RWR-RGCN : A Novel Framework for Fraud Detection via Node Context Aggregation

Anonymous authors
Paper under double-blind review

Abstract

The integrity of online reviews is crucial for businesses, yet widespread review fraud poses significant risks. This paper addresses this challenge by leveraging the power of multi-relational graph convolutional networks (RGCNs) for fraud detection. We introduce RWR-RGCN, a novel framework integrating a multi-layer RGCN architecture with Random Walks with Restart (RWR). The essential role of capturing critical connections lies in RWR generating node sequences, which can aggregate node features, enhancing the model's understanding of the local and global context within the review graph. To further refine fraud detection, we incorporate Louvain clustering for community identification, identifying high-modularity clusters indicative of coordinated fraudulent activity. Evaluated on the Yelp dataset, RWR-RGCN achieved an AUC of 82.58% and a recall of 94.56%, surpassing the state-of-the-art and baseline methods in AUC and recall. These results demonstrate the superior effectiveness of the proposed framework in detecting fraudulent activity within complex online review networks.

1 Introduction

The modern information era is seeing a shift in activity toward the World Wide Web. The Internet has become a vital component of contemporary life. Furthermore, since then, Internet services have encouraged a variety of fraudulent behaviors as they grow in popularity. Fraudsters pose as normal users to get over the anti-fraud systems, distribute fake information, or steal the private data of end users.

Additionally, fraud strategies are ever-changing. In order to minimize the likelihood of being uncovered, scammers adjust by developing ever-more-secret strategies. These constantly changing schemes make it harder to detect fraud, neglecting several dishonest practices that result in significant losses (Zhang et al., 2024). This emphasizes the significance of developing a fraud detection system capable of capturing such hidden behaviors and activities.

Manually filtering tens of millions, or even hundreds of millions, of complex web material to determine the fraudulent information is both ineffective and expensive process. There are important scientific ramifications for efficient fraud detection. Graph-based approaches have become an effective approach for detecting fraudulent behaviors in both the academic and industrial areas. Since fraudsters with similar objectives often link with one another, graph-based approaches connect entities through different relation types and identify suspicious patterns at the graph level.

With the advancement of graph studies, Graph Neural Networks (GNNs) have spawned numerous GNN-based fraud detectors that target various areas, including financial crimes (Kurshan et al., 2020), (Kurshan & Shen, 2020), fake news (Li & Li, 2024), (Xu et al., 2022), opinion fraud (Liu et al., 2020), (Li et al., 2019), and medical fraud Zhang et al. (2024) presented recently. Despite conventional graph-based techniques, GNN-based methods use neural networks to learn the representation of a focus node by aggregating neighboring features. They can be trained models in an end-to-end manner, which reduces the expense of feature engineering and data annotation (Hamilton et al., 2017). The ability of GNNs to comprehend intricate relationships and patterns is demonstrated in pioneering research, indicating the growing potential of this method to improve fraud detection (Zhang et al., 2024).

Graph-based fraud detection has become a crucial method for combating fraudulent activities. While initial graph neural network (GNN) Schlichtkrull et al. (2018) methodologies such as GCN, GAT, CARE-GNN (Dou et al., 2020), and GraphSAGE Hamilton et al. (2017) significantly improved detection accuracy, they are generally hindered by two primary challenges: class imbalance, where fraudulent nodes represent a minority, and heterophily, in which fraudulent nodes frequently connect with legitimate ones. Plenty of specialized models have been created over time to address these challenges from diverse perspectives. GraphConsis Liu et al. (2020) and CARE-GNN Dou et al. (2020) aims to mitigate the influence of disruptive or misleading neighbors by the implementation of consistency enforcement and adaptive neighbor selection, respectively. PC-GNN Liu et al. (2021) addresses the imbalance problem by carefully picking minority nodes, while FRAUDRE Zhang et al. (2021) uses relational modeling to detect diverse fraudulent patterns in inconsistent graphs. RioGNN Peng et al. (2021) directly integrates heterophily by disentangling relational signals or employing both homophilic and heterophilic edges. Recently, DOS-GNN Jing et al. (2024) progresses by employing dual-feature aggregation and embedding-space oversampling to simultaneously tackle heterophily and class imbalance.

Nevertheless, GNN-based fraud detection systems only employ GNNs within a specific context, disregarding the deceitful behaviors of fraudsters that have garnered significant interest from research and industry professionals. In the meantime, theoretical research demonstrates GNNs' shortcomings and susceptibilities in the presence of noisy edges and nodes in graphs. Consequently, the effectiveness of GNN-based fraud detectors would be compromised if the disguised criminals were not dealt with. Similar issues have been noted in surveys by (Dou et al., 2020), and their remedies either don't match the fraud detection difficulties or interfere with the GNNs' end-to-end learning approach.

In Dou et al. (2020) demonstrates the two categories of camouflage: as Li et al. (2019) first feature types, cunning con artists can modify their actions, utilize language generation models, or insert special characters into reviews (a tactic known as spamouflage) to mask overtly questionable consequences to facilitate eluding feature-based detectors. The second relation type is when the graph has more benign nodes than fraudsters. Fraudsters can investigate the defenders' graphs and modify their actions to reduce suspicion (Zhuang et al., 2024). More popular camouflage techniques include putting the poster in contact with high-credit users, eliminating critical remarks, and fabricating supportive comments (Yang et al., 2021).

GNNs power the graph information structure by feature aggregation from neighboring nodes. This technique takes advantage of the inherent homophily in data, which means that nodes with comparable characteristics are more likely to be connected (Gong et al., 2023). Although this strategy improves the learning process by allowing more precise predictions, it also brings about a notable drawback: the issue of oversmoothing (Liu et al., 2020). As information from neighboring nodes is aggregated hierarchically, the distinct properties of nodes within the graph may be lost, making them indistinguishable from one another. This issue is particularly problematic in the context of fraud detection, where fraudsters often hide within the network. These so-called "camouflage fraudsters" use the graph's homophily to their advantage, making it challenging for GNNs to discern between authentic and fraudulent nodes (Dou et al., 2020). The oversmoothing effect exacerbates this issue by masking even more subtle variations crucial for identifying fraudulent activity. Therefore, even though neighbor aggregation in GNNs is an effective tool, it must be carefully managed to avoid having the unintentional outcome of helping fraudsters avoid detection.

This paper is motivated by the observation that fraudsters tend to disguise their cunning behaviors by constructing real connections with normal users. Such behavior will add inaccurate information that GNN will learn from since GNN assumes the adjusted nodes share similar characteristics (Dou, 2022).

This study introduces an innovative unsupervised aggregation method that consolidates all behavior-related signals of nodes prior to classification, hence enhancing fraud detection resilience in multi-relational graphs. At its core, the framework introduces three complementary components. A Fraud Neighbors Selector utilizes random walk with restart (RWR) to autonomously identify and prioritize informative pathways, revealing the concealed behavioral patterns employed by fraudsters for concealment. Secondly, a Vertex Community Detection Based Aggregator utilizing Louvain clustering organizes nodes into cohesive structural communities, so maintaining collusive fraud rings as integrated substructures despite individual links appearing normal. Third, the enhanced behavioral embeddings are input into a Relation-aware Graph Convolutional Network

(RGCN), which utilizes relation-specific representation to more accurately differentiate between fraudulent and legitimate nodes. The methodology represents a novel approach that consolidates local walk-based signals, global community patterns, and multi-relational learning into a unified framework, integrating RWR, Louvain clustering, and RGCN for fraud detection. The proposed system provides a scalable, interpretable, and effective solution to combat the evolving evasion strategies employed by fraudsters.

The remainder of this paper is structured as follows: Section 2 provides background and related work. The methods we propose are outlined in Section 3. Section 4 presents the performance evaluation and results. Finally, we provide a discussion in Section 5 and a conclusion in Section 6.

2 Background and Related Work

This section provides a concise summary of the background and pertinent research on graphs, Graph Neural Networks (GNNs), and anomaly detection broadly. It also discusses the application of random walks in graph anomaly detection.

2.1 Graph

An undirected graph $\mathcal{G} = (V, E)$ is defined as a set of nodes or vertices V and a set of links or edges $E \subseteq V \times V$ that link the nodes together. Any two nodes i and j are adjacent if they are connected with a link. Edges are defined as (u, v), and it is supposed that the order of nodes in the pair does not matter. $\boldsymbol{A}$ an adjacency matrix, which is an $n \times n$ with $n = |V|$, is a popular method of representing a graph. If there is an edge between node i and node j, it is indicated by the (i, j) entry of the matrix $\mathbf{A}_{i,j}$. An edge's weight can be represented by the corresponding item in the adjacency matrix and can be either a real integer for a weighted graph or a binary value (0 or 1) for an unweighted graph. Both directed and undirected edges are possible. Directed edges contain information that has a direction; for instance, a road could be a one-way street. The geographical distance between two weather stations is an example of an undirected edge, which has no source.

Formally, an ordered tuple (u, v) represents a directed edge $e \in E$ between nodes $u \in V$ and $v \in V$, while the unordered variant $\{u, v\}$ represents an undirected edge. Nodes and edges can also have a feature vector $\boldsymbol{a} = (a_1, a_2, \ldots, a_n)$. These features could be informative about the characteristics of a node or edge; for instance, node features could contain review token counts, and edge features could indicate the type of edge that exists between two nodes (which could be more than 1). Any graph that has features (i.e., feature vectors) assigned to nodes and/or edges is referred to as an attributed graph. Many graph-based techniques, such as GNNs, rely on these attributes.

2.2 Graph Neural Network

Graph Neural Networks (GNN) is a class of deep learning models that was created especially to interpret graph-structured data and learn both attributed and structural graph information (Kipf & Welling, 2017), (Hamilton et al., 2017). GNN's primary function is to learn node representations by gathering and spreading data from nearby nodes to the center node. GNNs can capture intricate topology dependencies and contextual information among nodes through propagation, or message passing (Wu et al., 2021), (Bei et al., 2025).

There are two categories of GNNs: GNN spatial-based and GNN spectral-based. Spatial-based GNNs use the spatial information of the nodes and message-passing techniques Zhu et al. (2021) and (Hamilton et al., 2017). Spectral-based GNNs, on the other hand, use spectral graph theory and the graph's Laplacian matrix (Kipf & Welling, 2017), (Dong et al., 2024), and (He et al., 2024). Due to their ability to capture node properties and graph structure information, Graph Neural Networks (GNNs) have achieved reasonable success in various tasks (e.g., node classification, sub-graph classification, graph classification, link prediction) (Yu et al., 2022b).

Another direction was mentioned in Yu et al. (2022a): there has already been some work done on modelling the heterogeneous graph representation. For instance, metapath2vec Dong et al. (2017) and HERec Shi et al. (2019) are two studies that use random walks to build meta-paths over the heterogeneous network for node

embeddings. GNNs are becoming more and more popular as a way to encode methods for graph structures. Several heterogeneous GNN models have been developed to improve GNN architecture by enabling them to add the capacity of capturing heterogeneous contextual information at the node and edge levels. Multi-relation graphs represent a specific category of heterogeneous graph structures.

2.3 Anomaly Detection on Graph Data

GNN effectively depicts center nodes in graph-structured data by aggregating information from nearby nodes, which has sparked a lot of academic interest. Early anomaly detection methods used simple graph neural networks. To improve embedding learning in GCN-based Anti-Spam (GAS) (Li et al., 2019), various aggregators were predetermined to model various relationships, and homogenous graphs were erected based on similarity. FdGars Wang et al. (2019) used a preset tagging modality to classify users according to their behavior and material and then used multi-relations GCN to perceive fake users (Yu et al., 2024).

The Rayleigh Quotient Dong et al. (2024) demonstrates how to describe the graph's accumulated energy. It creates a unique spectral GNN and spectral-related pooling function to extract the spectrum energy data and figure out the anomalous graphs' underlying features. The Anomaly-Denoised Autoencoders for Graph Anomaly Detection (ADA-GAD) comprehensive structure He et al. (2024) has two stages of training architectures and anomaly-denoised augmentation to alleviate the damaging impact of the graphs anomalous shapes in addition to regularization that enhances the distribution of anomaly score.

Furthermore, Dou et al. (2020) anomaly detection unravels the anomalies in disguised behavior of fraudsters and divides them into two types from a feature and relation perspective. For features disguise, fraudsters can adapt and modify their behaviors, mask obvious questionable results with special characters in reviews, which is called spamouflage, or use generation models to handcraft the reviews' wordings or add unusual characters to evade feature-based systems that detect the fictitious reviews. Crowd workers enthusiastically carry out relational disguises on online social networks. They can investigate the defenders' graphs and modify their behaviors to allay their suspicious actions. These cunningly blend in by establishing connections with numerous normal users who can be found in the anomaly direct connection and are more benign users than anomalies.

Recently, countless GNN-based models have been created to tackle the distinct issues of fraud detection in graphs, including noise, camouflage, class imbalance, and heterophily. GraphConsis Liu et al. (2020) enhances resilience by ensuring consistency in embeddings among neighbors; hence it eliminates noisy and conflicting connections. Expanding on this concept, CARE-GNN Dou et al. (2020) presents an enhanced neighbor selection module that utilizes multi-relation graphs and uses reinforcement learning to aggregate node features dynamically from identifying informative neighbors, enabling the selection of best neighbors and effectively addressing camouflage wherein fraudulent nodes imitate real ones. Likewise, PC-GNN Liu et al. (2021) addresses the imbalance problem by segmenting the graph into substructures and implementing balanced sampling; hence it ensures sufficient representation of fraudulent minority nodes. FRAUDRE Zhang et al. (2021) collects various relational neighbors of a node to capture intricate interaction patterns and addresses class imbalance through the use of an imbalanced loss function. Recent methodologies, such as RioGNN (Peng et al., 2021), specifically address heterophily by disentangling relation-dependent representations or concurrently utilizing both homophilic and heterophilic edges; hence they augment expressiveness within mixed-label neighborhoods. RioGNN Peng et al. (2021) elucidates relation-dependent representations. Recent research advances DOS-GNN (Jing et al., 2024), which integrates dual-feature aggregation with embedding-space oversampling to concurrently tackle heterophily and class imbalance, therefore circumventing the drawbacks of noisy edge synthesis. Although these models provide substantial contributions, they frequently either entail considerable computing costs or concentrate exclusively on a singular facet of the issue. Despite attaining state-of-the-art results, DOS-GNN Jing et al. (2024) is confined to single-relational contexts and may exhibit poor generalization to more complex relational graphs.

For years, researchers have focused heavily on multi-relational GNNs as a forceful tool for improving knowledge representation (Tian & Meng, 2024). These networks intend to represent the intricate graph architecture that exists between links and entities to better capture the multi-relational information exchanges. Dou (2022) is inspired by the disguising actions of actual anomalies. According to their graph model, a cunning

scammer could establish links with normal users to reduce suspicion. The aforementioned concealing behavior may lead to the GNNs learning non-discriminative embeddings for anomalies who are disguised, as they are predisposed to believe that neighboring nodes share similar attributes.

Despite a considerable amount of research enhancing the understanding of anomaly detection in multi-relational graphs by examining the optimal selection of neighbors and concealing the behaviors of genuine anomalies, these studies frequently neglect the impact of fraudulent actions within the normal network through reinforcement learning or regularization. This error highlights a significant deficiency in the literature: the necessity for a thorough examination of the interaction between optimal neighbors selected for election and classification accuracy. This study introduces an innovative methodology that employs RWR to identify differentiable features by monitoring the adaptive camouflage patterns of fraudsters. Subsequently, it aggregates these features through vertex-community detection to encapsulate node behavior within localized subgraphs and utilizes Relation Graph Convolutional Network (RGCN) for feature learning across various graph relationships to generate informative features that differentiate fraudulent nodes from legitimate ones. This uniquely positions it to exceed existing state-of-the-arts in both imbalanced and heterophilic graph scenarios, while maintaining scalability and interpretability.

3 The RWR-RGCN Framework

Current graph neural network-based models for fraud detection have presented creative strategies to tackle issues such as camouflage, class imbalance, relation inconsistency, and heterophily. Presented solutions included selective aggregation, inductive neighbor sampling, resampling techniques, and the application of embedding alignment and disentanglement. Notwithstanding these advancements, these methodologies predominantly focus on nodes or edges in their aggregation processes, limiting their capacity to identify more extensive group-level structures essential for fraud detection, as fraudulent entities frequently establish collusive networks to avoid detection.

A notable shortcoming of these techniques is their susceptibility to camouflage and noise. Fraudsters frequently conceal their identities by generating links to normal entities or users, an issue that models like GraphSAGE (Hamilton et al., 2017), CARE-GNN (Dou et al., 2020), and RioGNN Peng et al. (2021) strive to address by selective neighbor aggregation. These models primarily function at the node-to-node relations level, potentially leading to the erroneous exclusion of informative edges or the retention of deceptive ones, so diminishing their overall detection efficacy. Louvain clustering, in contrast, focuses on the community level, aggregating nodes into clusters based on global structural coherence. This guarantees that fraud rings remain intact as unified entities, even when specific connections seem normal. Louvain is especially efficient in heterogeneous fraud graphs, where fraudsters may leverage various relationships, including items, users, and transactions, while still creating dense, dubious subcommunities that expose their collusion.

This section details the proposed framework RWR-RGNN to address these limitations, a novel system that integrates Louvain-enhanced community identification for modularity-driven community aggregation with random-walk-based feature selection and all its phases as depicted in figure 1 which shows a high-level architecture of the proposed framework for applying node classification methods to multi-relation graph data. RWR techniques are used firstly to produce node sequences or paths for better feature selection to monitor the adaptive camouflage tactics of fraudsters and confirm the retention of the most pertinent characteristics, which are then fed into an aggregation layer that sums the features in each random walk.

Subsequently, the Vertex Community Detection Based Aggregator has been incorporated with Louvain clustering, which segments the graph into dense communities that have high modularity indicative of probable fraud rings, hence facilitating aggregation at the community level instead of being limited to individual neighborhoods. Thereafter, the outcomes from random walk and Louvain clustering will be concatenated together to produce the new features. These features are used to generate the multi-embedding layers and learn from Yelp's multi-layer graph data. Afterwards, the aggregation layer has been used to combine the features of multi-embedding layers and feed them into the first RGCN layer to learn from the information produced in the representation learning. Finally, RGCN layers have been adopted to facilitate relation-aware feature propagation among these communities, therefore capturing heterogeneous interactions among items,

users, and transactions, with a Leaky Rectified Linear Unit (ReLU) layer, a dropout layer, and the Softmax activation function to decide whether the node is fraudster or normal node.

This design offers three fundamental advantages. Initially, by utilizing Louvain clustering, the model distinguishes structural communities that fraudsters cannot readily disguise, hence assuring robustness against camouflage and noise. The incorporation of RWR feature selection with RGCN improves adaptation across diverse fraud graphs, effectively capturing both local and global relational patterns. Third, Louvain's nearly linear scalability Huang et al. (2025) allows the framework to function efficiently on extensive, real-world graphs, while simultaneously providing interpretability, as the identified communities may be directly analyzed by investigators as potential fraud groups. Collectively, these advancements render RWR-RGNN a resilient, scalable approach for fraud detection. The following subsections formally define the problem statement and explain algorithm in detail.

3.1 Problem Definition

In this section, the graph $\mathcal{G} = (V, E)$ is defined by the set of nodes V and edges E. Each edge in E represents the connections among nodes in fraud detection graph-based problem. Thereafter, we present how to apply RGCN to multi-relation fraud detection problems.

Definition 1. Multi-Relation Graph

A multi-relation graph is defined as $\mathcal{G} = \{V, X, \{E_r\}|_{(r=1)}^{R}, Y\}$, where V is the set of nodes $(v_1, \ldots, v_n)$. Each node v_i has a d-dimensional feature vector $x_i \in R_d$ and $\boldsymbol{x} = \{x_1, \ldots, x_n\}$ represents a set of all node features. $e_{(i,j)}^{r} = (vi, vj) \in E_r$ is an edge between v_i and v_j with a relation $r \in \{1, \ldots, R\}$. Note that any edge can be associated by several relations, and there are R different relation types. Y is the label for each node in V.

Definition 2. Graph-Based Fraud Detection

The target entity in the fraud detection problem is represented by node v, whose suspicions should be vindicated, which can be for example, viewed in a trading system transaction or a review on the reviews website. Label assigned to the node is $y_v \in \{0, 1\} \in Y$, where 0 denotes normal and 1 denote suspicious or fraudster. The relation might be communications, rules, or any common properties across nodes for example, two reviews from the same user or transactions from the same devices. Node classification problem is a graph-based fraud detection model that is trained using both the multi-relationship graph and the labeled node data. Then, the converged models are used to predict the fraud unlabeled nodes.

3.2 The Proposed Framework Components

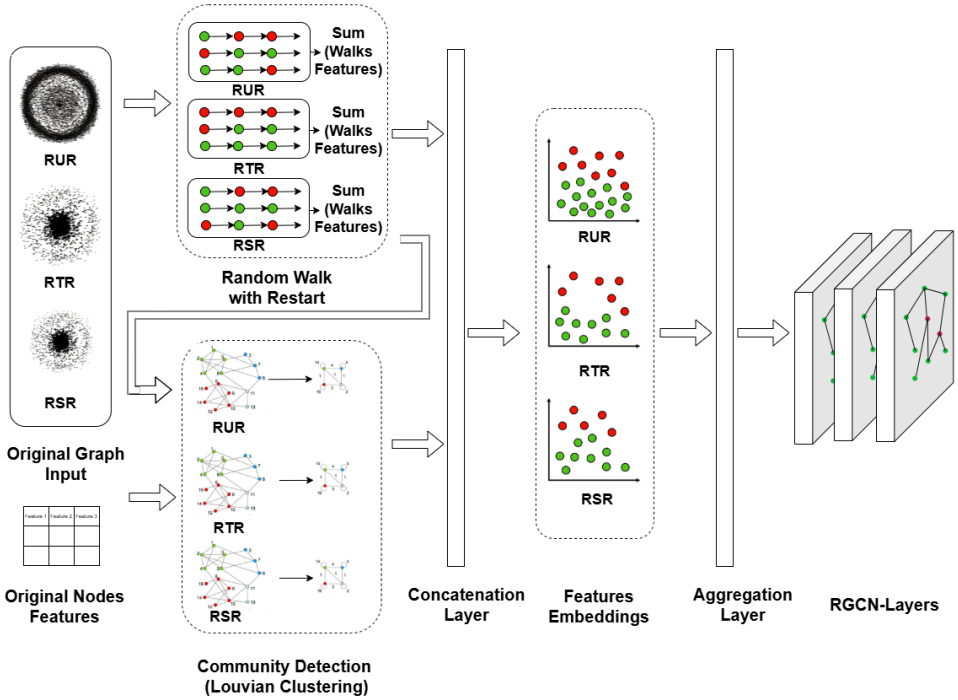
This subsection elucidates the rationale behind each step of the proposed fraud detection framework. The approach leverages three key components:

1. A depth-wise RWR-based feature selector that tracks the fraudster's evolving camouflage patterns,
2. A vertex-community-detection-based aggregator that summarizes node behavior within local subgraphs, and
3. A Relation Graph Convolutional Network (RGCN) for effective feature learning across diverse graph relationships. A primary challenge in graph-based fraud detection lies in generating informative features that reliably distinguish fraudulent nodes from legitimate ones. To address this challenge, a versatile graph processing framework capable of classifying nodes across various graph relationships is introduced.

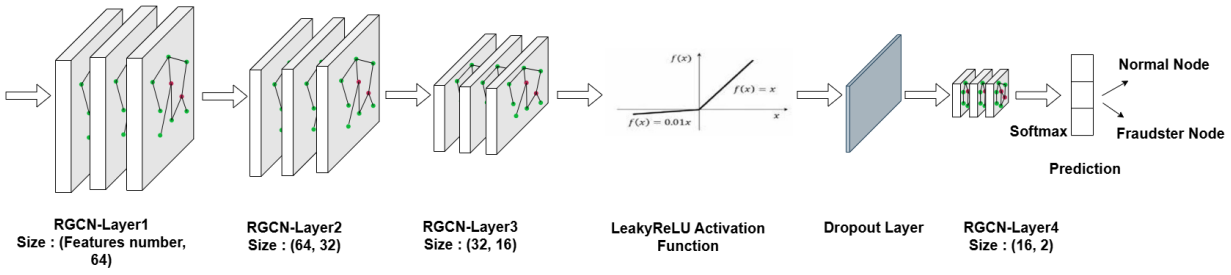
Each of the components is presented in the following before introducing the exact algorithm steps, which utilize all three components to classify the nodes either to be fraudsters or normal for an undirected graph.

3.2.1 Fraud Neighbors Selectors

RWR Jin et al. (2019) calculates each node's proximity to a specified query node s in a graph. It is often referred to as Personalized PageRank (PPR) with a single seed node. The random surfer assumed by RWR



(a) The initial phase of the RWR-RGCN architecture commences with the input layer, containing the original relational graph data and the node features linked to each entity (for further details, see Section 4.1 Dataset). This input undergoes processing via the Random Walk with Restart (RWR) for each relation type, generating walk-based feature representations. The original graph and RWR outputs are subsequently input into the community discovery module, employing Louvain clustering to ascertain structural communities. The outputs are concatenated and input into the feature embedding layer, followed by an aggregation phase that merges the learnt representations before they are processed through the stacked RGCN layers for additional relational reasoning illustrated in part (b).



(b) The output from (a) will be inputted to three consecutive RGCN layers that systematically diminish feature dimensionality: the first layer converts input data into 64-dimensional representations, the second compresses them to 32 dimensions, and the third further reduces them to 16 dimensions. A LeakyReLU activation function is utilized to introduce non-linearity, while a dropout layer is employed to mitigate overfitting. The processed characteristics are subsequently transmitted to a final RGCN layer, which transforms the 16-dimensional representations into a 2-dimensional output space. A softmax function is utilized on these outputs to produce a prediction, classifying each node as either Normal or Fraudster.

Figure 1: A high level architecture proposed applying in node classification methods for multi-relation graph data. It consists of two parts (a) and (b) (see Section 3.2 for details)

begins at node s. With a likelihood of $1 - c$, the surfer goes to one of its nearby nodes, or with a probability of c it restarts at node s. Every neighbor v that the surfer goes from u to its neighbor is chosen with a probability that is proportionate to the weight in the edge (u, v). A high score indicates that the nodes s and u are highly connected.

Since RWR offers a personalized ranking with respect to a node Jin et al. (2019), it has been utilized in numerous graph applications, including detecting community, graph partitioning, graph matching, graph sampling Nakajima & Shudo (2022), link prediction, and ranking. RWR ranking findings were utilized by Sun et al. (2005) to identify anomalies. Empirical research by Gleich & Seshadhri (2012) has demonstrated that random walk-based models are competitive with other cut-based methods for identifying graphs of local clustering. Our assumption when using RWR is to find the fraud versus normal paths within each nodes neighborhood.Through experimentation, we found that the features generated from the i-th path were more discriminative to classify each node as either fraud or normal.

3.2.2 Vertex Community Detection Based Aggregator

The Louvain algorithm can be used to identify communities from the networks (Baltsou et al., 2022). We assumed that the communities where fraudsters represent are the path to a group of fraudsters that tend to disguise themselves in normal behavior to be misclassified. This approach consists of two iteratively repeated steps in order to optimize the modularity called pass; defines as a scalar value measured by the density of links within communities relative to links between communities, which is the modularity of a partition, which is between -1 and 1 (Blondel et al., 2008).

The first step, such as figure 2 is to begin with N nodes that are fraud and normal nodes. Elect a distinct community for each network node. Thus, the number of communities in this first split equals the number of nodes. Then, it takes into account each node's neighbors j and calculate the increase in modularity that would result from shifting each node from its community and into the community of j. Then, and only if the increase is positive, node i is added to the community for which this improvement is maximizing the modularity. But if it is not positive, i node remains in the original community that it belongs to. Until no extra improvement is accomplished, this step is conducted repeatedly and sequentially for every node. Then the first step comes to the end when a local maxima of modularity is achieved, that is, when no single move may further increase modularity. The second step is constructing a new network in which its nodes are the communities found in the first step. A passes should be iterated until there are no more modifications and the maximum level of modularity is reached.

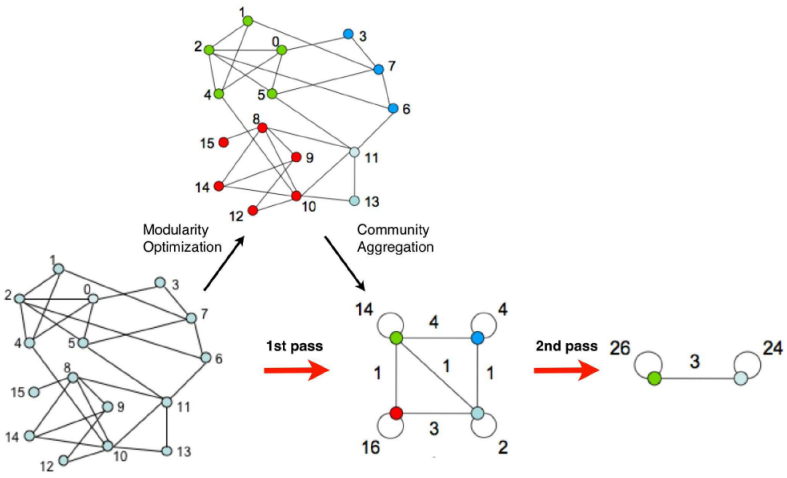


Figure 2: Blondel et al. (2008) visualizes the phases in order to optimize the modularity for louvian clustering model

3.2.3 Relational Graph Convolution Network (RGCN) As Node Classifier

RGCN is defined from labeled multi-graphs as $\mathcal{G} = (V; E; R)$ with nodes (entities) $v_i \in V$ and labeled edges (relations) $(v_i; r; v_j) \in E$, where $r \in R$ is a relation type. In Schlichtkrull et al. (2018) the authors innovate RGCN as a GCN extension to find the neighborhoods in local graph but for multi-relation data. We consider a node classification task to classify each node (review) in the graph fraud as or not and analyze the neighbors from each relation of the same node.

R-GCN layers are used for (semi-)supervised classification of nodes (entities) and a softmax layer (per node) on the final output layer. On every labeled node, we minimize the cross-entropy loss and then predict the test set for unseen data.

$$\mathcal{L} = -\sum_{i \in \mathcal{Y}} \sum_{k=1}^{K} t_{ik} \ln h_{ik}^{(L)} \tag{1}$$

In (Eq. 1) where $\ln h_{ik}^{(L)}$ is the k-th element of the model output for the i-th labeled node, and y is the set of node actual labels. The associated ground truth label is shown by t_{ik}. In the experiments, we use (full-batch) gradient descent methods to train the model.

3.2.4 Algorithm Details

Algorithm 1 presents the pseudocode for the proposed RWR-RGCN framework. The algorithm takes as input a multi-relation undirected graph, its edge index, a query node, random walk parameters (path length and restart probability), initial community assignments, a new community label, and node features. Initialization includes a path to store node sequences from random walks, edge weights (w), a restart vector (s) initialized with a single 1 at the query node, a row-normalized adjacency matrix $\boldsymbol{P}$, an iteration vector ($\boldsymbol{x}$) initialized to s, an error tolerance (e) for power iteration, and the initial community assignments (C). The RWR-RGCN algorithm begins by establishing a heterogeneous, multi-relational network, initialized with the nodes' original feature sets. Subsequently, it employs Random Walks with Restart (RWR) to trace potential fraud and normal pathways, using pre-defined threshold sequences. The restart probability, alpha (α), dictates the scope of these walks, favoring local structures with higher values. Next, the algorithm constructs enriched feature vectors by aggregating node features from these identified pathways. Then, community structures are detected using the Louvain clustering technique, which iteratively optimizes modularity until stable communities are formed. Finally, the augmented node features and community assignments are passed into a Relational Graph Convolutional Network (RGCN), where, for each layer or relation, the RGCN obtains unique feature embeddings for the existing nodes within that layer and calculates message passing to aggregate information from neighboring nodes; this processed information is then passed to a HeteroRGCN layer with a LeakyReLU activation function, which integrates information from all relations to produce logits that determine the class label (fraud or normal) for each node.

4 Performance Evaluation

Due to the inherent class imbalance in fraud detection, specialized evaluation metrics are crucial, as traditional accuracy can be misleading. To accurately assess model performance, we focus on recall, which measures the model's ability to identify all actual fraudulent instances, a critical metric in imbalanced scenarios where minimizing false negatives is paramount, and Receiver Operating Characteristic Area Under Curve (ROC-AUC), which quantifies the model's performance across various decision thresholds by depicting the trade-off between true positive and false positive rates. A high ROC-AUC indicates robust performance, demonstrating the model's ability to discriminate between fraudulent and legitimate activities regardless of the chosen threshold. As pointed out in (Sun et al., 2005), these metrics provide a comprehensive and objective evaluation of fraud detection models, particularly in the presence of significant class imbalance. And

Algorithm 1 RWR-Relation-GCN

Inputs: Edge index e, starting node v_0, path length t, restart probability α, graph $\mathcal{G}$, community C, new community new_C, feature input $\boldsymbol{f}$, layer L
Output: Logit for each node $v_0 \in (0, 1)$
Initialize: $path = [v_0]$ (path stores nodes in a walk), w (weight from v_i to v_{i+1}), s (restart vector with all entries 0 except 1 at v_0), $\mathbf{P}$ (row-normalized adjacency matrix), $x := s$, e (error tolerance for power iteration), C (initial communities)
for $v_0 = 1$ to V_n **do**
  for *path length* $<= t - 1$ and $v_i =!$ v_n **do**
    while x has not converged **do**
      $x := \alpha_s + (1 + \alpha){P_x}^T$
      $w = w(e(v_i, v_{i+1}); w \in (0; 1)$
      if w is the maximum **then**
        $rwr\_score = x$
        return rwr_score, $path+ = v_i$, $v_i + 1$
      else
        restart
      end if
    end while
  end for
end for
while $G.nodes == length(G.nodes)$ **do**
  $C = v0, j$
  $N =$ neighbors to each $v_{0,j}$
  $C = C.append(N)$
  $sum(modularity(N)); modularity \in (0; 1)$
  if Modularity After $N >=$ Modularity before N **then**
    Add node v_i to new_C
  else
    $v_i \in C$
    return new_C
  end if
end while
for n in $path$ with max(rwr_score) **do**:
  $\boldsymbol{f}(V) = sum(F$ for each $n)$
  $\boldsymbol{f}(V) = Append(new\_C(v_i))$
end for
for $L = 1$ to $|\boldsymbol{f}_i|$ **do**
  $e_f^{(l)} \leftarrow$ Embed$(\boldsymbol{f}, l)$ for all $\boldsymbol{f} \in \mathbf{F}, l \in L$
  $Wh = e_f^{(l)}$
  Calculate h message passing
  $h = Wh * h$
end for
while not convergence **do**
  $HeteroRGCNLayer(G, h)$
  $LeakyReLU$
  return $logit(n)$
end while

to ensure meaningful comparison with prior work, the evaluation has been conducted on the Yelp dataset, which is commonly used in the literature for benchmarking fraud detection models.

The comparative experiments have been performed between RWR and eight different baseline models and the latest state-of-the-art models, which were specifically developed for fraud detection or imbalanced graphs, which are briefly introduced as follows:

- GCN (Schlichtkrull et al., 2018): is a representation of the vanilla graph convolution approach, establishing a straightforward and well-structured hierarchical propagation rule for neural network models.
- GraphSAGE (Hamilton et al., 2017): is a representative non-spectrogram approach. This method offers a comprehensive inductive framework for each node, which samples and aggregates the features of its local neighbors to produce an embedding, rather than training an independent embedding. It enhances the scalability and adaptability of GNNs.
- GraphConsis (Liu et al., 2020): is a model that integrates context embedding with nodes, eliminates inconsistent neighbors, and produces related sample probabilities. The embeddings of sampled nodes from each relation are integrated utilizing a relation attention technique.
- CARE-GNN (Dou et al., 2020): a layer employing a label-perceived similarity metric is utilized to identify information-rich nearby nodes. Uses a special approach to select the most informative neighboring nodes for aggregation
- PC-GNN (Liu et al., 2021): uses samplers to build subgraphs and sample informative neighbors for aggregation.
- FRAUDRE (Zhang et al., 2021): a model that separates relation-specific embeddings and reconstructs them to improve resilience against camouflage and relationship inconsistencies in multi-relational graphs.
- RioGNN (Peng et al., 2021): a Label-aware similarity measure employs a two-layer framework for neighbor selection. Employs the Actor-Critic (AC) algorithm with a discrete strategy to iteratively determine the filter thresholds for various relationships, utilizing these thresholds as relational weights to aggregate neighboring entities across multiple relations.
- DOS-GNN (Jing et al., 2024): a dual-feature aggregation framework that maintains both similarity and dissimilarity signals while implementing oversampling in the embedding space to mitigate class imbalance in fraud detection.

4.1 Dataset

The proposed framework is evaluated the node classification task using the selected benchmark dataset, Yelp. Yelp review datasets focus on hotels and restaurants, which filtered the reviews as spam and recommended them as normal reviews, and the dataset has been used to study fraud detection in multi-relation graph convolution networks (Dou et al., 2020). Yelp includes 32 handcrafted features.

Dou et al. (2020) Yelp [1] datasets designated as multi-relations graph with three relation types. 1) R-S-R: It links reviews that belong to the same product that has the same star rating (15); 2) R-U-R: It links reviews written by the same user; 3) R-T-R: It links two reviews that were posted in the same month and belong to the same product. Yelp datasets contain 45,954 nodes with 14.5% of fraud nodes, meaning around 6,663 nodes are fraudsters with a total of 3,846,979 edges. It divided the relations into 49,315 edges, 573,616 edges, and 3,402,743 edges for R-U-R, R-T-R, and R-S-R, respectively.

4.2 Evaluation Metrics

In most classification tasks, the methods used to evaluate the models proposed are confusion matrices that can summarize the results and errors in the training pipeline, table 1 shows that all performance matrices match.

Gleich & Seshadhri (2012) mentioned that there are many matrices to evaluate the proposed model performance, and we used AUC and recall as popular assessments for binary classification.

[1] Yelp datasets link: https://github.com/YingtongDou/CARE-GNN/tree/master/data

Table 1: Confusion Matrix

	Actual positive	**Actual negative**
Predicted positive	True positive (TP)	False positive (FP)
Predicted negative	False negative (FN)	True negative (TN)

Recall in (Eq. 2), or the True Positive Rate (TPR), calculates the ratio of the positive class that the model accurately anticipated to be positive. Unbalance has no effect on recall since it solely depends on the positive class. The number of negative samples that are incorrectly identified as positive is not taken into account by recall, which can be problematic in situations where there is a class imbalance in the data and a large number of negative samples.

$$\text{Recall} = TPR = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{2}$$

The receiver operating characteristics (ROC) curve is a method for evaluation that plots the false positive rate over the true positive rate, putting together a visual representation of the trade-off between accurately and wrongly labeled positive and negative data points. Gleich & Seshadhri (2012) Thresholding can be employed in models that generate continuous probabilities to construct a series of points along ROC space. Using this, a solitary summary measure is the area under the ROC curve (AUC), which is frequently used to evaluate the effectiveness of different models as visualized in figure 3.

$$\text{G-Mean} = \sqrt{\text{TPR} \cdot \text{TNR}} = \sqrt{\frac{\text{TP}}{\text{TP} + \text{FN}} \cdot \frac{\text{TN}}{\text{TN} + \text{FP}}} \tag{3}$$

G-Mean (Eq. 3), mean of True Positive Rate (TPR) and True Negative Rate (TNR) instantaneously retaining both values relatively balanced. The higher scores of the G-Mean indicate a high performance of the approaches (Shaha et al., 2021).

4.3 Evaluation Setup

This section displays the results from the binary node classification task. Additionally, the proposed framework is tested in different random walks in figure 3 and figure 4. In the experiments, we used Yelp multi-relations undirected graph datasets for a review system that has 2 classes of nodes: 39,291 normal and 6,663 fraudsters, with 14.5% of fraud nodes. The Yelp dataset was split into three divisions: 40%, 30%, and 30% for training, validation, and testing, respectively.

Table 2: Presents the performance of proposed approach RWR-RGCN on the Yelp dataset (%) though the experiments for number of walks.

#Walks	**AUC**	**Recall**	**G-Mean**
2	50.04	50.00	50.00
3	51.25	50.18	50.00
4	50.02	50.00	50.00
5	**82.58**	92.46	**75.19**
6	82.36	**94.56**	74.89
7	81.79	92.61	74.77
8	82.09	91.06	75.24
9	69.52	79.83	68.67

The results of fraud detection are visible in table 2, which uses test data from the Yelp multi-relation graph to identify the fraud nodes. All the models are perform on random walks from 2 to 9. The performance of the

proposed system was assessed and evaluated using AUC, recall, and G-Mean results, which are demonstrated in figure 3 and figure 4 charts, respectively, highlighting the performance matrix across the epochs till the models are converged. With the aim of calculating the performance matrices, we first calculated True Positive (TP), False Positive (FP), False Negative (FN), and True Negative (TN) for the dataset Yelp.

Figure 3 and figure 4 show that the results are different in each random walk. It is fascinating to inspect when random walks are getting longer; the RGCN results become different. The result with random walks from 5 to 8 was the best.

Such behavior is expected since the random walk captures and selects the fraud features from the neighbors node, which differentiates between normal and not normal behavior in the representation learning. In contrast with the result in random walks 2 to 4 and 9, the worst is 9, as the small random walk has a few features to represent the differentiable behavior to be captured, or the higher random walk is the distraction for differentiable properties of the features deteriorating after 9 random walks and more, leading to reduced smoothness or continuity in capturing the fraud structure and properties. As in figure 5, the models' loss with random walks from 5 to 8 in both training and validation datasets were decreased with better recall and AUC.

All factors considered, these outcomes demonstrate the strength of RGCN layers and proposed implementation. Since the fraud selector and two aggregation layers, one from the community detection and one from RGCN across different relations, are made to share the features of nodes in the convolution process. These benefits are vital in fraud detection applications.

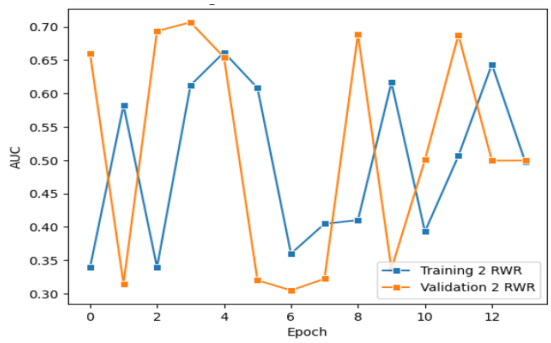
4.4 Evaluation Results

The comparison between our proposed RWR-RGNN and a wide range of baseline methods has been conducted to fully evaluate its effectiveness. These include standard GNNs like the Graph Convolutional Network (GCN) Schlichtkrull et al. (2018) and GraphSAGE (Hamilton et al., 2017), as well as more advanced graph-based techniques for detecting fraud and anomalies that are specifically designed for spatial heterophily, such as GraphConsis (Liu et al., 2020), CARE-GNN (Dou et al., 2020), PC-GNN (Liu et al., 2021), FRAUDRE (Zhang et al., 2021), RioGNN (Peng et al., 2021), and the latest DOS-GNN (Jing et al., 2024). These models represent the progress in mitigating noise, camouflage, imbalance, and heterophily in fraud detection.

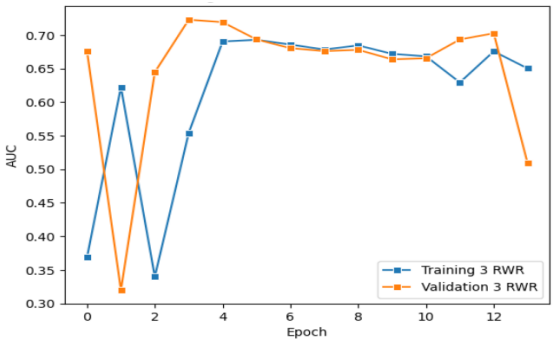
GraphConsis Liu et al. (2020) specifically addresses inconsistencies in context, features, and relationships, which makes it more robust in different situations. CARE-GNN Dou et al. (2020) is a model that is resistant to camouflage and uses reinforcement learning to adaptively sample neighbors. This technique solves the problem of fraudsters pretending to be real nodes. PC-GNN Liu et al. (2021) uses a two-step "pick and choose" method to resample neighbors and keep a balanced label distribution around nodes that are fraudulent. RioGNN Peng et al. (2021) uses relation-aware message passing with disentanglement to effectively capture heterophilic connections. FRAUDRE Zhang et al. (2021) propounds this concept by disentangling relation-specific embeddings and reconstructing them to improve resilience against relational inconsistency and complication. It is especially efficacious for multi-relational fraud graphs, wherein interactions encompass people, goods, and transactions. DOS-GNN Jing et al. (2024) introduces a dual-feature aggregation framework that clearly preserves both similarity and dissimilarity signals in embeddings, utilizing oversampling in the embedding space to address class imbalance.

The experimental results emphasize the progression from conventional GNNs to innovative fraud detection frameworks and illustrate the efficacy of the proposed RWR-RGCN. Baseline models like GCN Schlichtkrull et al. (2018), and GraphSAGE Hamilton et al. (2017) exhibit constrained performance, attaining AUCs of 53.38 and 54.39, respectively. Their dependence on local neighborhood aggregation and homophily assumptions undermines their efficacy in fraud detection, since fraudsters deliberately link to normal nodes, leading to suboptimal G-Mean values (47.36 for GCN Schlichtkrull et al. (2018) and merely 25.89 for GraphSAGE (Hamilton et al., 2017).

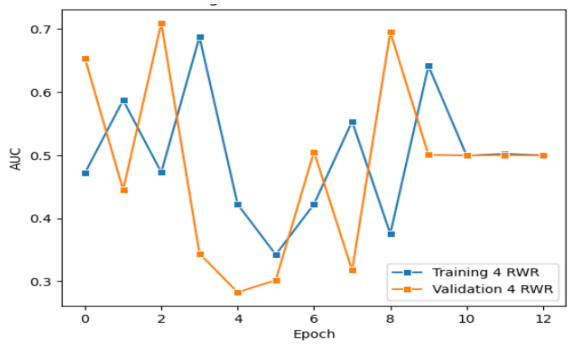
Modern specialized fraud detection models exhibit markedly improved performance by tackling specific difficulties mentioned in table 3. GraphConsis Liu et al. (2020) enhances consistency among relations and features, resulting in an increased AUC of 69.55 and recall of 66.20. CARE-GNN Dou et al. (2020) progresses



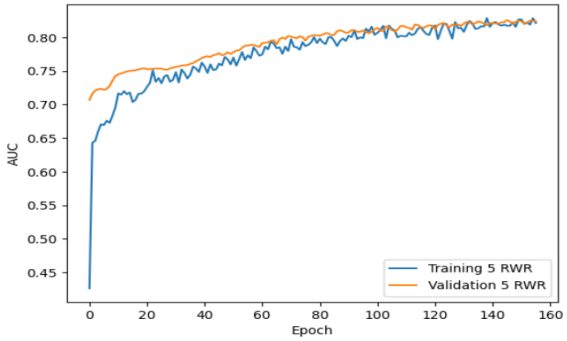
(a) Training and Validation AUC for 2 RWR



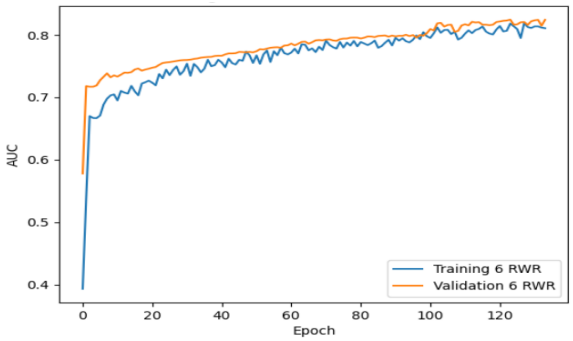
(b) Training and Validation AUC for 3 RWR



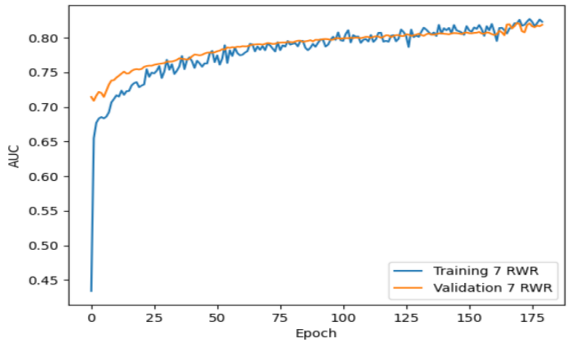
(c) Training and Validation AUC for 4 RWR



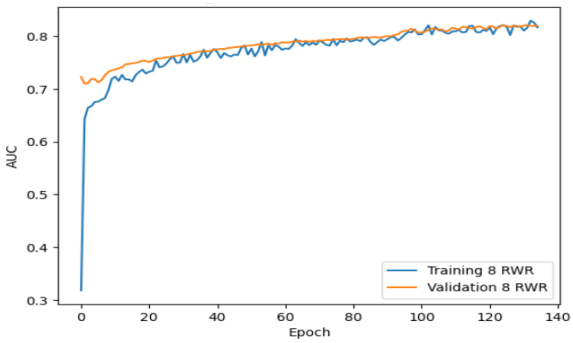
(d) Training and Validation AUC for 5 RWR



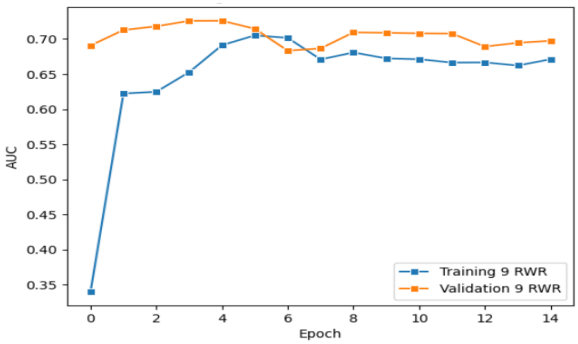
(e) Training and Validation AUC for 6 RWR



(f) Training and Validation AUC for 7 RWR



(g) Training and Validation AUC for 8 RWR



(h) Training and Validation AUC for 9 RWR

Figure 3: AUC performance matrix for different random walks from 2 to 9 walks starting from central nodes to select features as parameters in multi-relation node classification on both training and validation datasets

by using reinforcement learning for neighbor filtering, attaining an AUC of 75.70 and a G-Mean of 67.91. PC-GNN Liu et al. (2021) and FRAUDRE Zhang et al. (2021) exhibit competitive performance as well. PC-GNN Liu et al. (2021) equilibrates label distributions via resampling and achieves the greatest G-Mean of 70.88 among these models, whereas FRAUDRE Zhang et al. (2021) employs relational disentanglement to attain a balanced AUC of 72.22 and a G-Mean of 69.78. RioGNN Peng et al. (2021), which improves relation-aware message forwarding, achieves the greatest AUC among current baselines at 82.38, demonstrating the benefits of simulating multi-relational fraud interactions. DOS-GNN (Jing et al., 2024), employing dual-feature aggregation and an oversampling method, attains a well-balanced performance (AUC 81.15, Recall 82.14, G-Mean 81.66), establishing it as a formidable contender, especially in addressing imbalanced fraud situations.

Table 3: Experimental results (%) for node classification of different fraud detection methods on benchmark dataset. Some GNN models are highlighted bold denotes a significant improvement of results on the Yelp dataset.

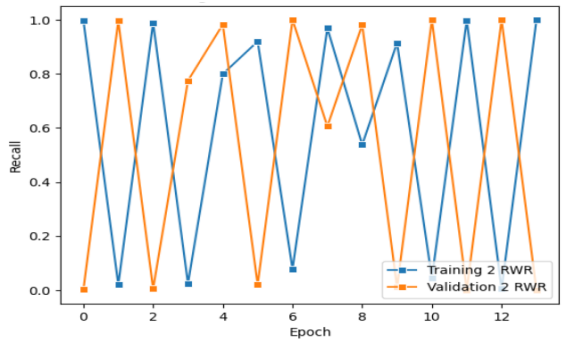
Method	**Metric**	**AUC**	**Recall**	**G-Mean**
Baselines	GCN Schlichtkrull et al. (2018)	53.38	50.43	47.36
	GraphSAGE Hamilton et al. (2017)	54.39	50.00	25.89
	GraphConsis Liu et al. (2020)	69.55	66.20	58.37
	CARE-GNN (Dou et al., 2020)	75.70	71.92	67.91
	PC-GNN Liu et al. (2021)	78.50	67.21	70.88
	FRAUDRE Zhang et al. (2021)	72.22	66.98	69.78
	RioGNN Peng et al. (2021)	82.38	75.08	-
	DOS-GNN Jing et al. (2024)	81.15	82.14	**81.66**
Ours	RWR-RGCN #walks=5	**82.58**	92.46	75.19
	RWR-RGCN #walks=6	82.36	**94.56**	74.89

The suggested RWR-GCN surpasses or equals these leading baselines or state-of-the-art models across multiple criteria. With five random walks, RWR-GCN reaches an AUC of 82.58, somewhat exceeding RioGNN Peng et al. (2021) and DOS-GNN (Jing et al., 2024), while significantly enhancing recall to 92.46. This model demonstrates its capacity to identify fraudulent nodes more thoroughly, which is essential in fraud detection applications where overlooking fraud incurs significant costs. The inclusion of six random walks elevates recall to 94.56, thus validating the efficacy of random-walk-based neighbor refinement in identifying fraudulent behavioral patterns. Despite the G-Mean values of RWR-GCN (75.19 for five walks and 74.89 for six walks) being marginally worse than DOS-GNNs Jing et al. (2024) 81.66, the lower accuracy is offset by a considerably higher recall, which is frequently more advantageous in fraud detection scenarios.

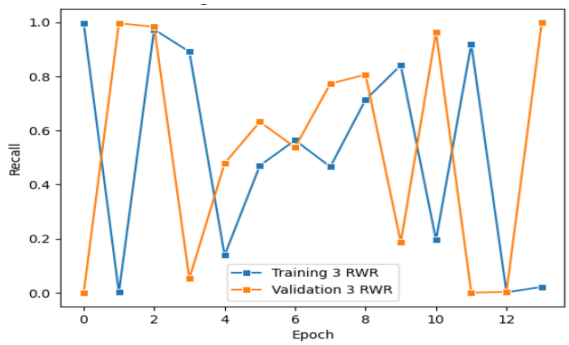
5 Discussion

In this work, we articulated a fair evaluation experimental setup for RGCN and accomplished extensive experiments on graph classification. Since the data used in the experiments for a multi-relation graph is a Yelp dataset, the RWR sequences for each node were the best choice for selecting the suggested nodes to extract the most informative features from neighbors, which made a good performance to highlight the disguise sequence or path that the fraudsters pattern follows. And its results outperform those of comparing with baseline models.

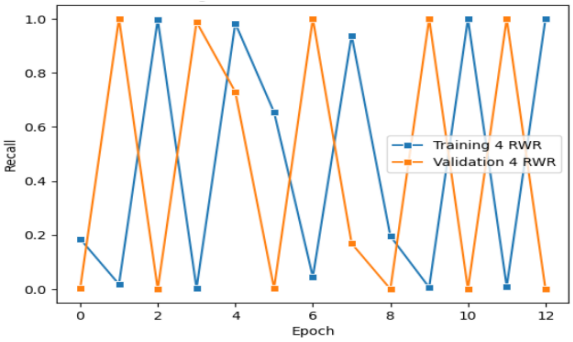
Overall, these results indicate that RWR-GCN proposals are an effective alternative that unifies sequence-level exploration with community and relation-aware aggregation. By relying on probabilistic neighbor weighting rather than reinforcement learning or oversampling, it avoids excessive computational complexity and synthetic distortion. The higher recall rates indicate its ability to reduce false negatives, while the high AUC values demonstrate balanced detection effectiveness. RWR-GCN is a robust framework that tackles



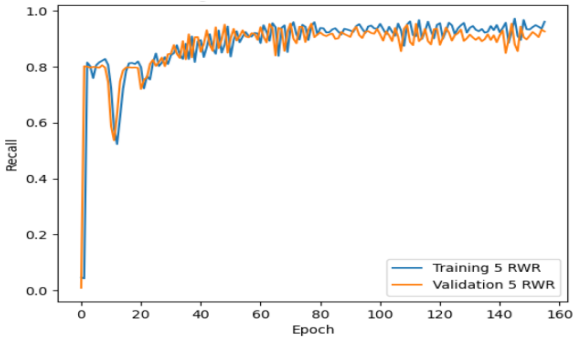
(a) Training and Validation Recall for 2 RWR



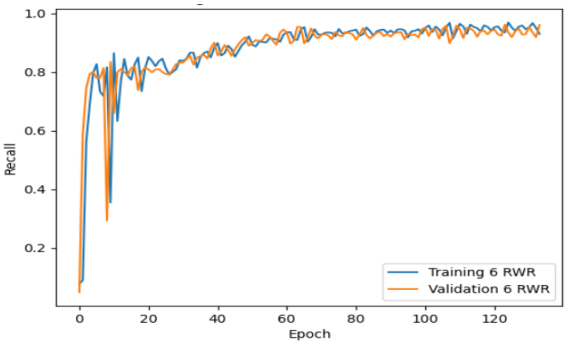
(b) Training and Validation Recall for 3 RWR



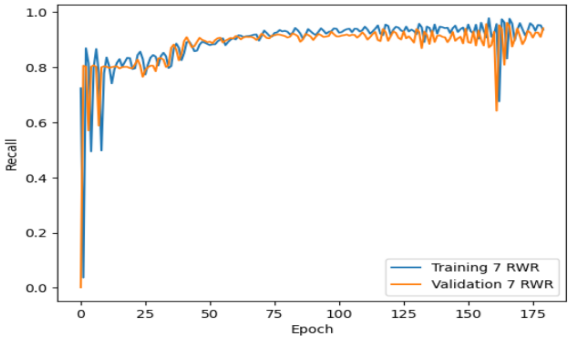
(c) Training and Validation Recall for 4 RWR



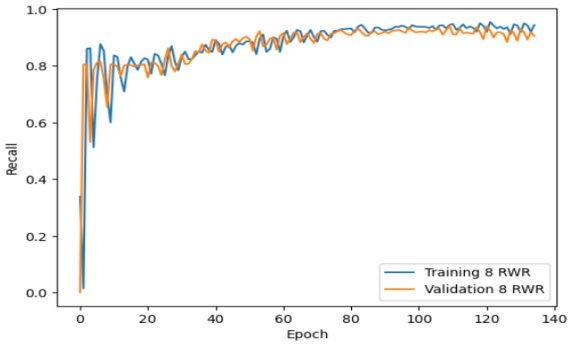
(d) Training and Validation Recall for 5 RWR



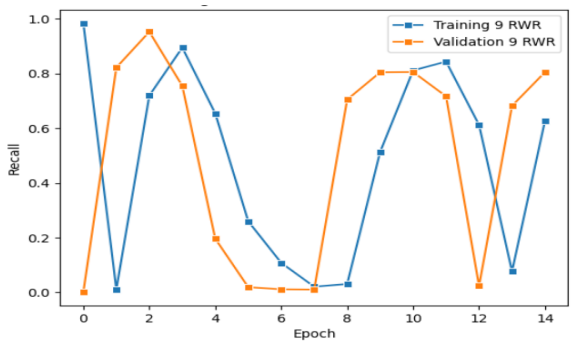
(e) Training and Validation Recall for 6 RWR



(f) Training and Validation Recall for 7 RWR



(g) Training and Validation Recall for 8 RWR



(h) Training and Validation Recall for 9 RWR

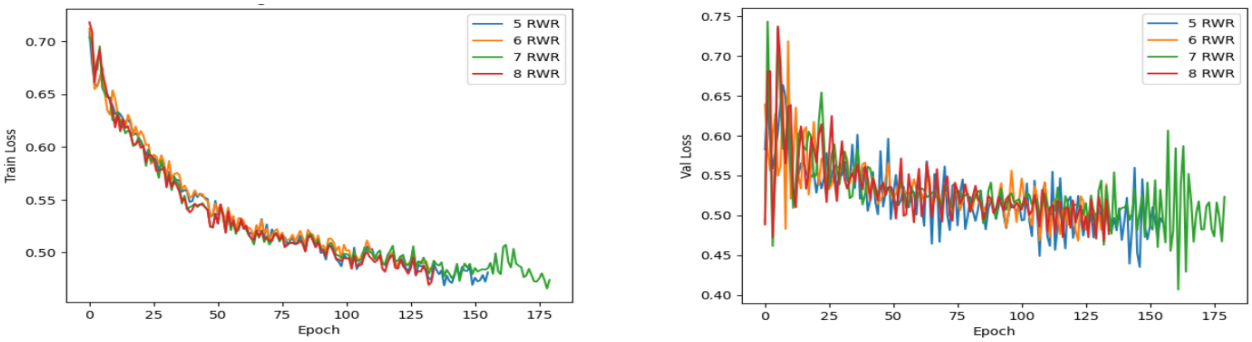
Figure 4: Recall performance matrix for different random walks from 2 to 9 walks starting from central nodes to select features as parameters in multi-relation node classification on both training and validation datasets

the issues associated with node, relation, and oversampling-based models, facilitating comprehension and scalability for extensive fraud detection.

These results indicate that RWR-GCN provides a scalable and efficient alternative that integrates path-level exploration with community and relation-aware aggregation. It utilizes probabilistic neighbor weighting instead of oversampling or reinforcement learning, hence circumventing excessive processing complexity and synthetic distortion. The elevated recall rates underscore its capacity to minimize false negatives, while the robust AUC values validate its balanced detection efficacy.

Lastly, it is interesting to see in figure 3 the relatively weak performance of the random walks between 2 to 4 and 9 walks. A possible explanation could be that the explicitly weak informative selection neighbors in the graph from each relation as long as the walks select informative sequences of nodes that provide features to the disguise behavior of the fraudster nodes. The extra aggregation layer that was added from the community detection and random walk models showed great improvements for multi-relation graph node classification. This will allow future work to concentrate on the node selection of fraud neighbors for better enhancement features of each node.

This draws attention to the innovative RGCN approach's shortcomings and raises several questions for further research and future work. Fraudsters who tend to disguise their deceptive behaviors by forming genuine connections with regular users pose a challenge to GNN. The suggested modifications to the local community model and random walks with restart, which have already been utilized as a model-based graph topology, produced significantly superior outcomes. For arbitrary node classification, the resulting model can serve as a more robust baseline. In addition, add an auto-encoder or node embedding layer to the architectures, which might enhance the model results. These developments should significantly help future node classification benchmarks.



(a) Training Loss when RWR from 5 to 8

(b) Validation Loss when RWR from 5 to 8

Figure 5: Training and validation loss for random walks from 5 to 8 walks starting from central nodes to select features as parameters in multi-relation node classification on both training and validation datasets

All in all, the proposed RWR-GNN has implemented a lightweight random-walk-based neighbor strategy that inherently prioritizes informative neighbors and equilibrates long-range dependencies, offering a scalable and cohesive solution for fraud detection in heterogeneous and imbalanced graphs. Unlike GraphConsis Liu et al. (2020) or CARE-GNN (Dou et al., 2020), it eschews heuristic or reinforcement-intensive neighbor selection. Unlike PC-GNN Liu et al. (2021) and DOS-GNN (Jing et al., 2024), it does not depend exclusively on oversampling or dual aggregation; rather, it utilizes RWR to inherently weight informative neighbors according to walk probabilities. RWR-GNN is positioned as a scalable and theoretically robust solution that integrates neighbor refinement with fraud detection. Unlike PC-GNN Liu et al. (2021) and DOS-GNN Jing et al. (2024), which mitigate class imbalance via sampling or oversampling, the model has diminished imbalance effects by consistently reinforcing minority nodes during propagation through random-walk weighting, thereby circumventing the potential for synthetic noise or structural distortion. Furthermore, FRAUDRE (Zhang et al., 2021), RioGNN (Peng et al., 2021), and RWR-GNN offers a lightweight yet robust alternative that effectively captures both homophilic and heterophilic signals without considerable computational bur-

den. By integrating these strengths, RWR-GNN provides a scalable, equitable, and resilient architecture that concurrently addresses heterophily, class imbalance, and noisy neighbor connections, attaining exceptional performance across various fraud detection contexts.

6 Conclusion

The reviewing systems witness fraudsters attitude to act as normal users to get over the anti-fraud systems, distribute fake information, or steal the private information of end users. The proposed framework RWR-RGCN in this paper was developed to enhance the fraud detection for the Yelp dataset and solve the challenges in baseline models by working with random walks with restart and community detection to expand the representation learning of the model to detect the fraud pattern; then the multi-RGCN classifier is used to predict the binary classification. The proposed framework leads to enhancement in AUC, as the main challenge is the processing power or hardware required to run the experiments, which could enhance the model results. The AUC resulted from the proposed model achieving 82.58% and the highest recall (over 92%94%) while maintaining competitive AUC and G-Mean scores, which provided a better prediction than the baseline. The experiments demonstrate remarkable features from different random walks. In conclusion, the study highlights the significance of fraud detection in multi-relation graph data. The choice of different random walks of the node selection has considerable control over the performance of RGCN results.

References

Georgia Baltsou, Konstantinos Christopoulos, and Konstantinos Tsichlas. Local community detection: A survey. *IEEE Access*, 10:110701–110726, 2022.

Yuanchen Bei, Sheng Zhou, Jinke Shi, Yao Ma, Haishuai Wang, and Jiajun Bu. Guarding graph neural networks for unsupervised graph anomaly detection. *IEEE Transactions on Neural Networks and Learning Systems*, 36:16840–16853, 2025.

Vincent D Blondel, Jean-Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre. Fast unfolding of communities in large networks. *Journal of statistical mechanics: theory and experiment*, 2008:P10008, 2008.

Xiangyu Dong, Xingyi Zhang, and Sibo Wang. Rayleigh quotient graph neural networks for graph-level anomaly detection. 2024.

Yuxiao Dong, Nitesh V. Chawla, and Ananthram Swami. Metapath2vec: Scalable representation learning for heterogeneous networks. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 135–144, 2017.

Yingtong Dou. Robust graph learning for misbehavior detection. In *Proceedings of the 15th ACM International Conference on Web Search and Data Mining*, WSDM '22, pp. 1545–1546, 2022.

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 and Knowledge Management (CIKM '20)*, pp. 315–324. Association for Computing Machinery, 2020.

David F Gleich and C Seshadhri. Vertex neighborhoods, low conductance cuts, and good seeds for local community methods. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 597–605, 2012.

Shengbo Gong, Jiajun Zhou, Chenxuan Xie, and Qi Xuan. Neighborhood homophily-based graph convolutional network. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*, pp. 3908–3912, 2023.

Will Hamilton, Zhitao Ying, and Jure Leskovec. Inductive representation learning on large graphs. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (eds.),

Advances in Neural Information Processing Systems, volume 30, pp. 1025–1035. Curran Associates, Inc., 2017.

Junwei He, Qianqian Xu, Yangbangyan Jiang, Zitai Wang, and Qingming Huang. Ada-gad: Anomaly-denoised autoencoders for graph anomaly detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 8481–8489, 2024.

Hailong Huang, Jiahong Yang, Hang Zeng, Yaqin Wang, and Liuming Xiao. Self-organizing maps-assisted variational autoencoder for unsupervised network anomaly detection. *Symmetry*, 17:520, 2025.

Woojeong Jin, Jinhong Jung, and U Kang. Supervised and extended restart in random walks for ranking and link prediction in networks. *PloS one*, 14:e0213857, 2019.

Shixiong Jing, Lingwei Chen, Quan Li, and Dinghao Wu. Dos-gnn: Dual-feature aggregations with over-sampling for class-imbalanced fraud detection on graphs. In *2024 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–8, 2024.

Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations*, 2017.

Eren Kurshan and Hongda Shen. Graph computing for financial crime and fraud detection: Trends, challenges and outlook. *International Journal of Semantic Computing*, 14:565–589, 2020.

Eren Kurshan, Hongda Shen, and Haojie Yu. Financial crime & fraud detection using graph computing. application considerations outlook. *In 2020 2nd international conference on transdisciplinary AI (transAI)*, pp. 125–130, 2020.

Ao Li, Zhou Qin, Runshi Liu, Yiqun Yang, and Dong Li. Spam review detection with graph convolutional networks. *In Proceedings of the 28th ACM international conference on information and knowledge management*, pp. 2703–2711, 2019.

Pei-Cheng Li and Cheng-Te Li. Tcgnn: Text-clustering graph neural networks for fake news detection on social media. *In Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pp. 134–146, 2024.

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 (WWW '21)*, pp. 3168–3177. Association for Computing Machinery, 2021.

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 43rd international ACM SIGIR conference on research and development in information retrieval*, pp. 1569–1572, 2020.

Kazuki Nakajima and Kazuyuki Shudo. Social graph restoration via random walk sampling. In *2022 IEEE 38th International Conference on Data Engineering (ICDE)*, pp. 01–14, 2022.

Hao Peng, Ruitong Zhang, Yingtong Dou, Renyu Yang, Jingyi Zhang, and Philip S. Yu. Reinforced neighborhood selection guided multi-relational graph neural networks. *ACM Transactions on Information Systems (TOIS)*, 40:1–46, 2021.

Michael Schlichtkrull, Thomas N. Kipf, Peter Bloem, Rianne Van Den Berg, Ivan Titov, and Max Welling. Modeling relational data with graph convolutional networks. *The Semantic Web. ESWC 2018. Lecture Notes in Computer Science*, 10843:593–607, 2018.

Ricko Shaha, Dipon Talukder, MD Asif Iqbal, and Md Mokammel Haque. Tos: A relative metric approach for model selection in machine learning solutions. In *2021 IEEE international conference on robotics, automation, artificial-intelligence and internet-of-things (RAAICON)*, pp. 26–31, 2021.

Chuan Shi, Binbin Hu, Wayne Xin Zhao, and Philip S. Yu. Heterogeneous information network embedding for recommendation. *IEEE Transactions on Knowledge and Data Engineering*, 31:357–370, 2019.

Jimeng Sun, Huiming Qu, Deepayan Chakrabarti, and Christos Faloutsos. Neighborhood formation and anomaly detection in bipartite graphs. In *5th IEEE international conference on data mining (ICDM'05)*, pp. 8–pp, 2005.

Xin Tian and Yuan Meng. Relgraph: A multi-relational graph neural network framework for knowledge graph reasoning based on relation graph. *Applied Sciences*, 14:3122, 2024.

Jianyu Wang, Rui Wen, Chunming Wu, Yu Huang, and Jian Xiong. 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.

Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and Philip S. Yu. A comprehensive survey on graph neural networks. *IEEE Transactions on Neural Networks and Learning Systems*, 32:4–24, 2021.

Weizhi Xu, Junfei Wu, Qiang Liu, Shu Wu, and Liang Wang. Evidence-aware fake news detection with graph neural networks. *In Proceedings of the ACM web conference 2022*, pp. 2501–2510, 2022.

Xiaoyu Yang, Yuefei Lyu, Tian Tian, Yifei Liu, Yudong Liu, and Xi Zhang. Rumor detection on social media with graph structured adversarial learning. In *Proceedings of the 29th International Conference on International Joint Conferences on Artificial Intelligence*, pp. 1417–1423, 2021.

Hang Yu, Zhengyang Liu, and Xiangfeng Luo. Barely supervised learning for graph-based fraud detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 16548–16557, 2024.

Pengyang Yu, Chaofan Fu, Yanwei Yu, Chao Huang, Zhongying Zhao, and Junyu Dong. Multiplex heterogeneous graph convolutional network. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 2377–2387, 2022a.

Shuo Yu, Jing Ren, Shihao Li, Mehdi Naseriparsa, and Feng Xia. Graph learning for fake review detection. *Frontiers in Artificial Intelligence*, 5:922589, 2022b.

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)*, pp. 867–876, 2021.

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

Meiqi Zhu, Xiao Wang, Chuan Shi, Houye Ji, and Peng Cui. Interpreting and unifying graph neural networks with an optimization framework. In *Proceedings of the Web Conference 2021*, pp. 1215–1226, 2021.

Wen-Ming Zhuang, Chih-Yao Chen, and Cheng-Te Li. Towards robust rumor detection with graph contrastive and curriculum learning. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 18: 1–21, 2024.