Relation-Aware Heterogeneous Graph Neural Network for Fraud Detection

Enxia Li[1], Jin Ouyang[2], Sheng Xiang[1(✉)], Lu Qin[1], and Ling Chen[1]

[1] University of Technology Sydney, Sydney, Australia
enxia.li@student.uts.edu.au, {sheng.xiang,lu.qin,ling.chen}@uts.edu.au
[2] Zhejiang Gong Shang University, Hangzhou, China

Abstract. Fraud detection is a crucial data-mining task in the fields of finance and social media. Traditional machine-learning approaches predict risk based solely on the features of individual nodes. Recent advancements in graph-based methods allow for the consideration of features across related nodes, enhancing predictive accuracy. Especially, Graph Neural Networks (GNNs) have shown high performance on graph-based fraud detection tasks. However, it presents significant challenges to performance and efficiency due to the complex and heterogeneous nature of social networks. This paper introduces a novel approach for fraud detection using a Relation-Aware Heterogeneous Graph Neural Network (RHGNN) model, which efficiently handles the intricacies of input data represented as heterogeneous graphs. Our model leverages a computation graph pre-process and hybrid propagation scheme that integrates both features and topology information for GNNs, allowing for precise and scalable fraud detection. Specifically, we first use a relation-aware node map-reduce to preprocess the computational graph. Then we use the hybrid propagation scheme, which optimizes the collection of neighborhood nodes with reduced complexity and remains the fraud pattern on graph data. This is achieved by alternating the focus between the 1-hop neighbor and 2-hop neighbor in the input graph, thereby enhancing the model without the typical computational overhead. We employ Stochastic Projection Reduction to manage feature dimensionality effectively, ensuring that the model remains efficient even with large-scale graph data. Experimental results on various datasets, including Amazon, Yelpchi, and T-Finance, demonstrate that our model outperforms existing methods in terms of fraud detection accuracy.

Keywords: Fraud Detection · Heterogeneous Graph · Graph Neural Networks

1 Introduction

Detecting fraudulent behavior is used in finance and social media to reduce economic loss. For example, credit card fraud is a major issue in the financial sector,

W. Zhang et al. (Eds.): APWeb-WAIM 2024, LNCS 14963, pp. 240–255, 2024.
https://doi.org/10.1007/978-981-97-7238-4_16

causing significant global economic losses each year [4,7]. As digital transactions grow, so does the complexity and volume of fraud, making effective detection systems necessary to protect consumer trust and financial integrity [32]. The need for robust fraud detection mechanisms is driven by the evolving nature of fraud techniques, which exploit the growth of e-commerce and digital banking [24]. Detecting fraud has thus become crucial in data mining.

Fraud detection systems are categorized based on their methodology. Early systems used rule-based methods, where typical fraudulent behaviors were hard-coded [33]. As fraud tactics evolved, machine learning techniques were incorporated, allowing automatic detection based on historical data [1]. Recently, deep learning methods have shown significant improvements in detecting complex fraud patterns [6]. These systems analyze large amounts of data to identify hidden patterns and anomalies indicating fraud. The use of Graph Neural Networks (GNNs) has introduced a new paradigm in fraud detection, leveraging relationships and connectivity patterns among nodes to enhance detection accuracy [34]. This shift towards relational data analysis emphasizes understanding not just individual nodes but also their connections. Traditional rule-based fraud detection systems define explicit conditions that trigger alerts when met. These systems, derived from known fraud patterns and expert knowledge, are straightforward but rigid, lacking the flexibility to adapt to new fraud tactics without manual updates [20]. Machine learning-based methods offer a dynamic approach by learning from historical data, automatically adapting and improving detection algorithms over time [2]. Techniques such as decision trees, neural networks, and support vector machines have been widely applied, enabling the detection of complex fraudulent behaviors not explicitly defined in rule sets [6,7]. This adaptability makes machine learning methods effective against evolving fraud strategies, continuously refining their models with new data.

Recent advancements in fraud detection have integrated graph-based methods, utilizing relationships between nodes to detect fraud more effectively. These methods consider not only individual node features but also their relationships within a network, capturing complex patterns typically missed by traditional methods. GNNs, in particular, have shown a remarkable ability to detect fraudulent activities by leveraging connected data in real-world networks [5,15]. GNNs apply deep learning principles to graph data, allowing information propagation across nodes and enabling the system to identify suspicious link patterns indicative of fraud [19,35]. This relational approach is advantageous for detecting organized fraud rings, where fraudulent transactions are interconnected. Innovations such as attention mechanisms and node embeddings have further enhanced the efficacy of these methods, providing nuanced understandings of node relationships within graphs [26].

Despite their success, existing graph-based fraud detection methods face several limitations. Many suffer from low efficiency, particularly when scaling to large datasets common in financial systems [35]. The computational complexity of processing large graphs, especially with multiple iterations, can lead to prohibitive processing times. Real-world graphs are usually heterogeneous, with various types of nodes and edges representing different entities [19]. For example,

according to Fig. 1, there are three types of nodes in financial fraud detection tasks. Most GNNs assume a homogeneous graph structure, oversimplifying real-life networks. This can result in a loss of critical information and less effective fraud detection.

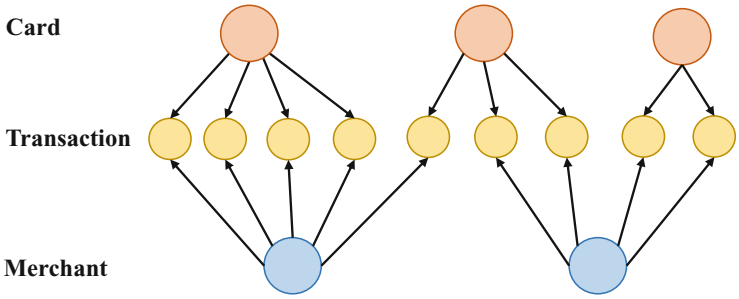

Fig. 1. An example of a heterogeneous graph in the financial field, showing multiple types of nodes/edges in real-world graphs.

In response to these limitations, this paper introduces a novel Relation-Aware Heterogeneous Graph Neural Network (RHGNN) model to address the shortcomings of existing graph-based fraud detection methods. Our model handles the heterogeneity of real-world graphs by recognizing and processing different types of nodes and edges as distinct but interconnected components. By leveraging computation graph preprocessing and a hybrid propagation scheme, our approach enhances the efficiency of information processing across the graph, reducing computational overhead and facilitating scalability. The integration of stochastic projection reduction manages feature dimensionality, maintaining efficiency even when scaling to large datasets. This dual focus on addressing heterogeneity and improving computational efficiency sets our method apart, offering a more robust and scalable solution for detecting fraudulent activities in expansive real-world networks. Finally, extensive experiments on benchmark datasets, including Amazon, Yelpchi, and T-Finance, demonstrate our method's superior performance over existing graph-based techniques in accuracy and efficiency in detecting fraudulent transactions.

The contributions of this paper are manifold and are itemized as follows:

- I) We delineated the complexities of fraud patterns in real-world networks, highlighting the need for models that can handle the heterogeneity of graph data effectively.
- II) We introduced a Relation-Aware Heterogeneous Graph Neural Network model. This model incorporates the computation graph pre-process and hybrid propagation scheme to enhance feature propagation efficiency and includes stochastic projection reduction to manage feature dimensionality without performance loss.

– III) Our method is rigorously tested on several benchmark datasets, including Amazon, Yelpchi, and T-Finance, where it demonstrates superior performance in fraud detection accuracy compared to existing state-of-the-art fraud detection techniques.

Table 1. Notation Summary.

Notation	Description
$\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{A}, \mathcal{R})$	A heterogeneous graph
$\mathcal{V}$	Set of nodes
$\mathcal{E}$	Set of edges
$\mathcal{A}$	Set of node types
$\mathcal{R}$	Set of edge types
$v \in \mathcal{V}$	A node in the graph
$e \in \mathcal{E}$	An edge in the graph
$A \in \mathcal{A}$	A node type
$R \in \mathcal{R}$	An edge type
$\mathbb{A}_{R_i}$	Adjacency matrix for edge type R_i
$P = A_1 \xrightarrow{R_1} A_2 \xrightarrow{R_2} \cdots \xrightarrow{R_{l-1}} A_l$	A meta-path
y_t	Label of transaction node t

2 Preliminaries

In this section, we define the notations and terminologies used throughout this paper. We focus on the problem of fraud detection within the context of the heterogeneous graphs. We also discuss the related works in this section.

2.1 Problem Definition

Unlike traditional graph-based methods that treat input data as homogeneous graphs to predict potential fraudulent activities, our model represents the real-world graph as a heterogeneous graph, automatically generated from real-world transactions or databases. In these heterogeneous graphs, nodes denote individual entities (e.g., users/transactions/merchants) and edges represent their relationships, encoded in the form of adjacency matrices $\mathbb{A}_{R_i}$ for each edge type R_i. Therefore, in this part, we formally defined the fraud detection problem in the context of heterogeneous graphs and the meta-path. All notations are summarized in the Table 1.

Heterogeneous Graph: A heterogeneous graph is defined as $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{A}, \mathcal{R})$, where $\mathcal{V}$ is the set of nodes, $\mathcal{E}$ is the set of edges, $\mathcal{A}$ is the set of node types, and $\mathcal{R}$ is the set of edge types. Each node $v \in \mathcal{V}$ belongs to one particular node type $A \in \mathcal{A}$, and each edge $e \in \mathcal{E}$ belongs to one particular edge type $R \in \mathcal{R}$. Each edge type R_i is associated with an adjacency matrix $\mathbb{A}_{R_i}$, which defines the connectivity between node types. For example, the financial heterogeneous graph has three types of nodes, i.e., card, merchant, and transaction.

Meta-path: A meta-path P is a sequence of node types and edge types in the form $A_1 \xrightarrow{R_1} A_2 \xrightarrow{R_2} \cdots \xrightarrow{R_{l-1}} A_l$, where $A_i \in \mathcal{A}$ and $R_i \in \mathcal{R}$. A meta-path captures a specific type of relationship between nodes in a heterogeneous graph. For example, in the financial heterogeneous graph, a meta-path can be composed by $card \xrightarrow{R_1} transaction \xrightarrow{R_2} merchant$.

Graph-based Fraud Detection: Given a heterogeneous graph $\mathcal{G}$ representing real-world relationships, where nodes represent entities such as users, merchants, and transactions, and edges represent relationships such as transaction occurrences and user-transaction associations, the goal is to predict whether a transaction node $t \in \mathcal{V}$ is fraudulent. This can be formulated as a binary classification problem where the model assigns a label $y_t \in \{0, 1\}$ to each transaction node t, with $y_t = 1$ indicating a fraudulent transaction and $y_t = 0$ indicating a legitimate one. The goal is to develop an end-to-end model that can accurately and efficiently identify fraudulent nodes by leveraging the heterogeneous nature of the graph and the complex relationships captured by meta-paths.

2.2 Related Works

Non-graph Based Fraud Detection. Non-graph based fraud detection has been extensively studied using various machine learning and statistical techniques. Traditional methods include supervised learning approaches such as logistic regression [10], decision trees [8], and support vector machines [22], which rely on hand-crafted features extracted from input data [11]. Recently, deep learning techniques have been employed to automatically learn complex features from raw fraud detection data, improving the accuracy of fraud detection systems [6]. However, these deep-learning approaches typically focus on individual entities and do not fully exploit the relationships between entities (such as users and transactions). Our work addresses this limitation by utilizing advanced heterogeneous graph neural networks (HGNNs) to capture "cross-user" information, providing a more comprehensive view of fraudulent activities.

Graph-Based Fraud Detection. Graph-based methods have gained popularity in fraud detection due to their ability to capture relationships between entities. These methods represent input data as graphs, where nodes represent entities (e.g., users, transactions) and edges represent relationships (e.g., transaction links). Techniques such as graph convolutional networks (GCNs) and

graph attention networks (GATs) have been applied to fraud detection, showing improved performance over traditional methods [29]. CARE-GNN [5] was proposed to select better neighbors for graph-based fraud detection tasks. PC-GNN [15] was designed to solve the node label imbalance problem in fraud detection. BW-GNN [25] was proposed to address the "right-shift" issue in the graph anomaly detection. GTAN [34] was designed for semi-supervised fraud detection in graphs. However, these approaches often assume a homogeneous graph structure and may not fully exploit the heterogeneity inherent in real-world financial transaction networks. Our work addresses this limitation by employing scalable heterogeneous graph neural networks (HGNNs) that can effectively handle the diverse types of nodes and edges present in financial graphs. By leveraging the heterogeneity of these graphs, our model captures more complex and realistic interactions between different types of entities. Importantly, this enhanced modeling capability is achieved without significantly increasing computational costs, making our approach both practical and efficient for large-scale credit card fraud detection.

Heterogeneous Graph Neural Networks. Heterogeneous graph neural networks (HGNNs) extend the concept of GNNs to heterogeneous graphs, which contain multiple types of nodes and edges. HGNNs have been used to model complex interactions in various domains, including recommendation systems, social networks, and fraud detection [14,23]. By incorporating meta-paths and relation-specific transformations, HGNNs can capture the rich semantics of heterogeneous graphs, leading to improved performance in fraud detection tasks [3]. Our work advances the fraud detection task by employing scalable HGNNs specifically designed for modeling real-world fraudulent activities. Unlike previous GNN-based approaches, we propose an effective solution tailored to the heterogeneity of real-life graphs. Our model leverages the diverse types of nodes and edges to capture more realistic and intricate interactions between different entities. This results in a robust and efficient fraud detection system that outperforms traditional methods and other GNN-based solutions, providing significant improvements in both detection accuracy and computational efficiency.

3 Proposed Method

In this section, we introduce our proposed method for fraud detection using a Relation-Aware Heterogeneous Graph Neural Network (RHGNN) model. This section includes the model architecture, computation graph pre-process, heterogeneous propagation, and fraud detection and optimization.

3.1 Model Architecture

Different from other graph-based fraud detection techniques, we consider real-world data as heterogeneous graphs in the fraud detection task. For example, in

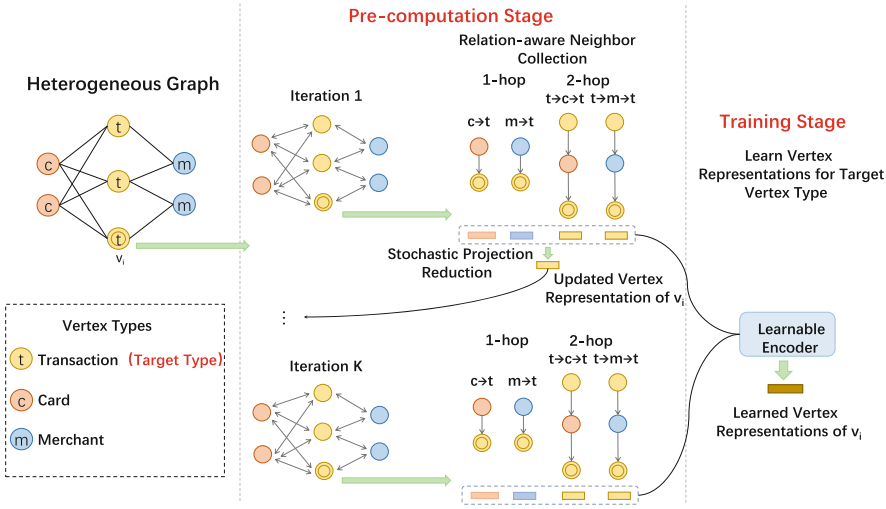

Fig. 2. Overall Framework of the proposed Relation-Aware Heterogeneous Graph Neural Network model (RHGNN). (1) Pre-computation Stage: To leverage the efficiency of hybrid propagation, the main framework of RHGNN consists of alternating propagation iterations, where we introduce a Stochastic Projection Reduction component to manage dimensionality reduction for updated vertex representations. To exploit the granularity of relation-specific information for low information loss, we introduce a hybrid propagation scheme, which alternates focus between 1-hop and 2-hop neighbors, collecting neighbor information based on different relational contexts separately. (2) Training Stage: After K iterations, the collected neighbor information (of target vertices) is used as the input for a heterogeneous graph encoder to generate the vertex representations of target vertices, enhancing the model's ability to detect fraudulent activities accurately and efficiently.

the application of credit card fraud detection, the transaction node is the *"target"* node, and card and merchant nodes are the other types of nodes. Thanks to the success of Random Projection Heterogeneous Graph Neural Network (RpHGNN) [9], our model architecture leverages the strengths of heterogeneous graph neural networks (HGNNs) to capture the complex and varied interactions present in fraud detection input data. The architecture integrates both efficient relation-aware pre-computation and heterogeneous graph neural networks, enabling precise and scalable fraud detection. The core components of our model include:

- **Relation-Aware Node Mapping:** This component involves a propagation scheme that utilizes different local relations to gather potential fraudulent information. For each node type, we identify relevant edge types and apply message passing to collect and aggregate neighbor information, which aids in constructing an enriched feature representation for each node.
- **Stochastic Node Reducing:** To manage the high dimensionality of aggregated neighbor information, this method employs a stochastic dimensional-

ity reduction technique. By applying stochastic node reducing, it effectively reduces the feature space dimensionality while preserving essential information, which is crucial for maintaining computational efficiency and model performance.

- **Intra-Relation Node Convolution:** Within the same type of relation, this convolution operation processes features through specified filters. It aggregates localized feature information from direct neighbors under the same relation type, enhancing the node's feature representation with context-specific information that is critical for identifying patterns indicative of fraud.
- **Heterogeneous Node Information Propagation:** This crucial step propagates information across various node types and relations within the heterogeneous graph. It employs meta-paths to guide the aggregation process, integrating diverse and complex interactions across the graph. This comprehensive aggregation allows the model to capture and leverage the nuanced dynamics characteristic of real-world data structures.

3.2 Computation Graph Pre-process

Before training the model, we preprocess the computation graph to optimize the efficiency and effectiveness of the heterogeneous graph neural network operations for fraud detection.

Relation-Aware Node Mapping. In the relation-aware node mapping phase, we collect neighbor information using a propagation scheme that incorporates various local relations to gather potential fraudulent information. For a node type $A_j \in \mathcal{A}$, we identify a set of edge types ending with it. Each edge type $R_m = (A_i, *, A_j)$ is treated as a local relation $A_i \xrightarrow{R_m} A_j$, and our model performs message passing along these edges to collect neighbor information:

$$(H_{R_m})^{(k)} = (D_{R_m})^{-1}\mathbb{A}_{R_m}(H_{A_i})^{(k-1)} \tag{1}$$

Here, $(H_{A_i})^{(k-1)}$ denotes the vertex representation matrix from the $(k-1)$-th iteration, or the raw feature matrix if $k = 1$. $\mathbb{A}_{R_m}$ is the adjacency matrix for edge type R_m, and D_{R_m} is the degree matrix for vertices of type A_j.

Stochastic Node Reduction. To manage high-dimensional neighbor information, we employ a stochastic node reduction technique, maintaining efficiency while reducing dimensionality. For each node type A_j and its collected neighbor information $\{(H_{R_m})^{(k)} | R_m = (*, *, A_j)\}$, we update the vertex representation of A_j as follows:

$$(H_{A_j})^{(k)} = \sum_{R_m=(*,*,A_j)} \text{Norm}((H_{R_m})^{(k)}(W_{R_m})^{(k)}) \tag{2}$$

where $W_{R_m}^{(k)}$ is the stochastic projection weight matrix, ensuring that the dimensionality of the updated vertex representations remains constant. This

approach avoids the efficiency issues of exponentially growing dimensionality seen in naive concatenation methods.

By leveraging the relation-aware node mapping and stochastic node reducing methods, we ensure that the collected neighbor information reflects the fine granularity of different relations, optimizing both the efficiency and the effectiveness of the graph neural network operations.

3.3 Heterogeneous Propagation

The heterogeneous propagation stage involves two key processes: intra-relation node convolution and heterogeneous node information propagation. Especially, we use 1-hop and 2-hop hybrid propagation scheme to ensure scalabiilty without performance loss.

Hybrid Propagation Scheme. In the traditional propagation scheme, each iteration collects neighbor information via 1-hop relations, requiring K iterations to capture relations within K hops. A simple way to reduce iterations is to use a 2-hop propagation scheme, extending local relations to 2 hops. This reduces the required iterations to $\frac{K}{2}$, lowering the frequency of untrainable vertex representation updates. However, this approach can be inefficient due to the potentially overwhelming number of 2-hop relations in certain graphs.

To address this efficiency issue, we design a hybrid 2-hop and 1-hop propagation scheme. This method selects twice as many 2-hop relations as local ones, including both 1-hop and 2-hop relations.

Intra-Relation Node Convolution. After confirming propagation scheme, in this step, we perform convolution operations within the same type of relation. For a given relation type $R \in \mathcal{R}$, we define the intra-relation convolution as:

$$\mathbf{H}_R^{(l+1)} = \sigma\left(\sum_{u \in \mathcal{N}_R(v)} \frac{1}{c_{vu}} \mathbf{H}_u^{(l)} \mathbf{W}_R\right), \tag{3}$$

where $\mathbf{H}_u^{(l)}$ is the feature vector of node u at layer l, $\mathbf{W}_R \in \mathbb{R}^{d \times d}$ is the weight matrix for relation R, c_{vu} is a normalization constant, and σ is an activation function. This intra-relation convolution extracts localized features specific to each type of relation.

Heterogeneous Node Information Propagation. This process propagates information across different types of nodes and edges, ensuring that the model captures the complex interactions that span various entities in the financial graph. Let $\mathbf{H}^{(l+1)}$ be the aggregated feature matrix at layer $l+1$:

$$\mathbf{H}^{(l+1)} = \sigma\left(\sum_{R \in \mathcal{R}} \sum_{u \in \mathcal{N}_R(v)} \frac{1}{c_{vu}} \mathbf{H}_u^{(l)} \mathbf{W}_R\right), \tag{4}$$

where the information is aggregated from multiple relations and nodes, guided by meta-paths that define the sequence of relations to be followed.

3.4 Fraud Detection and Optimization

The final stage of our method focuses on detecting fraudulent transactions and optimizing the model's performance.

After propagating information through the heterogeneous graph, we apply a classification layer to identify potentially fraudulent transactions. The classification layer uses the enriched node representations generated by the previous steps to make accurate predictions. The final node representation $\mathbf{H}^*$ is passed through a softmax function to obtain the probability of each node being fraudulent:

$$\hat{y}_v = \text{softmax}(\mathbf{H}_v^* \mathbf{W}_c), \tag{5}$$

where $\mathbf{W}_c$ is the weight matrix for the classification layer. To optimize the model, we employ techniques such as stochastic projection reduction to manage feature dimensionality, ensuring that the model remains efficient even with large-scale data. Additionally, we utilize loss functions tailored for imbalanced datasets, such as weighted cross-entropy, to improve the detection of rare fraudulent events:

$$\mathcal{L} = -\sum_{v \in \mathcal{V}} w_v y_v \log(\hat{y}_v), \tag{6}$$

where w_v is the weight assigned to node v based on class imbalance.
Our experimental results demonstrate that the proposed method not only improves fraud detection accuracy but also significantly reduces training time, making it suitable for real-time fraud detection systems.

4 Experiments

4.1 Experiments Settings

Datasets. To evaluate the fraud detection performance on financial networks, we conducted experiments on the T-Finance [25] dataset. The T-Finance dataset aims to find the anomaly accounts in transaction networks. The nodes are unique anonymized accounts with 10-dimension features related to registration days, logging activities and interaction frequency. The edges in the graph represent two accounts that have transaction records. Human experts annotate nodes as anomalies if they fall into categories like fraud, money laundering and online gambling. In the T-finance dataset, there are 39357 nodes, 21222543 edges and 10-dimension features.

Besides, we also conducted experiments on two public fraud detection datasets, the Yelp and Amazon dataset. The Yelp dataset consists of filtered spam and reviews recommending hotels and restaurants, while the Amazon dataset comprises reviews of musical instrument products. We extracted 45954

nodes, 3846979 edges, and 32 manual features from [21] as the original graph and node features of the Yelp dataset. We also extracted 11944 nodes, 4398392 edges, and 25 manual features from [18] as Amazon's original graph and node features.

Compared Methods. We compare our proposed techniques with the baselines: GCN [12], GEM [13], GAT [27], FdGars [30], Play2Vec [31], GeniePath [17], Graphconsis [16], SEMIGNN [28]. These methods are popular for graph-based node classification tasks or fraud detection.

Parameter Setting and Evaluation Metrics. In the experiment, each baseline is conducted separately and adopts the optimal parameters as they were originally proposed. For our model, we set the model embedding size to 256, the hidden size to 256, the learning rate to 0.003, and using Adam as the optimizer. Our model's hyperparameter setting is a combination of several hyperparameters, which include the input dropout rate, the hidden dropout rate, and squash k which is a hyperparameter that regulates the degree of feature dimensionality reduction achieved through stochastic node reducing techniques. We set the model input drop rate to 0.01, the drop rate to 0.01, and squash k to 6. In terms of the running environment, we train the models on a Linux machine with 2 Intel(R) Xeon(R) Silver 4208 CPU @ 2.10GHz, 128GB RAM, and an RTX 6000 with 48GB of GPU memory. We evaluate the experimental results on financial fraud detection and opinion fraud datasets by the area under the ROC curve (AUC), F1 score(F1-macro), and average precision (AP). For all three metrics, the higher score indicates the higher performance of the methods.

Table 2. Fraud Detection Experimental Results.

Model	YelpChi			Amazon			T-Finance		
	AUC	F1	AP	AUC	F1	AP	AUC	F1	AP
GCN	0.8207	0.7847	0.7575	0.8401	0.8057	0.7986	0.8303	0.7944	0.8077
GEM	0.8267	0.7897	0.7635	0.8461	0.8109	0.8046	0.8363	0.8003	0.8142
GAT	0.8372	0.7968	0.7705	0.8576	0.8175	0.8113	0.8474	0.8071	0.8210
FdGars	0.8515	0.8093	0.7832	0.8714	0.8309	0.8241	0.8610	0.8202	0.8348
Play2Vec	0.8603	0.8183	0.7934	0.8807	0.8396	0.8332	0.8703	0.8287	0.8429
GeniePath	0.8671	0.8250	0.8009	0.8872	0.8467	0.8404	0.8773	0.8351	0.8498
Graphconsis	0.8732	0.8301	0.8055	0.8938	0.8529	0.8473	0.8834	0.8440	0.8551
SEMIGNN	0.8806	0.8364	0.8118	0.9207	0.8591	0.8543	0.9109	0.8463	0.8612
RHGNN	**0.9431**	**0.8458**	**0.8156**	**0.9734**	**0.8788**	**0.8561**	**0.9573**	**0.8935**	**0.8641**

4.2 Fraud Detection Experiment

In the fraud detection experiment, we evaluate the performance of our proposed model against the baselines on the T-Finance, Amazon, and YelpChi datasets. The results are summarized in Table 2. Our model outperforms all baselines in terms of AUC, F1-macro, and AP, demonstrating its effectiveness in detecting fraudulent behaviors.

In the fraud detection experiment, we observed that our proposed model, RHGNN, consistently outperformed the baseline models across all three datasets: YelpChi, Amazon, and T-Finance. These results indicate the robustness and superior performance of RHGNN in detecting fraudulent activities across different types of networks. The significant improvement in metrics across all datasets demonstrates the effectiveness of our model's design and its applicability to various fraud detection scenarios.

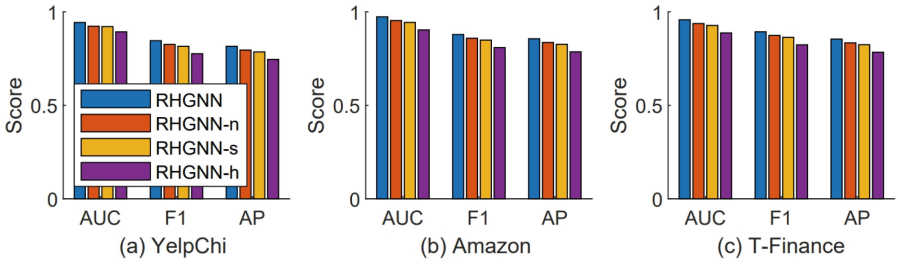

Fig. 3. Ablation study results on three datasets.

4.3 Ablation Study

We conduct an ablation study to understand the contributions of different components of our model. The results are shown in Fig. 3. Removing any of the components (hybrid neighbor scheme, stochastic node reduction, or heterogeneous propagation) results in a noticeable drop in performance, indicating the importance of each component. In this figure, RHGNN is our proposed method, RHGNN-n is the model with non-hybrid neighbor scheme, RHGNN-s denotes the model without stochastic node reduction, and RHGNN-h denotes the model without heterogeneous propagation (treat the edges as the same type). Performance metrics analyzed are the AUC, F1 Score, and Average Precision (AP).

As shown in Fig. 3(a), the base RHGNN model consistently outperforms other variants across all metrics on the YelpChi dataset. The RHGNN model achieves notably higher AUC, F1, and AP scores, indicating its comprehensive capability in handling the complexity and heterogeinity specific to YelpChi data. Both RHGNN-n and RHGNN-s variants show a slight reduction in performance, suggesting that the non-hybrid scheme and no stochastic node reduction slightly

hampers effectiveness. The RHGNN-h shows diminished effectiveness, particularly in F1 and AP scores, underscoring the importance of the utilization of heterogeneous information. We can obtain similar results in the experimental results of the other two datasets.

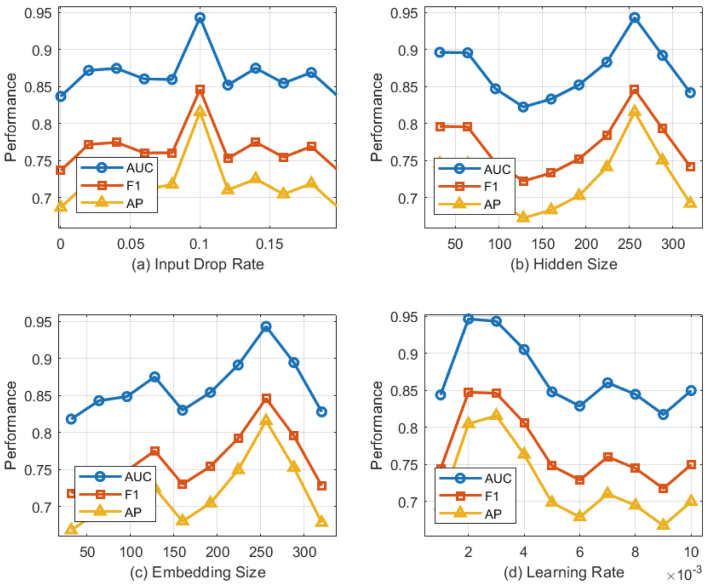

Fig. 4. Parameter sensitivity analysis for the RHGNN model.

4.4 Parameter Sensitivity

To examine the sensitivity of our model to its hyperparameters, we vary key parameters such as embedding size, learning rate, and dropout rates. Figure 4 illustrates the impact of these parameters on the model's performance.

Specifically, Fig. 4(a) shows the impact of varying the input drop rate on model performance. As the drop rate increases from 0 to 0.15, significant fluctuations in performance metrics are observed. The AUC metric remains relatively stable, maintaining high performance across different drop rates. However, both the F1 score and AP decline sharply as the drop rate exceeds 0.1, indicating a loss of critical information that adversely affects precision and recall balance. Figure 4(b) analyzes sensitivity to hidden size. The model's performance increases as the hidden size grows from 50 to around 150. Beyond this point, metrics peak and then slightly decline, particularly the F1 score and AP, suggesting an optimal range around 150–200. This is likely due to the model's ability

to capture more complex patterns with more hidden units without yet overfitting. The trends in embedding size, shown in Fig. 4(c), reveal a clear peak in performance at an embedding size of 150. Both the AUC and F1 scores improve significantly as the embedding size increases from 50 to 150, after which performance plateaus or slightly declines. This behavior suggests that up to a certain point, increased dimensionality provides sufficient representational space, beyond which additional dimensions do not contribute to better generalization. Lastly, Fig. 4(d) evaluates the effect of learning rate adjustments. A lower learning rate of around 0.002 to 0.004 appears optimal, as indicated by higher values in all performance metrics. The AP and F1 scores decrease substantially with higher learning rates, suggesting that too large a step may lead to instability in training or skipping over optimal solutions. We choose our best parameter settings according to these results.

5 Conclusion

In this study, we proposed a novel Relation-Aware Heterogeneous Graph Neural Network (RHGNN) for fraud detection in financial and social media data. Our approach leverages the intricate relationships between different types of entities to enhance the detection capabilities beyond traditional machine learning methods, which typically rely on the features of individual nodes alone. Through extensive experiments, we demonstrated that RHGNN significantly outperforms existing state-of-the-art methods across various metrics, including AUC, F1 score, and average precision. The results confirm the efficacy of incorporating relation-awareness in graph neural networks, providing a more nuanced understanding of fraudulent patterns. This advancement opens new avenues for developing even more sophisticated models that can handle the complexity and diversity of real-world data. Future work will focus on further enhancing the scalability of RHGNN and exploring its applicability to other domains beyond fraud detection. We also aim to investigate the integration of additional data sources and the impact of temporal dynamics on the model's performance.

References

1. Awoyemi, J.O., Adetunmbi, A.O., Oluwadare, S.A.: Credit card fraud detection using machine learning techniques: a comparative analysis. In: 2017 international conference on computing networking and informatics (ICCNI), pp. 1–9. IEEE (2017)
2. Bhattacharyya, S., Jha, S., Tharakunnel, K., Westland, J.C.: Data mining for credit card fraud: a comparative study. Decis. Support Syst. **50**(3), 602–613 (2011)
3. Bing, R., Yuan, G., Zhu, M., Meng, F., Ma, H., Qiao, S.: Heterogeneous graph neural networks analysis: a survey of techniques, evaluations and applications. Artif. Intell. Rev. **56**(8), 8003–8042 (2023)
4. Bolton, R.J., Hand, D.J.: Statistical fraud detection: a review. Stat. Sci. **17**(3), 235–249 (2002)

5. Dou, Y., Liu, Z., Sun, L., Deng, Y., Peng, H., Yu, P.S.: Enhancing graph neural network-based fraud detectors against camouflaged fraudsters. In: Proceedings of the 29th ACM International Conference on Information & Knowledge Management (2020)
6. Fu, K., Cheng, D., Tu, Y., Zhang, L.: Credit card fraud detection using convolutional neural networks. In: International Conference on Neural Information Processing (2016)
7. Goyal, R., Manjhvar, A.K.: Review on credit card fraud detection using data mining classification techniques & machine learning algorithms. Data Sci. Anal. eJournal **7**(1), 972–975 (2020)
8. Hájek, P., Abedin, M.Z., Sivarajah, U.: Fraud detection in mobile payment systems using an XGboost-based framework. Inf. Syst. Front. **25**(4), 1–19 (2022)
9. Hu, J., Hooi, B., He, B.: Efficient heterogeneous graph learning via random projection (2023). ArXiv abs/2310.14481
10. Itoo, F., Meenakshi, Singh, S.: Comparison and analysis of logistic regression, naïve bayes and KNN machine learning algorithms for credit card fraud detection. Int. J. Inf. Technol. **13**, 1503 – 1511 (2020)
11. Ke, G., et al.: LightGBM: a highly efficient gradient boosting decision tree. In: Neural Information Processing Systems (2017)
12. Kipf, T.N., Welling, M.: Semi-supervised classification with graph convolutional networks (2016). CoRR abs/1609.02907
13. Lin, W., Lan, H., Li, B.: Generative causal explanations for graph neural networks (2021). CoRR abs/2104.06643
14. Linmei, H., Yang, T., Shi, C., Ji, H., Li, X.: Heterogeneous graph attention networks for semi-supervised short text classification. In: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), pp. 4821–4830 (2019)
15. Liu, Y., Ao, X., Qin, Z., Chi, J., Feng, J., Yang, H., He, Q.: Pick and choose: a GNN-based imbalanced learning approach for fraud detection. In: Proceedings of the Web Conference 2021 (2021)
16. Liu, Z., Dou, Y., Yu, P.S., Deng, Y., Peng, H.: Alleviating the inconsistency problem of applying graph neural network to fraud detection (2020). CoRR abs/2005.00625
17. Liu, Z., Chen, C., Li, L., Zhou, J., Li, X., Song, L.: GeniePath: Graph neural networks with adaptive receptive paths (2018). CoRR abs/1802.00910
18. McAuley, J.J., Leskovec, J.: From amateurs to connoisseurs: modeling the evolution of user expertise through online reviews (2013). CoRR abs/1303.4402
19. Monti, F., Boscaini, D., Masci, J., Rodola, E., Svoboda, J., Bronstein, M.M.: Geometric deep learning on graphs and manifolds using mixture model CNNs. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 5115–5124 (2017)
20. Phua, C., Lee, V.C.S., Smith-Miles, K., Gayler, R.W.: A comprehensive survey of data mining-based fraud detection research (2010). ArXiv abs/1009.6119
21. Rayana, S., Akoglu, L.: Collective opinion spam detection: bridging review networks and metadata. In: Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 985–994 (2015)
22. Rtayli, N., Enneya, N.: Enhanced credit card fraud detection based on SVM-recursive feature elimination and hyper-parameters optimization. J. Inf. Secur. Appl. **55**, 102596 (2020)

23. Shi, J., Ji, H., Shi, C., Wang, X., Zhang, Z., Zhou, J.: Heterogeneous graph neural network for recommendation (2020). ArXiv abs/2009.00799
24. Sorournejad, S., Zojaji, Z., Atani, R.E., Monadjemi, A.H.: A survey of credit card fraud detection techniques: Data and technique oriented perspective (2016). ArXiv abs/1611.06439
25. Tang, J., Li, J., Gao, Z., Li, J.: Rethinking graph neural networks for anomaly detection. In: International Conference on Machine Learning, pp. 21076–21089. PMLR (2022)
26. Velivckovic, P., Cucurull, G., Casanova, A., Romero, A., Liò, P., Bengio, Y.: Graph attention networks (2017). arXiv preprint arXiv:1710.10903
27. Veličković, P., Cucurull, G., Casanova, A., Romero, A., Liò, P., Bengio, Y.: Graph attention networks (2018)
28. Wang, D., et al.: A semi-supervised graph attentive network for financial fraud detection (2020). CoRR abs/2003.01171
29. Wang, D., et al.: A semi-supervised graph attentive network for financial fraud detection. In: 2019 IEEE International Conference on Data Mining (ICDM), pp. 598–607 (2019)
30. Wang, J., Wen, R., Wu, C., Huang, Y., Xiong, J.: FdGars: fraudster detection via graph convolutional networks in online app review system. In: Companion Proceedings of the 2019 World Wide Web Conference, pp. 310–316 (2019)
31. Wang, Z., Long, C., Cong, G., Ju, C.: Effective and efficient sports play retrieval with deep representation learning. In: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pp. 499–509 (2019)
32. West, J., Bhattacharya, M.: Intelligent financial fraud detection: a comprehensive review. Comput. Secur. **57**, 47–66 (2016)
33. Whitrow, C., Hand, D.J., Juszczak, P., Weston, D.J., Adams, N.M.: Transaction aggregation as a strategy for credit card fraud detection. Data Min. Knowl. Disc. **18**, 30–55 (2009)
34. Xiang, S., et al.: Semi-supervised credit card fraud detection via attribute-driven graph representation. In: AAAI (2023)
35. Zhou, J., et al.: Graph neural networks: a review of methods and applications. AI Open **1**, 57–81 (2020). https://doi.org/10.1016/j.aiopen.2021.01.001