BIG DATA MINING AND ANALYTICS
ISSN 2096-0654 23/25 pp920–941
DOI: 10.26599/BDMA.2024.9020013
Volume 7, Number 3, September 2024

Multi-Relational Graph Representation Learning for Financial Statement Fraud Detection

Chenxu Wang*, Mengqin Wang, Xiaoguang Wang, Luyue Zhang, and Yi Long

Abstract: Financial statement fraud refers to malicious manipulations of financial data in listed companies' annual statements. Traditional machine learning approaches focus on individual companies, overlooking the interactive relationships among companies that are crucial for identifying fraud patterns. Moreover, fraud detection is a typical imbalanced binary classification task with normal samples outnumbering fraud ones. In this paper, we propose a multi-relational graph convolutional network, named FraudGCN, for detecting financial statement fraud. A multi-relational graph is constructed to integrate industrial, supply chain, and accounting-sharing relationships, effectively encapsulating the multidimensional and complex interactions among companies. We then develop a multi-relational graph convolutional network to aggregate information within each relationship and employ an attention mechanism to fuse information across multiple relationships. The attention mechanism enables the model to distinguish the importance of different relationships, thereby aggregating more useful information from key relationships. To alleviate the class imbalance problem, we present a diffusion-based under-sampling strategy that strategically selects key nodes globally for model training. We also employ focal loss to assign greater weights to harder-to-classify minority samples. We build a real-world dataset from the annual financial statement of listed companies in China. The experimental results show that FraudGCN achieves an improvement of 3.15% in Macro-recall, 3.36% in Macro-F1, and 3.86% in GMean compared to the second-best method. The dataset and codes are publicly available at: https://github.com/XNetLab/MRG-for-Finance.

Key words: financial statement fraud; class imbalance; Graph Neural Networks (GNN); multi-relational graphs

1 Introduction

Financial statement fraud refers to the intentional manipulation or misrepresentation of financial data in annual reports to conceal the true financial situation. This malpractice manifests in various forms to commit financial statement fraud, including overstatement of revenue, cost manipulation, inflation of reported financial outcomes, and other various forms of fraudulent ways[1]. Such deceptive practices compromise the functionality of capital markets[2], damage corporate reputation and stakeholder

• Chenxu Wang is with School of Software Engineering, and also with MoE Key Lab of Intelligent Networks and Network Security, Xi'an Jiaotong University, Xi'an 710049, China. E-mail: cxwang@mail.xjtu.edu.cn.

• Mengqin Wang, Xiaoguang Wang, and Luyue Zhang are with School of Software Engineering, Xi'an Jiaotong University, Xi'an 710049, China. E-mail: wmengqin@stu.xjtu.edu.cn; wangxg@stu.xjtu.edu.cn; zhangluyue@stu.xjtu.edu.cn.

• Yi Long is with Shenzhen Finance Institute, The Chinese University of Hong Kong, Shenzhen (CUHK-Shenzhen), Shenzhen 518026, China. E-mail: longyi@datago.com.hk.

∗ To whom correspondence should be addressed.

Manuscript received: 2024-01-02; revised: 2024-03-02; accepted: 2024-03-04

interests[3], and adversely impact the entire economic ecosystem[4]. Recently, there has been an increasing incidence of financial statement fraud among listed companies, resulting in significant adverse consequences. For example, in 2020, Luckin Coffee Co. Ltd. engaged in financial statement fraud, whose fraudulent activities triggered widespread repercussions, leading to substantial job losses. Eventually, Luckin Coffee Co. Ltd. was delisted from the list of listed companies. Thus, the prompt and accurate detection of financial statement fraud is imperative. Furthermore, such fraud in a listed company can adversely affect its stock value and may result in severe legal ramifications[5–7].

Annually, numerous companies engage in fraudulent activities. The Securities Regulatory Commission performs random audits on selected listed companies' financial statements and discloses the outcomes publicly. However, it suffers inefficiencies in manually reviewing financial reports. The vast number of companies necessitates extensive labor and resources for comprehensive spot checks. Therefore, the development of an efficient and precise algorithm to detect fraud risks in publicly listed companies is essential. Such a tool can minimize investor losses by facilitating the pre-auditing of financial reports, thereby enhancing audit efficiency and accuracy. However, some fraudulent companies remain undetected, either not chosen for audits or evading detection through sophisticated tactics. Hence, an effective and accurate algorithm is desired to identify financial statement fraud, overcoming the limitations of the Securities Regulatory Commission's random audit approach.

Previous research relies on data-driven methods for detecting fraud[8]. Data of various types, informed by domain expertise, are extracted and input into machine learning classifiers[9]. These data fall into two primary categories: structured and unstructured. Structured data, usually stored in tabular or relational databases, possess a defined data model and format, exemplified by raw financial statements data[10] and financial ratio index[11]. In contrast, unstructured data, often found as text, images, and audio[12], lack such defined models. Machine learning classifiers and neural networks[13, 14] analyze these diverse financial data to pinpoint potential fraud in financial statements, aiding regulatory agencies, and business managers in identifying risks.

Nevertheless, current data-driven methods concentrate on the internal characteristics of individual companies, overlooking the intricate interconnections among companies and the potential diffusion of fraudulent behaviors across them. Prior studies have demonstrated that the supply chain plays a crucial role in investigating the propagation of financial risks among enterprises[15]. Similarly, other relationships exist among companies, which also contribute to identifying the propagation of financial fraud risks within companies. For instance, from 2014 to 2019, Huarong encountered severe financial statement fraud issues, and its auditing firm, Deloitte Touche Tohmatsu CPA Ltd., had significant audit deficiencies[16]. Inference can be drawn that companies audited by Deloitte may face a higher risk of financial statement fraud than those audited by other entities.

Additionally, current methods suffer from the issue of class imbalance, with normal samples outnumbering fraudulent ones significantly. Models trained on imbalanced data tend to classify samples as non-fraudulent, thereby diminishing the accuracy of identifying fraudulent samples. Previous methods rely on either under-sampling or oversampling techniques to address class imbalance issues[17]. However, random under-sampling leads to the loss of valuable information on the training data and constrains the generalization of trained models. Oversampling techniques create fraudulent samples by either duplicating existing fraud samples or generating synthetic fraud samples. However, it could introduce bias in the training samples and consequently lead to performance degradation. The imbalanced data make detecting fraud more challenging.

Various relationships exist among companies, including upstream-downstream transactions and investment connections. Companies may collectively participate in fraudulent activities through these relationships, underscoring the significance of investigating unstructured connections between companies to uncover concealed fraud. We investigate three relationships among listed companies, constructing three distinct sub-graphs accordingly. **Industry relationship** links companies belonging to the same industry, which serves various roles and functions within an industrial value chain. It aids in comprehending the competition and cooperation among companies, unveiling the dynamic changes in industry development. Comparing companies within the same industry and analyzing performance changes over time can help identify potential signs of financial statement fraud. **Supply-chain relationship** connects upstream, mid-stream, and downstream enterprises on product lines. Companies within the same supply chain exhibit strong correlations, offering insights into the spread of risks among companies[18]. **Accounting-sharing relationship** connects the listed companies that accept audit services from the same auditing firm within the same year. Companies that engage the services of the same accounting firm for auditing often have closer cooperation. Specifically, certain accounting firms may collude with companies to commit financial statement fraud, increasing the likelihood of fraudulent activities among the companies they have served. Therefore, an accounting-sharing sub-graph can assist in identifying organized fraud. The three sub-graphs are fused into a single multi-relational graph for statements fraud detection.

Subsequently, we propose a novel multi-relational graph neural network model, named FraudGCN, to identify financial statement fraud. FraudGCN is composed of a diffusion-based under-sampling model, a multi-relational Graph Convolutional Network (GCN) encoder, and a Multi-Layer Perceptron (MLP) classifier with focal loss. The original dataset exhibits a significant class imbalance, requiring under-sampling to mitigate this issue. Additionally, under-sampling improves model reusability. However, using a basic random under-sampling method to retrieve a balanced dataset might discard numerous normal samples, leading to information loss. In this paper, we employ a diffusion-based under-sampling technique to select training samples based on their importance in the graph. We calculate a diffusion matrix to obtain a global view of node importance. Leveraging this diffusion matrix, we perform node under-sampling, focusing on nodes that are relatively more influential from the global perspective.

The multi-relational GCN encoder learns node embeddings that capture rich semantic information from multiple relationships and discover the inter and intra-relational neighborhood information. Compared to traditional GCNs, multi-relational GCNs provide greater flexibility as they can simultaneously consider various relationships or edges, enabling the assignment of distinct weights. This scheme enables the encoder to capture diverse node relationships more effectively, making them well-suited for complex graph data. During training, we employ the multi-relational GCN encoder based on the propagation scheme. Different from transductive learning, this approach's advantage lies in its capability to handle new nodes without retraining the algorithms. We can directly employ the model to predict labels for previously unseen instances, thus providing an inductive learning approach.

However, the mini-batch data obtained by diffusion-based under-sampling still exhibits imbalance, though to a reduced degree. We introduce the focal loss function to address the class imbalance problem in the mini-batch. By assigning higher weights to fraud samples through adjustments to the focusing parameters, the focal loss prioritizes challenging instances that possess more discriminative information. This policy enhances the model's acquisition of crucial information, thereby improving classification performance. The main contributions of this work are as follows:

- We introduce FraudGCN, an innovative multi-relational graph convolutional network for detecting financial statement fraud. It explores various kinds of connections between companies to aggregate comprehensive neighborhood information. To the best of our knowledge, this is the first attempt to apply multi-relational graphs in the field of financial statement fraud detection.
- We propose a diffusion-based node sampling method to address the class imbalance problem in fraud detection, which can select globally crucial nodes for model training. Additionally, we introduce Focal Loss to adjust weights for the sparse fraudulent samples.
- We conduct extensive experiments to evaluate the performance of FraudGCN on a real-world dataset from Chinese listed companies. Experimental results show that our model outperforms state-of-the-art approaches by 3.15% in Macro-recall, 3.36% in Macro-F1, and 3.86% in GMean.

The rest of this paper is organized as follows: Section 2 presents the related work. Section 3 describes our proposed method. Section 4 presents the conducted experiments, and Section 5 concludes this work with future research directions.

2 Related Work

2.1 Data-driven methods

Fraud detection in financial statements is typically formulated as a binary classification challenge. Various approaches have been proposed, employing conventional machine learning classifiers, including Logistic Regression (LR)[19], Decision Trees (DT)[19], and Support Vector Machine (SVM)[20] for financial statement fraud detection. These methods can be broadly classified into two categories based on the types of data utilized in the models: structured and unstructured.

Some studies have found that financial-related structured data can effectively improve the predictive performance of machine learning classifiers such as LR and SVM[20]. These structured data include raw financial data and financial ratio indicators[21]. Research has shown that using an SVM model based on raw financial data outperforms financial indicators based prediction models[22]. DT models based on both raw financial data and financial indicators can quickly draw effective conclusions from large datasets[23, 24]. However, these methods may be prone to overfitting. Random forests identify fraudulent activities in the financial statements of listed companies and offer better robustness and stability than DT[25]. Some studies suggest that fraudulent companies often misrepresent their true financial data, limiting classification performance[26].

Non-structured data, such as textual data from annual reports and audio samples from earnings calls, can better reflect the true business conditions of companies. Consequently, deep learning algorithms are employed to quantify the textual data from annual reports, significantly improving the predictive accuracy[26, 27]. Some studies leverage ensemble methods, which outperform other machine learning methods in accurately classifying fraudulent companies[28, 29]. Extracting linguistic indicators from the Management's Discussion and Analysis (MD&A) section of companies' annual reports can serve as an effective complement to structured data, enabling early warning for financial statement fraud[30]. Experiments have demonstrated that obtaining audio samples from Chief Executive Officers (CEOs) during earnings calls and generating audio markers are helpful in detecting financial fraud[31]. Dong et al.[32] proposed a text analysis framework based on Systemic Functional Linguistics (SFL) for automatically extracting text data from financial social media to assess companies' fraud risks. These methods extract features of companies from non-structured data from various perspectives, complementing structured financial data. Table 1 summarizes prior research related to financial statement fraud detection in the literature. Most of these studies use machine learning algorithms to detect financial fraud, with the primary data sources being a company's financial data and textual information. However, these methods primarily concentrate on the characteristics of individual companies, overlooking the wealth of interaction information of companies.

2.2 GNN-based approaches

Neural networks[39, 40], with their powerful data modeling and feature extraction capabilities, have been widely applied in various fields such as Natural Language Processing (NLP) and Computer Vision (CV). Concurrently, significant advancements in neural networks have paved the way for tackling complex scientific challenges[41], especially in addressing nonlinear problems[42–44]. Among these developments, Graph Neural Networks (GNNs), as a critical branch of neural networks, have been specifically optimized and

Table 1 Previous studies on financial statement fraud detection.

Reference	Year	Data type	Classification method
Cecchin et al.[22]	2010	Raw financial data	SVM
Cecchin et al.[26]	2010	Textual data from MD&A, financial ratio indicators	SVM
Chen et al.[33]	2014	Corporate governance indexes, financial ratio indicators	RF, RST, DT, BNP
Hajek and Henriques[30]	2017	Textual data from MD&A, financial ratio indicators	LR, NB, BBN, DT, SVM, MLP, Bagging, RF, Adaboost
Ozdagoglu et al.[34]	2017	Financial ratio indicators	DT, LR, ANN
Rizki et al.[35]	2017	Financial ratio indicators	SVM
Tang et al.[36]	2018	Raw financial data	C4.5 DT
Bao et al.[37]	2020	Raw financial data	RUSBoost
Craja et al.[12]	2020	Textual data from MD&A, financial ratio indicators	HAN
Wu and Du[38]	2022	Textual data from MD&A, financial ratio indicators	RNN, CNN, LSTM, GRU

designed for the structural characteristics of graph data. GNNs excel in processing data with network-like structures, including but not limited to social networks, knowledge graphs, and protein interaction networks, by efficiently leveraging the connections between nodes[45, 46].

Various GNN-based models have been proposed to tackle financial fraud detection and enterprise risk prediction. Liu et al.[47] constructed heterogeneous account-device graphs and introduced Graph Embeddings for Malicious accounts (GEM) for detecting abnormal accounts in Alipay. Feng et al.[48] considered the interaction of internal company features and explicitly modeled feature interactions using CCR-GNN for corporate credit rating. Yang et al.[15] proposed ST-GNN for extracting supply chain relationships among enterprises to predict enterprise risks. Additionally, GraphConsis[49] has been recognized for its balanced sampling optimization technique, devised to mitigate inconsistencies in risk control scenarios. The incorporation of Related Party Transactions (RPTs) node degree with financial data has been demonstrated to significantly boost fraud detection efficacy[50].

Traditional GNNs are confined to single-relationship graph data, yielding a limited scope of structural insights. Multi-relational GCNs emerge as a more versatile solution. These networks are adept at integrating multifaceted information from diverse sources, offering a richer data representation and enhanced analytical flexibility by concurrently evaluating multiple relational contexts. This capability renders them particularly effective for complex graph data analyses. In contrast to traditional fraud detection methods, our approach focuses on utilizing multi-relational graphs to discern complex cooperative dynamics among corporations.

3 Our Method

3.1 Problem definition

A multi-relational graph is an extended graph structure composed of nodes and multi-relational edges designed to represent complex relationships among nodes. Unlike traditional graphs, a multi-relational graph allows for multiple types of edges between nodes. A multi-relational imbalanced graph can be formalized as $G=\{V, \mathcal{E}, A, X, Y\}$, where $V=\{v_1, v_2, \ldots, v_N\}$ is the set of nodes, N is the number of nodes; $\mathcal{E}=\varepsilon_1 \cup \varepsilon_2 \cup \cdots \cup \varepsilon_R$ is the set of edges over all relations, ε_R is the set of edges of the R-th relation, R is the number of relations; $A=\{\boldsymbol{A}_1, \boldsymbol{A}_2, \ldots, \boldsymbol{A}_R\}$ is the set of multiple adjacency matrices, $\boldsymbol{A}_R$ is the adjacency matrix of the R-th relation; and X and Y are the sets of node features and labels, respectively. $e^r_{u,v}=(u,v)\in\varepsilon_r$ denotes an edge connecting nodes u and v with the relation r, and ε_r is the set of edges of the r-th relation. For each node v, $\boldsymbol{x}_v \in X^L$ is a d-dimensional feature vector, X^L is the set of features of labeled nodes, $y_v \in Y$ is the label of node v, $y_v=1$ means the node is fraud, and $y_v=0$ means the node is normal. Fraud detection aims to predict the label of a node in unlabeled nodes $X^U=X\backslash X^L$. In practice, we consider individual companies as nodes, with the node set V comprising all the listed companies. A multi-relational graph is constructed based on the relationships of these companies. Some companies have been subject to inspections by regulatory authorities and have their results disclosed. These nodes consist of labeled data X^L, while the remaining companies are considered unlabeled nodes in X^U. Our objective is to perform fraud detection on these unlabeled nodes X^U, i.e., those companies have not been subject to inspections.

The fraud detection model leverages intrinsic node features and multi-relational graphs to identify fraud nodes that significantly deviate from normal ones. This process is formulated as a binary classification task with imbalanced classes on multi-relational graphs. Class imbalance refers to the disproportionate distribution of node categories, with normal nodes substantially outnumbering fraudulent ones.

3.2 Overview of the proposed method

Figure 1 illustrates the architecture of our proposed model FraudGCN, which comprises three components: a diffusion-based under-sampling module for addressing class imbalance, a multi-relational GCN encoder for generating node embeddings with comprehensive semantic information derived from multiple relations, and an MLP for fraud prediction. In the training set X^L, the number of normal samples significantly exceeds that of fraudulent ones, resulting in class imbalance. The under-sampling module utilizes a diffusion matrix to select training nodes X^{L_d}. During training, nodes are grouped into mini-batches. Within each mini-batch, the multi-relational GCN encoder processes these nodes to extract their embeddings. The obtained node embeddings are then combined with the

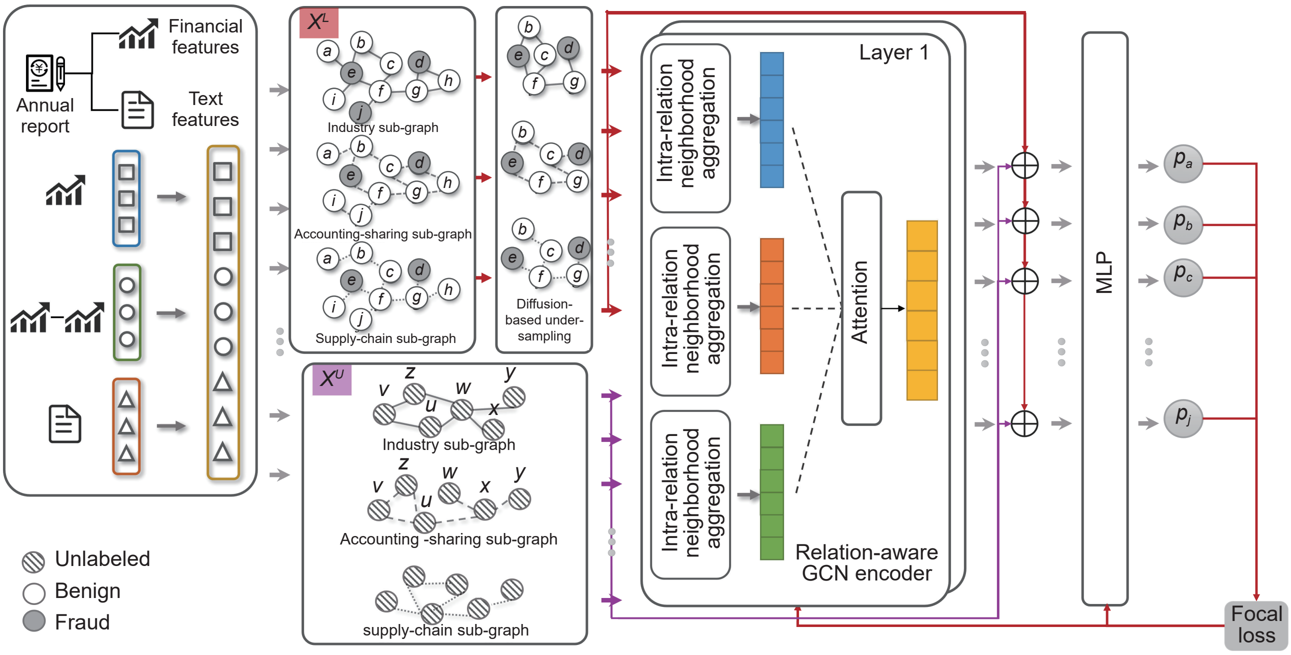
Financial features
Text features
Annual report
Unlabeled
Benign
Fraud
X^L
Industry sub-graph
Accounting-sharing sub-graph
Supply-chain sub-graph
Diffusion-based under-sampling
X^U
Industry sub-graph
Accounting -sharing sub-graph
supply-chain sub-graph
Layer 1
Intra-relation neighborhood aggregation
Intra-relation neighborhood aggregation
Intra-relation neighborhood aggregation
Attention
Relation-aware GCN encoder
MLP
p_a
p_b
p_c
p_j
Focal loss

Fig. 1 Overview of the proposed method. Based on the diverse interconnected relationships between companies, we construct three types of sub-graphs: industry, accounting-sharing, and supply-chain. Node features are extracted from the financial data and textual content of companies' annual reports. In these sub-graphs, gray nodes represent companies involved in fraud, white nodes depict normal companies, and striped nodes are for those unlabeled. The training process is illustrated with red arrows, whereas the testing phase is indicated by purple arrows. Within the training set X^L, a diffusion-based under-sampling technique is employed to obtain quasi-balanced subsets. Subsequently, the multi-relational GCN encoder initially aggregates neighborhood information within each relationship and subsequently integrates them with an attention mechanism. The resultant embeddings are then fed into an MLP to calculate the probabilities of node fraud $p_a - p_j$. For unlabeled nodes, fraud probabilities are directly inferred using the trained model.

original node features through skip connections to form the final embeddings. Subsequently, an MLP is employed to predict whether a node is fraudulent. The GCN encoder and MLP parameters are optimized using focal loss to mitigate class imbalance in a mini-batch. For testing, the model predicts fraud for each node without under-sampling.

3.3 Multi-relational graph construction

The multi-relational graph encompasses three distinct types of relations: industry, accounting-sharing, and supply-chain.

The construction of the industry sub-graph involves linking companies within the same industry during the same year. Industries play various roles and functions in an industry value chain. Industry sub-graph is instrumental in elucidating the interactions between competition and cooperation among companies, revealing dynamic changes in an industry's evolution. The process of constructing Industry sub-graph G_{Industry} involves multiple steps, with the corresponding edge set denoted as $\varepsilon_{\text{Industry}}$. Initially, data about all listed companies and their respective industry categories are collected, followed by establishing connections among companies by matching their stock symbols with the same industry categories.

The principal aim of audit services is to ensure the accuracy and transparency of a company's financial reports, thereby protecting the interests of both investors and stakeholders. Engaging the same accounting firm for auditing can cultivate a reciprocal relationship, potentially yielding various business synergies. When numerous companies choose to employ the same auditing firm, it can lead to diverse collaborative benefits and synergistic business opportunities, allowing them to mutually benefit in various ways. The Accounting-sharing sub-graph $G_{\text{Accounting-share}}$ connects listed companies that utilize audit services from the same auditing firm within a specific year. The edge set of $G_{\text{Accounting-share}}$ is denoted as $\varepsilon_{\text{Accounting-share}}$. This sub-graph is instrumental in identifying organized fraud.

Financial interactions among companies within the same supply chain are typically interdependent. If one company in the supply chain commits financial fraud, the likelihood of fraudulent activities in other companies within the same chain tends to be higher than average[15]. The Supply-chain sub-graph is constructed by linking enterprises along the same supply chain, including both upstream and downstream entities, denoted as $G_{\text{Supply-chain}}$. The associated edge set for this graph is denoted as $\varepsilon_{\text{Supply-chain}}$. The final multi-relational graph comprises three distinct graphs: $G=\{V, \mathcal{E}, A, X, Y\}$, where $\mathcal{E}=\varepsilon_{\text{Industry}} \cup \varepsilon_{\text{Accounting-share}} \cup \varepsilon_{\text{Supply-chain}}$.

3.4 Diffusion-based under-sampling method

We propose a diffusion-based sampler for node selection from the training set X^L. The key idea entails integrating a global perspective of node importance into the sampling process where nodes deemed significant globally are more likely to be sampled. Initially, we employ graph diffusion techniques[51, 52] to learn the structure of a graph, enabling the quantification of a node's importance from a global standpoint. In a multi-relational graph comprising multiple adjacency matrices, we calculate the diffusion matrix for each relationship and then compute a comprehensive result. Given the i-th relationship's adjacency matrix A_i, we calculate its graph diffusion matrix S_i as follows:

$$S_i=\sum_{k=0}^{\infty}\theta_k T_i^k \tag{1}$$

where T_i denotes the generalized transition matrix, T_i^k represents the k-th power of T_i, θ_k is a weighted coefficient that modulates the balance between global and local information, and S_i signifies the diffusion matrix for the i-th relationship. By adopting specific values for T_i and θ_k, different instantiations of graph diffusion can be obtained. This paper employs Personalized PageRank (PPR)[53] as an instantiation of graph diffusion. Specifically, PPR selects $T_i=\boldsymbol{A}_i\boldsymbol{D}_i^{-1}$ and $\theta_k=\alpha_d(1-\alpha_d)^k$, where $\boldsymbol{D}_i$ is the diagonal matrix of adjacency matrix $\boldsymbol{A}_i$, and $\alpha_d\in(0,1)$ is the propagation probability during PPR random walks. To circumvent multiple iterative steps, we calculate the PPR graph diffusion matrix as follows:

$$\boldsymbol{S}_i^{\text{PPR}}=\alpha_d\,(\boldsymbol{I}_n-(1-\alpha_d)\boldsymbol{D}_i^{-1/2}\boldsymbol{A}_i\boldsymbol{D}_i^{-1/2})^{-1} \tag{2}$$

where $\boldsymbol{I}_n$ is the identity matrix. In the diffusion matrix, the t-th column offers a global perspective on the connectivity between node v_t and other nodes. Consequently, we can employ the diffusion matrix to select nodes that are relatively important in the global context.

For achieving a balanced training set, nodes belonging to minority classes should be assigned a greater sampling probability compared to those from majority classes. Hence, the final sampling probability is determined as follows:

$$P(v)=\begin{cases}\dfrac{\boldsymbol{S}_1^{\text{PPR}}(:,v)+\boldsymbol{S}_2^{\text{PPR}}(:,v)+\boldsymbol{S}_3^{\text{PPR}}(:,v)}{N_1}, & y_v=1;\\[2ex] \dfrac{\boldsymbol{S}_1^{\text{PPR}}(:,v)+\boldsymbol{S}_2^{\text{PPR}}(:,v)+\boldsymbol{S}_3^{\text{PPR}}(:,v)}{N_0}, & y_v=0\end{cases} \tag{3}$$

where $\boldsymbol{S}_i^{\text{PPR}}(:,v)$ represents the sum of the column in $\boldsymbol{S}_i^{\text{PPR}}$ of the i-th sub-graph where node v is located, and N_1 and N_0 denote the number of nodes with class $y_v=1$ and $y_v=0$, respectively. The ratio of normal to fraudulent samples is defined as sr. Through under-sampling, sr is reduced to mitigate severe data imbalance.

To align the normal-to-fraudulent sample ratio with a specific target value of sr, a two-step process is required. Initially, the diffusion-based under-sampling method is employed to uniformly sample all categories, obtaining both normal and fraudulent samples. At this juncture, the ratio may not align with sr. Subsequently, the sample ratio is fine-tuned by selectively removing excess fraudulent samples until the target sr is attained.

3.5 Multi-relational GCN encoder

Considering the challenges faced by traditional GNNs in processing multi-relational graphs, we propose a multi-relational GCN encoder comprising two components: intra- and inter-relation neighborhood aggregations. In a single-relation sub-graph, the aggregation process incorporates neighborhood information pertinent to that specific relation. Each relation represents an important aspect within the multi-relational graph. Aggregating neighborhood information within each relation enables the effective utilization of diverse relational data, thereby facilitating a more thorough and integrated understanding.

In each relation of a multi-relational graph, the aggregated features encompass only the neighborhood information pertinent to that specific relation, thus reflecting a single facet of the information from a specific kind of neighbors. In a multi-relational graph, a node's neighbors vary across different relations, indicating that the node can aggregate more diverse information by considering multiple relationships. For a more holistic representation, integrating the aggregated features from different relations is crucial, enabling the utilization of rich semantic information within the multi-relational graph. The aggregation process in a multi-relational graph consists of two phases: (1) the intra-relation aggregation that aggregates neighborhood information in a single sub-graph, and (2) the inter-relation aggregation that fuses information from different relationships.

3.5.1 Intra-relation aggregation

Within a single sub-graph of the multi-relational graph, similarity-based top-p sampling is utilized for aggregation operations on neighbors within the neighborhood[54]. Node embedding is calculated by aggregating information from these sampled neighbors.

The process of updating node embeddings involves two phases. Initially, a similarity measurement mechanism is devised to select neighbors with higher similarity. This approach prevents the over-aggregation of normal samples, which might otherwise diminish the detectability of fraudulent nodes. Within each relationship, we perform top-p sampling for a target node, predicated on its similarity to neighboring nodes. In a single-relation graph, the similarity between u and v is calculated as follows:

$$\mathrm{sim}\,(v,u) = \frac{\boldsymbol{h}_v^{(l)\mathrm{T}}\boldsymbol{h}_u^{(l)}}{\left\|\boldsymbol{h}_v^{(l)}\right\|_2\left\|\boldsymbol{h}_u^{(l)}\right\|_2},\ (v,u)\in\varepsilon_r^{(l)} \tag{4}$$

where $\boldsymbol{h}_v^{(l)}$ is the embedding of node v at the l-th layer, $\boldsymbol{h}_v^{(0)}=\boldsymbol{x}_v$, $\boldsymbol{x}_v$ is the feature of node v, and $\varepsilon_r^{(l)}$ is the set of edges of node v at the l-th layer under relation r.

Subsequently, the embeddings of the selected neighbors are aggregated to obtain a comprehensive representation of the target node. The aggregation operation for the neighbors of node v is denoted as follows:

$$\boldsymbol{h}_{v,r}^{(l+1)} = \mathrm{Aggregate}\,(\{\boldsymbol{h}_{u,r}^{(l)}, u\in N_r^{(l)}(v)\}) \tag{5}$$

where $N_r^{(l)}(v)$ is the set of sampled neighbors of node v at the l-th layer under relation r, and $\boldsymbol{h}_{u,r}^{(l)}$ denotes the representation of node u at the l-th layer under relation r, $\boldsymbol{h}_{v,r}^{(l+1)}$ represents the aggregated embedding of the target node v obtained through top-p sampling of neighbors, and Aggregate $(\cdot)$ refers to the mean pooling function. This approach effectively captures the contextual information of nodes in each corresponding relationship.

3.5.2 Inter-relation aggregation

Inter-relation aggregation facilitates the integration of features across various relationships, thereby uncovering interconnections between companies. Nonetheless, the significance of each relation varies depending on the target node. To ensure a thorough fusion of information, an attention mechanism is employed to compute attention coefficients for different relations. The calculation proceeds as follows:

$$\alpha_{v,r}^{(l)} = \frac{\exp\,(\sigma\,(a\cdot[\boldsymbol{W}_r{}^{(l)}\boldsymbol{h}_v^{(l-1)}\,\|\,\boldsymbol{W}_r{}^{(l)}\boldsymbol{h}_{v,r}^{(l)}]))}{\sum\limits_{\phi=1}^{R}\exp\,(\sigma\,(a\cdot[\boldsymbol{W}_r{}^{(l)}\boldsymbol{h}_v^{(l-1)}\,\|\,\boldsymbol{W}_r{}^{(l)}\boldsymbol{h}_{v,\phi}^{(l)}]))} \tag{6}$$

where "||" is the concatenation operation, σ denotes the activation function, a is a learnable weight vector, $\boldsymbol{W}_r{}^{(l)}$ is a trainable weight matrix, and $\alpha_{v,r}^{(l)}$ represents the attention coefficient of relation r for node v at the l-th layer, and R is the number of relations.

Ultimately, the formula for aggregating features across various relations is as follows:

$$\boldsymbol{h}_v^{(l)} = \sigma\,(\boldsymbol{W}_v{}^{(l)}\boldsymbol{h}_v^{(l-1)}\oplus\sum_{r=1}^{R}\alpha_{v,r}^{(l)}\boldsymbol{W}_v{}^{(l)}\boldsymbol{h}_{v,r}^{(l)}) \tag{7}$$

where $\boldsymbol{W}_v{}^{(l)}$ is a trainable weight matrix, $\boldsymbol{h}_v^{(l-1)}$ denotes the node embedding in the $(l-1)$-th layer of node v, $\boldsymbol{h}_v^{(l)}$ represents the node embedding from relation r in the l-th layer, and "$\oplus$" denotes the summation operation of embeddings.

3.6 Classifier and model optimization

In our model, node features $\boldsymbol{x}$ are fed into the MLP layer via a skip connection. Specifically, we arrange node embeddings obtained from the multi-relational GCN encoder in conjunction with raw node features in a concatenated manner. This concatenated representation serves as the ultimate node vector representation. For each node v, the ultimate embedding of the node is computed as follows:

$$z_v = \boldsymbol{h}_v^{(N_{\mathrm{lay}})}\,\|\,\boldsymbol{x}_v \tag{8}$$

where $\boldsymbol{h}_v^{(N_{\mathrm{lay}})}$ is the node embedding output from the relational-aware GCN, N_{lay} is total number of layers, $\boldsymbol{x}_v$ represents the original features of node v, and z_v is the final embedding of node v. Subsequently, an MLP is utilized for predicting node labels,

$$\mathrm{MLP}\,(z_v) = \sigma\,(\boldsymbol{b}_2+\boldsymbol{W}_2\,(\sigma\,(\boldsymbol{b}_1+\boldsymbol{W}_1 z_v))) \tag{9}$$

where $\sigma(\boldsymbol{b}_1 + \boldsymbol{W}_1 z_v)$ represents the output of the hidden layer, $\boldsymbol{b}_1$ and $\boldsymbol{W}_1$ are the bias vector and weight matrix from the output layer to the hidden layer, with σ being the activation function. Similarly, $\boldsymbol{b}_2$ and $\boldsymbol{W}_2$ denote the bias vector and weight matrix from the hidden layer to the output layer. The MLP is configured with two hidden layers.

Loss functions are used to calculate the difference between the predicted and true values[55]. By optimizing the loss function, the model adapts its parameters effectively[56]. Even in a mini-batch training set, class imbalance persists due to the number of normal over fraudulent samples, thereby complicating the task of fraud detection. To mitigate the impact of class imbalance during training, we incorporate the focal loss. The main idea behind focal loss is to reduce the weight of easy examples and increase the weight of hard examples, thereby emphasizing the challenging samples for the model. We introduce the focal loss to optimize the model as follows:

$$\begin{aligned}\text{Loss} = \sum_{v\in X^{L_d}} -(\alpha(1-p_v)^{\gamma} y_v \log(p_v) + \\ (1-\alpha)(p_v)^{\gamma}(1-y_v)\log(1-p_v))\end{aligned} \tag{10}$$

where $p_v = \text{MLP}(z_v)$, X^{L_d} is the set of nodes obtained by diffusion-based under-sampling from the training set X^L during the current epoch, α is a balance factor to balance the importance of fraud/normal samples and alleviate the class imbalance issue, and γ is an adjustable parameter that controls the balance between the weights of easy and hard samples. γ acts as a modulating factor, diminishing the weight of easily classified samples while amplifying that of more difficult ones, thereby directing the model's focus towards minority classes and challenging cases. Focal loss can bolster the predictive accuracy of the MLP and counter overfitting by reducing the model's inclination to classify a test sample as normal.

Algorithm 1 shows the pseudocode of the training process. And our model employs 98 neurons and 26 722 trainable parameters to optimize performance. In the relational aggregation module, 12 160 trainable parameters are used to adjust the nodes' embedding dimension from an initial feature dimension of 190 to a standardized embedding dimension of 64. For the attention mechanism, 128 trainable parameters are used for linear transformation. Furthermore, the final MLP module comprises 98 neurons and 14 454 trainable parameters, allocated over three linear layers. This design has been carefully tuned based on the complexity of the task and characteristics of the dataset to achieve optimal performance.

Algorithm 1 FraudGCN

```
Input: $G = \{V, \mathcal{E}, A, X, Y\}$, $N_{\text{batch}}$: number of training
    batch size, $L$, $R$, and sr.
Output: Fraud probability for each node in $X^L$.
Initialization $\boldsymbol{h}_v^{(0)} \leftarrow \boldsymbol{x}_v, P(v)$, and $v \in X^L$ ;
for $e = 1, 2 \ldots, N_{\text{epoch}}$ do
    Sample nodes according to the probability based on
    diffusion-based under-sampling defined in Eq. (3) to
    obtain $V_P$;
    //Set of sampling nodes is marked as $V_P$
    Decide the number of training batches $B = \left\lceil \frac{|V_P|}{N_{\text{batch}}} \right\rceil$;
    for $b = 1, 2 \ldots, B$ do
        for $l = 1, 2 \ldots, N_{\text{lay}}$ do
            for $r = 1, 2 \ldots, R$ do
                sim $(v, u)$ is calculated by Eq. (4);
                $N_r^{(l)}(v)$ is obtained by Top-$p$ sampling;
                $\boldsymbol{h}_{v,r}^{(l)}$ is obtained by Eq. (5);
                //Intra-relation aggregation
            end
            $\alpha_{v,r}^{(l)}$ is obtained by Eq. (6);
            $\boldsymbol{h}_v^{(l)}$ is obtained by Eq. (7);
            //Inter-relation aggregation
            $z_v$ is obtained by Eq. (8);
            Focal loss is obtained by Eq. (10);
        end
    end
end
```

4 Experiment and Result

4.1 Experimental settings

4.1.1 Datasets

A real-world dataset is gathered from 4043 Chinese listed companies in 2021. This dataset comprises financial statements, industry categorization, auditing firm affiliations, and supply chain details of these companies. The data are sourced from the annual financial statements of Chinese listed companies and the China Stock Market & Accounting Research database (CSMAR).

In light of the limitations identified in prior works, we have extracted three types of features: raw financial statements data, changes in financial statements data, and annual report text data. Raw financial statement data are compiled from three distinct tables: the balance sheet, income statement, and cash flow statement. This data accurately reflect a company's financial status for a given year, such as its liabilities and profitability. Financial indicators with excessive missing values are omitted, resulting in the retention of 91 raw financial indicators. These 91-dimensional original financial features are denoted as financial features (ff). The details of financial features are elaborated in Appendix in Table A1. Changes in financial statements data are determined by calculating variations between the raw financial data of two successive years, thereby capturing the dynamic nature of a company's financial condition. We compute the difference between a company's financial features of two consecutive years, denoted as Δff. Consequently, the feature dimensionality expands to 182. Annual report text data are sourced from the MD&A section of the company's annual report[26]. Text features (tf) encompass various textual indicators from the MD&A section, including text similarity compared to the previous year, the number of positive words, the number of negative words, the total vocabulary, the number of sentences, the word count, and sentiment analysis. The unstructured textual data from the annual report offer comprehensive and objective insights into the company's business operations within the given year. The detailed composition and calculation methods of text features are detailed in Table 2, where "mean" represents the average value of the corresponding feature, and "standard deviation" represents the standard deviation of the corresponding feature.

We denote the three kinds of features as ff, Δff, and tf, respectively. The raw feature vector of a single company is obtained as follows:

$$\boldsymbol{x} = \mathrm{ff} \parallel \Delta\mathrm{ff} \parallel \mathrm{tf} \tag{11}$$

where $\boldsymbol{x}$ is the raw feature vector. Table 3 illustrates the statistical characteristics of these features.

The annual financial statements offer a clear depiction of the financial and operational conditions of the listed companies for the given year. In this dataset, we classify companies based on their audit opinions. Companies with an audit opinion of "Unqualified opinion" are labeled as normal samples, whereas the others are marked as fraud samples. "Unqualified opinion" refers to the auditors' conclusion that, following the audit of a company's financial statements, the financial information presented in the statements is true, accurate, and complete. Among the 4 043 companies, 235 companies are labeled as fraud, leading to a significant imbalance. The Imbalance Ratio (IR), defined as the ratio of majority to minority class samples, is 16.20. Table 4 provides a summary of the dataset, which contains a total of 828 497 edges, and average degree refers to the average value of the degrees of all nodes in the graph. In subsequent experiments, a stratified sampling method is employed to create training and test sets at a ratio of 3 to 1. The use of stratified sampling provided by Scikit-learn[57] ensures that the imbalance proportions of samples within each subset remain consistent with those in the original dataset. The experiment is repeated five times, and the average results are reported. The dataset will be publicly available via Github.

4.1.2 Baseline methods

We employ the following twelve widely-recognized methods as baselines to highlight the effectiveness of

Table 2 Statistics of text features.

Feature	Meaning	Calculation	Mean	Standard deviation
TextualSimilarity	Text similarity compared to the previous year	Utilizing Latent Semantic Indexing (LSI) cosine similarity calculation algorithm	91.39	13.13
PositiveVocabularyNum	Number of positive words	–	740.78	348.34
NegativeVocabularyNum	Number of negative words	–	369.10	152.90
TotalWordsNum	Total number of words	–	8997.16	3877.60
SentencesNum	Number of sentences	–	219.71	99.95
WordsNum	Number of the characters	–	17 778.47	7593.60
EmotionTone1	Emotional intonation 1	(PositiveVocabularyNum–NegativeVocabularyNum)/ TotalWordsNum	0.04	0.01
EmotionTone2	Emotional intonation 2	(PositiveVocabularyNum–NegativeVocabularyNum)/ (PositiveVocabularyNum+NegativeVocabularyNum)	0.32	0.11

Table 3 Feature dimension statistics.

Feature	Dimensionality
ff	91
Δff	91
tf	8
Total	190

Table 4 Statistics of text features.

Relation	Number of relation edges	Avgeage degree
Industry sub-graph	334 193	165.32
Accounting-sharing sub-graph	521 360	257.91
Supply-chain sub-graph	4587	2.27

our proposed model.

- LR[58]: LR is a classical classification model for binary classification tasks.
- DT[59]: DT is a tree-based classification model that learns rules based on decisive features to partition unknown instances.
- KNN[60]: K-Nearest Neighbors (KNN) is an instance-based learning algorithm that categorizes a sample into the majority class of the k-nearest neighbors.
- RF[61]: Random Forest (RF) is an ensemble learning method that improves performance by training and predicting on numerous decision trees.
- SVM[62]: SVM is one of the most popular machine learning algorithms. It aims to find an optimal hyperplane that maximizes the margin between data points of different classes.
- XGBoost[63]: Extreme Gradient Boosting (XGBoost) classifier is a gradient boosting tree model that improves predictive performance through iterative training of decision trees.
- LightGBM[64]: LightGBM is a lightweight machine learning framework based on gradient boosting trees.
- GBDT[65]: Gradient Boosting Decision Tree (GBDT) is an ensemble learning algorithm which iteratively trains multiple decision trees to enhance predictive performance.
- GCN[66]: GCN applies convolution operations on graphs to learn effective node representations.
- GraphSAGE[54]: GraphSAGE is an inductive GCN that utilizes aggregation functions to learn node representations from local neighbors of nodes.
- GraphConsis[49]: GraphConsis is a GNN solving three types of inconsistencies present in graphs: context inconsistency, feature inconsistency, and relation inconsistency.
- CARE-GNN[67]: CARE-GNN provides a solution to the issue of fraudster impersonation by incorporating a label-aware similarity measure and a similarity-aware neighbor selector.

GCN and GraphSAGE are employed on the Supply-chain sub-graph, which achieves the best performance among the three relationships. GraphConsis and Care-GNN are applied to the multi-relational graph by treating all relations equally.

4.1.3 Parameter Settings

The model employs the Adam optimizer for parameter optimization, using a learning rate of 0.001 and a weight decay of 0.1. Additionally, the number of layers in the multi-relational GCN model is set to 1. Hyper-parameters α and γ in the focal loss are assigned values of 0.95 and 3, respectively. Top-p sampling selects the top 50% most similar neighbors of a target node. In our experiments, FraudGCN is trained using mini-batch training sets with a batch size of 512, and the number of epochs is 1000. During training, we closely observe the loss function, initiating the premature conclusion of training upon observing loss convergence without notable fluctuations. Furthermore, training ceases if the maximum number of epochs elapses prior to the convergence of the loss function.

4.2 Evaluation metrics

Financial statement fraud detection is an imbalance binary classification problem. The model's performance is assessed using the following evaluation metrics: Macro-precision[68], Macro-recall[68], Macro-F1[68], and GMean[69].

Precision is the ratio of the number of samples correctly predicted as a particular class by the classifier to the total number of samples predicted as that class. It measures the accuracy of the classifier in predicting a specific class as follows:

$$\text{Precision} = \frac{\text{TP}}{\text{TP}+\text{FP}} \tag{12}$$

where TP represents cases where the model correctly predicts positive samples as positive, and FP represents cases where the model incorrectly predicts negative samples as positive. Macro-precision calculates the precision for normal and fraud classes and then takes the average to obtain the macro-average precision.

Recall represents the proportion of samples with a positive actual label for which the model correctly predicted as positive. Its calculation is as follows:

$$\text{Recall} = \frac{\text{TP}}{\text{TP}+\text{FN}} \tag{13}$$

where FN represents cases where the model incorrectly predicts positive samples as negative. Macro-recall is calculated by computing the recall for each class and then taking the average for normal and fraud classes. It offers a measure of the overall recall performance for the model across two classes.

F1-score is a metric employed to evaluate the performance of binary classification models. It represents the harmonic mean of precision and recall:

$$\text{F1-score} = \frac{2\times\text{Precision}\times\text{Recall}}{\text{Precision}+\text{Recall}} \tag{14}$$

The F1-score varying between 0 and 1, serves as an indicator of model performance, where higher values signify enhanced effectiveness. Macro-F1, representing an aggregate assessment of the model's F1-score for each class, offers an objective and precise evaluation of its classification capabilities. It is calculated by first computing the F1-score for each class and subsequently averaging these scores for both normal and fraudulent classes.

GMean is a widely employed metric in the classification of imbalanced datasets, as it accounts for the likelihood of accurate classification for both fraudulent and normal samples. The formula for calculating GMean is as follows:

$$\text{GMean} = \sqrt{\frac{\text{TN}}{\text{TN}+\text{FP}}\times\frac{\text{TP}}{\text{TP}+\text{FN}}} \tag{15}$$

where TN denotes the cases where the model correctly predicts negative samples as negative.

4.3 Effectiveness evaluation

Table 5 displays the performance of different methods, with the optimal results emphasized in bold. To mitigate the impact of the bias from random data divisions on experimental outcomes, the experiments are conducted five times, and their average values are reported.

The experimental results illustrate that FraudGCN attains superior performance for three evaluation metrics: Macro-recall, Macro-F1, and GMean, which are pivotal for evaluating the model's ability to handle imbalanced datasets. FraudGCN achieves the best performance because it excels in leveraging diverse inter-company relationships, thereby enhancing the overall performance of the model. For instance, within the Accounting-sharing sub-graph, the presence of fraud in a company audited by a certain firm suggests a high fraud risk for other companies audited by the same firm. FraudGCN can effectively uncover these relationships. The results further reveal that GNN-based approaches outperform traditional machine learning and ensemble learning methods, highlighting the significant advantage of including inter-company relationships in fraud detection. Furthermore, FraudGCN's enhanced performance over other GNN-based methods suggests that analyzing multiple types of relationships among companies is crucial for accurate fraud detection.

GCN and GraphSAGE achieve higher Macro-precision in analyzing the supply-chain sub-graph. An interpretation is that graph neural networks are able to capture structural information more effectively.

Table 5 Performance for financial statement fraud detection.

Model	Macro-recall	Macro-F1	GMean	Macro-precision
LR	0.6057±0.0218	0.6390±0.0260	0.4693±0.0458	0.7243±0.0308
DT	0.5783±0.0480	0.4230±0.1287	0.5409±0.0603	0.5356±0.0409
SVM	0.6124±0.0932	0.4254±0.2008	0.5432±0.1451	0.5537±0.0546
KNN	0.5431±0.0109	0.5615±0.0168	0.3028±0.0353	0.6911±0.0323
RF	0.6445±0.0851	0.5654±0.0853	0.5205±0.2132	0.6779±0.1028
XGBoost	0.6458±0.1064	0.6148±0.0745	0.5112±0.2258	0.6981±0.0997
LightGBM	0.6301±0.1103	0.5843±0.0593	0.4659±0.2494	0.6971±0.0890
GBDT	0.6413±0.0715	0.4610±0.1275	0.5833±0.1101	0.5719±0.0632
GCN	0.5894±0.0287	0.6259±0.0368	0.4260±0.0648	0.7785±0.0752
GraphSAGE	0.6163±0.0206	0.6588±0.0284	0.4883±0.0422	**0.7797±0.0533**
GraphConsis	0.7428±0.0547	0.6242±0.0360	0.7188±0.0912	0.6155±0.0337
CARE-GNN	0.7963±0.0532	0.6942±0.0442	0.7858±0.0606	0.6567±0.0377
FraudGCN	**0.8222±0.0209**	**0.7175±0.0287**	**0.8161±0.0221**	0.6737±0.0257

However, data imbalance may predispose these models to favor the majority class and overlook minority class instances (e.g., fraudulent activities), diminishing Macro-recall. Additionally, the graph with a single relationship might be tainted with noise, leading to a reduction in Macro-recall and GMean. Comprehensive metrics, such as Macro-F1 and GMean, consider the model's overall performance in distinguishing fraud and normal samples. Our model's outstanding performance in these two metrics demonstrates its optimal functionality in fraudulent company detection.

Compared to GCN and GraphSAGE, CARE-GNN and GraphConsis demonstrate improved performance, reflecting the benefit of integrating multiple inter-company relationships for a more thorough analysis. Notably, FraudGCN outperforms both CARE-GNN and GraphConsis because our proposed diffusion-based sampling method allows for the selection of globally significant nodes and the multi-relational GCN comprehensively captures information from diverse relationships. In addition, the implementation of Focal Loss mitigates the adverse effects of class imbalance on model performance. Compared to the second-best method, our method demonstrates an improvement of 3.15% in Macro-recall, 3.36% in Macro-F1, and 3.86% in GMean.

4.4 Ablation study

To assess the effect of feature compositions on the efficacy of the model, we develop multiple variants of FraudGCN: FraudGCN$_{/\Delta\mathrm{ff}}$ removes the changes of financial statements Δff, FraudGCN$_{/\mathrm{tf}}$ removes tf, FraudGCN$_{/\Delta\mathrm{ff+tf}}$ removes Δff and tf. As depicted in Fig. 2, the performance of FraudGCN$_{/\Delta\mathrm{ff}}$ is lower than that of FraudGCN, indicating that incorporating the annual change of the original financial data enables the model better to understand fluctuations in a company's financial profile. Moreover, the model's efficacy is notably reduced when both Δff and tf are removed, indicating that the integration of these two kinds of features effectively captures the characteristics and historical trends of a company's financial situations.

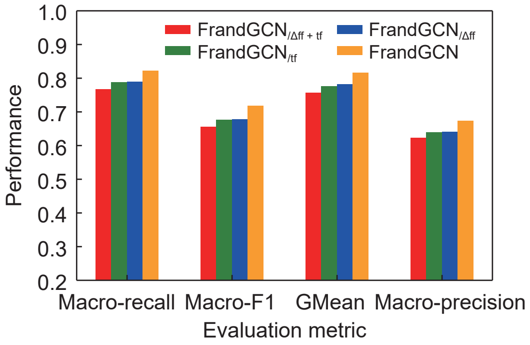

Fig. 2 Ablation study on feature compositions of FraudGCN.

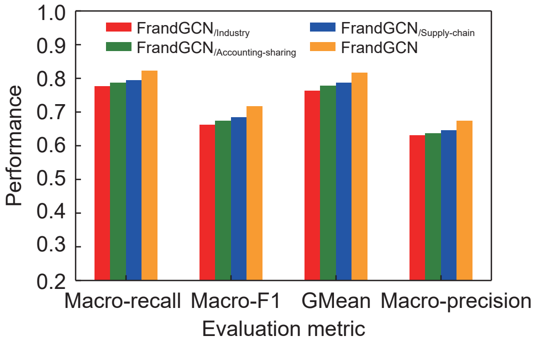

Fig. 3 Ablation study on relational graphs compositions of FraudGCN.

Ablation experiments are conducted to examine the impact of the three relational graphs on FraudGCN. FraudGCN$_{/\mathrm{Industry}}$ denotes the exclusion of the Industry sub-graph, FraudGCN$_{\mathrm{Accounting\text{-}sharing}}$ removes the Accounting-sharing sub-graph, and FraudGCN$_{/\mathrm{Supply\text{-}chain}}$ removes the Supply-chain sub-graph. Figure 3 shows that FraudGCN achieves optimal performance when all three relationships are integrated. Notably, the elimination of the Industry sub-graph leads to the most pronounced decline in performance, indicating that the inclusion of this relationship is more beneficial for detecting fraud. This result may be attributed to the fact that industry relationship aids the model in comprehending a company's financial status and overarching trend. Furthermore, the Industry sub-graph aids in understanding the interactions of competition and collaboration among companies, revealing the dynamic shifts in industry evolution.

To validate the efficacy of our proposed under-sampling method, we conduct a comparative analysis by substituting our diffusion-based under-sampling method with random sampling. Simultaneously, within the Care-GNN model, the standard random under-sampling is replaced with the diffusion-based under-sampling method. Care-GNN-random and FraudGCN-random represent models utilizing random under-sampling, while Care-GNN-diffusion and FraudGCN-diffusion denote the use of under-sampling based on the diffusion matrix. Figure 4 illustrates that the

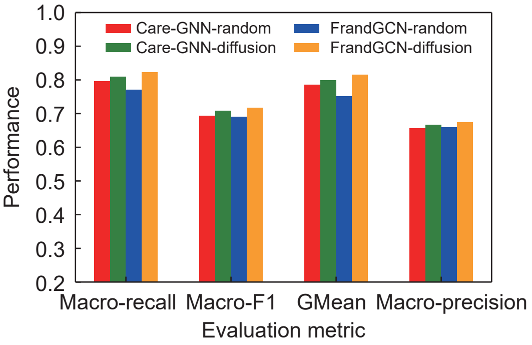

Fig. 4 Ablation study for models with different sampling methods.

diffusion-based under-sampling method significantly improves the model's performance compared to random sampling. This finding indicates that leveraging the diffusion matrix for node selection, i.e., focusing on globally more significant nodes, can augment the model's capability in fraud detection. Notably, FraudGCN outperforms Care-GNN when the same sampling technique is applied. The possible reasons are that FraudGCN incorporates top-*p* sampling to selectively choose neighbor nodes and the application of focal loss to boost the predictive accuracy of the MLP.

To study the influence of the focal loss and under-sampling schemes on the model, multiple variants of FraudGCN are implemented: FraudGCN/FL removes the focal loss, FraudGCN/sampling removes the under-sampling module, and FraudGCN/FL+sampling eliminates the Focal Loss and under-sampling. Figure 5 shows that FraudGCN outperforms the other three variants. Notably, FraudGCN/FL+sampling exhibits a markedly lower GMean than both FraudGCN/FL and FraudGCN/sampling. The inclusion of the Focal Loss and under-sampling modules improves the model performance in terms of GMean. Conversely, the exclusion of both Focal Loss and sampling might predispose the model to favor the majority class, typically the normal samples, thereby augmenting the overall macro-precision.

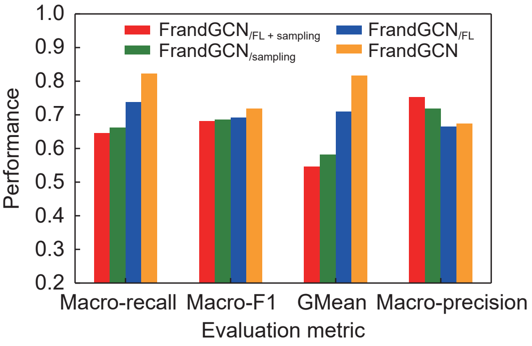

Fig. 5 Ablation study on focal loss and under-sampling schemes of FraudGCN.

4.5 Parameter sensitivity

This section details a sensitivity analysis conducted on various parameters of our model. These parameters encompass the top-*p* sampling, sampling ratio sr, embedding size d, as well as α and γ in the focal loss. Additionally, the number of layers in the multi-relational GCN model and the learning rate are also examined.

Figure 6 presents a sensitivity analysis for top-*p* sampling. When $p = 0$, the model randomly aggregates half of the neighbors; when $p = 10\%$, the model aggregates the top 10% of most similar neighbors for each target node; and when $p = 100\%$, the model aggregates all neighbors. Observations indicate that the model's efficiency initially improves but subsequently declines with the increase of p. When $p = 50\%$, using the top-*p* sampling method yields the best performance. During the aggregation phase, although neighbors contribute essential insights, they may also introduce noise and redundant data. Choosing too few neighbors for aggregation helps filter out noise but might lead to the loss of valuable information. Conversely, aggregating too many neighbors could lead to information redundancy. When $p = 50\%$, the application of top-*p* sampling adeptly eliminates extraneous data while preserving critical information, thereby achieving the highest performance.

We define sr as the ratio of normal samples to fraudulent samples in a mini-batch. Figure 7 depicts the impact of varying the sampling ratio for under-

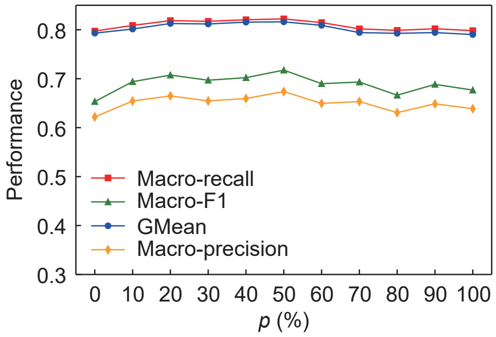

Fig. 6 Parameter sensitivity analysis for the top-*p* sampling.

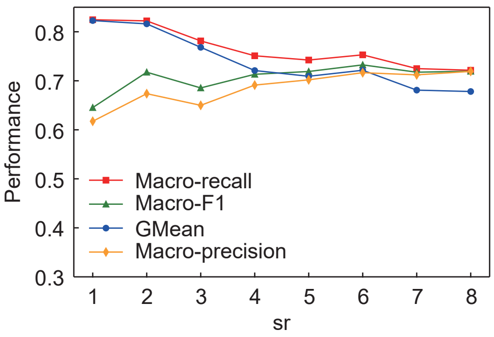

Fig. 7 Parameter sensitivity analysis for the sampling ratio.

sampling. Observations reveal that as the sampling ratio increases, both Macro-F1 and Macro-precision initially rise but subsequently diminish, while Macro-recall and GMean consistently decrease. When sr = 2, i.e., the number of normal samples is twice the number of fraud samples, the model attains relatively high values across all four metrics. When sr = 1, the model demonstrates superior performance in Macro-recall and GMean, but its performance in Macro-F1 and Macro-precision diminishes. One possible reason is that the model has a balanced distribution of positive and negative samples when sr = 1. However, the focal loss guides the model to overly focus on fraudulent samples, ultimately leading to the misclassification of many normal samples as fraud. When sr = 2, the model performs well across all metrics. Therefore, in this study, we set sr = 2 for mini-batch training sets.

To further investigate the effects of focal loss, we set sr = 1 and substitute focal loss with the standard Binary Cross-Entropy Loss (BCELoss). As illustrated in Fig. 8, the model's performance using BCELoss is marginally inferior compared to that achieved with focal loss even when the quantities of normal and fraudulent samples are equal. This observation implies that focal loss contributes to performance enhancement irrespective of data balance.

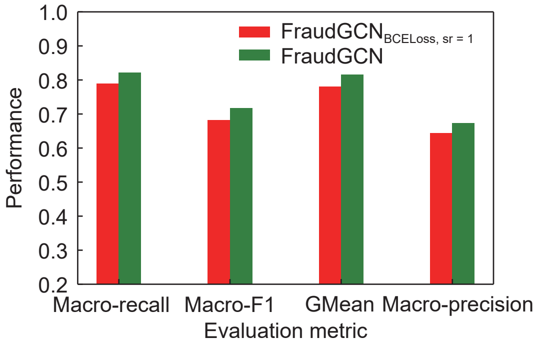

Fig. 8 Parameter sensitivity analysis of FraudGCN with BCELoss, sr=1.

Figure 9 indicates that optimal outcomes are obtained when γ = 3. When γ = 3, the loss for easy samples becomes smaller, and the weight for hard samples increases. This adjustment enables the model to concentrate more effectively on those challenging samples, particularly on companies adept at concealing fraudulent activities, thus enhancing its detection capabilities.

We evaluate the performance of FraudGCN versus the hyperparameter α, which is responsible for balancing the significance of diverse class samples. As illustrated in Fig. 10, FraudGCN exhibits improved performance at α = 0.95. Given the scarcity of fraud samples in the dataset, increasing α amplifies the prominence of these samples. This strategy directs the model's focus more intensely toward the limited fraud samples, thereby mitigating the effects of data imbalance. Nonetheless, when α is too high, the focal loss places excessive emphasis on negative samples, culminating in overfitting to the minority class and a pronounced decline in model performance.

We conduct a further assessment of FraudGCN,

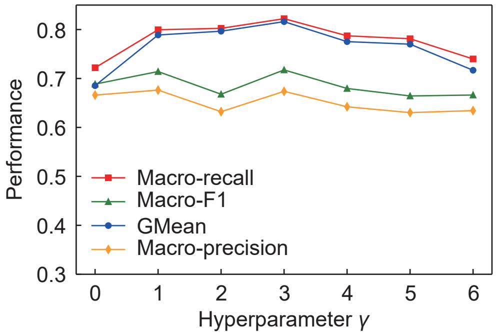

Fig. 9 Parameter sensitivity analysis for the hyperparameter γ of the focal loss.

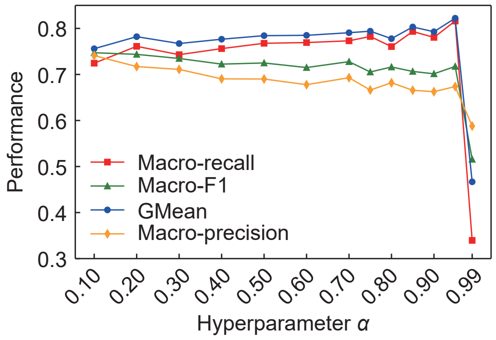

Fig. 10 Parameter sensitivity analysis for the hyperparameter α of the focal loss.

focusing on varying training ratios. The proportion of training samples is adjusted incrementally from 5% to 60%. Figure 11 presents the outcomes corresponding to the adjustments in training ratios. With the expansion of the training set, there is a notable enhancement in the model's performance, signifying its increased proficiency in obtaining high-quality embeddings. Notably, even at a training percentage of 5%, our model demonstrated effective training with limited supervisory signals, demonstrating strong robustness concerning training ratios.

The dimensionality of embeddings notably influences the performance of the model. As depicted in Fig. 12, the performance initially improves and then deteriorates as the embedding dimensionality increases. The optimal performance is observed when the embedding size is 64. This trend can be explained by two key factors: when the embedding size is too small, it is insufficient to capture the complete node information. Conversely, an excessively large embedding size may lead to overfitting issues. Consequently, we set the embedding dimensionality in this study to 64.

Figure 13 displays the model's performance across various learning rates, highlighting the influence of the learning rate on model training. It controls the magnitude of parameter updates during training iteration. A smaller learning rate leads to slower convergence, as parameter updates are smaller, thus requiring a greater number of iterations to reach optimal performance. In contrast, an elevated learning rate might cause instability in the model or prevent effective convergence. The sensitivity analysis reveals that the optimal performance for all four metrics is attained at a learning rate of 0.001. Therefore, we set the model's learning rate to 0.001.

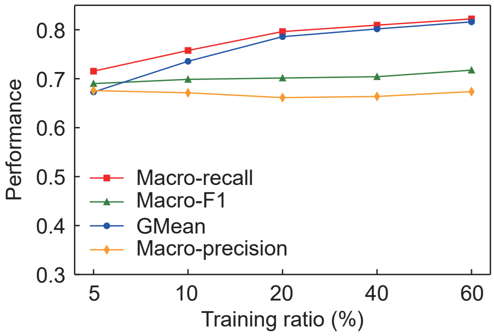

Fig. 11 Parameter sensitivity analysis for the hyper-parameter training ratio.

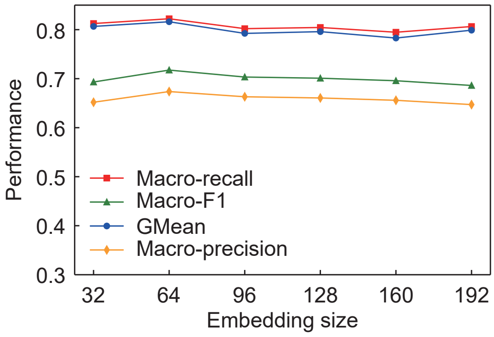

Fig. 12 Parameter sensitivity analysis for embedding dimensionality.

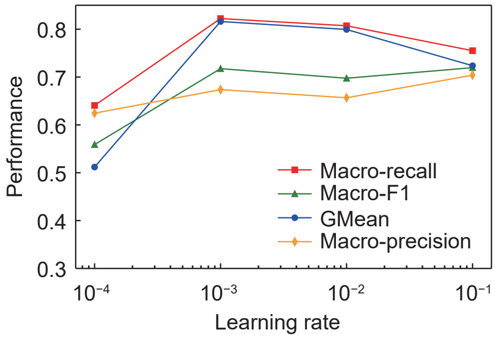

Fig. 13 Parameter sensitivity analysis for learning rate.

Figure 14 illustrates that a single-layered FraudGCN, which aggregates solely one-hop neighbors, successfully captures ample information. Multiple layers of FraudGCN are better suited for sparse graphs. As indicated in Fig. 14, augmenting the number of layers does not enhance the model's overall performance. When the number of layers is 3, the model's recall significantly decreases, indicating slightly poorer performance in fraud detection. Meanwhile, a single-layered model not only saves computational costs but also achieves better detection results.

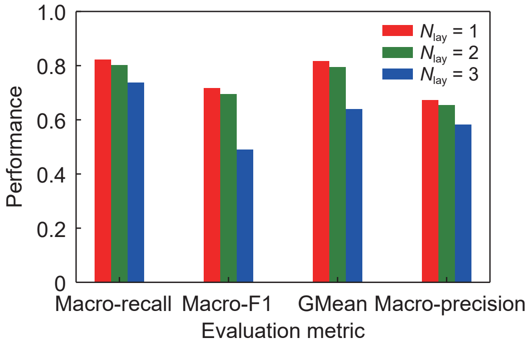

Fig. 14 Parameter sensitivity analysis for the number of layers.

4.6 Visualization

The node embeddings generated by various models, including FraudGCN, GCN, and GraphSAGE, are visualized to facilitate a comparative analysis of their performances. We use t-SNE[70] to project the acquired node embeddings onto a two-dimensional plane, thereby enabling an effective visualization.

For a streamlined and effective presentation, we only display the results of the test set. Figure 15 demonstrates the experimental results. GCN and GraphSAGE only employ a single relational graph. From Figs. 15a and 15b, it can be observed that the fraud and normal nodes are mixed together for both GCN and GraphSAGE. GraphSAGE performs slightly better than GCN. Notably, the fraud nodes are dispersed and predominantly isolated, encircled by numerous normal nodes. This may lead the model to predict fraudulent nodes towards the majority class, yielding a higher Macro-precision but lower Macro-recall. In our model, there is a clear boundary between fraud and normal nodes, and they are not mixed together as in other models. This result suggests that our model, which utilizes information from multiple relations, is adept at more precisely differentiating fraud nodes from normal ones.

Figure 16 illustrates the visualization of the three relational sub-graphs obtained by FraudGCN after performing intra-relation aggregation. We can observe that the separation between fraudulent and normal nodes is not very prominent across all three relationships. Nevertheless, subsequent to inter-relation aggregation, where information from the three relations is consolidated via attention mechanisms as depicted in Fig. 15b, the separation between fraud and normal nodes becomes significantly more pronounced, demonstrating an improved separation effect. This finding highlights that inter-relation aggregation can effectively utilize the neighborhood information from these three relations, enabling the model to aggregate a more diverse set of information and thus enhancing the effectiveness of node classification.

5 Conclusion and Future Work

This paper proposes a novel financial statement fraud

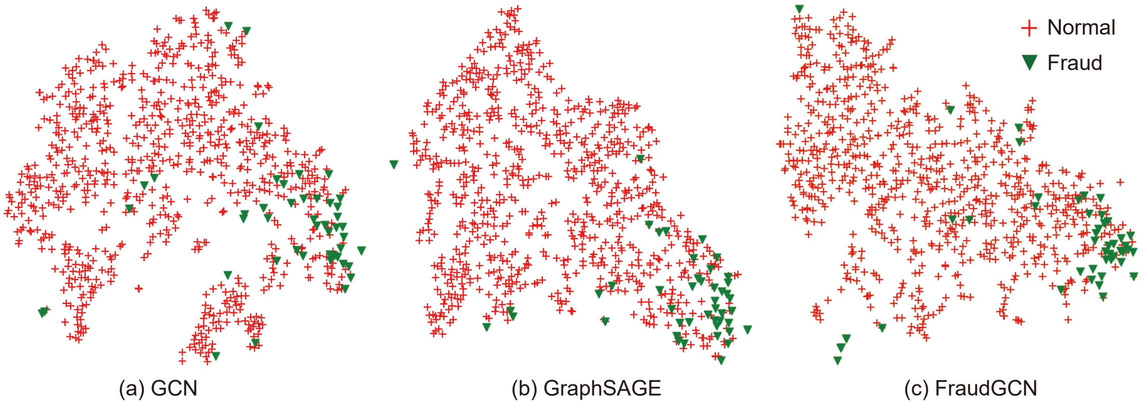

Fig. 15 Visualization of node embeddings. Red pluses and green triangles represent normal and fraud nodes, respectively.

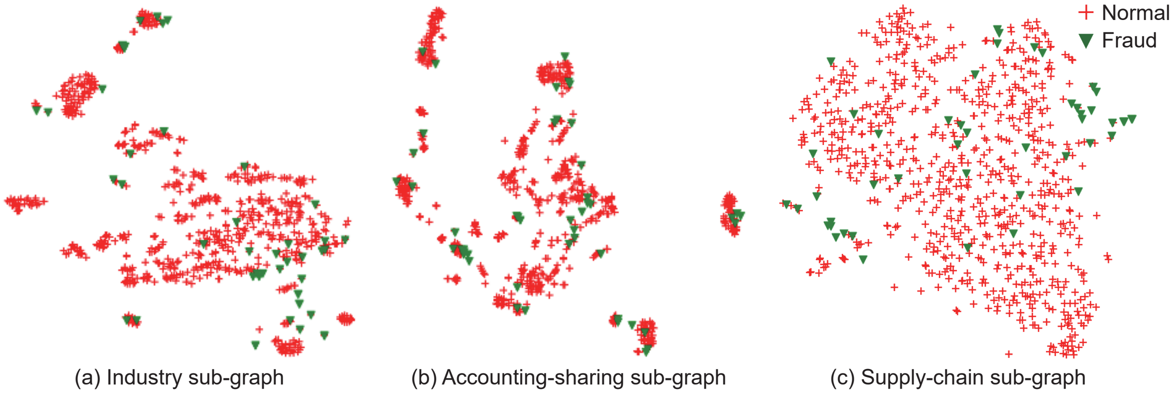

Fig. 16 Effects of different relational graphs. Red pluses and green triangles represent normal and fraud nodes, respectively.

detection model named FraudGCN based on a multi-relational graph. We construct three kinds of sub-graphs, namely the industry sub-graph, supply-chain sub-graph, and accounting-sharing sub-graph. We build a multi-relational graph neural network model for effective node representation learning, which aggregates various kinds of information effectively. Moreover, we propose a novel under-sampling method to select more important nodes from a global perspective for model training, which mitigates the class imbalance issue in the original training set. In addition, we incorporate focal loss to mitigate the challenges posed by class imbalance in the mini-batch training set. The efficacy of FraudGCN is rigorously validated through comprehensive experiments based on a real-world dataset.

This paper still has some limitations that warrant further investigation. Firstly, we would like to explore dynamic graphs to better capture the temporal information for financial statement fraud detection. Additionally, it is interesting to migrate our model to medium-sized enterprises. Nevertheless, the availability of data for medium-sized enterprises poses a conspicuous impediment to subsequent research endeavors.

Appendix

Acknowledgment

The research was supported by the National Natural Science Foundation of China (Nos. 62272379, T2341003, U22B2019, and 62102310), the Natural Science Basic Research Plan in Shaanxi Province (No. 2021JM-018), the Key R&D in Shaanxi Province (No. 2023-YBGY-269), and the Fundamental Research Funds for the Central Universities (No. xzy012023068).

References

[1] P. Ravisankar, V. Ravi, G. R. Rao, and I. Bose, Detection of financial statement fraud and feature selection using data mining techniques, *Decis. Support Syst.*, vol. 50, no. 2, pp. 491–500, 2011.

[2] S. Barman, U. Pal, A. Sarfaraj, B. Biswas, A. Mahata, and P. Mandal, A complete literature review on financial fraud detection applying data mining techniques, *Int. J. Trust Manage. Comput. Commun.*, vol. 3, no. 4, pp. 336–359, 2016.

[3] G. Niu, L. Yu, G. Z. Fan, and D. Zhang, Corporate fraud, risk avoidance, and housing investment in China, *Emerg. Mark. Rev.*, vol. 39, pp. 18–33, 2019.

[4] M. Jusup, P. Holme, K. Kanazawa, M. Takayasu, I. Romić, Z. Wang, S. Geček, T. Lipić, B. Podobnik, L. Wang, et al., Social physics, *Phys. Rep.*, vol. 948, pp. 1–148, 2022.

[5] L. G. A. Alves, H. Y. D. Sigaki, M. Perc, and H. V. Ribeiro, Collective dynamics of stock market efficiency, *Sci. Rep.*, vol. 10, no. 1, p. 21992, 2020.

[6] D. Fister, M. Perc, and T. Jagrič, Two robust long short-term memory frameworks for trading stocks, *Appl. Intell.*, vol. 51, no. 10, pp. 7177–7195, 2021.

[7] A. A. B. Pessa, M. Perc, and H. V. Ribeiro, Age and market capitalization drive large price variations of cryptocurrencies, *Sci. Rep.*, vol. 13, no. 1, p. 3351, 2023.

[8] B. Baesens, S. Höppner, and T. Verdonck, Data engineering for fraud detection, *Decis. Support Syst.*, vol. 150, p. 113492, 2021.

[9] K. G. Al-Hashedi and P. Magalingam, Financial fraud detection applying data mining techniques: A comprehensive review from 2009 to 2019, *Comput. Sci. Rev.*, vol. 40, p. 100402, 2021.

[10] P. S. Stanimirovic, Fraud detection in publicly traded us firms using beetle antennae search: A machine learning approach, *Expert Systems with Applications*, vol. 191, p. 116148, 2022.

[11] P. M. Dechow, W. Ge, C. R. Larson, and R. G. Sloan, Predicting material accounting misstatements, *Contemp. Account. Res.*, vol. 28, no. 1, pp. 17–82, 2011.

[12] P. Craja, A. Kim, and S. Lessmann, Deep learning for detecting financial statement fraud, *Decis. Support Syst.*, vol. 139, p. 113421, 2020.

[13] Z. Sabir, H. A. Wahab, S. Javeed, and H. M. Baskonus, An efficient stochastic numerical computing framework for the nonlinear higher order singular models, *Fractal Fract.*, vol. 5, no. 4, p. 176, 2021.

[14] Z. Sabir, K. Nisar, M. A. Z. Raja, A. A. B. A. Ibrahim, J. J. P. C. Rodrigues, K. S. Al-Basyouni, S. R. Mahmoud, and D. B. Rawat, Heuristic computational design of morlet wavelet for solving the higher order singular nonlinear differential equations, *Alex. Eng. J.*, vol. 60, no. 6, pp. 5935–5947, 2021.

[15] S. Yang, Z. Zhang, J. Zhou, Y. Wang, W. Sun, X. Zhong, Y. Fang, Q. Yu, and Y. Qi, Financial risk analysis for SMEs with graph-based supply chain mining, in *Proc. Twenty-Ninth Int. Joint Conf. on Artificial Intelligence*, Yokohama, Japan, 2021, p. 643.

[16] D. T. Ngaa, N. T. Le Thi Khanh Hoaa, P. Anha, T. V. Anha, L. P. Thaoa, and D. T. Haa, The impact of auditor's emotional intelligence and leadership style on audit quality: A, in *Proc. ICAEFM 2023*, Nha trang, Vietnam 2023, p. 124.

[17] R. Barandela, R. M. Valdovinos, J. S. Sánchez, and F. J. Ferri, The imbalanced training sample problem: Under or over sampling? in *Proc. Joint IAPR Int. Workshops*, Lisbon, Portugal, 2004, pp. 806–814.

[18] S. C. L. Koh, M. Demirbag, E. Bayraktar, E. Tatoglu, and S. Zaim, The impact of supply chain management

Table A1 Dimensions of ff.

ID	Financial raw data	ID	Financial raw data
A001101000	Cash at bank and on hand	B001000000	Income before tax
A001110000	Notes receivable	B002100000	Less: income tax
A001111000	Accounts receivable	B002000000	Net profit
A001112000	Advances to suppliers	B002000101	Net profit attributable to parent company owners
A001121000	Other receivables	B002000201	Minority interest profit/loss
A001123000	Inventory	B003000000	Basic earnings per share
A001125000	Other current assets	B004000000	Diluted earnings per share
A001100000	Total current assets	B006000000	Total comprehensive income
A001212000	Fixed assets	B006000101	Comprehensive income attributable to parent company owners
A001213000	Construction in progress	B006000102	Comprehensive income attributable to minority shareholders
A001218000	Intangible assets	B001207000	Taxes and additions
A001221000	Long-term prepaid expense	B001209000	Selling expense
A001222000	Deferred tax assets	B001210000	General and administrative expense
A001200000	Total non-current assets	B001211000	Financial expense
A001000000	Total attets	B001212000	Assets impairment losses
A002101000	Short-term borrowings	B001302000	Investment income
A002108000	Accounts receivable	B001300000	Operating income
A002109000	Advances to suppliers	B001400000	Add: non-operating income
A002112000	Employee benefits payable	B001500000	Less: non-operating expense
A002113000	Taxes payable	C001001000	Cash received from sales of goods or rendering of services
A002120000	Other payables	C001012000	Refund of tax and fee received
A002100000	Total current liabilities	C001013000	Other operating cash inflows
A002200000	Total non-current liabilities	C001014000	Cash paid for commodities or labor
A002000000	Total liabilities	C001020000	Cash paid to and on behalf of employees
A003101000	Paid-in capital	C001021000	Taxes and fees paid
A003102000	Capital surplus; additional paid-in capital	C001022000	Other cash paid relating to operating activities
A003103000	Surplus reserve	C001000000	Cash flows from operating activities
A003105000	Undistributed profits	C002003000	Cash from disposal of fixed assets, intangible assets, and otherlong-term assets
A003100000	Total equity attributable to parent company owners	C002006000	Cash paid to acquire fixed assets, intangible assets, Long-termassets
A003200000	Minority interests	C002000000	Cash flows from investing activities
A003000000	Total owners' equity	C003002000	Borrowings
A004000000	Total liabilities and owners' equity	C003005000	Cash paid for debt
B001100000	Total operating revenue	C003006000	Cash paid for dividend, profit, or interest
B001101000	Revenue	C003007000	Other cash paid related to financing activities
B001200000	Total operating costs	C003000000	Cash flows from financial activities/net cash provided byfinancing activities
B001201000	Cost of sales	C004000000	effect of exchange rate changes on cash and cash equivalents
C005000000	Net increase in cash and cash equivalents	D000109000	Finance costs/income
C005001000	cash and cash equivalents at the beginning of period	D000110000	Investment losses/gains

(To be continued)

Table A1 Dimensions of ff.

(Continued)

ID	Financial raw data	ID	Financial raw data
C006000000	cash and cash equivalents at the end of period	D000111000	Decrease/increase in deferred income tax assets
D000101000	Net profit	D000113000	Decrease/increase in inventory
D000102000	Asset impairment provision	D000114000	Decrease/increase in operating receivables
D000103000	Depreciation of fixed assets, depletion of oil and gas assets,amortization of productive biological assets	D000115000	Increase/decrease in operating payables
D000104000	Amortization of intangible assets	D000100000	Cash flows from operating activities
D000105000	Amortization of long-term prepaid expenses	D000204000	Ending cash balance
D000106000	Losses/gains from disposal of fixed assets, intangible assets,and other long-term assets	D000205000	Beginning cash balance
D000200000	Net increase in cash and cash equivalents		

practices on performance of SMEs, *Ind. Manage. Data Syst.*, vol. 107, no. 1, pp. 103–124, 2007.

[19] M. Abed and B. Fernando, E-commerce fraud detection based on machine learning techniques: Systematic literature review, *Big Data Mining and Analytics*, vol. 7, no.2, pp. 419–444, 2024.

[20] J. Perols, Financial statement fraud detection: An analysis of statistical and machine learning algorithms, *Audit. : A J. Pract. Theory*, vol. 30, no. 2, pp. 19–50, 2011.

[21] W. H. Beaver, Financial ratios as predictors of failure, *J. Account. Res.* vol. 4, no. 1, pp. 71–111, 1966.

[22] M. Cecchini, H. Aytug, G. J. Koehler, and P. Pathak, Detecting management fraud in public companies, *Manage. Sci.*, vol. 56, no. 7, pp. 1146–1160, 2010.

[23] S. Kotsiantis, E. Koumanakos, D. Tzelepis, V. Tampakas, Forecasting fraudulent financial statements using data mining, *Int. J. Comput. Intell.*, vol. 3, no. 2, pp. 104–110, 2006.

[24] H. C. Koh and C. K. Low, Going concern prediction using data mining techniques, *Manag. Audit. J.*, vol. 19, no. 3, pp. 462–476, 2004.

[25] C. Liu, Y. Chan, S. H. A. Kazmi, and H. Fu, Financial fraud detection model: Based on random forest, *Int. J. Econ. Finance*, vol. 7, no. 7, pp. 178–188, 2015.

[26] M. Cecchini, H. Aytug, G. J. Koehler, and P. Pathak, Making words work: Using financial text as a predictor of financial events, *Decis. Support Syst.*, vol. 50, no. 1, pp. 164–175, 2010.

[27] Y. Y. Chen, Forecasting financial distress of listed companies with textual content of the information disclosure: A study based MD&A in Chinese annual reports, (in Chinese), *Chin. J. Manage. Sci.*, vol. 27, no. 7, pp. 23–34, 2019.

[28] T. K. Hwang, W. C. Chen, W. C. Chiang, and Y. M. Li, Machine learning detection for financial statement fraud, in *Information Systems and Technologies*, A. Rocha, H. Adeli, G. Dzemyda, and F. Moreira, eds. Cham, Switzerland: Springer, 2022, pp. 148–154.

[29] A. Dyck, A. Morse, and L. Zingales, Who blows the whistle on corporate fraud? *J. Finance*, vol. 65, no. 6, pp. 2213–2253, 2010.

[30] P. Hajek and R. Henriques, Mining corporate annual reports for intelligent detection of financial statement fraud—A comparative study of machine learning methods, *Knowl. -Based Syst.*, vol. 128, pp. 139–152, 2017.

[31] J. L. Hobson, W. J. Mayew, and M. Venkatachalam, Analyzing speech to detect financial misreporting, *J. Account. Res.*, vol. 50, no. 2, pp. 349–392, 2012.

[32] W. Dong, S. Liao, and Z. Zhang, Leveraging financial social media data for corporate fraud detection, *J. Manage. Inf. Syst.*, vol. 35, no. 2, pp. 461–487, 2018.

[33] F. H. Chen, D. J. Chi, and J. Y. Zhu, Application of random forest, rough set theory, decision tree and neural network to detect financial statement fraud-taking corporate governance into consideration, in *Proc. 10th Int. Conf. on Intelligent Computing*, Taiyuan, China, 2014, pp. 221–234.

[34] G. Ozdagoglu, A. Ozdagoglu, Y. Gumus, and G. Kurt Gumus, The application of data mining techniques in manipulated financial statement classification: The case of turkey, *J. AI Data Mining.*, vol. 5, no. 1, pp. 67–77, 2017.

[35] A. A. Rizki, I. Surjandari, and R. A. Wayasti, Data mining application to detect financial fraud in Indonesia's public companies, in *Proc. 2017 3rd Int. Conf. on Science in Information Technology*, Bandung, Indonesia, 2017, pp. 206–211.

[36] X. B. Tang, G. C. Liu, J. Yang, and W. Wei, Knowledge-based financial statement fraud detection system: Based on an ontology and a decision tree, *Knowl. Org.*, vol. 45, no. 3, pp. 205–219, 2018.

[37] Y. Bao, B. Ke, B. Li, Y. J. Yu, and J. Zhang, Detecting accounting fraud in publicly traded U. S. firms using a machine learning approach, *J. Account. Res.*, vol. 58, no. 1, pp. 199–235, 2020.

[38] X. Wu and S. Du, An analysis on financial statement fraud detection for Chinese listed companies using deep learning, *IEEE Access*, vol. 10, pp. 22516–22532, 2022.

[39] Z. Sabir, M. A. Z. Raja, A. S. Alnahdi, M. B. Jeelani, and

M. A. Abdelkawy, Numerical investigations of the nonlinear smoke model using the Gudermannian neural networks, *Math. Biosci. Eng*, vol. 19, no. 1, pp. 351–370, 2022.

[40] Z. Sabir, M. A. Z. Raja, J. L. G. Guirao, and T. Saeed, Meyer wavelet neural networks to solve a novel design of fractional order pantograph lane-emden differential model, *Chaos Solitons Fractals*, vol. 152, p. 111404, 2021.

[41] Z. Sabir, M. A. Z. Raja, H. A. Wahab, M. Shoaib, and J. F. G. Aguilar, Integrated neuro-evolution heuristic with sequential quadratic programming for second-order prediction differential models, *Numer. Methods Part. Differ. Equations*, vol. 40, no. 1, p. e22692, 2024.

[42] K. Nisar, Z. Sabir, M. A. Z. Raja, A. A. A. Ibrahim, F. Erdogan, M. R. Haque, J. J. P. C. Rodrigues, and D. B. Rawat, Design of morlet wavelet neural network for solving a class of singular pantograph nonlinear differential models, *IEEE Access*, vol. 9, pp. 77845–77862, 2021.

[43] Z. Sabir, Neuron analysis through the swarming procedures for the singular two-point boundary value problems arising in the theory of thermal explosion, *Eur. Phys. J. Plus*, vol. 137, no. 5, p. 638, 2022.

[44] Z. Sabir, T. Botmart, M. A. Z. Raja, R. Sadat, M. R. Ali, A. A. Alsulami, and A. Alghamdi, Artificial neural network scheme to solve the nonlinear influenza disease model, *Biomed. Signal Process. Control*, vol. 75, p. 103594, 2022.

[45] J. Zhou, G. Cui, S. Hu, Z. Zhang, C. Yang, Z. Liu, L. Wang, C. Li, and M. Sun, Graph neural networks: A review of methods and applications, *AI Open*, vol. 1, pp. 57–81, 2020.

[46] H. Yang, AliGraph: A comprehensive graph neural network platform, in *Proc. 25th ACM SIGKDD Int. Conf. on Knowledge Discovery* & *Data Mining*, Anchorage, AK, USA, 2019, pp. 3165–3166.

[47] Z. Liu, C. Chen, X. Yang, J. Zhou, X. Li, and L. Song, Heterogeneous graph neural networks for malicious account detection, in *Proc. 27th ACM Int. Conf. on Information and Knowledge Management*, Torino, Italy, 2018, pp. 2077–2085.

[48] B. Feng, H. Xu, W. Xue, and B. Xue, Every corporation owns its structure: Corporate credit rating via graph neural networks, in *Proc. 5th Chinese Conf. on Pattern Recognition and Computer Vision*, Shenzhen, China, 2022, pp. 688–699.

[49] Z. Liu, Y. Dou, P. S. Yu, Y. Deng, and H. Peng, Alleviating the inconsistency problem of applying graph neural network to fraud detection, in *Proc. 43rd Int. ACM SIGIR Conf. on Research and Development in Information Retrieval*, New York, NY, USA, 2020, pp. 1569–1572.

[50] X. Mao, H. Sun, X. Zhu, and J. Li, Financial fraud detection using the related-party transaction knowledge graph, *Procedia Comput. Sci.*, vol. 199, pp. 733–740, 2022.

[51] J. Gasteiger, S. Weißenberger, and S. Günnemann, Diffusion improves graph learning, in *Proc. 33rd Int. Conf. on Neural Information Processing Systems*, Vancouver, Canada, 2019, p. 1197.

[52] K. Hassani and A. H. Khasahmadi, Contrastive multi-view representation learning on graphs, in *Proc. 37th Int. Conf. on Machine Learning*, Virtual Event, 2020, pp. 4116–4126.

[53] L. Page, S. Brin, R. Motwani, and T. Winograd, The PageRank citation ranking: bringing order to the web, *Stanford Digital Libraries Working Paper*, doi: 10.1007/978-3-319-08789-4_10.

[54] W. L. Hamilton, Z. Ying, and J. Leskovec, Inductive representation learning on large graphs, in *Proc. 31st Conf. on Neural Information Processing Systems*, Long Beach, CA, USA, 2017, pp. 1024–1034.

[55] P. Christoffersen and K. Jacobs, The importance of the loss function in option valuation, *J. Financ. Econ.*, vol. 72, no. 2, pp. 291–318, 2004.

[56] U. Ruby and V. Yendapalli, Binary cross entropy with deep learning technique for image classification, *Int. J. Adv. Trends Comput. Sci. Eng*, vol. 9, no. 4, pp. 5393–5397, 2020.

[57] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, et al., Scikit-learn: Machine learning in python, *J. Mach. Learn. Res.*, vol. 12, pp. 2825–2830, 2011.

[58] W. Caesarendra, A. Widodo, and B. S. Yang, Application of relevance vector machine and logistic regression for machine degradation assessment, *Mech. Syst. Signal Process.*, vol. 24, no. 4, pp. 1161–1171, 2010.

[59] W. Tong, H. Hong, H. Fang, Q. Xie, and R. Perkins, Decision forest: Combining the predictions of multiple independent decision tree models, *J. Chem. Inf. Comput. Sci.*, vol. 43, no. 2, pp. 525–531, 2003.

[60] L. Y. Hu, M. W. Huang, S. W. Ke, and C. F. Tsai, The distance function effect on k-nearest neighbor classification for medical datasets, *SpringerPlus*, vol. 5, no. 1, p. 1304, 2016.

[61] V. Y. Kulkarni and P. K. Sinha, Pruning of random forest classifiers: A survey and future directions, in *Proc. 2012 Int. Conf. on Data Science* & *Engineering*, Cochin, India, 2012, pp. 64–68.

[62] X. Zhao, Z. Ma, and M. Yin, Using support vector machine and evolutionary profiles to predict antifreeze protein sequences, *Int. J. Mol. Sci.*, vol. 13, no. 2, pp. 2196–2207, 2012.

[63] T. Chen and C. Guestrin, XGBoost: A scalable tree boosting system, in *Proc. 22nd ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining*, San Francisco, CA, USA, 2016, pp. 785–794.

[64] G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, and T. Y. Liu, LightGBM: A highly efficient gradient boosting decision tree, in *Proc. 31st Int. Conf. on Neural Information Processing Systems*, Red Hook, CA, USA, 2017, pp. 3149–3157.

[65] J. H. Friedman, Greedy function approximation: A gradient boosting machine, *Ann. Statist.*, vol. 29, no. 5, pp. 1189–1232, 2001.

[66] T. N. Kipf and M. Welling, Semi-supervised classification with graph convolutional networks, in *Proc. 5th Int. Conf. on Learning Representations*, Toulon, France, 2017, doi: 10.48550/arXiv.1609.02907.

[67] Y. Dou, Z. Liu, L. Sun, Y. Deng, H. Peng, and P. S. Yu, Enhancing graph neural network-based fraud detectors against camouflaged fraudsters, in *Proc. 29th ACM Int. Conf. on Information & Knowledge Management*, Virtual Event, 2020, pp. 315–324.

[68] M. Sokolova and G. Lapalme, A systematic analysis of performance measures for classification tasks, *Inf. Process. Manage.*, vol. 45, no. 4, pp. 427–437, 2009.

[69] Y. Liu, X. Ao, Z. Qin, J. Chi, J. Feng, H. Yang, and Q. He, Pick and choose: A GNN-based imbalanced learning approach for fraud detection, in *Proc. Web Conf. 2021*, Ljubljana, Slovenia, 2021, pp. 3168–3177.

[70] L. van der Maaten and G. Hinton, Visualizing data using t-SNE, *J. Mach. Learn. Res.*, vol. 9, no. 86, pp. 2579–2605, 2008.

Chenxu Wang received the BEng degree in communication engineering and the PhD degree in control science and engineering from Xi'an Jiaotong University, China in 2009 and 2015, respectively. He was a postdoctoral researcher at Hong Kong Polytechnic University, China from 2016 to 2017. He is currently an associate professor at School of Software Engineering, Xi'an Jiaotong University, China. His current research interests include graph data mining, fraud detection, complex network analysis, and network representation learning.

Yi Long received the PhD degree in engineering from The University of Hong Kong, China in 2015, and then worked as a postdoctoral researcher in FinTech at The Chinese University of Hong Kong, China. He has rich experience in financial data mining, and has published dozens of academic papers in internationally renowned conferences and journals. He is currently the co-founder and CEO of Datago Technology Limited (Shenzhen), and an adjunct professor at Shenzhen Finance Institute, The Chinese University of Hong Kong (Shenzhen), China. His current research interests include financial data mining and financial risk analysis.

Mengqin Wang received the BEng degree in computer science and technology from Sichuan University, China in 2022. Currently, she is a master student at School of Software Engineering, Xi'an Jiaotong University, China. Her current research focuses on financial fraud detection.

Xiaoguang Wang received the MEng degree in finance from Xi'an Jiaotong University, China in 2021. He is currently a PhD candidate at School of Software Engineering, Xi'an Jiaotong University, China. His research focuses on financial fraud detection and graph neural networks.

Luyue Zhang received the BEng degree in information management and information systems from Central University of Finance and Economics, China in 2021. She is currently a master student at School of Software Engineering, Xi'an Jiaotong University, China. Her current research focuses on financial fraud detection.