fraud Detection

Mingjiang Duan[1], Tongya Zheng[2,1], Yang Gao[1], Gang Wang[3,4], Zunlei Feng[1,5*], Xinyu Wang[1,4]

[1]Zhejiang University
[2]Hangzhou City University
[3]Bangsheng Technology Co,Ltd.
[4]ZJU-Bangsun Joint Research Center
[5]Shanghai Institute for Advanced Study of Zhejiang University
{duanmj, tyzheng, 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 $3\% \sim 16\%$ 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. H^2-FDetector (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 arithmeticmean 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

*Corresponding author.

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 $3\% \sim 16\%$.

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 H^2-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 ($x_n \in R^d$) 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 $\mathcal{A}$=$\{a_1, ..., a_d\}$. Additionally, the set of node features and GT label can be denoted as $\mathcal{X} = \{x_1, x_2, ..., x_n, ..., x_N\}$ and $\mathcal{Y} = \{y_1, y_2, ..., y_n, ..., y_N\}$, 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

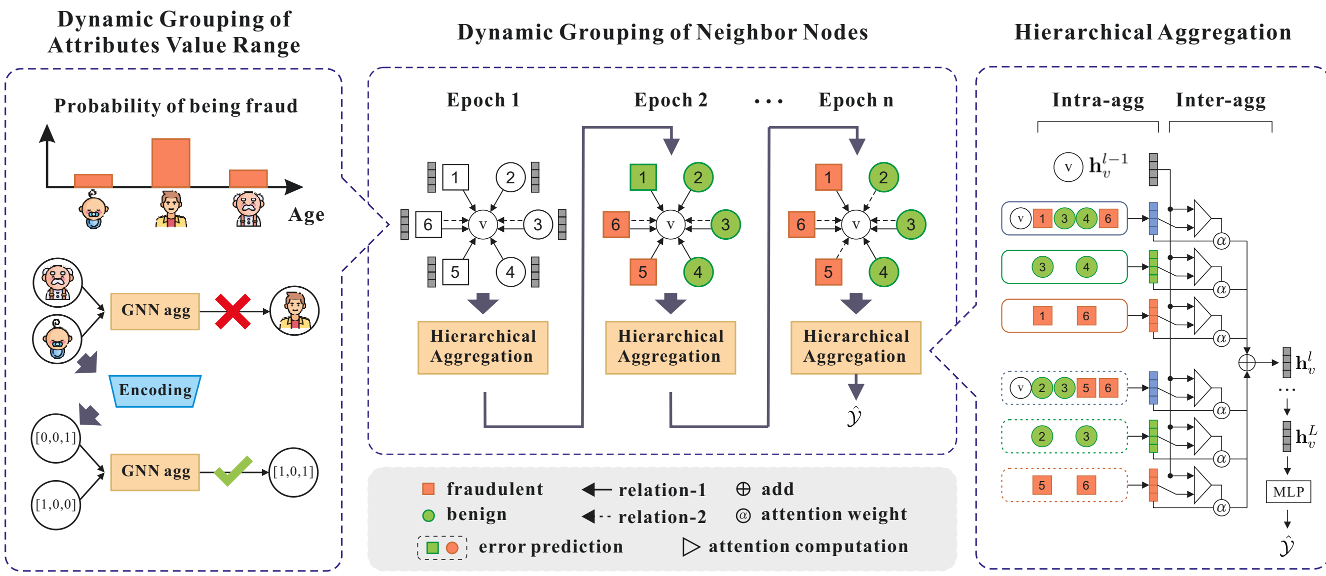

Figure 1: The framework of DGA-GNN comprises three parts: dynamic grouping of attributes value range, dynamic grouping of neighbor nodes, and hierarchical aggregation. The former converts on-additive attributes to bin vectors, which are well-matched with GNN's aggregation operation and avoid nonsensical 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.

the number of groups, and the split condition values determine the range of each group. With the trained decision tree for each attribute, all the split condition values are sorted in ascending order and then used to divide the whole range into K mutually exclusive bins. Based on the learned groups for each attribute, the original features x_n can be converted into $\widetilde{x}_n$. The set of node features will be updated into $\widetilde{\mathcal{X}} = \{\widetilde{x}_1, \widetilde{x}_2, ..., \widetilde{x}_n, ..., \widetilde{x}_N\}$. Algorithm 1 describes the pseudocode of decision tree binning encoding.

To better elucidate, consider an example. Take the age attribute, which varies between 0 to 100 years. It can be divided into 10 distinct groups, with each representing a decade. Consequently, a 5-year-old individual would be represented by the one-hot vector $[1, 0, 0, ..., 0]$. When aggregating the data for two 5-year-olds alongside a 95-year-old, the resulting vector becomes $[2, 0, 0, ..., 1]$, effectively preserving the features from neighbors indicative of a low likelihood of committing fraud. An illustration of the decision tree binning encoding is given in Figure 2.

Dynamic Grouping of Neighbor Nodes

To improve the distinguishability of grouped messages from neighboring nodes, we need to dynamically classify these nodes into fraudulent and benign groups at each layer separately. For each convolution layer, there are two steps to obtain $\mathbf{H}^l$ from $\mathbf{H}^{l-1}$: dynamic grouping and hierarchical aggregation, where $\mathbf{H}^0 = \boldsymbol{MLP}(\widetilde{X})$. $\mathbf{H}^{l-1} = \{\mathbf{h}_1^{l-1}, \mathbf{h}_2^{l-1}, \mathbf{h}_3^{l-1}, \ldots, \mathbf{h}_N^{l-1}\}$ is the set of node embedding at $l-1$-th layer. $\mathbf{H}^L$ represents node embedding output on the last layer. The following additional Multi-Layer Perceptron neural network (MLP) serves as the prediction head.

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, $r \in \{1, 2, ..R\}$ 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:

$$\mathcal{N}_{r,*}(v) = \bigcup_{\overline{g}} \mathcal{N}_{r,\overline{g}}(v), \overline{g} \in \{0, 1\}, \tag{1}$$

Algorithm 1: Decision Tree Binning Encoding

Input: Feature matrix and labels $(\mathcal{X}, \mathcal{Y})$, attribute set $\mathcal{A}$, the number of bins k in the decision tree
Output: Decision tree binning encoded feature matrix $\widetilde{\mathcal{X}}$

$\widetilde{\mathcal{X}} \leftarrow \mathcal{X}$
for a in $\mathcal{A}$ **do**
  Build a Decision Tree (**DT**) base on $(\mathcal{X}^a, \mathcal{Y})$
  Extract all split values as **split_list** from **DT**
  Sort **split_list** and build k bins
  Replace origin values of $\mathcal{X}^a$ to the $serial\_number$ of bins, $0 <= serial\_number < k$
  One-hot encode $\mathcal{X}^a$ to get $\widetilde{\mathcal{X}}^a$
  $\widetilde{\mathcal{X}} \leftarrow \widetilde{\mathcal{X}} || \widetilde{\mathcal{X}}^a$
end for
return $\widetilde{\mathcal{X}}$

$\mathcal{N}_{r,*}(v)$ denotes the entire set of neighbors of the central node v under relation r. $\mathcal{N}_{r,\overline{g}}(v)$ represents the set of neighbors associated with the central node v under relation r for the $\overline{g}$-th group. $u \in \mathcal{N}_{r,0}(v)$ **if** $\hat{y}_u = 0, u \in \mathcal{N}_{r,1}(v)$ **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 $g \in \{0, 1, *\}$.

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:

$$\mathbf{h}'^{,l}_{v,r,*} = \boldsymbol{Agg}_{mean}\left(\mathbf{h}_u^{l-1}\right), \forall u \in \mathcal{N}_{r,*}(v), \tag{2}$$

$$\mathbf{h}^{l}_{v,r,*} = \boldsymbol{ReLU}\left(\mathbf{W}^l_{intra\_r}\left(\mathbf{h}_v^{l-1} \,||\, \mathbf{h}'^{,l}_{v,r,*}\right)\right), \tag{3}$$

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

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

$$\mathbf{h}'^{,l}_{v,r,\overline{g}} = \boldsymbol{Agg}_{mean}\left(\mathbf{h}_u^{l-1}\right), \forall u \in \mathcal{N}_{r,\overline{g}}(v), \tag{4}$$

$$\mathbf{h}^{l}_{v,r,\overline{g}} = \boldsymbol{ReLU}\left(\mathbf{W}^l_{intra\_g}\mathbf{h}'^{,l}_{v,r,\overline{g}}\right), \tag{5}$$

where $\mathbf{W}^l_{intra\_g}$ 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 $\mathbf{h}_v$ of the l-th layer as follows:

$$\mathbf{h}^l_v = \sum_r \sum_g \alpha^l_{r,g}\mathbf{h}^l_{v,r,g}, \tag{6}$$

$$\alpha^l_{r,g} = \frac{\exp\left(\omega^l_{r,g}\right)}{\sum_m \exp\left(\omega^l_{r,g}\right)}, \tag{7}$$

$$\omega^l_{r,g} = \boldsymbol{q}^T \cdot \tanh\left(\mathbf{W}^l_{inter1}\mathbf{h}^{l-1}_v + \mathbf{W}^l_{inter2}\mathbf{h}^l_{v,r,g}\right), \tag{8}$$

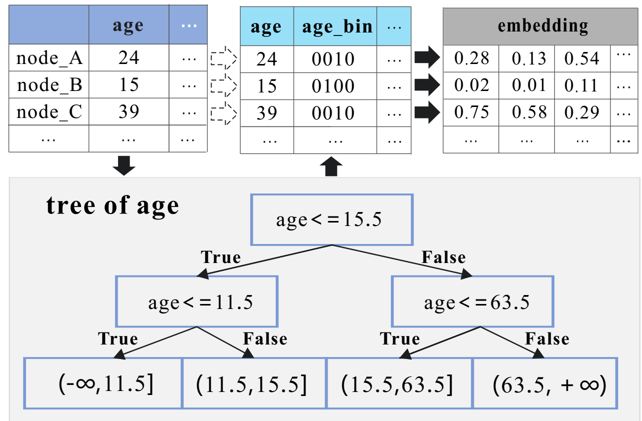

Figure 2: An illustration of decision tree binning encoding. Age is one of the node attributes. Building a decision tree for the age column. In the tree of age, the feature space is divided into four bins. The feature split points are 11.5, 15.5 and 63.5. The age of node_A is in the third bin. Therefore, its one-hot bin vector is denoted as '[0,0,1,0]'.

where $\mathbf{W}^l_{inter1}$ and $\mathbf{W}^l_{inter2}$ are the corresponding weight matrix for the inter-aggregation. $\boldsymbol{q}$ is a learnable parameter vector, which decides the value size of the attention parameter α^l_g for group g and relation r. T denotes the transposition operation.

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:

$$p_v = \boldsymbol{Sigmoid}\left(\boldsymbol{MLP}\left(\mathbf{h}^L_v\right)\right), \tag{9}$$

$$\mathcal{L} = -\sum_{v \in \mathcal{V}} \left[y_v \log p_v + (1 - y_v) \log(1 - p_v)\right]. \tag{10}$$

At the end of each epoch, we evaluate the category estimation for all nodes using the current model, represented as $\hat{\mathcal{Y}} = \{\hat{y}_1, \hat{y}_2, ..., \hat{y}_n, ..., \hat{y}_N\}$. 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 ($\hat{y}_v = 0$):

$$\hat{y}_v = \begin{cases} 1 & \text{if } p_v > z \\ 0 & \text{otherwise} \end{cases}. \tag{11}$$

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. Algorithm 2 shows the pseudocode of the training procedure.

Method	Scenario	Finance				Social		Review			
	Dataset	Elliptic		T-Finance		T-Social		YelpChi		Amazon	
	Metrics	AP	AUC	AP	AUC	AP	AUC	AP	AUC	AP	AUC
Non-GNN Baseline	MLP (Hinton 1990)	61.71	80.08	65.95	91.31	5.28	67.78	29.18	68.27	66.23	92.62
	RF (Breiman 2001)	65.20	82.69	81.16	94.42	42.50	80.38	69.77	89.47	83.02	94.47
Homophilic	GCN (Kipf and Welling 2017)	38.27	85.28	78.47	92.34	56.35	91.72	24.89	60.99	29.68	80.89
	GAT (Veličković et al. 2018)	27.15	76.54	78.94	92.89	52.26	93.40	34.86	73.98	85.16	94.50
Spectral Heterophilic	AMNet (Chai et al. 2022)	69.49	88.89	83.41	95.72	\	\	52.85	83.57	87.87	97.90
	BWGNN (Tang et al. 2022)	48.39	89.63	85.50	96.29	79.53	97.77	68.05	90.64	88.52	98.08
	GHRN (Gao et al. 2023)	55.20	89.95	83.78	96.76	87.66	98.26	66.68	91.51	89.84	98.18
Spatial Heterophilic	CARE-GNN (Liu et al. 2020)	37.19	87.84	61.84	89.95	41.15	78.32	52.96	83.99	85.64	96.96
	PC-GNN (Liu et al. 2021b)	34.70	88.52	65.11	89.90	51.40	89.29	49.06	81.75	84.85	95.46
	RioGNN (Peng et al. 2021)	29.06	86.35	62.60	91.32	17.62	81.74	56.45	85.72	87.62	96.91
	H^2-FDetector (Shi et al. 2022)	10.53	63.18	\	\	\	\	57.48	89.88	84.94	96.05
	GAGA (Wang et al. 2023)	28.01	82.72	84.13	96.53	78.52	97.80	76.58	93.86	88.34	97.18
Ours	**DGA-GNN**	**81.12 +11.63**	**94.22 +4.27**	**90.51 +5.01**	**97.77 +1.01**	**98.19 +10.53**	**99.88 +1.62**	**92.80 +16.22**	**97.95 +4.09**	**92.97 +3.13**	**98.39 +0.21**

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 $\mathrm{E}_{\text{epoch}}$, and an attribute graph represented as $\mathcal{G} = (\mathcal{V}, \mathcal{X}, \mathcal{A}, \mathcal{E}, \mathcal{Y})$.

```
Use (X_train, Y_train) to fit binning encoder
Use binning encoder to get X~ ;
H^0 = MLP(X~);
while e < E_epoch do
    Get node grouping from dynamic grouping buffer;
    for l = 0, 1, ..., L do
        for r = 0, 1, ..., R do
            Calculate h^l_{v,r,*} by Eq.(2)(3)
            Calculate h^l_{v,r,g} by Eq.(4)(5) for each group;
        end for
        Update h^l by Eq.(6)(7)(8);
        Update the total loss using Eq.(9)(10);
        Use back-propagation to update model parameters;
    end for
    Update dynamic grouping buffer with output;
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 Multilayer 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

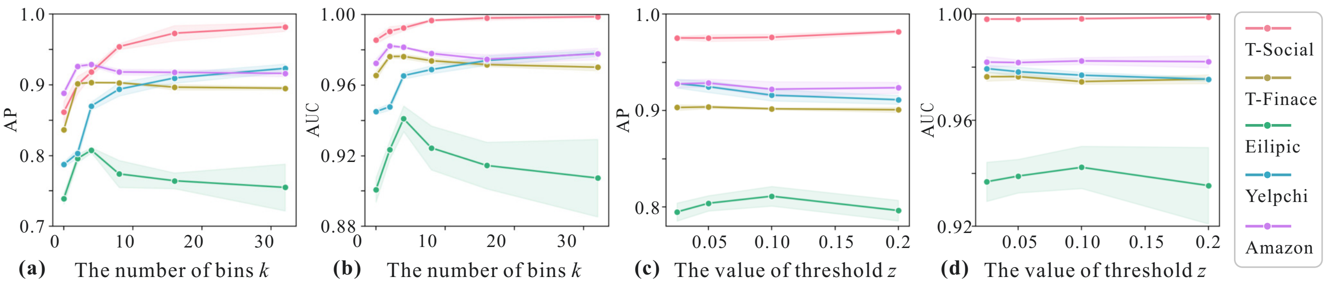

Figure 3: Parameter analysis with emphasis on the number of bins k and the threshold value z.

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

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.

AMNet, BWGNN, GHRN (Chai et al. 2022; Tang et al. 2022; Gao et al. 2023), while the latter comprises CARE-GNN, PC-GNN, RioGNN, H2, and GAGA (Liu et al. 2020; Dou et al. 2020; Liu et al. 2021b; Peng et al. 2021; Shi et al. 2022; Wang et al. 2023). The last two groups were implemented using the official versions supplied by the authors. We present the mean result of ten trials across all datasets for each method, excluding the T-Social dataset, where we performed and reported the average of five trials. All experiments are run on a NVIDIA A100 GPU and an Intel i9-13900K processor @5.80 GHz.

Metrics. For fraud detection tasks, the evaluation indicators should balance precision and recall. Therefore, we use two widely used indicators to measure the performance of all comparative methods, which are AP and AUC.

Data Split. For all datasets excluding Elliptic, the proportions for training, validation, and testing are distributed in a 4:2:4 ratio. The partitioning is performed with utilities from the sklearn package, and we maintain a consistent random seed as per prior work. In the case of the Elliptic dataset, the partitioning respects transaction entity timestamps, conforming to official recommendations for dataset division.

Hyperparameters. For all methods involving neural networks, we employ the Adam optimizer with a learning rate of 0.001 and a weight decay of 0.001. The maximum number of iterations is set to 1000. The model achieving the lowest validation loss is saved and subsequently utilized for test set predictions. All baseline models are fine-tuned post-initialization using the officially recommended parameters. For the DGA-GNN, the number of bins k and the decision threshold z are determined based on the validation set score. The detailed settings are given in supplementary materials.

Dataset	T-Social		YelpChi	
Metric	AP	AUC	AP	AUC
w/o encoding	86.15	98.55	78.73	94.47
w/ equality	92.09	98.96	87.02	96.24
w/o grouping	79.97	98.14	80.72	94.90
w/ static	90.16	99.03	88.21	96.79
DGA-GNN	98.19	99.88	92.80	97.95

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.

Quantitative Performance Comparison

Table 1 shows that the proposed method has shown promising results on both evaluation measures across all data, achieving about $3\% \sim 16\%$ increases on different metrics of different datasets. Non-GNN baseline methods do not consider the information from the graph. The superior performance of the random forest indicates the significant ability of decision trees to utilize features in fraud detection tasks.

The traditional GNN methods (GCN, GAT) are based on the homophilic hypothesis, so they show poor performance. The superior performance of GAT could be attributed to its use of the attention mechanism. This mechanism effectively weights information from different neighbors, thereby mitigating the effects of heterophily.

The improved heterophilic GNN algorithms for fraud detection take into account the non-homophily of the fraud graph. So that they achieve better results. In the spectral domain, GHRN (Gao et al. 2023) leads due to its dynamic consideration of label distribution. In the spatial domain, the GAGA method (Wang et al. 2023) excels as it consolidates the original information and label distribution information from different category neighbors into fixed groups and learns through an attention mechanism.

The success of GAGA demonstrates the efficacy of group-

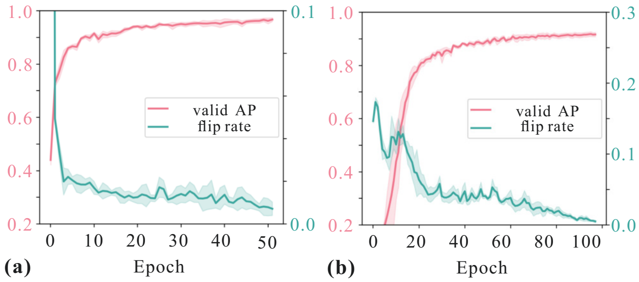

Figure 4: As the number of training epochs increases, there is a variation in the flip rate of $\hat{y}$ and the accuracy of the validation set. Subfigures (a) and (b) correspond to the T-Social dataset and YelpChi dataset, respectively.

Method	non-additivity	additivity	DGA-GNN
AP	90.02	81.86	92.80
AUC	97.27	96.03	97.95

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

ing strategies. The proposed method is also rooted in the grouping, but due to the introduction of dynamic grouping, coupled with the consideration of non-additivity in original features, our results surpass those of GAGA.

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 sub-optimal 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.

Number of Bins k. Figure 3(a) and Figure 3(b) present the AUC and AP scores of DGA-GNN on five datasets when varying k from 0 to 32. The experimental outcomes reveal that the datasets' performance typically reaches its peak with an increasing k. Specifically, the apex values for the Amazon, T-Finance, and Elliptic datasets are observed at $k = 4$. The T-Social and YelpChi datasets attain their maximum value at $k = 32$. This discrepancy can likely stem from the varying degrees of non-additivity inherent to each dataset.

Value of threshold z. The results are not highly sensitive to the value of the threshold z. For certain datasets, optimal values tend to cluster around the fraud proportion in the dataset, as depicted in Figure 3(c) and Figure 3(d).

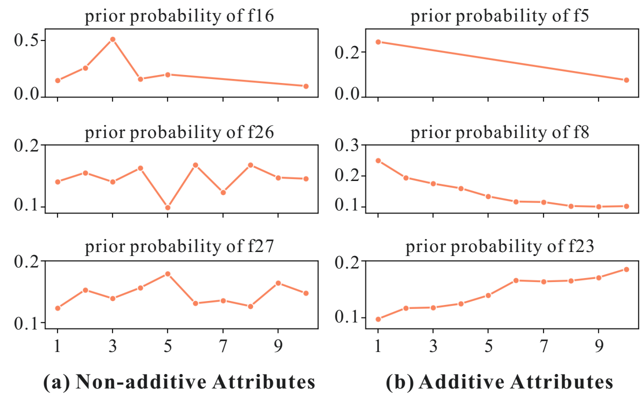

Figure 5: The study of non-additive attributes. Attributes are binned equally, and then ordered numerically. In Subfigures (a) and (b), the x-axis represents the bin index and the y-axis represents averages of y values for each bin. Attributes are divided into non-additive and additive subsets, depending on monotonicity. Subfigures (a) and (b) exhibit three representative attributes for each subset.

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

Bo, D.; Wang, X.; Shi, C.; and Shen, H. 2021. Beyond low-frequency information in graph convolutional networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 3950–3957.

Breiman, L. 2001. Random Forests. *Machine Learning*, 45(1): 5–32.

Chai, Z.; You, S.; Yang, Y.; Pu, S.; Xu, J.; Cai, H.; and Jiang, W. 2022. Can Abnormality be Detected by Graph Neural Networks? In *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence*, 1945–1951.

Defferrard, M.; Bresson, X.; and Vandergheynst, P. 2016. Convolutional Neural Networks on Graphs with Fast Localized Spectral Filtering. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, 3844–3852.

Ding, K.; Li, J.; Bhanushali, R.; and Liu, H. 2019. Deep anomaly detection on attributed networks. In *Proceedings of the 2019 SIAM International Conference on Data Mining*, 594–602.

Dou, Y.; Liu, Z.; Sun, L.; Deng, Y.; Peng, H.; and Yu, P. S. 2020. Enhancing graph neural network-based fraud detectors against camouflaged fraudsters. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, 315–324.

Gao, Y.; Wang, X.; He, X.; Liu, Z.; Feng, H.; and Zhang, Y. 2023. Addressing heterophily in graph anomaly detection: A perspective of graph spectrum. In *Proceedings of the ACM Web Conference 2023*, 1528–1538.

Hamilton, W. L.; Ying, R.; and Leskovec, J. 2017. Inductive Representation Learning on Large Graphs. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, 1025–1035.

Hinton, G. E. 1990. Connectionist learning procedures. In *Machine learning*, 555–610. Elsevier.

Kipf, T. N.; and Welling, M. 2017. Semi-Supervised Classification with Graph Convolutional Networks. In *International Conference on Learning Representations*.

Krizhevsky, A.; Sutskever, I.; and Hinton, G. E. 2012. ImageNet Classification with Deep Convolutional Neural Networks. In *Proceedings of the 26th International Conference on Neural Information Processing Systems*, 1097–1105.

LeCun, Y. 1998. The mnist database of handwritten digits. http://yann.lecun.com/exdb/mnist/.

LeCun, Y.; Bottou, L.; Bengio, Y.; and Haffner, P. 1998. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11): 2278–2324.

Li, A.; Qin, Z.; Liu, R.; Yang, Y.; and Li, D. 2019. Spam review detection with graph convolutional networks. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, 2703–2711.

Liu, C.; Sun, L.; Ao, X.; Feng, J.; He, Q.; and Yang, H. 2021a. Intention-aware heterogeneous graph attention networks for fraud transactions detection. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, 3280–3288.

Liu, Y.; Ao, X.; Qin, Z.; Chi, J.; Feng, J.; Yang, H.; and He, Q. 2021b. Pick and Choose: A GNN-Based Imbalanced Learning Approach for Fraud Detection. In *Proceedings of the Web Conference 2021*, 3168–3177.

Liu, Z.; Chen, C.; Li, L.; Zhou, J.; Li, X.; Song, L.; and Qi, Y. 2019. Geniepath: Graph neural networks with adaptive receptive paths. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 4424–4431.

Liu, Z.; Dou, Y.; Yu, P. S.; Deng, Y.; and Peng, H. 2020. Alleviating the Inconsistency Problem of Applying Graph Neural Network to Fraud Detection. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, 1569–1572.

Luan, S.; Hua, C.; Xu, M.; Lu, Q.; Zhu, J.; Chang, X.-W.; Fu, J.; Leskovec, J.; and Precup, D. 2023. When Do Graph Neural Networks Help with Node Classification? Investigating the Homophily Principle on Node Distinguishability. In *Advances in Neural Information Processing Systems*.

McAuley, J. J.; and Leskovec, J. 2013. From amateurs to connoisseurs: modeling the evolution of user expertise through online reviews. In *Proceedings of the 22nd international conference on World Wide Web*, 897–908.

Peng, H.; Zhang, R.; Dou, Y.; Yang, R.; Zhang, J.; and Yu, P. S. 2021. Reinforced Neighborhood Selection Guided Multi-Relational Graph Neural Networks. *ACM Transactions on Information Systems*, 40(4): 69:1–69:46.

Rayana, S.; and Akoglu, L. 2015. Collective Opinion Spam Detection: Bridging Review Networks and Metadata. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 985–994.

Shi, F.; Cao, Y.; Shang, Y.; Zhou, Y.; Zhou, C.; and Wu, J. 2022. H2-FDetector: A GNN-Based Fraud Detector with Homophilic and Heterophilic Connections. In *Proceedings of the ACM Web Conference 2022*, 1486–1494.

Tang, J.; Li, J.; Gao, Z.; and Li, J. 2022. Rethinking graph neural networks for anomaly detection. In *International Conference on Machine Learning*, 21076–21089.

Veličković, P.; Cucurull, G.; Casanova, A.; Romero, A.; Liò, P.; and Bengio, Y. 2018. Graph Attention Networks. In *International Conference on Learning Representations*.

Wang, D.; Lin, J.; Cui, P.; Jia, Q.; Wang, Z.; Fang, Y.; Yu, Q.; Zhou, J.; Yang, S.; and Qi, Y. 2019a. A semi-supervised graph attentive network for financial fraud detection. In *2019 IEEE International Conference on Data Mining*, 598–607.

Wang, X.; He, X.; Wang, M.; Feng, F.; and Chua, T.-S. 2019b. Neural graph collaborative filtering. In *Proceedings of the 42nd international ACM SIGIR conference on Research and development in Information Retrieval*, 165–174.

Wang, Y.; Zhang, J.; Huang, Z.; Li, W.; Feng, S.; Ma, Z.; Sun, Y.; Yu, D.; Dong, F.; Jin, J.; et al. 2023. Label Information Enhanced Fraud Detection against Low Homophily in Graphs. In *Proceedings of the ACM Web Conference 2023*, 406–416.

Weber, M.; Domeniconi, G.; Chen, J.; Weidele, D. K. I.; Bellei, C.; Robinson, T.; and Leiserson, C. E. 2019. Anti-Money Laundering in Bitcoin: Experimenting with Graph Convolutional Networks for Financial Forensics. arXiv:1908.02591.

Wu, J.; Zhang, C.; Liu, Z.; Zhang, E.; Wilson, S.; and Zhang, C. 2022. Graphbert: Bridging graph and text for malicious behavior detection on social media. In *2022 IEEE International Conference on Data Mining*, 548–557.

Xu, K.; Hu, W.; Leskovec, J.; and Jegelka, S. 2019. How Powerful are Graph Neural Networks? In *International Conference on Learning Representations*.

Yang, L.; Li, M.; Liu, L.; Niu, B.; Wang, C.; Cao, X.; and Guo, Y. 2021. Diverse Message Passing for Attribute with Heterophily. In *Advances in Neural Information Processing Systems*.

Ying, R.; He, R.; Chen, K.; Eksombatchai, P.; Hamilton, W. L.; and Leskovec, J. 2018. Graph convolutional neural networks for web-scale recommender systems. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 974–983.

Zhang, G.; Wu, J.; Yang, J.; Beheshti, A.; Xue, S.; Zhou, C.; and Sheng, Q. Z. 2021. Fraudre: Fraud detection dual-resistant to graph inconsistency and imbalance. In *2021 IEEE International Conference on Data Mining*, 867–876.

Zhang, M.; and Chen, Y. 2018. Link Prediction Based on Graph Neural Networks. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, 5171–5181.

Zhu, J.; Rossi, R. A.; Rao, A.; Mai, T.; Lipka, N.; Ahmed, N. K.; and Koutra, D. 2021. Graph Neural Networks with Heterophily. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

Zhu, J.; Yan, Y.; Zhao, L.; Heimann, M.; Akoglu, L.; and Koutra, D. 2020. Beyond Homophily in Graph Neural Networks: Current Limitations and Effective Designs. In *Advances in Neural Information Processing Systems*.