Generative Dynamic Graph Representation Learning for Conspiracy Spoofing Detection

Sheng Xiang
Tongji University
Shanghai, China
xiangsheng218@gmail.com

Yidong Jiang
Tongji University
Shanghai, China
2253899@tongji.edu.cn

Yunting Chen
University of Technology Sydney
Sydney, NSW, Australia
yunting.chen@uts.edu.au

Dawei Cheng*
Tongji University & Shanghai Artificial Intelligence Laboratory
Shanghai, China
dcheng@tongji.edu.cn

Guoping Zhao
China Futures Market Monitoring Center
Beijing, China
zhaogp@cfmmc.com

Changjun Jiang
Tongji University & Shanghai Artificial Intelligence Laboratory
Shanghai, China
cjjiang@tongji.edu.cn

ABSTRACT

Spoofing detection in financial trading is crucial, especially for identifying complex behaviors such as conspiracy spoofing. Traditional machine-learning approaches primarily focus on isolated node features, often overlooking the broader context of interconnected nodes. Graph-based techniques, particularly Graph Neural Networks (GNNs), have advanced the field by leveraging relational information effectively. However, in real-world spoofing detection datasets, trading behaviors exhibit dynamic, irregular patterns. Existing spoofing detection methods, though effective in some scenarios, struggle to capture the complexity of dynamic and diverse, evolving inter-node relationships. To address these challenges, we propose a novel framework called the Generative Dynamic Graph Model (GDGM), which models dynamic trading behaviors and the relationships among nodes to learn representations for conspiracy spoofing detection. Specifically, our approach incorporates the generative dynamic latent space to capture the temporal patterns and evolving market conditions. Raw trading data is first converted into time-stamped sequences. Then we model trading behaviors using the neural ordinary differential equations and gated recurrent units, to generate the representation incorporating temporal dynamics of spoofing patterns. Furthermore, pseudo-label generation and heterogeneous aggregation techniques are employed to gather relevant information and enhance the detection performance for conspiratorial spoofing behaviors. Experiments conducted on spoofing detection datasets demonstrate that our approach outperforms state-of-the-art models in detection accuracy. Additionally, our spoofing detection system has been successfully deployed in one of the largest global trading markets, further validating the practical applicability and performance of the proposed method.

*Corresponding author: Dawei Cheng.

WWW '25, April 28-May 2, 2025, Sydney, NSW, Australia
© 2025 Copyright held by the owner/author(s). Publication rights licensed to ACM.
ACM ISBN 979-8-4007-1274-6/25/04
https://doi.org/10.1145/3696410.3714518

CCS CONCEPTS

• **Applied computing → Secure online transactions**; • **Information systems → Data mining**.

KEYWORDS

Graph Representation Learning, Spoofing Detection, Dynamic Graph

ACM Reference Format:
Sheng Xiang, Yidong Jiang, Yunting Chen, Dawei Cheng, Guoping Zhao, and Changjun Jiang. 2025. Generative Dynamic Graph Representation Learning for Conspiracy Spoofing Detection. In *Proceedings of the ACM Web Conference 2025 (WWW '25), April 28-May 2, 2025, Sydney, NSW, Australia.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3696410.3714518

1 INTRODUCTION

Spoofing transaction [31], commonly known as "deceptive trading" in financial markets, represents a sophisticated form of market manipulation characterized by strategic order placement. As illustrated in Figure 1, traders intentionally submit non-executable orders (e.g., large-volume bids or asks in a small period) to fabricate artificial market signals about supply/demand equilibrium. This manipulative practice distorts asset pricing mechanisms for stocks, currencies, and derivatives by creating illusory liquidity or volatility patterns that enable illicit gains through subsequent counter-trades [4]. Functioning as a deceptive market intervention, spoofing behavior erodes market fairness and efficiency, contributing to systemic risks and substantial investor losses [3, 12]. The proliferation of algorithmic trading platforms has exponentially amplified both the occurrence rate and economic consequences of spoofing activities, compelling global regulators to implement stringent surveillance frameworks and punitive measures [13, 32]. Consequently, developing advanced detection paradigms for spoofing transactions has emerged as a paramount research priority across financial trading institutions and academic communities.

Early systems for detecting spoofing in financial transactions relied heavily on rule-based methods, where alerts were triggered by predefined behavioral thresholds [39, 41–43]. While effective in static scenarios, these methods lacked the adaptability needed for the evolving tactics of fraudsters. Machine learning and deep learning approaches have since emerged as more dynamic solutions, offering data-driven methodologies that continuously refine

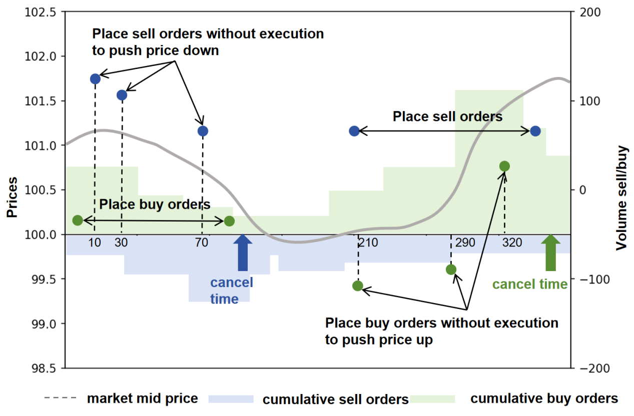

Figure 1: A typical example of spoofing transactions. Traders place deceptive sell (ask) or buy (bid) orders without execution to influence the market price by misleading other traders about the market demand or supply.

detection capabilities [2]. Techniques such as convolutional neural networks (CNNs) [17], recurrent neural networks (RNNs) [40], and attention mechanisms [11, 44] have been utilized to capture complex transaction patterns, marking significant progress in fraud detection [50]. However, these methods often fall short of capturing the intricate relationships and dynamics within interconnected transactions, ignoring the rich information from relations.

Graph-based methods, particularly Graph Neural Networks, have transformed fraud detection by effectively leveraging relational structures and connectivity patterns among transactions [10]. GNNs, such as CARE-GNN and SemiGNN, have demonstrated notable success in modeling dependencies across nodes and incorporating both relational and temporal information for improved detection performance [15, 38]. Recent advancements have introduced models like RTG-Trans, which integrates deep graph learning with temporal analysis to enhance spoofing detection accuracy [23]. Additionally, GPEGNN combines local and global community features to better identify and analyze conspiracy spoofing behaviors [24].

Despite the significant progress made by machine learning and graph-based approaches in detecting spoofing trading, several limitations remain that hinder their applicability to real-world financial trading data. First, the temporal relationships between transactions are highly irregular, with varying distances and dependencies that cannot be effectively modeled by trivial recurrent neural networks (RNNs) or standard transformer architectures. These methods rely on assumptions of regular or sequential data patterns, e.g., words and image patches, making them ill-suited for capturing the dynamic and erratic nature of trading behaviors. Second, the structural relationships in transaction networks exhibit inherent heterogeneity, as the interactions between transactions form non-homophily graph structures [48]. Conventional Graph Neural Networks (GNNs), such as Graph Convolutional Networks (GCNs) and Graph Attention Networks (GATs), are limited in their ability to distinguish between non-homophily graph representations, leading to reduced performance when analyzing the diverse and complex patterns required for conspiracy spoofing detection.

To address these challenges, we proposed Generative Dynamic Graph Model (GDGM). Specifically, our method comprises four steps: First, we convert raw trading data into structured time-stamped sequences for generative dynamic data encoding. During encoding, we introduce neural ordinary differential equation (Neural ODE) as the latent space generative method to obtain temporal representations from transaction sequences. Second, we introduce a Transaction Graph Pseudo-Label Generation mechanism to assign pseudo-labels for unlabeled nodes, thereby improving the model's capability to better identify patterns associated with conspiracy spoofing transactions. Finally, heterogeneous aggregation integrates different types of information across the graph, enabling the model to capture the diverse and non-homophily graph structures inherent in trading data. The experiments conducted on real-world spoofing detection datasets demonstrate that GDGM achieves superior detection performance when compared to existing state-of-the-art methods. Furthermore, the deployment of our system in one of the largest global trading markets, coupled with a case study, highlights the practical effectiveness of GDGM in addressing conspiracy spoofing detection.

The contributions of our work can be summarized as follows:

- We transform raw trading data into time-stamped sequences. Then, by utilizing generative neural ordinary differential equations, our method effectively captures temporal dependencies in transaction representations.
- To handle the non-homophily and diverse relationships in trading data, we generate pseudo-labels and design a heterogeneous aggregation mechanism. This mechanism enables our model to adaptively integrate relational information across different types of connections in trading data.
- Experimental results on real-world spoofing detection datasets demonstrate that GDGM outperforms state-of-the-art models in terms of detection accuracy. Furthermore, our method has been successfully deployed in one of the largest global trading markets, and a case study highlights its superior capability in uncovering spoofing patterns.

2 RELATED WORKS

This section reviews existing methods for spoofing detection and graph learning on financial transactions, highlighting their key contributions and main limitations.

2.1 Spoofing Detection

Since spoofing poses a major threat to the stability of financial markets, a variety of detection methodologies have been introduced in recent years to detect this issue. Early approaches mainly relied on statistical analysis of transaction behaviors to identify spoofing patterns [6]. Machine learning-based methods, such as Linear Regression [18], Decision Tree [16, 46] and Multi-Layer Perceptron [21] were broadly implemented in real-world spoofing detection systems. As spoofing tactics evolved, researchers started to adopt deep learning techniques, such as recurrent neural networks (RNNs) and attention-based neural networks, to capture more complex trading patterns[1, 50]. While these approaches offered advancements, they still had limitations in capturing the intricate interactions within transactions. More recently, graph neural networks (GNNs) have gained attention for their effectiveness in spoofing detection by leveraging inter-connected behavior within transaction graphs [15]. For instance, RTG-Trans combines deep graph learning with temporal analysis to enhance spoofing detection [23]. However, many GNN models focus primarily on local context, relying on adjacent nodes, which restricts their ability to identify complex, coordinated spoofing activities that require a global perspective on transaction relationships. Apart from graph-based techniques, other approaches have also been developed to tackle spoofing detection from different angles. Rule-based algorithms, for instance, set specific parameters to identify suspicious trading patterns. A good example of this can be found in studies analyzing transactions across stocks in indices like the Ibovespa, as shown in [30]. Moreover, researchers have also explored spoofing from a micro-structural perspective. They have introduced variables such as multilevel imbalances in price action and delved into the optimization strategies that potential spoofers might employ, as described in [45]. Despite these continuous advancements, existing methods still struggle when it comes to capturing the temporal relationships in trading behaviors. These relationships are inherently dynamic and do not fit well with traditional RNNs or transformer-based approaches.

2.2 Graph Learning on Financial Transaction

Graph-based machine learning has become instrumental in analyzing financial transactions and detecting fraudulent activities, drawing on its effectiveness in fields such as image processing, natural language processing, and knowledge graph construction[5, 9, 14, 20, 49]. Within financial applications, graph models have proven adept at addressing complex tasks like credit risk assessment [27] and financial fraud detection [26]. For example, vulnerable nodes on the guarantee loan network can be detected by graph models [7]. One significant challenge in credit risk assessment for small and medium-sized enterprises is the limited sample size. To address this, Wang et al. proposed an adaptive heterogeneous multi-view graph learning model, integrating multiple data perspectives to aggregate heterogeneous information for a comprehensive evaluation of credit risks[37]. Based on the idea of leveraging diverse data sources, SemiGNN utilizes both labeled and unlabeled data to capture dependencies across data views and neighboring nodes through a hierarchical attention mechanism [38]. Another major challenge is that fraudulent activities often involve sophisticated tactics such as disguising features and relationships within the data. To address this, Dou et al. introduced CARE-GNN, which incorporates a label-aware similarity measure and a reinforcement-learning-driven neighbor selection strategy, dynamically focusing on relevant nodes based on label information to improve detection accuracy[15]. In addition to relational data, temporal patterns play a crucial role in identifying anomalies [28]. GADBench offers a benchmark framework for sequence-based anomaly detection, capturing user behavior patterns over time, facilitating the detection of anomalies within temporal transaction data[33]. While these approaches address challenges in fraud detection broadly, they neglect the specific requirements of spoofing detection, particularly the need to distinguish between non-homophily graph structures. To date, no dedicated solutions have been proposed to model the heterogeneous graph relationships unique to spoofing scenarios.

Table 1: Notations used in this paper.

Symbol	Definition
n	the total number of nodes
m	the total number of edges
r	the number of relationships in graph $\tilde{G}$
d	the number of dimension
Z_i	the transformed representation using the i-th wavelet kernel
H	the comprehensive node representation aggregated from Z_0 to Z_C
p_i	the anomaly probability for node i
$X = \{x_1, x_2, ..., x_n\}$	the set of node features
$\hat{y}_i$	the pseudo-label for node i

3 METHODOLOGY

In this section, according to Figure 2, the proposed framework, GDGM, consists of four main components: (1) Generative Dynamic Data Encoding, (2) Pseudo-Labeled Graph Generation, (3) Heterogeneous Graph Attention, and (4) Classification. These components work together to capture generative dynamics data representations and heterogeneous graph relationships in financial transaction data, effectively modeling conspiracy spoofing behaviors.

3.1 Generative Dynamic Data Encoding

To effectively model the irregular temporal dynamics inherent in financial transaction data, especially in spoofing detection, we first explicitly leverage the timestamp of each transaction. Then, we employ a generative representation learning method, namely Neural Ordinary Differential Equation-based Recurrent Neural Network (ODERNN). This approach combines the continuous-time generative modeling capability of ODEs with the sequential representation power of GRU cells. The encoding process captures both temporal irregularities and sequential dependencies in the trading data.

3.1.1 Neural ODE Dynamics. We set the initial hidden state $h(0)$ as zero tensors. Then, the temporal evolution of the hidden state $h(t)$ in the ODE module is generated and governed by:

$$\frac{dh(t)}{dt} = f_\theta(h(t), t), \tag{1}$$

where f_θ is a neural network parameterized by θ that defines the dynamics of the system. The hidden state $h(t)$ at a future time t_1 is obtained by solving the initial value problem:

$$h(t_1) = h(t_0) + \int_{t_0}^{t_1} f_\theta(h(t), t)\, dt, \tag{2}$$

which is approximated numerically using the ODE solver:

$$h(t_1) = \text{ODEInt}(f_\theta, h(t_0), [t_0, t_1]), \tag{3}$$

where the function ODEInt computes the current hidden state, i.e., solution of the ODE over the time interval $[t_0, t_1]$.

3.1.2 GRU-based Sequential Updates. For each transaction sequence, we initialize the hidden state as h_0. Then we iteratively update it using the observations $\mathbf{x}_i$ at each time step with a gated recurrent unit (GRU) [8, 47]. The sequence update combines ODE dynamics

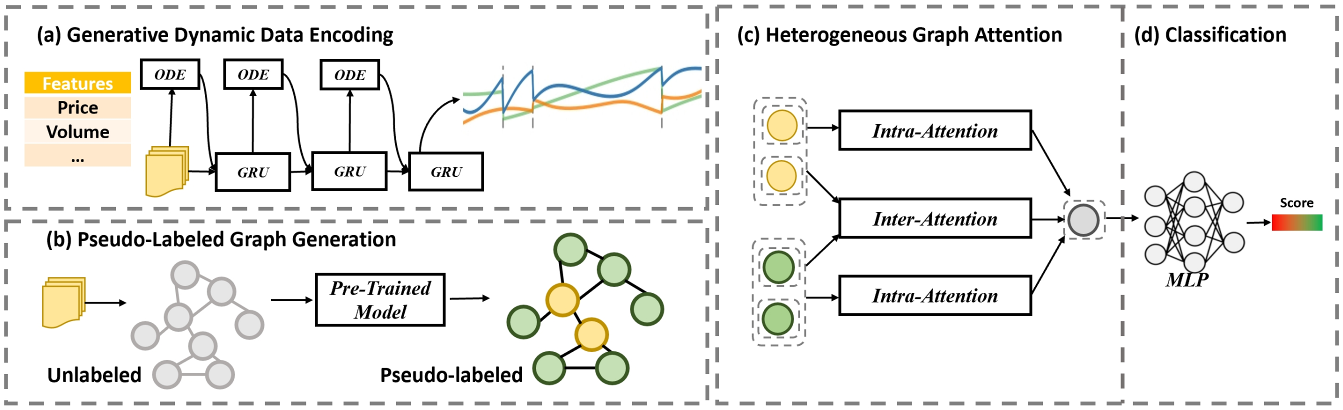

Figure 2: The proposed Generative Dynamic Graph Model (GDGM) architecture for Conspiracy Spoofing Detection. The first part is the historical transaction encoding of input time series data, which builds the embedding of irregular transaction series. The second part is the temporal graph attention. The third part is the heterogeneous graph attention layer, which aggregates the information for different types of neighbors. The fourth part is the classification layer, which is a multi-layer perception that gives the prediction of whether this transaction is spoofing.

with a GRU to handle the final observations, which is formulated as:

$$h(t_j) = \text{ODEInt}(f_\theta, h(t_{j-1}), [t_{j-1}, t_j]), \tag{4}$$

$$h_j = \text{GRUCell}(\mathbf{x}_j, h(t_j)), \tag{5}$$

where $h(t_j)$ is the hidden state after solving the ODE, and h_j is the updated state incorporating the observation $\mathbf{x}_j$ via the GRU cell.

3.1.3 Mini-batch Encoding. Given a batch of transaction sequences, the model processes each transaction independently. The final hidden states $h_{i,T}$ for all transactions are aggregated into a batch representation, which is formulated as follows:

$$H_{\text{batch}} = \{h_{1,T}, h_{2,T}, \ldots, h_{N,T}\}, \tag{6}$$

where $h_{i,T}$ represents the last hidden state of the i-th sequence, and N is the batch size. The last hidden state will be used as the input of the GNN-based classifier. Overall, the neural ODE is responsible for modeling dynamics and generating hidden variable candidates. Then the gated recurrent unit cells are leveraged to generate neural representations incorporated with temporal information.

3.2 Pseudo-Labeled Graph Generation

To effectively utilize unlabeled data, we propose a dynamic labeling mechanism integrated with the Pre-Trained Beta Wavelet Graph Neural Network (BWGNN) [34]. Leveraging its robust classification capabilities, BWGNN serves as the backbone for generating and refining pseudo labels, ensuring an effective and adaptive labeling process that enhances the model's robustness and generalization.

The process begins with the transformation of the node features X through a Multi-Layer Perceptron (MLP), which captures the intrinsic node characteristics. The raw node features X_{raw} were concatenated by the last hidden state H_{batch} of the encoding module, which is formulated as: $X = X_{\text{raw}}||H_{\text{batch}}$. These features then processed by a set of Beta wavelet kernels $\mathcal{W}_{i,C-i}$, each designed to extract spectral information at specific frequency scales, resulting in transformed representations Z_i:

$$Z_i = \mathcal{W}_{i,C-i}(\text{MLP}(X)), \quad i \in \{0, 1, \ldots, C\}. \tag{7}$$

The outputs from the wavelet kernels are aggregated into a comprehensive node representation H:

$$H = \text{AGG}([Z_0, Z_1, \ldots, Z_C]), \tag{8}$$

where AGG combines multi-scale spectral features. This representation is further processed by another MLP with a Sigmoid activation to compute anomaly probabilities p_i for each node:

$$p_i = \text{Sigmoid}(\text{MLP}(H)). \tag{9}$$

Based on these probabilities, pseudo labels are generated by applying a threshold z, which is formulated as follows:

$$\hat{y}_i = \begin{cases} 1 & \text{if } p_i > z \\ 0 & \text{otherwise.} \end{cases} \tag{10}$$

The transaction labeling mechanism provides a fully labeled graph for each batch of nodes to have graph data for heterogeneous graph attention during the training process.

3.3 Heterogeneous Graph Attention

In multi-relational graphs, redundant features can hinder the learning process by introducing noise. To address this challenge, we adopt a heterogeneous graph attention mechanism that consists of two main components: **intra-attention** and **inter-attention**. This approach efficiently handles the complexity of multi-relational transaction graphs by attending to information both within individual nodes and between different types of nodes, ensuring all meaningful relationships are captured.

3.3.1 Intra-Attention. The intra-attention mechanism is designed to consolidate features from neighboring nodes connected by the same relation type. This ensures that the model can choose important neighbors and capture shared characteristics within each type of node. The process includes the following steps:

1. *Attention-based Neighbor Aggregation* within each relation:

$$h'^{,l}_{v,r,*} = \sum_{u \in \mathcal{N}_{r,*}(v)} \alpha_{vu,r} h_u^{l-1} \tag{11}$$

The attention weights $\alpha_{vu,r}$ are calculated using the softmax function to normalize the importance of neighboring nodes:

$$\alpha_{vu,r} = \frac{\exp\left(\text{LeakyReLU}\left(a_r^\top \cdot \left[W_r^l h_v^{l-1} \parallel W_r^l h_u^{l-1}\right]\right)\right)}{\sum_{k \in \mathcal{N}_{r,*}(v)} \exp\left(\text{LeakyReLU}\left(a_r^\top \cdot \left[W_r^l h_v^{l-1} \parallel W_r^l h_k^{l-1}\right]\right)\right)} \tag{12}$$

Here, a_r is a learnable vector specific to the relation r, W_r^l is a relation-specific weight matrix, and $\parallel$ denotes concatenation.

2. *Feature Update* using attended neighbor features:

$$h^l_{v,r,*} = \text{ReLU}\left(W^l_{\text{intra},r} \cdot \left(h_v^{l-1} \parallel h'^{,l}_{v,r,*}\right)\right) \tag{13}$$

where $W^l_{\text{intra},r}$ is the learnable weight matrix for intra-attention at layer l, h_v^{l-1} represents the central node's features at the previous layer, $h'^{,l}_{v,r,*}$ is the aggregated feature from the neighbors under relation r, and ReLU introduces non-linearity to the updated features.

The intra-attention mechanism focuses on leveraging homogeneous node relationships among the same type of nodes to refine the neighbor node feature representations.

3.3.2 Inter-Attention. After introducing intra-attention, the model proceeds to inter-attention, where it collects information across different types of nodes. The process is as follows: (1) Features within each type are aggregated using a mean operation, ensuring each type's features are combined based on their relationships; (2) The attended features from different types of nodes are then passed through an attention mechanism to compute the attention weights, allowing the model to focus on the most informative relationships between the central node and different types of relationships.

The inter-attention process is performed as follows:

1. *Mean Aggregation* within each relation:

$$h'^{,l}_{v,r,g} = \text{Agg}_{\text{mean}}(h_u^{l-1}), \quad \forall u \in \mathcal{N}_{r,g}(v) \tag{14}$$

where $\mathcal{N}_{r,g}(v)$ represents the set of neighboring nodes of v connected by relation r and group g, Agg_{mean} denotes the function to compute the mean values of neighboring node features h_u^{l-1}, and h_u^{l-1} is the feature of neighbor u at layer $l-1$.

2. *Inter-Attention* between different relations:

$$h_v^l = \sum_r \sum_g \alpha^l_{r,g} h^l_{v,r,g} \tag{15}$$

where h_v^l is the updated feature of the central node v at layer l, $h^l_{v,r,g}$ is the feature aggregated from relation r and group g, and $\alpha^l_{r,g}$ is the attention weight for relation r and group g. The attention weights $\alpha^l_{r,g}$ are computed using the softmax function:

$$\alpha^l_{r,g} = \frac{\exp(\omega^l_{r,g})}{\sum_m \exp(\omega^l_{r,g})} \tag{16}$$

Here, $\omega^l_{r,g}$ is the unnormalized attention score for relation r and group g, and $\sum_m$ normalizes the scores across all relations and groups. The weight $\omega^l_{r,g}$ is determined by the interaction between the node's features and the attended features from each relation:

$$\omega^l_{r,g} = q^T \cdot \tanh(W^l_{\text{inter }1} h_v^{l-1} + W^l_{\text{inter }2} h^l_{v,r,g}) \tag{17}$$

In this equation, q is a learnable parameter vector that projects the interaction into a scalar, $W^l_{\text{inter }1}$ and $W^l_{\text{inter }2}$ are learnable weight matrices for the central node and the relation-specific features, respectively, and tanh introduces non-linearity to the interaction. The inter-attention mechanism ensures that information is aggregated across relations and groups, enabling the model to capture complex dependencies between different types of nodes and relations. After both intra-attention and inter-attention, the final node representation is obtained by concatenating the features from all relational representations with the central node's feature. This step ensures that the model incorporates information from both the central node and the different relationships:

$$h_v^{\text{final}} = h_v^l \parallel \text{concat}\left(\{h^l_{v,r,g}\}_{r,g}\right) \tag{18}$$

3.4 Optimization Objective

After the graph attention process, the embedding h_v^{final} obtained from the final layer is input into a Multilayer Perceptron (MLP), which produces a classification score p_v representing the probability of node v belonging to each category. The probabilities are computed using the softmax function, enabling the model to estimate the likelihood for each class. The training process minimizes the cross-entropy loss, which is defined as:

$$\mathcal{L} = -\sum_{v \in \mathcal{V}} \sum_{c=1}^{C} y_{v,c} \log p_{v,c}, \tag{19}$$

where $y_{v,c}$ represents the ground-truth label for node v in class c, $p_{v,c}$ is the predicted probability, and C is the total number of classes.

At the end of each training epoch, for spoofing detection, i.e., binary classification tasks, a threshold z is often used for decision-making, where the prediction is determined as follows:

$$\hat{y}_v = \begin{cases} 1 & \text{if } p_v > z, \\ 0 & \text{otherwise.} \end{cases} \tag{20}$$

This binary decision and prediction probability are used for model performance comparison experiments and building real-world spoofing detection systems.

4 EXPERIMENTS

In this section, we first show the detail of experimental settings. Then we report the performance comparison results on spoofing detection task. After that, we introduce the ablation study, parameter sensitivity, case studies, and implementations.

4.1 Experimental Settings

4.1.1 Datasets. We have created a new dataset called Spoofing Detection Dataset, consisting of 40,072 transaction records collected from our partners between January 5, 2018, and November 14, 2018. The dataset contains 49 feature dimensions, which can be categorized into four types of information: order-related details (e.g., order price, order balance, and order date), market and price data (e.g., today's trading volume and value), as well as order positions and profit or loss figures. The ground truth labels are based on cases reported by traders and verified by financial domain experts. Transactions are labeled as 1 if identified as fraudulent, and 0 otherwise.

During the data preprocessing stage, we remove irrelevant columns, such as trader ID and customer ID. Then, the features are normalized based on the train set statistics. To construct the graph, each transaction is treated as a node. Edges between nodes are established using a sliding window based on the transaction date.

4.1.2 Compared Methods. The compared methods can be classified into two parts: (1) Traditional learning-based methods; and (2) Advanced Spoofing Detection Methods. The traditional learning-based methods include Logistic Regression (LR) [18], Random Forest (RF) [46], Adaboost [25], Gradient Boosting Decision Tree (GBDT) [16], Hybrid Multi-layer Perceptron (HMLP) [21], Long Short Term Memory (LSTM) [19], and BiTransformer [35], all of which are implemented with their default hyperparameter settings.

For the advanced spoofing detection methods, we compare with EigenGCN [29], a graph convolutional network designed for transaction relationship learning; RetaGNN [22], a relational temporal attention-based graph neural network; GRU-DM [36], a Gated Recurrent Unit framework for spoofing detection using market indicators; RTG-Trans [23], a temporal gating method for detecting dynamic interactions within spoofing detection graphs; GPEGNN [24], a multi-layer graph attention-based method for local and global context learning; and GDGM, our proposed method, with hyperparameter settings detailed in section 4.4.

4.1.3 Evaluation Metrics. To evaluate the the performance of our model on spoofing detection, we leverage five widely recognized metrics to assess: Area Under the ROC Curve (AUC), Precision, Recall, F1 Score and Accuracy. The AUC measures evaluates the model's ability to differentiate between classes at various threshold settings. Precision (P) measures the proportion of true positive predictions among all positive predictions. It is computed as: $P = \frac{N_{TP}}{N_{TP}+N_{FP}}$ where N_{TP} represents the number of true positives, and N_{FP} represents the number of false positives. Recall (R) quantifies the model's ability to identify all true positive instances. It is calculated as: $R = \frac{N_{TP}}{N_{TP}+N_{FN}}$ where N_{FN} denotes the number of false negatives. The F1 Score provides a harmonic mean of Precision and Recall, balancing the trade-off between these two metrics. It is expressed as: $F1 = \frac{2\times P\times R}{P+R}$ Accuracy evaluates the overall correctness of the model by considering both positive and negative classes. It is defined as: $Accuracy = \frac{N_{TP}+N_{TN}}{N_{TP}+N_{TN}+N_{FP}+N_{FN}}$ where N_{TN} denotes the number of true negatives.

4.2 Performance Comparison

In this section, we evaluate the performance of our proposed model against various baseline methods on the spoofing detection task. Table 2 provides a comprehensive comparison using metrics such as Area Under the ROC Curve (AUC), Accuracy, F1 Score, Precision, and Recall. Each model was tested over multiple iterations, and the average results are reported. The first six rows of Table 2 present traditional machine learning methods, including Logistic Regression (LR), Random Forest (RF), and gradient boosting models (Adaboost and GBDT). Among these, RF achieves the highest AUC (0.8985), significantly outperforming other traditional methods. However,

Table 2: Experimental results for different machine learning methods on spoofing detection task.

Method	AUC	Accuracy	F1 Score	Precision	Recall
LR	0.7828	0.8119	0.5320	0.7077	0.4262
RF	0.8985	0.8578	0.7066	0.7322	0.6826
Adaboost	0.8716	0.8538	0.6564	0.7994	0.5568
GBDT	0.8911	0.8615	0.6720	0.8284	0.5653
HMLP	0.7587	0.7893	0.5894	0.7785	0.5852
LSTM	0.7759	0.8393	0.7483	0.8149	0.7058
BiTransformer	0.8957	0.8529	0.7504	0.8109	0.7205
EigenGCN	0.8904	0.8491	0.7405	0.8059	0.7102
RetaGNN	0.8935	0.8552	0.7705	0.8201	0.7409
GRU-DM	0.8805	0.8483	0.7601	0.8152	0.7308
RTG-Trans	0.8998	0.8602	0.7807	0.8255	0.7507
GPEGNN	0.8989	0.8578	0.7852	0.8309	0.7558
Ours	**0.9029**	**0.8701**	**0.8002**	**0.8508**	**0.7702**

these baseline models, while effective, demonstrate limitations in capturing complex relational patterns, leading to suboptimal recall values, particularly for LR (0.4262) and Adaboost (0.5568). The middle section of Table 2 introduces more advanced graph-based and sequence-based models, such as BiTransformer, EigenGCN, and RTG-Trans. These methods show marked improvements over traditional baselines, with BiTransformer achieving an AUC of 0.8957 and EigenGCN delivering competitive precision (0.8059). Notably, RTG-Trans outperforms most other baselines in recall (0.7507), indicating its effectiveness in detecting spoofing cases with minimal false negatives. However, even these models exhibit limitations in balancing all performance metrics, as seen in slightly lower F1 scores for some methods. Our proposed model, listed as "Ours" in Table 2, achieves the best overall performance across all metrics. It records the highest AUC (0.9029), Accuracy (0.8701), and F1 Score (0.8002), along with superior Precision (0.8508) and Recall (0.7702). These results demonstrate the effectiveness of our approach in leveraging intricate graph-based relationships and temporal dependencies for spoofing detection. Compared to the best-performing baseline (RTG-Trans), our model achieves a 0.31% improvement in AUC and a 2.4% increase in Accuracy, showcasing its robustness and reliability in handling complex spoofing scenarios. This significant performance boost highlights the capability of our model to capture nuanced patterns and relationships in graph-structured data, making it particularly well-suited for detecting spoofing and other fraudulent activities in real-world applications.

4.3 Implementation and Online Deployment

In addition to the comparison on historical data, testing the accuracy on future transactions is more important to real-world applications. As to real-world deployment of spoofing detection, transactions are processed through a distributed message queue for real-time evaluation. Initially, they are checked against blacklist entries and fraud detection rules (in-process detection). Transactions that hit any blacklist or fraud rule are blocked immediately. If no matches are found, user and symbol features are then extracted and passed to an online predictive model (GDGM, in this case) as part of the post-process detection. Throughout this process, historical transaction data flagged by the detection systems is stored in an in-memory

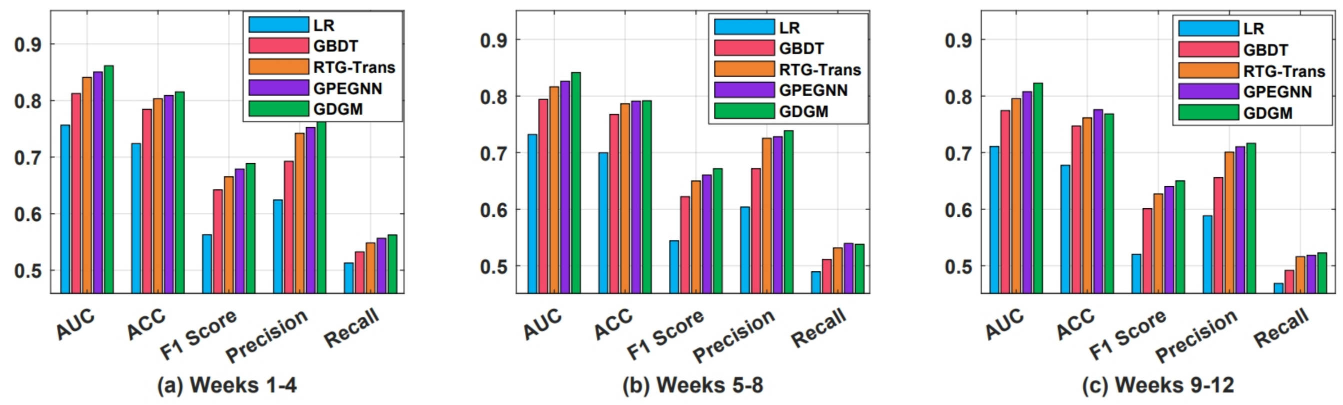

Figure 3: The experimental results of spoofing detection methods under queue-based study. During the experiment, pre-trained models were employed to detect spoofing transactions over four weeks. At the end of each four-week interval, newly observed cases were merged with the historically labeled database, and the models were retrained to improve their performance.

database to support large-scale detection. High-risk transactions are escalated to domain experts for verification, with feedback on these cases stored in the historical database. The predictive model is periodically retrained in batches based on the latest expert feedback, allowing it to learn from recent spoofing patterns. Newly identified fraud cases are also incorporated to refine the blacklist and fraud detection rules, ensuring both detection layers remain adaptive and robust. To evaluate the performance of our model under real-world conditions, we tested 105 confirmed spoofing cases using data collected between July and September through a 12-weeks queue-based study. We compared our proposed GDGM model with two widely-used baselines, Logistic Regression (LR) and Gradient Boosting Decision Tree (GBDT), as well as two state-of-the-art (SOTA) methods specifically designed for spoofing detection, RTG-Trans and GPEGNN. The comparison results of the online experiments are summarized in Figure 3. As shown in Figure 3, in the first 4 weeks, GDGM consistently outperforms all baseline and state-of-the-art methods across all evaluation metrics. Notably, GDGM achieves the highest AUC (0.8614), ACC (0.8152), F1 score (0.6887), precision (0.7624), and recall (0.5623). These results highlight the ability of our model to effectively identify spoofing cases in real-world settings, even when dealing with noisy or irregular data. Similar conclusions can be derived from the experimental results of the next 8 weeks. The superior performance of GDGM can be attributed to its ability to model irregular trading behaviors and heterogeneous relationships among transactions, as well as its integration of pseudo-labeling and graph aggregation techniques. By leveraging these advanced capabilities, GDGM captures subtle patterns indicative of conspiracy spoofing that are often missed by traditional baselines and even existing SOTA methods. Furthermore, the deployment of GDGM in a real-time production environment demonstrates its practicality and scalability. The integration with a distributed message queue ensures low-latency processing, while the adaptive learning framework enables the model to evolve based on the latest spoofing trends. These features make GDGM a robust and reliable solution for combating financial fraud in dynamic, high-stakes trading markets.

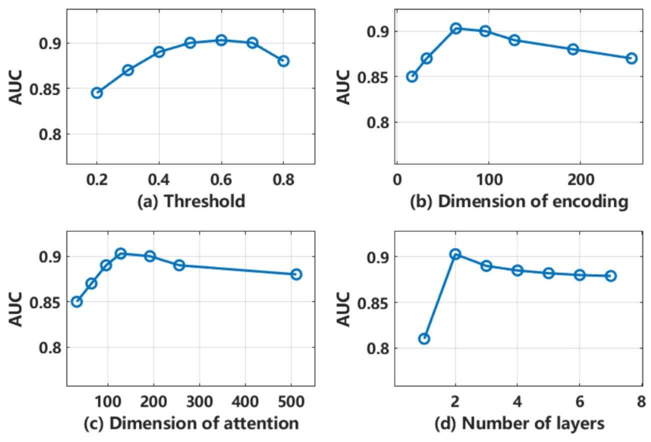

Figure 4: AUC of our method, in terms of threshold z, dimension of encoding output h, dimension of the attention vector q, and number of heterogeneous aggregation layers.

4.4 Parameter Sensitivity

One of the important research questions is to analyze the sensitivity of our model to its hyperparameters. In this experiment, we examine key parameters such as the threshold for pseudo-labeling, the dimension of encoding output, the dimension of the attention vector, and the number of graph aggregation layers. Figure 4 illustrates the impact of adjusting these parameters on the model's performance on the spoofing detection dataset.

Specifically, Figure 4(a) shows the effect of varying the threshold for pseudo-labeling on model performance. As the threshold increases from 0.2 to 0.8, the AUC metric steadily improves, peaking when the threshold is set to 0.6. Beyond this point, performance begins to decline, highlighting the importance of selecting an appropriate threshold to balance noise suppression and label informativeness. Figure 4(b) evaluates the sensitivity to the dimension of encoding output. The model's performance increases significantly

Table 3: Ablation study results, showing the impact of different model components on spoofing detection task.

Model Variant	AUC	F1 Score	Precision
GDGM	**0.9029**	**0.8002**	**0.8508**
w/o Pseudo Label	0.8998	0.7807	0.8255
w/o Neural ODE	0.8965	0.7513	0.8126
w/o Hete. GNN	0.8901	0.7413	0.8041

as the dimension grows from 16 to 64, with the AUC peaking for a dimension of 64. However, further increases in the dimension lead to slight performance degradation, likely due to overfitting or an inability to efficiently utilize the additional capacity. The impact of the dimension of the attention vector is shown in Figure 4(c). The AUC steadily improves as the dimension increases from 32 to 128, reaching a peak. Beyond this point, larger attention vectors result in marginally lower performance, indicating that 128 is an optimal choice for balancing expressiveness and computational efficiency. Figure 4(d) examines the effect of the number of graph aggregation layers. The performance improves as the number of layers increases from 1 to 2, achieving the highest AUC at 2 layers. More layers lead to a decline in performance, suggesting that excessive stacking of graph aggregation layers may introduce noise or redundancy. Based on these findings, we select the best parameter settings as follows: a threshold of 0.6 for pseudo-labeling, an encoding output dimension of 64, an attention vector dimension of 128, and 2 graph aggregation layers. These configurations enable our model to achieve optimal performance on the spoofing detection task.

4.5 Ablation Study

To evaluate the contribution of each component in our model, we conducted an ablation study by removing specific features from the model: **(1) Without Pseudo Label:** This variant directly uses the training set labels as input to the heterogeneous graph neural network instead of pseudo labels, which reduces the model's ability to leverage inferred label consistency. **(2) Without Neural ODE:** This configuration replaces our ODE-RNN encoder with a standard RNN encoder, limiting the model's capacity to capture continuous temporal dynamics effectively. **(3) Without Heterogeneous GNN:** In this variant of the model, instead of employing the heterogeneous graph neural network, we chose to use a homogeneous graph neural network, specifically the Graph Attention Network (GAT), for the classification task. When we switch to a homogeneous GNN like GAT, the model's expressiveness in accurately representing and modeling these diverse node and edge types is reduced.

The experimental results obtained from the spoofing detection dataset, which are clearly presented in Table 3, demonstrate the significance of each of these components. The full model, which incorporates all the components including the pseudo labels, neural ODEs, and heterogeneous GNNs, achieves the best performance across all the metrics that were considered for evaluation. This outstanding performance of the full model serves to highlight the importance of integrating these specific components together. Notably, when we replace the pseudo label mechanism, we observe a moderate decline in the performance of the model. Moreover, when either the neural ODE encoder or the heterogeneous GNN is removed from the model, the effectiveness of the model is further degraded to an even greater extent. These experimental outcomes clearly underline the critical role that each component plays in enabling the model to achieve robust spoofing detection capabilities.

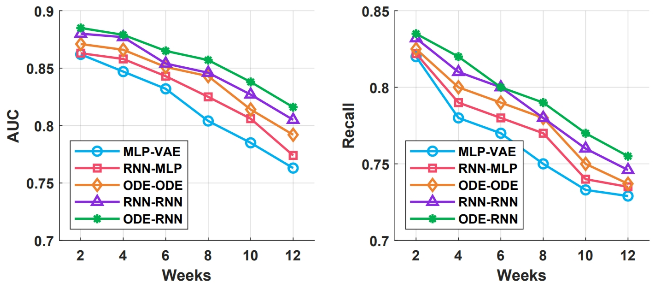

Figure 5: Performance comparison for models with different generative data encoding modules. We fix the trained model and make predictions for the next 12 weeks.

In addition to the ablation components that were mentioned above, we also extended our investigation to explore the performance of various encoding modules that are commonly used in the context of spoofing detection. The experimental comparison of these different encoding modules is visually depicted in Figure 5. As can be clearly observed from the figure, the ODE-RNN encoding module outperforms all the other models that were included in the comparison. Following closely behind in terms of performance is the RNN-RNN model, and then the Neural ODE. On the other hand, the MLP-VAE and RNN-MLP models display relatively weaker performance. This is primarily due to their inherent inability to fully capture both the temporal and sequential dynamics of the data. These dynamics are essential for accurately understanding and processing the information within the spoofing detection dataset, and the failure to effectively capture them results in the observed suboptimal performance of these particular encoding modules.

5 CONCLUSION

Spoofing detection in financial trading, particularly intricate behaviors like conspiracy spoofing, is a critical yet challenging task. Conventional machine learning methods often neglect the interconnected and heterogeneous nature of trading data, while existing graph-based techniques struggle to capture the irregularities inherent in real-world trading behaviors. Therefore, we proposed the Generative Dynamic Graph Model (GDGM), a novel framework designed to model both temporal trading behaviors and dynamic, heterogeneous relationships among nodes. Our approach leverages neural ordinary differential equations and gated recurrent units to represent irregular trading patterns. Then, we employ the pre-trained model for pseudo-labeling and heterogeneous attention-based aggregation mechanisms to effectively capture conspiratorial spoofing signals. The results of extensive experiments demonstrate that GDGM outperforms state-of-the-art models in detecting spoofing behaviors, highlighting its effectiveness and robustness. Furthermore, the successful deployment of our system in one of the largest global trading markets underscores its practical applicability, validating its exceptional performance in real-world scenarios.

ACKNOWLEDGMENTS

The work is supported by the National Natural Science Foundation of China (62472317), the Fundamental Research Funds for the Central Universities and the Shanghai Science and Technology Innovation Action Plan Project (Grant no. 22YS1400600 and 24692118300).

REFERENCES

[1] Shefali Arora and Mahinder Pal Singh Bhatia. 2019. Fingerprint Spoofing Detection to Improve Customer Security in Mobile Financial Applications Using Deep Learning. *Arabian Journal for Science and Engineering* 45 (2019), 2847 – 2863. https://api.semanticscholar.org/CorpusID:210749350

[2] Siddhartha Bhattacharyya, Sanjeev Jha, Kurian Tharakunnel, and J Christopher Westland. 2011. Data mining for credit card fraud: A comparative study. *Decision Support Systems* 50, 3 (2011), 602–613.

[3] Richard J. Bolton and David J. Hand. 2002. Statistical fraud detection: A review. *Quality Engineering* 49 (2002), 313–314. https://api.semanticscholar.org/CorpusID:15208227

[4] Zinelabidine Boulkenafet, Jukka Komulainen, and Abdenour Hadid. 2016. Face Spoofing Detection Using Colour Texture Analysis. *IEEE Transactions on Information Forensics and Security* 11, 8 (2016), 1818–1830. doi:10.1109/TIFS.2016.2555286

[5] Longbing Cao. 2022. AI in Finance: Challenges, Techniques, and Opportunities. *ACM Comput. Surv.* 55, 3, Article 64 (Feb. 2022), 38 pages. doi:10.1145/3502289

[6] Yi Cao, Yuhua Li, Sonya Coleman, Ammar Belatreche, and Thomas Martin McGinnity. 2015. Adaptive Hidden Markov Model With Anomaly States for Price Manipulation Detection. *IEEE Transactions on Neural Networks and Learning Systems* 26, 2 (2015), 318–330. doi:10.1109/TNNLS.2014.2315042

[7] Dawei Cheng, Chen Chen, Xiaoyang Wang, and Sheng Xiang. 2021. Efficient top-k vulnerable nodes detection in uncertain graphs. *IEEE Transactions on Knowledge and Data Engineering* 35, 2 (2021), 1460–1472.

[8] Dawei Cheng, Ye Liu, Zhibin Niu, and Liqing Zhang. 2018. Modeling similarities among multi-dimensional financial time series. *IEEE Access* 6 (2018), 43404–43413.

[9] Dawei Cheng, Zhibin Niu, Jie Li, and Changjun Jiang. 2022. Regulating systemic crises: Stemming the contagion risk in networked-loans through deep graph learning. *IEEE Transactions on Knowledge and Data Engineering* 35, 6 (2022), 6278–6289.

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

[11] Dawei Cheng, Sheng Xiang, Chencheng Shang, Yiyi Zhang, Fangzhou Yang, and Liqing Zhang. 2020. Spatio-temporal attention-based neural network for credit card fraud detection. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 34. 362–369.

[12] Dawei Cheng, Fangzhou Yang, Sheng Xiang, and Jin Liu. 2022. Financial time series forecasting with multi-modality graph neural network. *Pattern Recognition* 121 (2022), 108218.

[13] Dawei Cheng, Yujia Ye, Sheng Xiang, Zhenwei Ma, Ying Zhang, and Changjun Jiang. 2023. Anti-money laundering by group-aware deep graph learning. *IEEE Transactions on Knowledge and Data Engineering* 35, 12 (2023), 12444–12457.

[14] Dawei Cheng, Yao Zou, Sheng Xiang, and Changjun Jiang. 2025. Graph neural networks for financial fraud detection: a review. *Frontiers of Computer Science* 19, 9 (2025), 1–15.

[15] Yingtong Dou, Zhiwei Liu, Li Sun, Yutong Deng, Hao Peng, and Philip S. Yu. 2020. Enhancing Graph Neural Network-based Fraud Detectors against Camouflaged Fraudsters. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management* (Virtual Event, Ireland) *(CIKM '20)*. Association for Computing Machinery, New York, NY, USA, 315–324. doi:10.1145/3340531.3411903

[16] Jerome H. Friedman. 2001. Greedy function approximation: A gradient boosting machine. *Annals of Statistics* 29 (2001), 1189–1232. https://api.semanticscholar.org/CorpusID:39450643

[17] Kang Fu, Dawei Cheng, Yi Tu, and Liqing Zhang. 2016. Credit card fraud detection using convolutional neural networks. In *Neural Information Processing: 23rd International Conference, ICONIP 2016, Kyoto, Japan, October 16–21, 2016, Proceedings, Part III 23*. Springer, 483–490.

[18] Jibing Gong and Shengtao Sun. 2009. A New Approach of Stock Price Prediction Based on Logistic Regression Model. In *2009 International Conference on New Trends in Information and Service Science*. 1366–1371. doi:10.1109/NISS.2009.267

[19] Alex Graves, Santiago Fernández, and Jürgen Schmidhuber. 2005. Bidirectional LSTM Networks for Improved Phoneme Classification and Recognition. In *Artificial Neural Networks: Formal Models and Their Applications – ICANN 2005*, Włodzisław Duch, Janusz Kacprzyk, Erkki Oja, and Sławomir Zadrożny (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 799–804.

[20] William L. Hamilton, Rex Ying, and Jure Leskovec. 2017. Inductive representation learning on large graphs. In *Proceedings of the 31st International Conference on Neural Information Processing Systems* (Long Beach, California, USA) *(NIPS'17)*. Curran Associates Inc., Red Hook, NY, USA, 1025–1035.

[21] Ali Asghar Heidari, Hossam Faris, Ibrahim Aljarah, and Seyedali Mirjalili. 2019. An efficient hybrid multilayer perceptron neural network with grasshopper optimization. *Soft Computing* 23, 17 (2019), 7941–7958. doi:10.1007/s00500-018-3424-2

[22] Cheng Hsu and Cheng-Te Li. 2021. RetaGNN: Relational Temporal Attentive Graph Neural Networks for Holistic Sequential Recommendation. *CoRR* abs/2101.12457 (2021). arXiv:2101.12457 https://arxiv.org/abs/2101.12457

[23] Le Kang, Tai-Jiang Mu, and Xiaodong Ning. 2023. Conspiracy Spoofing Orders Detection with Transformer-Based Deep Graph Learning. In *Advanced Data Mining and Applications*, Xiaochun Yang, Heru Suhartanto, Guoren Wang, Bin Wang, Jing Jiang, Bing Li, Huaijie Zhu, and Ningning Cui (Eds.). Springer Nature Switzerland, Cham, 489–503.

[24] Le Kang, Tai-Jiang Mu, and XiaoDong Ning. 2024. Spoofing Transaction Detection with Group Perceptual Enhanced Graph Neural Network. In *Machine Learning and Knowledge Discovery in Databases. Applied Data Science Track*, Albert Bifet, Tomas Krilavičius, Ioanna Miliou, and Slawomir Nowaczyk (Eds.). Springer Nature Switzerland, Cham, 106–122.

[25] Rihab Salah Khairy, Ameer Saleh Hussein, and Haider Th. Salim Alrikabi. 2021. The Detection of Counterfeit Banknotes Using Ensemble Learning Techniques of AdaBoost and Voting. *International Journal of Intelligent Engineering and Systems* (2021). https://api.semanticscholar.org/CorpusID:233843556

[26] Enxia Li, Jin Ouyang, Sheng Xiang, Lu Qin, and Ling Chen. 2024. Relation-aware heterogeneous graph neural network for fraud detection. In *Asia-Pacific Web (APWeb) and Web-Age Information Management (WAIM) Joint International Conference on Web and Big Data*. Springer, 240–255.

[27] Charles Z Liu, Sheng Xiang, Dawei Cheng, Junyi Liu, Ying Zhang, and Lu Qin. 2023. Transformed Graph Attention for Credit Rating. In *2023 IEEE 18th Conference on Industrial Electronics and Applications (ICIEA)*. IEEE, 1011–1016.

[28] Jiacheng Ma, Sheng Xiang, Qiang Li, Liangyu Yuan, Dawei Cheng, and Changjun Jiang. 2024. Parallel Graph Learning with Temporal Stamp Encoding for Fraudulent Transactions Detections. *IEEE Transactions on Big Data* (2024).

[29] Yao Ma, Suhang Wang, Charu C. Aggarwal, and Jiliang Tang. 2019. Graph Convolutional Networks with EigenPooling. *CoRR* abs/1904.13107 (2019). arXiv:1904.13107 http://arxiv.org/abs/1904.13107

[30] Luís F. Mendonça and Alan de Genaro. 2020. Detection and analysis of occurrences of spoofing in the Brazilian capital market. *Journal of Financial Regulation and Compliance* 28 (2020), 369–408. https://api.semanticscholar.org/CorpusID:216325996

[31] Mark L. Psiaki and Todd E. Humphreys. 2016. GNSS Spoofing and Detection. *Proc. IEEE* 104, 6 (2016), 1258–1270. doi:10.1109/JPROC.2016.2526658

[32] Praveen Kumar Sadineni. 2020. Detection of Fraudulent Transactions in Credit Card using Machine Learning Algorithms. In *2020 Fourth International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC)*. 659–660. doi:10.1109/I-SMAC49090.2020.9243545

[33] Jianheng Tang, Fengrui Hua, Ziqi Gao, Peilin Zhao, and Jia Li. 2023. GADBench: Revisiting and Benchmarking Supervised Graph Anomaly Detection. In *Advances in Neural Information Processing Systems*, Vol. 36. 29628–29653. https://proceedings.neurips.cc/paper_files/paper/2023/file/5eaafd67434a4cfb1cf829722c65f184-Paper-Datasets_and_Benchmarks.pdf

[34] Jianheng Tang, Jiajin Li, Zi-Chao Gao, and Jia Li. 2022. Rethinking Graph Neural Networks for Anomaly Detection. In *International Conference on Machine Learning*. https://api.semanticscholar.org/CorpusID:249209972

[35] Murat Tezgider, Beytullah Yildiz, and Galip Aydin. 2021. Text classification using improved bidirectional transformer. *Concurrency and Computation: Practice and Experience* (2021). doi:10.1002/CPE.6486

[36] Jean-Noël Tuccella, Philip Nadler, and Ovidiu Şerban. 2021. Protecting Retail Investors from Order Book Spoofing using a GRU-based Detection Model. arXiv:2110.03687 [q-fin.ST] https://arxiv.org/abs/2110.03687

[37] Cong Wang, Fangyue Yu, Zaixu Zhang, Jian Zhang, and Baogui Xin. 2021. Multiview Graph Learning for Small- and Medium-Sized Enterprises' Credit Risk Assessment in Supply Chain Finance. *Complex.* 2021 (Jan. 2021), 13 pages. doi:10.1155/2021/6670873

[38] Daixin Wang, Yuan Qi, Jianbin Lin, Peng Cui, Quanhui Jia, Zhen Wang, Yanming Fang, Quan Yu, Jun Zhou, and Shuang Yang. 2019. A Semi-Supervised Graph Attentive Network for Financial Fraud Detection. *2019 IEEE International Conference on Data Mining (ICDM)* (2019), 598–607.

[39] Christopher Whitrow, David J. Hand, Piotr Juszczak, David John Weston, and Niall M. Adams. 2009. Transaction aggregation as a strategy for credit card fraud detection. *Data Mining and Knowledge Discovery* 18 (2009), 30–55.

[40] Sheng Xiang, Dawei Cheng, Chencheng Shang, Ying Zhang, and Yuqi Liang. 2022. Temporal and heterogeneous graph neural network for financial time series prediction. In *Proceedings of the 31st ACM international conference on information & knowledge management*. 3584–3593.

[41] Sheng Xiang, Dawei Cheng, Jianfu Zhang, Zhenwei Ma, Xiaoyang Wang, and Ying Zhang. 2022. Efficient learning-based community-preserving graph generation. In *2022 IEEE 38th International Conference on Data Engineering (ICDE)*. IEEE, 1982–1994.

[42] Sheng Xiang, Dong Wen, Dawei Cheng, Ying Zhang, Lu Qin, Zhengping Qian, and Xuemin Lin. 2022. General graph generators: experiments, analyses, and improvements. *The VLDB Journal* (2022), 1–29.

[43] Sheng Xiang, Chenhao Xu, Dawei Cheng, and Ying Zhang. 2025. Scalable Learning-Based Community-Preserving Graph Generation. *IEEE Transactions on Big Data* 01 (2025), 1–14.

[44] Sheng Xiang, Mingzhi Zhu, Dawei Cheng, Enxia Li, Ruihui Zhao, Yi Ouyang, Ling Chen, and Yefeng Zheng. 2023. Semi-supervised Credit Card Fraud Detection via Attribute-driven Graph Representation. In *AAAI*.

[45] Lan Ling Xuan Tao, Andrew Day and Samuel Drapeau. 2022. On detecting spoofing strategies in high-frequency trading. *Quantitative Finance* 22, 8 (2022), 1405–1425. doi:10.1080/14697688.2022.2059390

[46] Wengang Zhang, Chongzhi Wu, Haiyi Zhong, Yongqin Li, and Lin Wang. 2021. Prediction of undrained shear strength using extreme gradient boosting and random forest based on Bayesian optimization. *Geoscience frontiers* 12 (2021), 469–477. https://api.semanticscholar.org/CorpusID:216513042

[47] Peng Zhu, Yuante Li, Yifan Hu, Sheng Xiang, Qinyuan Liu, Dawei Cheng, and Yuqi Liang. 2024. MCI-GRU: Stock Prediction Model Based on Multi-Head Cross-Attention and Improved GRU. *arXiv preprint arXiv:2410.20679* (2024).

[48] Yao Zou and Dawei Cheng. 2024. Effective High-order Graph Representation Learning for Credit Card Fraud Detection. In *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence, IJCAI-24*, Vol. 8. 7581–7589.

[49] Yao Zou, Sheng Xiang, Qijun Miao, Dawei Cheng, and Changjun Jiang. 2024. Subgraph patterns enhanced graph neural network for fraud detection. In *International Conference on Database Systems for Advanced Applications*. Springer, 375–384.

[50] Sebastian Jaimungal Álvaro Cartea and Yixuan Wang. 2020. Spoofing and Price Manipulation in Order-Driven Markets. *Applied Mathematical Finance* 27, 1-2 (2020), 67–98. doi:10.1080/1350486X.2020.1726783 arXiv:https://doi.org/10.1080/1350486X.2020.1726783