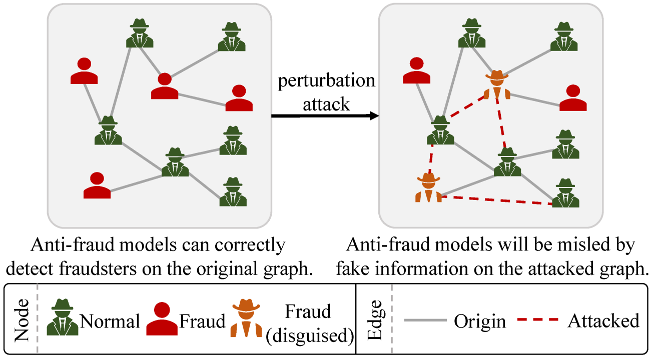
Safeguarding Fraud Detection from Attacks: A Robust Graph Learning Approach

Jiasheng Wu[1] , **Xin Liu**[1] , **Dawei Cheng**[1,2*] , **Yi Ouyang**[3] , **Xian Wu**[3] and **Yefeng Zheng**[3]
[1]Department of Computer Science and Technology, Tongji University, Shanghai, China
[2]Shanghai Artificial Intelligence Laboratory, Shanghai, China
[3]Jarvis Research Center, Tencent YouTu Lab, Shenzhen, China
{2331929, 2051277, dcheng}@tongji.edu.cn, {yiouyang, kevinxwu, yefengzheng}@tencent.com

Abstract

Financial fraud is one of the most significant social issues and has caused tremendous property losses. Graph neural networks (GNNs) have been applied to anti-fraud practices and achieved decent results. However, recent researches have discovered flaws in the robustness of fraud-detection models based on GNNs, enabling fraudsters to mislead them through attacks like data poisoning. In addition, most existing attack-defense models tend to study on ideal settings and lose information during truncation or filtering, which lowers their performances in complicated financial fraud cases. Therefore, in this paper, we propose a novel robust anti-fraud GNN model. In particular, we first design an attack algorithm tampering with both features and structures of graph data to simulate fraudsters' attacking behaviors in real-life complex fraud scenarios. Then we apply singular value decomposition to the graph and learn the decomposed matrices in a GNN model with specifically designed joint losses. This enables our model to learn the graph patterns in low-rank subspaces without losing too much detailed information and fit the graph structure to characteristics including class-homophily and sparsity to guarantee robustness. The proposed approach is experimented on real-world fraud datasets, which demonstrates its advantages in fraud detection and robustness compared with the state-of-the-art baselines.



Anti-fraud models can correctly detect fraudsters on the original graph.

Anti-fraud models will be misled by fake information on the attacked graph.

Figure 1: The illustration of perturbation attack on the graph. Fraud nodes will imitate the features of normal nodes and modify the structure to disturb the detector's judgment.

1 Introduction

In recent years, financial fraud has become a serious social problem that affects not only the property security of individuals and enterprises, but also the stability and credit of the financial system. The survey [PricewaterhouseCoopers, 2022] reveals that 46% of surveyed organisations reported suffering from some form of fraud or economic crime within the last 24 months. Increasingly, it appears that financial fraud has moved from the fringes of financial market activity to become a widespread type of behavior [Reurink, 2019], which has posed great threats to economic growth and social well-being like employment and personal safety. As the number of fraud cases has intensively risen in recent years, fraud detection is becoming an increasingly critical topic [Khedmati *et al.*, 2020]. Since graphs can effectively capture long-range correlations among inter-dependent data objects [Akoglu *et al.*, 2015], recent advanced researches have started to model and identify fraud behaviors on the basis of graphs [Xiang *et al.*, 2023; Zhang *et al.*, 2024]. Among many graph-based research methods, Graph Neural Networks (GNNs) have become a widely applied graph analysis method [Liang *et al.*, 2023]. These methods can achieve a certain level of accuracy and efficiency in identifying financial fraud in various fields.

However, whenever it becomes known that one detection method is in place, criminals will adjust their fraud strategies accordingly and attempt alternative ones [Bolton and Hand, 2002]. The survey [PricewaterhouseCoopers, 2022] also reveals a trend that could threaten the social order to a great extent: organized crime groups are becoming more specialized and professional. This can be illustrated by their attempts to conceal their fraudulent behavior as legitimate. As is shown in Figure 1, fraud nodes will imitate the features of normal nodes and modify the graph structure to disturb the detector's judgment. Such a perturbation attack on the graph often leads to a decrease in the accuracy of the detector's judgment, which has become an urgent crisis and challenge for the finance industry and academia. If not properly

[*]Corresponding Author is Dawei Cheng.

addressed, fraudsters can exploit the vulnerability of those fraud-detecting models, causing severe property losses and social distrust of the existing financial system.

Existing researches have made decent progress in two critical aspects: some of them have improved the accuracy of models in fraud detection, while others have enhanced the robustness of models in universal scenarios. However, as real-world fraudsters continually evolving their fraud methods and carrying out purposed attacks on models, there still exist new challenges to be settled. First, for most existing GNN-based anti-fraud models, they do not take into account the increasingly complex financial-fraud scenarios consisting of adversarial attacks from different aspects. They rely heavily on the original graph which may be artificially perturbed by fraudsters and therefore become vulnerable during the process of node aggregation. On the other side, for those models specializing in the defense of graph adversarial attacks, they tend to study on ideal and unpractical environments and datasets, failing to focus on real-life based financial settings. And mainstream methods based on direct truncation or filtering [Wu *et al.*, 2022; Huang *et al.*, 2023] may lose parts of detailed information of the graph, which will to some extent affect their overall performance on fraud-detection missions in spite of their high robustness.

These motivate us to design a novel robust anti-fraud graph neural network framework to address the two main dilemmas. In this paper, we first design an algorithm which simultaneously perturbs node features and topological structures of graph data to simulate the possible multifaceted attacks from fraudsters, and then propose a novel GNN-based model with graph learning and singular value decomposition (SVD) called GLSGNN to defend against the attacks. In particular, there are two key points in the model: (i) design novel joint loss functions for the model by exploiting the properties of pure graph data to improve its robustness while maintaining classification precision; and (ii) decompose the graph and apply them as learnable parameters, which enables the model to learn more patterns of the graph in low-rank subspaces with sufficient details. Specifically, we first employ the method of SVD to decompose the adjacency matrix of the attacked graph into three matrices. Each of these matrices will be independently learned as a parameter during the update of the model. Then, we optimize the model with a specially designed loss function containing weighed penalization based on sparsity, class-homophily, and classification results and try to recover the pure graph. For the attack part, we select different proportions of fraud nodes. We not only falsify parts of their features to disguise them as non-fraud ones, but also reconnect them to innocent nodes to perturb the graph structure. We conduct experiments on a real-world medical insurance dataset [Gupta, 2019] and Amazon fraud dataset [Zhang *et al.*, 2020], which demonstrates our model's superior performance on both fraud detection and robustness. Contributions of our work are summarized as follows:

- To the best of our knowledge, this is the first work that addresses the crucial anti-fraud problem in financial fields with multifaceted attacks on data by modeling the conspiracy defraud in a learnable robust graph neural network, which provides a solution to the newest and most advanced fraud patterns.
- In order to simulate real-world fraud scenarios, we design poisoning attacks on both the features of nodes and the edge structure. To defend against the attacks, We propose a novel anti-fraud graph neural network model learning decomposed adjacent matrix under the constraint of an elaborate joint loss function, which is capable of handling the variability of attack patterns while keeping high classification precision.
- Extensive experiments on real-world fraud datasets show that our proposed method outperforms the compared state-of-the-art baselines in fraud detection.

2 Related Works

2.1 Graph Neural Networks for Fraud Detection

By mining the hidden features of nodes and focusing on the complex relationships between nodes, graph neural networks (GNNs) have become pivotal in fraud detection missions [Cheng *et al.*, 2020; Ma *et al.*, 2023], with recent researches adopting fraud detection strategies such as SemiGNN [Wang *et al.*, 2019a] using attention mechanisms and information aggregation, [Zhang *et al.*, 2023] introducing a rule mining module refined with expert knowledge, CARE-GNN [Dou *et al.*, 2020], PC-GNN [Liu *et al.*, 2021] utilizing neighbor selector for imbalanced supervised learning on graphs and so on. Other works also focus on different node types and various relationships between nodes. They utilize heterogeneous information networks to enhance their capabilities in fraud detection. FdGars [Wang *et al.*, 2019b], HACUD [Hu *et al.*, 2019], FRAUDRE [Zhang *et al.*, 2021] focus on the features of different nodes or edges in real-world fraud detection scenarios. H2-FDetector [Shi *et al.*, 2022] considers both homophilic and heterophilic connections and applies aggregation strategies separately. GAGA [Wang *et al.*, 2023], a novel group aggregation enhanced transformer, provides a portable method to cope with the low homophily issue. However, these works rely heavily on the original graph and thus make it difficult to cope with fraudsters' disguises and perturbation, which poses a great threat to the security of the anti-fraud system. In contrast, we aim to devise our model and improve its robustness from the perspective of possible attacks on graph data in real-world financial fraud cases, which fills the gap in the field and can inspire more work in the community.

2.2 Graph Adversarial Learning Method

In order to reduce the influence of adversarial attacks on graphs, researches have been made on defense methods. The defense methods on graph data can be mainly divided into four parts: disturbance detection, graph purification, adversarial training, and robust model. DONE [Bandyopadhyay *et al.*, 2020] uses two parallel autoencoders to deal with structural and attribute anomalies respectively and detect abnormal disturbances of nodes. RTGNN [Qian *et al.*, 2023] generates pseudo-labels on unlabeled nodes and supervises nodes with different types of labels adaptively for effective learning while minimizing the impact of noisy labels. GraphDefense [Wang *et al.*, 2019c], BVAT [Deng *et al.*, 2019], LAT-GCN [Jin and Zhang, 2019] construct a powerful perturbed

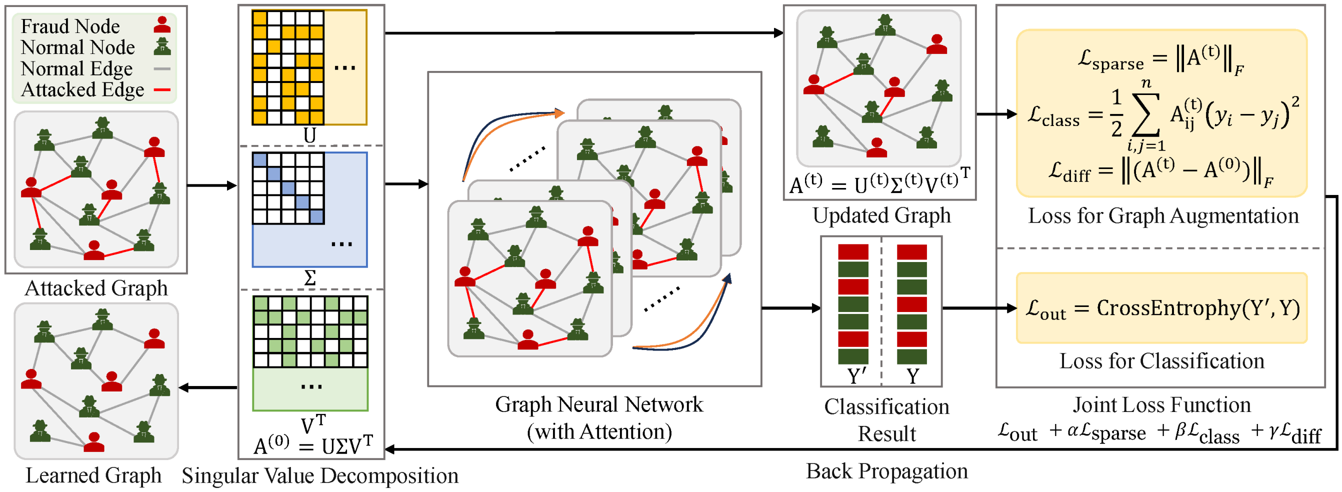

Fraud Node
Normal Node
Normal Edge
Attacked Edge
Attacked Graph
U
Σ
V^T
$A^{(0)} = U\Sigma V^T$
Learned Graph
Singular Value Decomposition
Graph Neural Network
(with Attention)
$A^{(t)} = U^{(t)}\Sigma^{(t)}V^{(t)T}$
Updated Graph
Y′
Y
Classification
Result
$\mathcal{L}_{sparse} = \|A^{(t)}\|_F$
$\mathcal{L}_{class} = \frac{1}{2}\sum_{i,j=1}^{n} A^{(t)}_{ij}(y_i - y_j)^2$
$\mathcal{L}_{diff} = \|(A^{(t)} - A^{(0)})\|_F$
Loss for Graph Augmentation
$\mathcal{L}_{out} = \mathrm{CrossEntropy}(Y', Y)$
Loss for Classification
Joint Loss Function
$\mathcal{L}_{out} + \alpha\mathcal{L}_{sparse} + \beta\mathcal{L}_{class} + \gamma\mathcal{L}_{diff}$
Back Propagation

Figure 2: The illustration of the proposed GLSGNN's architecture. We first employ SVD to decompose the adjacency matrix. The three matrices decomposed from the input graph will be independently updated as parameters and projected back to approximate the graph structure in the GNN model. Then, we optimize the model with a joint loss function containing weighed penalization to make the graph close to the patterns of a clean graph while learning the classification criteria.

data close to the original data through adversarial training, and ultimately obtain a more robust model on these perturbations. Recent typical robust models employ various strategies such as RGCN [Zhu *et al.*, 2019], RoHe [Zhang *et al.*, 2022] using attention mechanisms, Pro-GNN [Jin *et al.*, 2020] learning the clean graph structure, RT-GCN [Wu *et al.*, 2022], GSR [Zhao *et al.*, 2023] considering graph structure refinement, Mid-GCN [Huang *et al.*, 2023] exploring mid-frequency signals to enhance their effectiveness against adversarial attacks in graph-based scenarios. However, most existing defense methods still have flaws in our situation. In spite of their robustness, popular graph refinement algorithms based on truncation and filtering will break the graph structure and cause information losses, which makes them less competitive than those fraud detection models in overall performances on less disturbed graphs. Hence, we introduce a novel anti-fraud model that learns a decomposed adjacent matrix, constrained by an elaborate joint loss function, which makes full use of the graph details to enhance the accuracy of fraud detection while keeping its robustness.

3 Proposed Method

In this section, we introduce the method proposed in detail. First, we present the definition of the problem. Second, we show the overall structure of our framework GLSGNN. Third, we explain the design of every part of GLSGNN, including the singular value decomposition, the convolutional layer, and the learning of decomposed adjacent matrix. Lastly, we introduce the optimization strategy.

3.1 Problem Formulation

In this paper, we denote the graph as $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V} = \{\nu_0, \nu_1, \ldots, \nu_{N_v}\}$ represents the set of the nodes in the graph and $\mathcal{E} = \{e_{0,0}, e_{0,1}, \ldots, e_{i,j}, \ldots, e_{N_v,N_v}\}$ represents the set of the edges in the graph. $N_v = |\mathcal{V}|$ is the number of nodes, $N_e = |\mathcal{E}|$ is the number of edges, and $e_{i,j}$ denotes the edge from node ν_i to node ν_j, representing the relationship between those two nodes. Considering the two significant aspects of a graph: node features and topological structure, a graph can also be denoted as $\mathcal{G} = (\mathbf{A}, \mathbf{X})$, where $\mathbf{A} \in \mathbb{R}^{N_v \times N_v}$ is the adjacent matrix of the graph and $\mathbf{X} \in \mathbb{R}^{N_v \times N_f}$ is the feature matrix of the nodes. Simultaneously, the features of nodes can be denoted as $X = \{\boldsymbol{x_0}, \boldsymbol{x_1}, \ldots, \boldsymbol{x_{N_v}}\}$, with $\boldsymbol{x_i}$ being the feature of node ν_i and N_f being the number of features of one node. Furthermore, the set of the labels is denoted as $Y = \{y_0, y_1, ..., y_{N_v}\}$, in which y_i represents the label of node ν_i. The label of a normal node is 0, while the label of a fraud node is 1.

Based on reality, we design two different kinds of attacks on the graph to simulate the behavior of fraudsters. Firstly, since the features of nodes are important factors to classification and parts of them can be deliberately controlled by fraudsters, we perturb some of the values to perform feature attacks. Secondly, as fraudsters may impersonate others or participate with outsiders to conceal their identities by attaching themselves to non-fraudulent ones, we tend to add fake edges to the adjacent matrix to perform structure attacks. Different from most previous studies, we simultaneously apply attacks to features and graph structure, which is more challenging and closer to real life. Therefore, the problem we study in this paper can be presented as follows: given an attacked graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, we aim to get an ideal classification result on unlabeled nodes by learning both GNN parameters and the graph $\mathcal{G}$ in our proposed framework. In particular, our model needs to learn the classification criteria while purifying the graph to defend against attacks on data.

3.2 Model Architecture

The overall structure of our proposed framework is illustrated in Figure 2. In order to defend against the feature attacks and structure attacks mentioned above, we construct a novel model structure called GLSGNN. Firstly, we apply the singular value decomposition to the attacked graph to get the decomposed matrices of the adjacent matrix. Then, in addition to the common parameters of graph convolutional layers in GNN, we also set the decomposed matrices as learnable parameters in the model, aiming to fit the attacked graph to certain characteristics of pure ones. Besides the penalty for misclassified results, we design several loss functions specifically for the adjacent matrix to learn its sparsity and class-homophily. Besides, to avoid the learned graph being too different from the original one and containing too many noises, we add an extra restriction to it. Benefit from the combination of those methods, our model can make effective corrections to the attacked graph. At the same time, we can obtain a less interfered GNN model, which therefore performs better on the final classification mission.

3.3 Adversarial Learning with Structure Decomposition

We first decompose the adjacent matrix by using singular value decomposition (SVD), and the results will be applied as the initial values of three learnable parameters in the following GNN model which will be continuously updated and projected back as the approximation of the input graph. The motivation of this design is based on the fact that most existing low-rank methods merely preserving top-k singular components will inevitably impair the details of the graph [Deng *et al.*, 2022]. And several recent researches in other fields like [Wang *et al.*, 2021] have demonstrated that learning with decomposed matrices will help to get a low-rank structure and a better spectrum decay pattern of the original matrix by learning in low-rank subspaces, which further inspires us to perform the decomposition on the adjacent matrix of the graph. The SVD process can be formulated as:

$$\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^{\top}. \tag{1}$$

$\mathbf{U}$ is the left singular vector matrix. Every its column vector $\boldsymbol{u_i}$ is the eigenvector of $\mathbf{A}\mathbf{A}^{\top}$. The eigenvalue λ_i^u of $\mathbf{A}\mathbf{A}^{\top}$ satisfies:

$$\left(\mathbf{A}\mathbf{A}^{\top}\right)\boldsymbol{u_i} = \lambda_i^u \boldsymbol{u_i}. \tag{2}$$

$\mathbf{V}$ is the right singular vector matrix. Every its column vector $\boldsymbol{v_i}$ is the eigenvector of $\mathbf{A}^{\top}\mathbf{A}$. The eigenvalue λ_i^v of $\mathbf{A}^{\top}\mathbf{A}$ satisfies:

$$\left(\mathbf{A}^{\top}\mathbf{A}\right)\boldsymbol{v_i} = \lambda_i^v \boldsymbol{v_i}. \tag{3}$$

$\mathbf{\Sigma}$ is the singular value matrix and it's a diagonal matrix. Every element σ_i on its diagonal is the singular value of $\mathbf{A}$. The eigenvalue λ_i^u of $\mathbf{A}\mathbf{A}^{\top}$ satisfies:

$$\sigma_i = \sqrt{\lambda_i^u}. \tag{4}$$

Therefore SVD can be expressed in another equivalent form, in which r represents the rank of $\mathbf{A}$:

$$\mathbf{A} = \sum_{i=1}^{r} \boldsymbol{u_i}\sigma_i\boldsymbol{v_i}^{\top}. \tag{5}$$

It has to be additionally mentioned that choosing SVD as the method of matrix decomposition has more advantages. Each singular value multiplied by its corresponding left and right singular vectors will result in a matrix with rank 1. Each of them contains one part of the graph's information, and larger singular values refer to matrices representing more important patterns of the original adjacent matrix of the graph. Hence, by learning those decomposed matrices, the model is able to view the graph in different subspaces, which helps it obtain more information that may not be easily extracted from the original adjacent matrix.

We denote the initial value of the approximate adjacent matrix as $\mathbf{A^{(0)}}$. Considering the meaning of adjacent matrix in real life, we pass it through a ReLU function to eliminate the negative values, which can be formulated as:

$$\mathbf{A^{(0)}} = \mathrm{ReLU}\left(\mathbf{U}\mathbf{\Sigma}\mathbf{V}^{\top}\right). \tag{6}$$

3.4 Graph Convolutional Layer with Attention

Among all these GNN models, we choose a two layer GAT [Velickovic *et al.*, 2017] as the base of our model. By assigning different weights to neighboring nodes, the attention mechanism can significantly enhance the robustness of graph convolutional layers, which aligns with our ultimate goal. Except from the weights $\mathbf{W}$ in convolutional layers, we set other three learnable parameters $\mathbf{U^{(t)}}$, $\mathbf{\Sigma^{(t)}}$, $\mathbf{V^{(t)}}$ and initialize $\mathbf{U^{(0)}}$, $\mathbf{\Sigma^{(0)}}$, $\mathbf{V^{(0)}}$ with the $\mathbf{U}$, $\mathbf{\Sigma}$, $\mathbf{V}$ obtained in SVD. The approximation of the adjacent matrix $\mathbf{A^{(t)}}$ used in the learning of the model can be formulated as:

$$\mathbf{A^{(t)}} = \mathrm{ReLU}\left(\mathbf{U^{(t)}}\mathbf{\Sigma^{(t)}}\mathbf{V^{(t)}}^{\top}\right), \tag{7}$$

where t is the iteration number.

The attention mechanism can be calculated as follows. Firstly, we evaluate the similarity between nodes, where s_{ij} is the similarity coefficient, NN is a single layer feed-forward neural network, and $\boldsymbol{h_i}$ is the embedding of node ν_i:

$$s_{ij} = \mathrm{NN}\left(\mathbf{W}\boldsymbol{h_i}, \mathbf{W}\boldsymbol{h_j}\right). \tag{8}$$

We can get the attention coefficient α_{ij} through the following formulation after passing the similarity coefficient s_{ij} through a LeakyReLU function to activate it, where $\mathcal{N}_i$ is the set of neighbors of ν_i:

$$\alpha_{ij} = \frac{\exp\left(\mathrm{LeakyReLU}\left(s_{ij}\right)\right)}{\sum_{k\in\mathcal{N}_i}\exp\left(\mathrm{LeakyReLU}\left(s_{ik}\right)\right)}. \tag{9}$$

Finally, the output of node embedding $\boldsymbol{h_i'}$ after passing through the convolution layer can be formulated as follows, where we choose ELU as the activation function and K is the number of all the attention heads:

$$\boldsymbol{h_i'} = \|_{k=1}^{K}\mathrm{ELU}\left(\sum_{\mathrm{j}\in\mathcal{N}_\mathrm{i}}\alpha_{\mathrm{ij}}^{\mathrm{k}}\mathbf{W^k}\boldsymbol{h_j}\right). \tag{10}$$

3.5 Graph Structure Learning

In those real-life pure graph data, most of the nodes are only connected to a small number of other nodes [Zhou *et al.*, 2013]. The scale of connection between nodes is relatively small compared to the size of the whole graph, therefore the adjacency matrix of the entire graph will remain sparse. As analyzed above, fraudsters tend to connect themselves to other innocent nodes and add edges to the graph, which will to some extent make the graph denser. In order to learn the characteristics of sparsity, we consider adding a penalty for the extra edges in the graph. Therefore we introduce the frobenius form, and denote this loss function as $\mathcal{L}_{\text{sparse}}$.

$$\mathcal{L}_{\text{sparse}} = \left\|\mathbf{A^{(t)}}\right\|_F = \sqrt{\sum_i \sum_j \mathbf{A_{ij}^{(t)^2}}}. \tag{11}$$

The sparsity of the graph in a normal adjacency matrix means that elements with the value of 1 are as few as possible. Although the adjacency matrix $\mathbf{A^{(t)}}$ no longer only consists of binary values of 0 and 1 in the learning process, having fewer large values in the adjacent matrix still represents that the graph is sparser. Therefore, using frobenius norm as a constraint can have a decent effect.

Latest researches [Choi *et al.*, 2022; Zhao *et al.*, 2021] show that it is more helpful for GNNs to learn node classification criteria when there exist many edges in the graph connecting nodes of the same class. This kind of connection pattern between nodes is called class-homophily. This is also easy to understand in real-life circumstances. For example, in a social network, similar people with the same backgrounds and behaviors will have relationships with each other.

In our anti-fraud practice, the characteristic of class-homophily still exists. Firstly, conspiracy fraud involving gangs has become increasingly common in recent years. Secondly, an individual on suspicion of fraud is likely to behave fraudulently in several other records, which are usually connected in the graph structure. Hence, we evaluate the class similarity between nodes by designing a loss function denoted as $\mathcal{L}_{\text{class}}$. If the connected nodes have different labels, $\mathcal{L}_{\text{class}}$ is going to increase for punishment.

$$\mathcal{L}_{\text{class}} = \frac{1}{2}\sum_{i,j=1}^{n} \mathbf{A_{ij}^{(t)}}\left(y_i - y_j\right)^2. \tag{12}$$

3.6 Loss Function and Optimization

Except from $\mathcal{L}_{\text{sparse}}$ and $\mathcal{L}_{\text{class}}$, the entire loss function contains two more parts: Firstly, we denote the loss of classification as $\mathcal{L}_{\text{out}}$. This is the common loss function of GNNs, which is applied to evaluate the gap between the original labels and the predicted results. We choose to use the cross-entropy function to calculate it, where $\hat{y}_i$ is the predicted label, because the cross-entropy function can not only present the performance of the model on node classification, but also keep a decent rate in the gradient descent process.

$$\mathcal{L}_{\text{out}} = -\sum_{i=1}^{n}\left(y_i \log\left(\hat{y}_i\right) + \left(1 - y_i\right)\log\left(1 - \hat{y}_i\right)\right). \tag{13}$$

Secondly, we don't want the adjacency matrix to change too much during the learning of the model, which will introduce excessive noises and make the learned graph unreal. Therefore, we add another loss function $\mathcal{L}_{\text{diff}}$ to make a constraint to the changes. By calculating the differences between the adjacent matrix in the learning process and the initial one, we can keep it in an appropriate range.

$$\mathcal{L}_{\text{diff}} = \left\|\left(\mathbf{A^{(t)}} - \mathbf{A^{(0)}}\right)\right\|_F = \sqrt{\sum_i \sum_j \left(\mathbf{A_{ij}^{(t)}} - \mathbf{A_{ij}^{(0)}}\right)^2}. \tag{14}$$

Dataset	# Edges	# Nodes	# Features	# Frauds
Medical Insurance	89218	13643	26	5141
Amazon Fraud	4,398,392	11944	25	821

Table 1: Dataset statistics including the number of edges, the number of nodes, the feature dimensions and the number of fraudulent nodes of the chosen two datasets.

Finally, we add weights between all the parts mentioned above to formulate the entire joint loss function $\mathcal{L}$, where α, β, γ are hyper-parameters:

$$\mathcal{L} = \mathcal{L}_{\text{out}} + \alpha\mathcal{L}_{\text{sparse}} + \beta\mathcal{L}_{\text{class}} + \gamma\mathcal{L}_{\text{diff}}. \tag{15}$$

The model can be optimized through standard stochastic gradient descent-based methods. We use a default Adam optimizer [Kingma and Ba, 2014] with a learning rate of 2×10^{-3} to execute the optimization process.

4 Experiments

4.1 Experiment Settings

Datasets. To demonstrate the effectiveness of our model, we choose two real-world datasets. The first one is a medical insurance dataset based on real medicare claims [Gupta, 2019] containing inpatient data, outpatient data, and beneficiary details data. Due to the limit of computing resources, we choose a subset of the original dataset and join different kinds of data according to the claimID. Experts from our cooperating organization help us analyze the data and label fraud behaviors. We view each claim as a node, and if two claims within a certain time period have the same provider or beneficiary, we connect these two nodes with an edge. We also filter out several unhelpful features. The other one is the Amazon fraud dataset, a multi-relational graph dataset built upon the Amazon review dataset [Dou *et al.*, 2020]. 25 handcrafted features from [Zhang *et al.*, 2020] are used in the dataset including product data, user data, etc. Users are nodes in the graph, and have three relations: U-P-U, U-S-U, and U-V-U, which respectively connect users reviewing same products, users giving same ratings and users with similar reviews. We use the union of all these relations in our experiment. Table 1 shows the detailed statistics of the datasets.

Baselines. We employ the following state-of-the-art methods on our benchmark dataset to highlight the effectiveness of the proposed GLSGNN, which are mainly divided into two categories. Firstly, we select popular GNN-based fraud detection methods including GraphSage [Hamilton *et al.*, 2017], GEM [Liu *et al.*, 2018], FdGars [Wang *et al.*, 2019b], GraphConsis [Liu *et al.*, 2020], CARE-GNN [Dou *et al.*, 2020], PC-GNN [Liu *et al.*, 2021], FRAUDRE [Zhang *et al.*, 2021], H^2-FDetector [Shi *et al.*, 2022], GAGA [Wang *et al.*, 2023]. Secondly, we also select several graph adversarial defense methods including GCN [Kipf and Welling, 2016], GAT [Velickovic *et al.*, 2017], RGCN [Zhu *et al.*, 2019], ProGNN [Jin *et al.*, 2020], GARNET [Deng *et al.*, 2022], RT-GCN [Wu *et al.*, 2022], Mid-GCN [Huang *et al.*, 2023]. In these experiments, the tasks are learned independently.

Evaluation Metrics and Parameter Settings. In the experiment, we utilize area under curve (AUC), macro average of F1 score (F1-macro), recall score and accuracy to evaluate the effectiveness of our model, which ensures the validity of our experiment. For the hyper-parameters, in consideration of the overall performance, we set the weight α of $\mathcal{L}_{\text{sparse}}$ to 0.0005, the weight β of $\mathcal{L}_{\text{class}}$ to 0.0005, and the weight γ of $\mathcal{L}_{\text{diff}}$ to 0.1. We adopt a GNN framework with 2 graph convolutional layers, 8 hidden features and 5 attention heads.

Adversarial Attack Method. We adopt a poisoning attack rather than an evasion attack, since the former perturbs the data before the training process of the model while the latter only performs attacks on test data. In most cases, it is more difficult to defend a poisoning attack because it will directly affects the learning of parameters of the model, which is also closer to real-life scenarios. Specifically, We simultaneously attack the features and structures of the graph data to simulate complex fraud behaviors, and we will first introduce our strategy on the medical insurance dataset:

- **Feature Attack:** we first randomly select some fraud nodes in an attempt to disguise them as normal nodes. Then, we manually choose parts of the features which have the risk of being controlled by fraudsters like reimbursements, deductible amount, etc. After calculating the data distribution of normal nodes, we randomly modify those picked features of the selected fraud nodes to random values between 40%-60% quantiles of the feature attributes of normal nodes.
- **Structural Attack:** we are inspired by real-world crimes and base on the connection rules mentioned above to perturb the structure. For the fraud nodes previously chosen, we change the provider of it to the one of a random normal claim within the limited time period. Therefore these nodes will be reconnected to other normal ones, thereby perturbing the entire graph structure.

And for the Amazon fraud dataset, since the data are well-encapsulated and thus being hard for us to distinguish the specific features and nodes, we slightly modify the attack patterns by perturbing several random features of those selected fraud nodes to the range of normal ones and creating fake edge links directly on the adjacent matrix.

4.2 Fraud Detection Performance

We compare our model with the state-of-the-art anti-fraud methods to validate its performance in the original fraud detection task. Table 2 shows the result of the experiment. From the statistics, it can be seen that our proposed GLSGNN model achieves excellent performances in the cases, and surpasses most of the baseline methods. Especially in the medical insurance dataset, our model consistently outperforms the baseline with AUC values ranging from 5.9% to 45.6%, F1 score ranging from 5.0% to 79.0%, recall score ranging from 10.8% to 46.4%, and accuracy ranging from 5.0% to 64.6%, representing a notable enhancement. Due to the design of learning matrices decomposed from the full input graph, our model doesn't suffer from the flaws of existing defense models of missing important graph details, thus showing its strong ability in fraud detection precision.

	Model	AUC	F1	Recall	Accuracy
Amazon Fraud	GraphSage	0.9184	0.7913	0.7902	0.9668
	GEM	0.8962	0.6038	0.5722	0.9193
	FdGars	0.7388	0.4586	0.4599	0.7089
	GraphConsis	0.9241	0.7702	0.7147	0.9381
	CARE-GNN	0.9437	0.8966	0.8927	0.9654
	PC-GNN	0.9660	0.8791	0.9072	0.9211
	FRAUDRE	0.9502	0.8942	0.8953	0.9635
	H^2-FDetector	0.9596	0.7848	**0.9085**	0.8981
	GAGA	0.9665	0.9060	0.8927	0.9688
	GLSGNN	**0.9672**	**0.9119**	0.8940	**0.9711**
Medical Insurance	GraphSage	0.6891	0.6356	0.6319	0.6793
	GEM	0.5012	0.3888	0.5000	0.6362
	FdGars	0.5540	0.5338	0.5540	0.6472
	GraphConsis	0.5190	0.4837	0.5119	0.6070
	CARE-GNN	0.5882	0.4954	0.5218	0.6165
	PC-GNN	0.5290	0.5010	0.5064	0.6348
	FRAUDRE	0.6623	0.4138	0.4782	0.4332
	H^2-FDetector	0.6690	0.6627	0.4851	0.6344
	GAGA	0.6070	0.6150	0.6136	0.6358
	GLSGNN	**0.7295**	**0.6958**	**0.7002**	**0.7130**

Table 2: Experiment on the performance of fraud detection on different datasets. We compare our model with other popular GNN-based fraud detection models.The result proves that our method significantly outperforms recent anti-fraud baselines in most metrics.

4.3 Adversarial Attack Performance

In the experiment, we apply the attack algorithm introduced above on both of the datasets and set the attack rate at every 5% from 0% to 25%. From table 3, compared with popular graph adversarial defense methods, our model shows a better performance, which maintains a stable level on all metrics with small fluctuations within an acceptable range. Our model keeps high AUC scores over 0.96 and F1 scores over 0.9 on the amazon-fraud dataset. And simultaneously, on all attack ratios on the medical insurance dataset, our model reaches AUC scores over 0.72 and F1 scores over 0.69, which shows a greater advantage over the defense baselines.

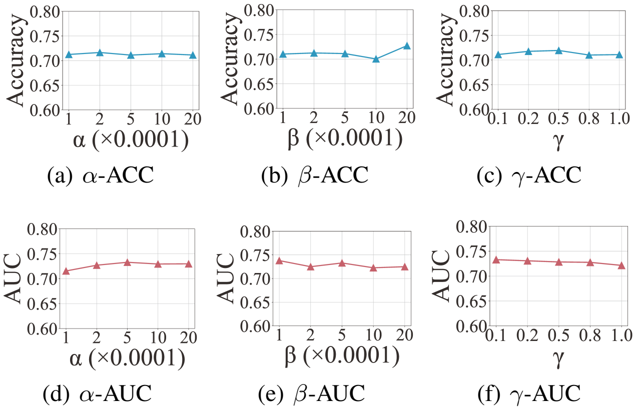
And it is worth mentioning that many popular baselines can not get satisfying results on the medical insurance dataset. The most probable reason is that this dataset has only 89218 edges, far less than the Amazon fraud dataset, which makes some models struggle to aggregate information. However in real-world fraud cases, the number of edges in the data varies a lot, and the superior performance of our model on the medical insurance dataset further improves the ability and practical value of it on different real-world scenarios. In conclusion, these outcomes strongly affirm the superiority of our proposed model, showing its high precision in detecting fraud nodes and the stability against intentional perturbations.

4.4 Parameter Sensitivity and Ablation Study

We first vary the important hyper-parameters of α, β, and γ in the loss function to discover their effects on classification results. We make this experiment on the medical insurance dataset and set the attack rate to a peak value of 25%. Figure

	Model	0%		5%		10%		15%		20%		25%	
		AUC	F1	AUC	F1	AUC	F1	AUC	F1	AUC	F1	AUC	F1
Amazon Fraud	GCN	0.8296	0.5615	0.8236	0.5568	0.8241	0.5591	0.8243	0.5657	0.8235	0.5512	0.8238	0.5642
	GAT	0.9160	0.8655	0.8971	0.8622	0.8864	0.8597	0.9096	0.8582	0.8888	0.8646	0.8832	0.8622
	RGCN	0.8272	0.6035	0.8175	0.5828	0.8169	0.5862	0.8168	0.5828	0.8166	0.5799	0.8153	0.5729
	ProGNN	0.7910	0.5675	0.7913	0.5700	0.7909	0.5669	0.7907	0.5673	0.7909	0.5660	0.7906	0.5647
	GARNET	0.9130	0.8278	0.9131	0.8231	0.9138	0.8148	0.9163	0.8113	0.9161	0.8158	0.9134	0.8142
	RT-GCN	**0.9759**	0.9114	0.9761	0.9093	0.9572	0.8987	0.9683	0.9008	0.9632	0.8993	**0.9660**	0.9024
	Mid-GCN	0.9424	0.7810	0.9429	0.7845	0.9412	0.7535	0.9400	0.7558	0.9402	0.7637	0.9403	0.7669
	GLSGNN	0.9672	**0.9119**	**0.9736**	**0.9129**	**0.9681**	**0.9004**	**0.9657**	**0.9075**	**0.9660**	**0.9100**	0.9646	**0.9034**
Medical Insurance	GCN	0.6942	0.6005	0.6429	0.5353	0.6958	0.5686	0.6735	0.6070	0.6401	0.5064	0.6457	0.5790
	GAT	0.6883	0.6708	0.7106	0.6539	0.6822	0.6655	0.6865	0.6541	0.6852	0.6098	0.6861	0.6594
	RGCN	0.6095	0.6030	0.6078	0.6039	0.6022	0.6044	0.6179	0.6272	0.6148	0.6454	0.5569	0.5620
	ProGNN	0.5713	0.4828	0.5484	0.4853	0.5711	0.4687	0.6136	0.4395	0.6063	0.4345	0.6189	0.4418
	GARNET	0.6657	0.6523	0.6611	0.6410	0.6636	0.6236	0.6564	0.6293	0.6708	0.6129	0.6627	0.6452
	RT-GCN	0.5343	0.5297	0.5324	0.5087	0.5456	0.5044	0.5336	0.5128	0.5453	0.5043	0.5350	0.5363
	Mid-GCN	0.6434	0.6519	0.6384	0.6659	0.6325	0.6507	0.6392	0.6534	0.6275	0.6474	0.6309	0.6428
	GLSGNN	**0.7295**	**0.6958**	**0.7252**	**0.6938**	**0.7289**	**0.6950**	**0.7517**	**0.6916**	**0.7364**	**0.6950**	**0.7329**	**0.6924**

Table 3: Results of experiments on attack ratios ranging from 0% to 25%. We compare our model with other popular graph adversarial defense methods. The result proves that our method have excellent performances under each attack ratio in most metrics.



(a) α-ACC (b) β-ACC (c) γ-ACC

(d) α-AUC (e) β-AUC (f) γ-AUC

Figure 3: The result of classification with different hyper-parameters of α, β, and γ in the loss function.

3(a), 3(d) show the influence of the weight α of $\mathcal{L}_{\text{sparse}}$ varying from 0.0001 to 0.002. The result shows that our model performs better when increasing α from 0.0001 to 0.0005, and then remains stable. Figure 3(b), 3(e) show the influence of the weight β of $\mathcal{L}_{\text{class}}$ varying from 0.0001 to 0.002. The result slightly fluctuates when β exceeds 0.0005 as focusing too much on edges connecting similar nodes may introduce unnecessary noises. Figure 3(c), 3(f) show the influence of the weight γ of $\mathcal{L}_{\text{diff}}$ varying from 0.1 to 1. The AUC score shows a overall decline when we increase the value of γ, which is probably due to the fact that limiting changes to the graph will also to some extent restrict the model's ability to purify the graph. In conclusion, we can observe that our model shows a strong stability with different hyper-parameters. It implies that different loss items in our joint loss function can keep a dynamic balance during the learning of the model, which guarantees the capability of our model in complicated and volatile situations.

In order to demonstrate the effectiveness of the important components in our proposed method, we also conduct an ablation study. After ablating the SVD module, the AUC score falls by 4.6% and the F1 score falls by 3.0%. We also remove the additional items in the joint loss function and only keep the cross-entropy loss. In this situation, the AUC score falls by 2.9% and the F1 score falls by 2.1%. The result of this experiment validates the necessity of the components of our GLSGNN in robust fraud detection missions.

5 Conclusion

In this paper, in order to solve the two novel critical challenges on financial fraud detection: (i) anti-fraud GNN models are fragile facing attacks from fraudsters on graph data; and (ii) robust defense models have unsatisfying classification precision due to information losses during the process of truncation or filtering, we propose a novel robust anti-fraud framework GLSGNN. To mimic the behavior of real-life fraudsters, we simultaneously implement attacks on graph structure and features. In order to keep both high robustness and superior performance on classification, our model learns the adjacent matrix decomposed by SVD in a graph convolutional network with attention mechanism and specially designed loss functions, which enables the model to learn better patterns with a graph being continuously purified without losing too much details. Experiments on real-world fraud datasets demonstrate the outstanding performance of our GLSGNN on fraud detection and robustness compared with other baselines. The effectiveness of GLSGNN will also provide the real-life anti-fraud systems and practitioners with new ideas and inspirations to improve social security and justice, which will make contributions to economic growth and social well-being including employment, health and so forth.

Acknowledgements

This work was supported by the National Key R&D Program of China (Grant no. 2022YFB4501704), the National Natural Science Foundation of China (Grant no. 62102287), and the Shanghai Science and Technology Innovation Action Plan Project (Grant no. 22YS1400600 and 22511100700).

References

[Akoglu *et al.*, 2015] Leman Akoglu, Hanghang Tong, and Danai Koutra. Graph based anomaly detection and description: a survey. *Data mining and knowledge discovery*, 29:626–688, 2015.

[Bandyopadhyay *et al.*, 2020] Sambaran Bandyopadhyay, Lokesh N, Saley Vishal Vivek, and M Narasimha Murty. Outlier resistant unsupervised deep architectures for attributed network embedding. In *Proceedings of the 13th international conference on web search and data mining*, pages 25–33, 2020.

[Bolton and Hand, 2002] Richard J Bolton and David J Hand. Statistical fraud detection: A review. *Statistical science*, 17(3):235–255, 2002.

[Cheng *et al.*, 2020] Dawei Cheng, Xiaoyang Wang, Ying Zhang, and Liqing Zhang. Graph neural network for fraud detection via spatial-temporal attention. *IEEE Transactions on Knowledge and Data Engineering*, 34(8):3800–3813, 2020.

[Choi *et al.*, 2022] Yoonhyuk Choi, Jiho Choi, Taewook Ko, Hyungho Byun, and Chong-Kwon Kim. Finding heterophilic neighbors via confidence-based subgraph matching for semi-supervised node classification. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, pages 283–292, 2022.

[Deng *et al.*, 2019] Zhijie Deng, Yinpeng Dong, and Jun Zhu. Batch virtual adversarial training for graph convolutional networks. *arXiv preprint arXiv:1902.09192*, 2019.

[Deng *et al.*, 2022] Chenhui Deng, Xiuyu Li, Zhuo Feng, and Zhiru Zhang. Garnet: Reduced-rank topology learning for robust and scalable graph neural networks. In *Learning on Graphs Conference*, pages 3–1. PMLR, 2022.

[Dou *et al.*, 2020] Yingtong Dou, Zhiwei Liu, Li Sun, Yutong Deng, Hao Peng, and Philip S Yu. Enhancing graph neural network-based fraud detectors against camouflaged fraudsters. In *Proceedings of the 29th ACM international conference on information & knowledge management*, pages 315–324, 2020.

[Gupta, 2019] Rohit Anand Gupta. Healthcare provider fraud detection analysis. Website, 2019. https://www.kaggle.com/datasets/rohitrox/healthcare-provider-fraud-detection-analysis/data.

[Hamilton *et al.*, 2017] William L. Hamilton, Rex Ying, and Jure Leskovec. Inductive representation learning on large graphs. In *NIPS*, 2017.

[Hu *et al.*, 2019] Binbin Hu, Zhiqiang Zhang, Chuan Shi, Jun Zhou, Xiaolong Li, and Yuan Qi. Cash-out user detection based on attributed heterogeneous information network with a hierarchical attention mechanism. In *The Thirty-Third AAAI Conference on Artificial Intelligence*, 2019.

[Huang *et al.*, 2023] Jincheng Huang, Lun Du, Xu Chen, Qiang Fu, Shi Han, and Dongmei Zhang. Robust mid-pass filtering graph convolutional networks. In *Proceedings of the ACM Web Conference 2023*, pages 328–338, 2023.

[Jin and Zhang, 2019] Hongwei Jin and Xinhua Zhang. Latent adversarial training of graph convolution networks. In *ICML workshop on learning and reasoning with graph-structured representations*, volume 2, 2019.

[Jin *et al.*, 2020] Wei Jin, Yao Ma, Xiaorui Liu, Xianfeng Tang, Suhang Wang, and Jiliang Tang. Graph structure learning for robust graph neural networks. In *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 66–74, 2020.

[Khedmati *et al.*, 2020] Mohamad Khedmati, Masoud Erfani, and Mohammad GhasemiGol. Applying support vector data description for fraud detection. *arXiv preprint arXiv:2006.00618*, 2020.

[Kingma and Ba, 2014] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[Kipf and Welling, 2016] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.

[Liang *et al.*, 2023] Yongquan Liang, Qiuyu Song, Zhongying Zhao, Hui Zhou, and Maoguo Gong. Ba-gnn: Behavior-aware graph neural network for session-based recommendation. *Frontiers of Computer Science*, 17(6):176613, 2023.

[Liu *et al.*, 2018] Ziqi Liu, Chaochao Chen, Xinxing Yang, Jun Zhou, Xiaolong Li, and Le Song. Heterogeneous graph neural networks for malicious account detection. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, pages 2077–2085, 2018.

[Liu *et al.*, 2020] Zhiwei Liu, Yingtong Dou, Philip S. Yu, Yutong Deng, and Hao Peng. Alleviating the inconsistency problem of applying graph neural network to fraud detection. In *Proceedings of the 43nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2020.

[Liu *et al.*, 2021] Yang Liu, Xiang Ao, Zidi Qin, Jianfeng Chi, Jinghua Feng, Hao Yang, and Qing He. Pick and choose: a gnn-based imbalanced learning approach for fraud detection. In *Proceedings of the web conference 2021*, pages 3168–3177, 2021.

[Ma *et al.*, 2023] Jiacheng Ma, Fan Li, Rui Zhang, Zhikang Xu, Dawei Cheng, Yi Ouyang, Ruihui Zhao, Jianguang Zheng, Yefeng Zheng, and Changjun Jiang. Fighting against organized fraudsters using risk diffusion-based parallel graph neural network. In *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence*, pages 6138–6146, 2023.

[PricewaterhouseCoopers, 2022] PricewaterhouseCoopers. Pwc's global economic crime and fraud survey 2022. 2022. available at: https://www.pwc.com/gx/en/services/forensics/economic-crime-survey.html.

[Qian *et al.*, 2023] Siyi Qian, Haochao Ying, Renjun Hu, Jingbo Zhou, Jintai Chen, Danny Z Chen, and Jian Wu. Robust training of graph neural networks via noise governance. In *Proceedings of the Sixteenth ACM International Conference on Web Search and Data Mining*, pages 607–615, 2023.

[Reurink, 2019] Arjan Reurink. Financial fraud: A literature review. *Contemporary Topics in Finance: A Collection of Literature Surveys*, pages 79–115, 2019.

[Shi *et al.*, 2022] Fengzhao Shi, Yanan Cao, Yanmin Shang, Yuchen Zhou, Chuan Zhou, and Jia Wu. H2-fdetector: A gnn-based fraud detector with homophilic and heterophilic connections. In *Proceedings of the ACM Web Conference 2022*, pages 1486–1494, 2022.

[Velickovic *et al.*, 2017] Petar Velickovic, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, Yoshua Bengio, et al. Graph attention networks. *stat*, 1050(20):10–48550, 2017.

[Wang *et al.*, 2019a] Daixin Wang, Jianbin Lin, Peng Cui, Quanhui Jia, Zhen Wang, Yanming Fang, Quan Yu, Jun Zhou, Shuang Yang, and Yuan Qi. A semi-supervised graph attentive network for financial fraud detection. In *2019 IEEE International Conference on Data Mining (ICDM)*, pages 598–607. IEEE, 2019.

[Wang *et al.*, 2019b] Jianyu Wang, Rui Wen, Chunming Wu, Yu Huang, and Jian Xion. Fdgars: Fraudster detection via graph convolutional networks in online app review system. In *Companion Proceedings of The 2019 World Wide Web Conference*, pages 310–316, 2019.

[Wang *et al.*, 2019c] Xiaoyun Wang, Xuanqing Liu, and Cho-Jui Hsieh. Graphdefense: Towards robust graph convolutional networks. *arXiv preprint arXiv:1911.04429*, 2019.

[Wang *et al.*, 2021] Ruoxi Wang, Rakesh Shivanna, Derek Cheng, Sagar Jain, Dong Lin, Lichan Hong, and Ed Chi. Dcn v2: Improved deep & cross network and practical lessons for web-scale learning to rank systems. In *Proceedings of the web conference 2021*, pages 1785–1797, 2021.

[Wang *et al.*, 2023] Yuchen Wang, Jinghui Zhang, Zhengjie Huang, Weibin Li, Shikun Feng, Ziheng Ma, Yu Sun, Dianhai Yu, Fang Dong, Jiahui Jin, et al. Label information enhanced fraud detection against low homophily in graphs. In *Proceedings of the ACM Web Conference 2023*, pages 406–416, 2023.

[Wu *et al.*, 2022] Zhebin Wu, Lin Shu, Ziyue Xu, Yaomin Chang, Chuan Chen, and Zibin Zheng. Robust tensor graph convolutional networks via t-svd based graph augmentation. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 2090–2099, 2022.

[Xiang *et al.*, 2023] Sheng Xiang, Mingzhi Zhu, Dawei Cheng, Enxia Li, Ruihui Zhao, Yi Ouyang, Ling Chen, and Yefeng Zheng. Semi-supervised credit card fraud detection via attribute-driven graph representation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 14557–14565, 2023.

[Zhang *et al.*, 2020] Shijie Zhang, Hongzhi Yin, Tong Chen, Quoc Viet Nguyen Hung, Zi Huang, and Lizhen Cui. Gcn-based user representation learning for unifying robust recommendation and fraudster detection. In *Proceedings of the 43rd international ACM SIGIR conference on research and development in information retrieval*, pages 689–698, 2020.

[Zhang *et al.*, 2021] Ge Zhang, Jia Wu, Jian Yang, Amin Beheshti, Shan Xue, Chuan Zhou, and Quan Z Sheng. Fraudre: Fraud detection dual-resistant to graph inconsistency and imbalance. In *2021 IEEE International Conference on Data Mining (ICDM)*, pages 867–876. IEEE, 2021.

[Zhang *et al.*, 2022] Mengmei Zhang, Xiao Wang, Meiqi Zhu, Chuan Shi, Zhiqiang Zhang, and Jun Zhou. Robust heterogeneous graph neural networks against adversarial attacks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 4363–4370, 2022.

[Zhang *et al.*, 2023] Ya-Lin Zhang, Yi-Xuan Sun, Fangfang Fan, Meng Li, Yeyu Zhao, Wei Wang, Longfei Li, Jun Zhou, and Jinghua Feng. A framework for detecting frauds from extremely few labels. In *Proceedings of the Sixteenth ACM International Conference on Web Search and Data Mining*, pages 1124–1127, 2023.

[Zhang *et al.*, 2024] Rui Zhang, Dawei Cheng, Jie Yang, Yi Ouyang, Xian Wu, Yefeng Zheng, and Changjun Jiang. Pre-trained online contrastive learning for insurance fraud detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 22511–22519, 2024.

[Zhao *et al.*, 2021] Tong Zhao, Yozen Liu, Leonardo Neves, Oliver Woodford, Meng Jiang, and Neil Shah. Data augmentation for graph neural networks. In *Proceedings of the aaai conference on artificial intelligence*, volume 35, pages 11015–11023, 2021.

[Zhao *et al.*, 2023] Jianan Zhao, Qianlong Wen, Mingxuan Ju, Chuxu Zhang, and Yanfang Ye. Self-supervised graph structure refinement for graph neural networks. In *Proceedings of the Sixteenth ACM International Conference on Web Search and Data Mining*, pages 159–167, 2023.

[Zhou *et al.*, 2013] Ke Zhou, Hongyuan Zha, and Le Song. Learning social infectivity in sparse low-rank networks using multi-dimensional hawkes processes. In *Artificial Intelligence and Statistics*, pages 641–649. PMLR, 2013.

[Zhu *et al.*, 2019] Dingyuan Zhu, Ziwei Zhang, Peng Cui, and Wenwu Zhu. Robust graph convolutional networks against adversarial attacks. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 1399–1407, 2019.