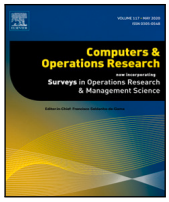
Computers & Operations Research 169 (2024) 106738

Contents lists available at ScienceDirect

Computers and Operations Research

journal homepage: www.elsevier.com/locate/cor



LG-GNN: Local and Global Information-aware Graph Neural Network for default detection

Yi Liu [a], Xuan Wang [a], Tao Meng [b,*], Wei Ai [b], Keqin Li [c]

[a] College of Finance and Statistics, Hunan University, Hunan, 410082, China
[b] College of Computer and Mathematics, Central South University of Forestry and Technology, Hunan, 410004, China
[c] Department of Computer Science, State University of New York, New Paltz, NY 12561, USA

ARTICLE INFO

Keywords:
Default detection
Graph neural network
Contrastive learning
Credit scoring

ABSTRACT

Default detection, a crucial aspect of individual credit scoring, has attracted considerable attention in research. Previous approaches typically focus on classifying applicants using only explicit attributes, overlooking the importance of latent relations among them. Concurrently, graph-based techniques have emerged as promising tools for credit scoring. However, existing graph-based methods often need to be more accurate in aggregating information from limited neighbors, which can lead to misclassification when the target node has differently labeled neighbors. Motivated by these challenges, we propose a Local and Global Information-aware Graph Neural Network (LG-GNN) approach for default detection. By leveraging the loan applicant relation graph, LG-GNN dynamically learns the representation of the target node from both local and global perspectives. Furthermore, it adaptively fuses the information from these perspectives and employs contrastive learning to enhance feature variations. Extensive experiments demonstrate the superiority of LG-GNN over mainstream methods across several widely used default detection datasets. Specifically, LG-GNN achieves an average relative performance improvement of 47.9% compared to baselines. Moreover, compared to the most competitive default detection methods, LG-GNN exhibits an average performance improvement of 11.9%. Our code is publicly available at https://github.com/BERA-wx/LG-GNN.

1. Introduction

Default detection, which is a process that involves the use of machine learning techniques to predict the likelihood of individuals or entities defaulting on their credit obligations, is an essential aspect of decision-making for consumer finance firms, commercial banks, and financial institutions (West, 2000; Crook, 2002). It involves analyzing an individual's financial status, credit history, and capability to repay a loan for making informed credit approval or denial decisions.

Approved applicants are denoted as approved samples, while those whose applications are denied are termed rejected samples. The lending institution can track the behavior of the approved samples throughout the loan term and classify timely repayments as non-default samples and instances of default behavior as default samples. The goal of default detection is to predict which applicants will become default samples. Traditional statistical methods like Logistic Regression (LR) (Sohn et al., 2016) and Linear Discriminant Analysis (LDA) (Altman, 1968) are frequently used due to their simplicity and interpretability. However, these methods face challenges when dealing with increasing data diversity and populations, which has led to the popularity of advanced machine learning techniques such as Back Propagation (BP) Neural Networks, K Nearest Neighbor (KNN), Support Vector Machines (SVM), and Multi-Layer Perceptron (MLP) for modeling default detection. Conventional machine learning methods for default detection rely heavily on manual feature engineering and expert knowledge to extract informative feature sets. On the other hand, deep learning methods automatically extract features using various end-to-end paradigms (Scarselli et al., 2008), resulting in the popularity of deep learning methods in default detection (Mancisidor et al., 2020; Dahooie et al., 2021; Zhang et al., 2022). For example, TSSMN (Guo et al., 2023) leverages the transitivity of labels in graph neural networks (GNNs) to infer loan status directly. Furthermore, previous research typically focuses on detecting defaulters using explicit attributes. However, nearly every applicant is connected to others through diverse and far-reaching connections, such as friendships, kinships, and other latent relations. Therefore, data for default detection is better represented using graphs to simulate latent relations (Wu et al., 2024; Shi et al., 2024).

* Corresponding author.
E-mail addresses: yiliu@hnu.edu.cn (Y. Liu), xuanw@hnu.edu.cn (X. Wang), mengtao@hnu.edu.cn (T. Meng), aiwei@hnu.edu.cn (W. Ai), lik@newpaltz.edu (K. Li).

https://doi.org/10.1016/j.cor.2024.106738
Received 28 November 2023; Received in revised form 23 April 2024; Accepted 12 June 2024
Available online 20 June 2024
0305-0548/© 2024 Elsevier Ltd. All rights are reserved, including those for text and data mining, AI training, and similar technologies.

Table 1
The mean values of the debt-to-income ratio (DTI-Ratio) and employment length (EL) of the defaulters and the non-defaulters in the Lending Club dataset across different years.

Year	DTI-Ratio (%)		EL (year)	
	Default	Non-Default	Default	Non-Default
2013	18.66	16.90	6.20	6.21
2014	19.55	17.52	6.03	6.08
2015	20.87	18.37	5.90	5.97
2016	20.45	18.11	5.85	6.02
2017	20.21	17.93	5.56	5.89
2018	19.10	17.87	5.64	5.98

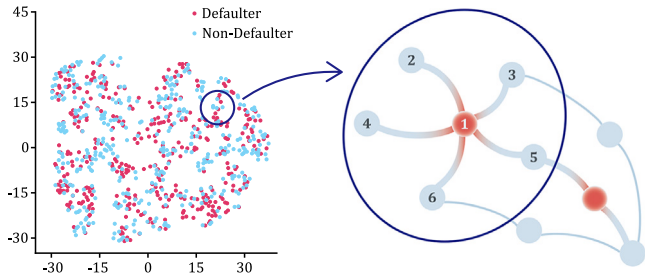


Fig. 1. Limited feature variability and context inconsistency. Limited feature variability is characterized by the constrained difference in features between defaulters and non-defaulters, illustrated by a substantial overlap of blue and red sample points in the left plot. Context inconsistency manifests as an abundance of non-defaulters among the neighbors of defaulters. In the case illustrated in the right plot, Node 1 is categorized as a defaulter, while its first-order neighbors (Nodes 2 to 6) are all non-defaulters.

Vanilla GNNs have two aspects that merit consideration. Firstly, the sole reliance on nearby information can lead to homogenization (Zhu et al., 2020). Herein, the target node continuously assimilates information from its neighbors, diminishing differences. Homogenization helps when neighbors belong to the same category but hinders when they belong to a different class. Secondly, neighbor-information aggregation is distance-dependent (Garg et al., 2020), causing rapid attenuation of contributions from distant neighbors. Consequently, the influence of far-distance neighbors diminishes quickly, facilitating short-distance information learning but impeding long-distance information learning.

Moreover, vanilla GNNs encounter two challenges in default detection. Firstly, there is limited feature variability between default and non-default samples, as defaulters align their features with non-defaulters to meet loan requirements, limiting the generalization ability of graph models. Table 1 in the Lending Club dataset shows slight differences in average *debt-to-income ratio* and *employment length* between defaulters and non-defaulters. Additionally, Fig. 1 reveals significant overlap in data distribution, indicating minimal differences between the two groups. Secondly, there is an issue of context inconsistency in default detection graphs. Vanilla GNNs find it hard to represent the node when its nearby neighbors belong to different categories. Homogenization and distance dependency aggravate the issue, limiting feature variability and hindering learning from long-distance neighbors. Context inconsistency often arises when defaulters' neighbors are mainly non-defaulters since they outnumber defaulters. Node 1 in Fig. 1 is an example of this, which prevents it from obtaining second-order neighbors classified similarly to it. Considering these challenges, vanilla GNNs that heavily rely on nearby neighbor information need help accurately identifying defaulters in default detection.

Accordingly, we developed LG-GNN, a GNN that combines local and global perspectives to represent target nodes more comprehensively. Simultaneously, we employ a contrastive learning strategy to facilitate the aggregation of similar samples, reducing the impact of context inconsistency and emphasizing feature disparities among different class samples. LG-GNN first acquires node-embedded representations through an embedding layer and then constructs the loan applicant relation graph. Subsequently, it constructs local and global information filtering networks to obtain local and global representations of target nodes. The different representations are combined by dynamically taking a weighted average. Ultimately, the target node features are updated with the help of contrastive learning, yielding the final representation.

Overall, LG-GNN has proven effective across seven different datasets. The results demonstrate significant performance in identifying potential defaulters within the loan applicants. Evaluated using the common Kolmogorov–Smirnov (KS) metric and compared to traditional classifiers such as LR, MLP, and XGB, LG-GNN exhibited a relative average performance increase of 97.15%. Compared to the most competitive default detection methods, LG-GNN showed an average performance improvement of 32.85%.

Our main contributions can be summarized as follows:

- We propose a novel LG-GNN method that integrates local and global information within a graph data structure. Additionally, we enhance the GNN model with contrastive learning to improve graph representation learning.
- Our approach effectively mines latent relations from personal pre-loan attributes using local and global networks. These latent relations are incorporated into the learning process to enhance node representation, allowing for effective discrimination even in context inconsistency.
- Experiments conducted on seven datasets under various conditions consistently demonstrate the effectiveness of LG-GNN compared to mainstream methods for default detection.

The remainder of this paper is structured as follows. Section 2 provides an overview of recent related research. Section 3 describes the default detection task. Section 4 elaborates on our LG-GNN framework. In Section 5, we empirically assess LG-GNN's effectiveness through experiments. Finally, in Section 7, we summarize our work and explore potential directions for future research.

2. Related work

In this section, we review some works on default detection and two important related research aspects: graph neural networks and contrastive learning.

2.1. Default detection

In default detection, prediction performance mainly relies on high-quality applicant features and advanced techniques. Various studies analyze discriminant information, including intrinsic features and external data, for accurate default prediction. Intrinsic features are the inherent characteristics of an individual that can be used to predict default. Research such as Liang et al. (2016) and Jones (2017) identified crucial intrinsic features for corporate default prediction. Similarly, Xiong et al. (2013) determinant personal features for default prediction. As for external data, Ma et al. (2018b) utilized phone usage data, and Sukharev et al. (2020) employed individual transaction records for personal default detection. The external information serves the purpose of discovering helpful knowledge in credit analysis and could also be beneficial complements in default detection.

Some researchers aim to develop advanced detection methods. There are three categories of mainstream default detection methods: expert scoring, statistical techniques, and machine learning. Expert scoring involves evaluating the creditworthiness of an applicant based on relevant factors. Conventional statistical methods such as LR and PR are simple and transparent but face challenges when dealing with extensive and diverse data sets. As more powerful predictions require more information and better predictors (Ohlson, 1980), machine learning techniques such as KNN, MLP, and gradient boosting (Jones, 2017) have become popular methods for default detection. Moreover, there

is a growing trend toward utilizing semi-supervised (Xiao et al., 2020), counterfactual (Bueff et al., 2022), and multi-task learning (Liu et al., 2022) methods to model default detection. However, conventional machine learning methods for default detection rely heavily on manual feature engineering and expert knowledge to extract informative feature sets.

Recent studies have demonstrated that Graph Neural Networks (GNNs) (Scarselli et al., 2008), a type of deep learning model, are highly effective in processing extensive, diverse, and high-dimensional data. In contrast to traditional machine learning algorithms, which assume that instances are independent of one another, default detection requires a deeper understanding of the complex relationships between instances. Graphs are a more appropriate way to represent default detection data due to their expressive power and ability to capture intricate connections between instances, such as friendship and kinship. As a result, researchers are actively exploring the potential of GNNs in default detection to improve credit assessment accuracy. For instance, they investigate integrated graph representation learning methods (Shi et al., 2024) and heterogeneous GNNs (Wu et al., 2024).

Despite the proficiency of existing machine and deep learning models, there are still two pivotal challenges: limited feature diversity and context inconsistency.

2.2. Graph neural networks

Graph Neural Networks (GNNs) (Scarselli et al., 2008) aggregate neighboring node features iteratively to model target node representations in graphs, ensuring stable node states. This inherent characteristic and strong interpretability (Ying et al., 2019) have made GNNs a powerful tool for various applications, including credit scoring (Guo et al., 2023; Shi et al., 2024) and fraud detection (Lakhan et al., 2022).

Diverse methods have been proposed to elucidate the predictions of GNN-based models, such as XGNN (Yuan et al., 2020), GNNExplainer (Ying et al., 2019), and SubgraphX (Yuan et al., 2021). These methods offer transparent insights into the decision-making process, enabling users to understand the rationale behind GNN predictions and fostering trust in their outcomes.

The Graph Attention Network (GAT) (Veličković et al., 2017) refines message passing by selectively aggregating features from first-order neighbors onto target nodes using attention coefficients. This approach enhances interpretability as the attention mechanism directly reflects the importance of different regions within the graph (Ying et al., 2019). However, GAT employs static attention coefficients, limiting its adaptability. To address this, GATv2 (Brody et al., 2021) introduced dynamic attention. Similarly, GraphSage (Hamilton et al., 2017) improves dynamic graph node feature learning by extending GCN to incorporate generalized aggregation functions, achieved through sampling and inductive learning from neighbors. However, GATs and GraphSage to learn target node representations are limited by relying solely on information from nearby neighbors (Zhu et al., 2020; Garg et al., 2020).

Recent works, such as LGD-GCN (Guo et al., 2022), LGM-GNN (Li et al., 2023b), and LGGNet (Ding et al., 2023), introduce an innovative approach to acquiring target nodes' representations through the integration of local and global perspectives. The method helps to overcome the limitation of distance in obtaining information. These methods integrate neighbors' information with different distances by aggregating filtered information from local and global perspectives through concatenation. However, these methods face two issues in default detection. First, excessive information filtering leads to a loss of useful information. Second, the concatenation approach may overly emphasize the effectiveness of the original features, hindering the learning of node information where original features are less critical in specific scenarios.

2.3. Contrastive learning

Unsupervised contrastive learning (UCL) (Hadsell et al., 2006) is a popular idea that has been successful in various domains, such as image classification (Wang et al., 2022; Li et al., 2023a), emotion recognition (Yang et al., 2023; Shen et al., 2022), and computer vision (Chen et al., 2020). It achieves representation learning by creating similar and dissimilar instances and then learning a representation model that ensures similar instances are projected closely in the embedding space while dissimilar instances are projected farther apart. To achieve this, maximizing mutual information is typically used (Hjelm et al., 2018). One of the examples of UCL is Deep InfoMax (Hjelm et al., 2018), which maximizes the mutual information between input data and high-level feature vectors for effective feature representations.

Supervised contrastive learning (SCL) (Khosla et al., 2020) is an extension of unsupervised contrastive methods. It incorporates label information to construct positive and negative samples, which take samples with the same label as positive pairs to calculate contrastive loss, thus facilitating the model to find the decision boundaries (Gao et al., 2021; Gunel et al., 2020). The process involves two stages: first, learning representations through a contrastive loss and then training a classifier using cross-entropy loss.

Several studies have explored the integration of contrastive learning with the graph learning process, resulting in the creation of graph contrastive learning (GCL) (Zhu et al., 2021; You et al., 2020). This approach has proven particularly beneficial when capturing graph structure and node attribute information. In the context of default detection, contrastive learning can utilize structure information, attribute information, and loan status labels from the relation graph formed by loan applicants, which can help capture the characteristics of different types of applicants and lead to more accurate inferences about their loan status.

3. Problem statement

All approved samples are denoted as a set of samples $\mathcal{A} = \{a_1, \ldots, a_N\}$, where N signifies the total number of approved samples. Each approved sample a_i corresponds to a feature vector $x_i \in \mathbb{R}^d$, where d is the feature dimensionality of approved samples. These approved samples can be further categorized into two subsets: default samples, indicated as $x_i^d \in \mathbb{R}^d$, and non-default samples, denoted as $x_i^n \in \mathbb{R}^d$. For each approved sample a_i, the ground-truth label for its default/non-default status is assigned, where $y_i \in \{0, 1\}$. Specifically, $y_i = 0$ represents non-default samples (i.e., $x_i^n \in \mathbb{R}^d$), while $y_i = 1$ corresponds to default samples (i.e., $x_i^d \in \mathbb{R}^d$). Therefore, the default detection can be formally framed as a binary classification problem, where the model is required to perform well on approved samples for default loan prediction.

4. Methodology

4.1. The design of the LG-GNN architecture

This section aims to provide an in-depth explanation of the LG-GNN architecture's design. As depicted in Fig. 2, the framework of the LG-GNN model proposed in this paper is depicted. The LG-GNN model is composed of seven components: (1) Features embedding: Using an embedding method, we transform high-dimensional, sparse feature vectors of loan applicants into low-dimensional, dense feature vectors. This enhances the model's learning efficiency by more effectively capturing applicant characteristics; (2) Loan applicant relation graph construction: We construct a loan applicant relation graph to capture inter-applicant relationships and feature similarities; (3) Local information filtering: Introducing an information filtering mechanism for local neighbors helps eliminate interference from dissimilar neighbor nodes in the context of partial context inconsistency, thus refining target node

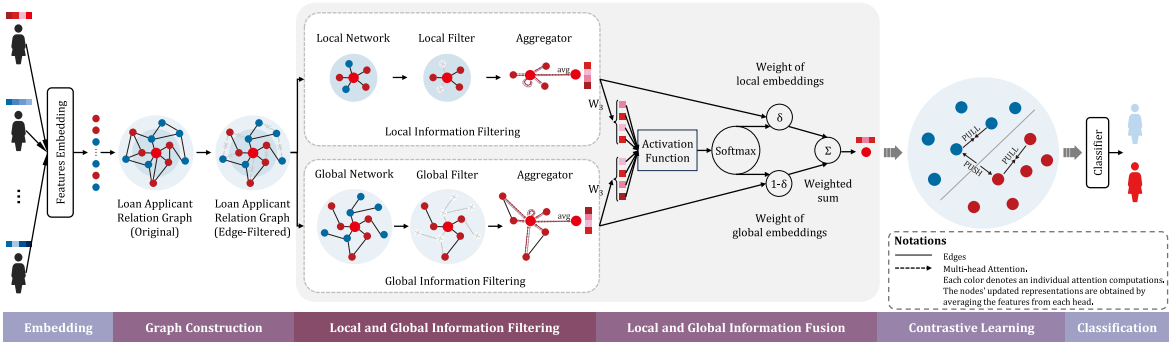


Fig. 2. Detailed architecture of the proposed LG-GNN. Firstly, it transforms the raw features of applicants into low-dimensional dense vectors through feature embedding. Secondly, a loan applicant relation graph is constructed based on the similarity of applicant features, serving as an initial noise filter. Thirdly, to address the challenge of context inconsistency, the model employs a graph to specify the top-k values for local and global information filtering, further refining noise filtering. Fourthly, an adaptive fusion mechanism aggregates the filtered local and global information, facilitating the update of node features. Fifthly, a loss optimization strategy based on contrastive learning is applied to tackle the limited feature variability issue. Finally, an MLP-based classifier is employed for loan status inference.

feature learning; (4) Global information filtering: From a global perspective, we comprehensively capture target node features to mitigate limitations arising from context inconsistency. Simultaneously, we filter helpful information for target node updates from global neighbors, reducing noise interference; (5) Local and global information fusion: Adaptive weight learning is employed to aggregate locally and globally acquired information for the target node; (6) Default/non-default contrastive learning: Contrastive learning is introduced to amplify feature differences between default and non-default samples; (7) Default/non-default classification: This component is dedicated to inferring loan applicants' loan status.

4.2. Features embedding

This section embeds representations of loan applicants' original features to map high-dimensional discrete features into lower-dimensional continuous vectors. The goal is to preserve relationships while reducing dimensionality. First, given the coexistence of discrete and continuous features and the unsuitability of continuous features for direct embedding processing, it is imperative to discretize continuous features before embedding operations. Following discretization, the robustness of features to anomalous data is significantly strengthened (Liu et al., 2002). Simultaneously, the discretization operation enhances model stability and mitigates overfitting risks by reducing the model's excessive reliance on features with larger weights (Chapelle et al., 2014). Therefore, we conduct discretization on continuous features before proceeding with embedding operations. For convenience, the equal-width-based binning (Holte, 1993) is employed. Subsequently, an embedding layer is utilized to encode the discretized feature vectors. Specifically, the embedding layer transforms the feature vector $x_i \in \mathbb{R}^d$ of each node into $emb_i \in \mathbb{R}^{d\times k}$, where k represents the embedding dimension. Following this transformation, for the ease of passing the embedded features to subsequent computational networks and to prevent overfitting, we flatten the embedded features to one dimension, i.e., $emb_i \in \mathbb{R}^{dk}$. Consequently, after the features embedding, all sample features, initially denoted as $\mathcal{A} \in \mathbb{R}^{N\times d}$, are transformed into $E \in \mathbb{R}^{N\times dk}$.

4.3. Loan applicant relation graph construction

The loan applicant relation graph, denoted as $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$, is an undirected graph that aims to capture complex relationships among loan applicants. Nodes in the graph represent embedding representations of both default and non-default samples, i.e., $emb_i \in \mathcal{V}$. Edges between nodes are constructed based on the similarity between according node pairs, with similar nodes connected to form the original graph. Considering the density of edges is crucial for effective representation learning of nodes in graphs and the problem of limited feature variability and context inconsistency, we incorporate an edge threshold, p, to control edge density and primarily filter useless relationship information. When the similarity between nodes is high enough, nodes are likelier to belong to the same type. Therefore, edges remain only if node similarity exceeds the threshold, p. The original graph is optimized for connectivity and meaningful information propagation, followed by an edge-filtered graph. The process of constructing edges is as follows:

$$e_{ij} = \begin{cases} 1 & sim(emb_i, emb_j) > p \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

where $e_{ij} \in \mathbb{R}$ ($e_{ij} \in \mathcal{E}$) indicates the presence or absence of an edge between the node pairs. The similarity function, denoted as $sim(\cdot)$, is chosen as the cosine similarity function.

4.4. Local information filtering

In constructing the loan applicant relation graph, we initially perform rudimentary clustering based on numerical feature similarity using a threshold p. However, due to context inconsistency, this results in nodes of different types within similar groups. Therefore, precise information filtering becomes crucial when aggregating information for target nodes. We approach information filtering by mining information on applicant features.

We focus on local information filtering by considering first-order neighbors, as their influence on target nodes is significant. The first-order neighbors comprise the local neighbors whose associations collectively constitute the local network.

To measure the influence of neighbors on target nodes, we introduce an attention coefficient. This coefficient, denoted as $r_{ij}^{(l,k)}$, is computed using a two-layer approach incorporating linear transformations and an activation function. A multi-head mechanism is implemented to enhance stability in the learning process. The attention coefficient indicates the relevance of information between node pairs, with a higher value indicating greater relevance. The attention coefficient $r_{ij}^{(l,k)}$ between node v_i and node v_j in the lth graph layer and kth head is calculated as follows:

$$r_{ij}^{(l,k)} = (\mathbf{W}_1^{(l,k)})^\top (leakyrelu(\mathbf{W}_2^{(l,k)}[\phi_i^{(l)} \oplus \phi_j^{(l)}])) \tag{2}$$

where $\mathbf{W}_1^{(l,k)}$ and $\mathbf{W}_2^{(l,k)}$ are learnable parameters, $r_{ij}^{(l,k)} \in \mathbb{R}$, and $(\cdot)^\top$ denotes transposition. The symbol $\oplus$ represents the concatenation operation. When $l = 1$, $\phi_i^{(l)} = emb_i$ and $\phi_j^{(l)} = emb_j$.

Subsequently, we use the $softmax$ function to normalize attention coefficients between node pairs to ensure a consistent comparison. This normalization process yields the normalized local coefficient, $\omega_{ij,l}^{(l,k)} \in \mathbb{R}$,

in the local network. It focuses solely on the target nodes and their first-order neighbors. The normalization formula is as follows:

$$\omega_{ij,l}^{(l,k)} = \sigma(r_{ij}^{(l,k)}) = \frac{\exp(r_{ij}^{(l,k)})}{\sum_{j\in\mathcal{N}_{i,l}} \exp(r_{ij}^{(l,k)})} \tag{3}$$

where $\mathcal{N}_{i,l}$ represents the first-order neighbors of node v_i, including v_i, obtained by masking attention coefficients based on the first-order adjacency matrix. A larger $\omega_{ij,l}^{(l,k)}$ indicates stronger relevance and higher similarity, and $\sigma(\cdot)$ denotes the softmax function.

Secondly, we employ a top-k-based node importance filtering method to identify relatively important neighbors of a target node on the local network, guided by the normalized attention coefficient. This operation effectively screens out unimportant local neighbors and eliminates noisy local information. The selection criteria are defined as follows:

$$\begin{aligned} score_{ij,l}^{(l,k)}, idx_{ij,l}^{(l,k)} &= TopK(\omega_{ij,l}^{(l,k)}, K_{local}) \\ p_l^{(l,k)} &= \min(score_{ij,l}^{(l,k)}) \\ L_{ij}^{(l,k)} &= \begin{cases} \omega_{ij,l}^{(l,k)} & \omega_{ij,l}^{(l,k)} > p_l^{(l,k)} \\ 0 & \text{otherwise} \end{cases} \end{aligned} \tag{4}$$

where $score_{ij,l}^{(l,k)}$ and $idx_{ij,l}^{(l,k)}$ are the scores and corresponding indexes after $TopK$ pooling and filtering. The hyperparameter K_{local} is the ratio for filtering the first-order neighbors of the target node. $L_{ij}^{(l,k)}$ represents the selected first-order neighbors, and $p_l^{(l,k)}$ is the corresponding local threshold. Information from a neighbor node contributes to the target node's feature update only when $\omega_{ij,l}^{l,k}$ exceeds $p_l^{(l,k)}$; otherwise, it is disregarded during feature aggregation.

Thirdly, we update the target node's local feature embedding using a graph neural network (GNN) with *elu* activation function. This process entails computing a weighted linear combination of corresponding feature vectors, where the weights are determined by the associated attention coefficients $L_{ij}^{(l,k)}$. The update is expressed through the following formula:

$$Emb_{i,l}^{(l)} = elu\left(\frac{1}{K}\sum_{k=1}^{K}\sum_{j\in\mathcal{N}_{i,l}} L_{ij}^{(l-1,k)}\mathbf{W}_L^{(l-1,k)} Emb_{i,l}^{(l-1)}\right) \tag{5}$$

where K represents the number of multi-attention layers, and $\mathbf{W}_L^{(l-1,k)} \in \mathbb{R}^{dk\times h_l^k}$ corresponds to the weight matrix for input linear transformations, with h_l^k indicating the dimensionality of the lth layer in the kth head. The term $Emb_{i,l}^{(l)} \in \mathbb{R}^{h_l^k}$ denotes the local representation of node v_i, when $l = 1$, $Emb_{i,l}^{(l)} = emb_i$.

4.5. Global information filtering

Due to context inconsistency and the constrained filtering capacity of local information filtering, local neighbors may still include nodes with different labels, leading to several issues. Firstly, solely aggregating information from local neighbors overlooks effective microscopic information from distant neighbors with the same label and information hidden in the macroscopic structure of the entire graph. Secondly, when the features of the target node and local neighbors with different labels share limited variability, the local information filter fails to effectively filter out interfering information from such neighbors. This results in exacerbating feature similarities between the target node and local neighbors with different labels. Additionally, the local information filter is prone to overfitting under interfering information and features from applicants in the context.

To address these concerns, we introduce global information filtering to enrich the representation of the target node by aggregating information comprehensively. Adopting a global viewpoint, we consider the impact of all neighbors on the target node, with all nodes of the graph collectively forming the global neighbors and their associations forming the global network. Utilizing attention coefficients, $r_{ij}^{(l,k)}$, we gauge the importance of information transmitted from global neighbors to the target node during information aggregation. The calculation of $r_{ij}^{(l,k)}$ for all node pairs is derived according to Eq. (2). To facilitate comparisons, a $softmax$ function is applied to the global neighbors, resulting in $\omega_{ij,g}^{(l,k)} \in \mathbb{R}$. The formula is as follows:

$$\omega_{ij,g}^{(l,k)} = \sigma(r_{ij}^{(l,k)}) = \frac{\exp(r_{ij}^{(l,k)})}{\sum_{j\in\mathcal{N}_{i,g}} \exp(r_{ij}^{(l,k)})}, \tag{6}$$

where $\mathcal{N}_{i,g}$ denotes the global neighbors of node v_i, obtained directly from the attention coefficients rather than based on high-order adjacency matrices to enhance the computational efficiency.

To improve generalization and prediction accuracy across varying data distributions of different samples, we employ a parameter normalization method. We compute a scaled parameter score $\beta_{ij,g}^{(l,k)} \in \mathbb{R}$ to measure node pair similarity, which involves a linear transformation using the *relu* function, followed by min–max scaling to constrain it within the $(0, 1)$ range, as illustrated in the formula below:

$$\begin{aligned} \alpha_{ij,g}^{(l,k)} &= relu(\left\|\rho(x_i) - \rho(emb_j)\right\|_1), j \in \mathcal{N}_{i,g} \\ \beta_{ij,g}^{(l,k)} &= \frac{\alpha_{ij,g}^{(l,k)} - \min(\alpha_{ij,g}^{(l,k)})}{\max(\alpha_{ij,g}^{(l,k)}) - \min(\alpha_{ij,g}^{(l,k)})} \end{aligned} \tag{7}$$

here, $\|\cdot\|_1$ calculates the $L1$ norm of the corresponding vector and $\rho(\cdot)$ calculates the $L2$ norm of the corresponding vector. $\beta_{ij,g}^{(l,k)}$ is the scaled parameter score, with $\min(\alpha_{ij,g}^{(l,k)})$ as the minimum value and $\max(\alpha_{ij,g}^{(l,k)})$ as the maximum value. Since the *sigmoid* function saturates at extreme values, we employ the min–max method to scale the values to the specified range while preserving differences instead of directly applying the *sigmoid* function.

Then, by combining the attention coefficients between nodes and the normalized scores, we compute the final score, $\gamma_{ij,g}^{(l,k)} \in \mathbb{R}$, for global neighbors:

$$\gamma_{ij,g}^{(l,k)} = \frac{1}{2}\left(\omega_{ij,g}^{(l,k)} + \left(1 - \beta_{ij,g}^{(l,k)}\right)\right) \tag{8}$$

Next, we filter the global neighbors based on the final scores, selecting those highly similar to the target node. These filtered global neighbors are considered to possess crucial information at the global level. We continue to use the top-k-based operation for filtering:

$$\begin{aligned} score_{ij,g}^{(l,k)}, idx_{ij,g}^{(l,k)} &= TopK(\omega_{ij,g}^{(l,k)}, K_{global}) \\ p_g^{(l,k)} &= \min(score_{ij,g}^{(l,k)}) \\ s_{ij,g}^{(l,k)} &= \begin{cases} \omega_{ij,g}^{(l,k)} & \omega_{ij,g}^{(l,k)} > p_g^{(l,k)} \\ 0 & \text{otherwise} \end{cases} \end{aligned} \tag{9}$$

here, $s_{ij,g}^{(l,k)} \in \mathbb{R}$ represents the ultimately chosen global neighbors, and K_{global} denotes the ratio for filtering the global neighbors of the target node with $p_g^{(l,k)}$ as the corresponding global threshold. Information from a neighbor node contributes to the target node's feature update only when $\omega_{ij,g}^{l,k}$ exceeds $p_g^{(l,k)}$; otherwise, it is disregarded during feature aggregation.

To evaluate the significance of the chosen global neighbors, we employ a softmax normalization process defined as follows:

$$G_{ij}^{(l,k)} = \frac{\sigma(s_{ij,g}^{(l,k)})}{t} = \frac{\exp(s_{ij,g}^{(l,k)})}{t\sum_{j\in E_i} \exp(s_{ij,g}^{(l,k)})} \tag{10}$$

here, $G_{ij}^{(l,k)} \in \mathbb{R}$ represents the weight of the last selected nodes, and t represents the global temperature coefficient, and in our experiment, we set its value to 0.001. To ensure numerical stability during the softmax operation, particularly when $s_{ij,g}^{(l,k)}$ values are close to zero, we introduce a temperature coefficient to control the smoothness, which helps maintain stability during the computation.

Subsequently, we employ the GNN framework to aggregate information from the global neighbor nodes and the target node, obtaining the global feature embedding for the target node:

$$Emb_{i,g}^{(l)} = elu\left(\frac{1}{K}\sum_{k=1}^{K}\sum_{j\in\mathcal{N}_{i,g}} G_{ij}^{(l-1,k)}\mathbf{W}_G^{(l-1,k)}Emb_{i,g}^{(l-1)}\right) \tag{11}$$

where $\mathbf{W}_G^{(l-1,k)} \in \mathbb{R}^{dk\times h_l^k}$ corresponds to the weight matrix for input linear transformations. The term $Emb_{i,g}^{(l)} \in \mathbb{R}^{h_l^k}$ denotes the global representation of node v_i, when $l = 1$, $Emb_{i,g}^{(l)} = emb_i$.

4.6. Local and global information fusion

So far, we have obtained two distinct representations of node v_i: the local representation, $Emb_{i,l}^{(l)}$, and the global representation, $Emb_{i,g}^{(l)}$. Our current task is to merge these two representations to obtain a more comprehensive node representation. However, due to the issue of inconsistency, the importance of local and global information varies among different nodes. Therefore, we need to dynamically learn the weights that balance the roles of local and global information in information fusion, and we employ the attention mechanism. The calculation of the weight balancing parameter is as follows:

$$\delta_i^{(l)} = \sigma(leakyrelu(\mathbf{W}_3^{(l)}[Emb_{i,l}^{(l)} \oplus Emb_{i,g}^{(l)}])) \tag{12}$$

where $\delta_i^{(l)} \in \mathbb{R}^{h_l^k}$ represents the weight of the local representation of node v_i, and $\mathbf{W}_3^{(l)} \in \mathbb{R}^{2dk\times h_l^k}$ represent the shared weight matrix.

Finally, we perform local and global information fusion based on the learned weights, $\delta_i^{(l)}$:

$$\phi_i^{(l)} = \delta_i^{(l)} Emb_{i,l}^{(l)} + \left(1-\delta_i^{(l)}\right) Emb_{i,g}^{(l)} \tag{13}$$

where $\phi_i^{(l)} \in \mathbb{R}^{h_l^k}$ represents the final representation of node v_i.

4.7. Default/non-default contrastive learning

In light of the challenges posed by the limited distinctions between default samples and non-default samples, accurately classifying such nodes has become a complex task. To amplify the differentiation among nodes with different loan statuses, we leverage contrastive learning. This approach aims to extract semantic information from nodes with diverse loan statuses to bring nodes sharing the same loan status labels into proximity while pushing apart nodes with dissimilar loan status labels. Contrastive learning, in turn, amplifies the disparities among features of different node categories. Within the realm of contrastive learning, we designate node pairs with identical loan status labels as positive pairs (i.e., $(emb_i^d, emb_j^d \mid v_i, v_j \in \mathcal{V}, i \neq j)$ or $(emb_i^n, emb_j^n \mid v_i, v_j \in \mathcal{V}, i \neq j)$) and those with different loan status labels as negative pairs (i.e., $(emb_i^d, emb_j^n \mid v_i, v_j \in \mathcal{V})$). The default/non-default contrastive network is formulated as follows:

$$\mathcal{L}_1 = -\mathbb{E}_s\left[\sum_{v_{j(i)}} \in \mathcal{V} \log \frac{\exp\left(sim\left(\phi_i^{(l)}, \phi_{j(i)}^{(l)}\right)/\tau\right)}{\sum_{v_z\in\mathcal{V}}\exp\left(sim\left(\phi_i^{(l)}, \phi_z^{(l)}\right)/\tau\right)}\right], \tag{14}$$

where $sim(\cdot)$ represents the similarity function, $\tau \in \mathbb{R}^+$ is a scalar temperature parameter. The index i denotes the sub-index of the anchor node v_i, and the index $j(i)$ is the sub-indices of the node with the same label. The index z denotes the sub-index of other nodes except the anchor node.

4.8. Default/non-default classification

Ultimately, we employ the default/non-default classification network to infer the loan status and compute default and non-default sample probabilities. This classification network comprises multiple linear layers integrated with an activation function. The first layer engages in the multiplication of the ultimate representations, $\phi_i^{(l)} \in \mathbb{R}^{h_l}$, with a weight matrix $\mathbf{W}_4^{(1)} \in \mathbb{R}^{h_l\times d_1}$, followed by activation through the rectified linear unit (ReLU) function. This layer yields the latent representation of the first layer, $p_{i,1} \in \mathbb{R}^{d_1}$, where d_t signifies the dimensionality of the tth layer. Apart from the final layer, the remaining layers iteratively update the latent representation in the same way as the first layer as

$$p_{i,t} = relu(p_{i,t-1}\mathbf{W}_4^{(t-1)}), \tag{15}$$

where $p_{i,t} \in \mathbb{R}^{d_t}$. When $t = 1$, $p_{i,1} = \phi_i^{(l)}$.

For the final layer, if we denote the number of layers as T, we will have $d_T = 2$. The final output of the default/non-default classification network will be the non-default probability and the default probability

$$p_{i,T} = \sigma(p_{i,T-1}\mathbf{W}_4^{(T)}), \tag{16}$$

where $p_{i,T} \in \mathbb{R}^2$. The first dimension, $p_{i,T}^{(0)}$, illustrates the probability of non-default and the second dimension, $p_{i,T}^{(1)}$, illustrates the probability of default.

4.9. Model training

For contrastive learning, given the default/non-default label y_i, we have contrastive loss defined as Eq. (14).

For the prediction of default and non-default labels, when provided with the default/non-default label y_i and the output from the default/non-default classifier, $p_{i,T}$, we obtain the following:

$$\mathcal{L}_2 = -\sum_{i=1}^{N} p_{i,T}^{(0)} log(y_i) + p_{i,T}^{(1)} log(1-y_i). \tag{17}$$

With η as a hyperparameter to represent the weight of contrastive loss and $(1-\eta)$ representing prediction loss, respectively, we denote the overall loss function as:

$$\mathcal{L} = \eta\cdot\mathcal{L}_1 + (1-\eta)\cdot\mathcal{L}_2. \tag{18}$$

5. Experiments

In this section, we commence by introducing pertinent datasets, followed by the presentation of empirical experiments to affirm the efficacy of the LG-GNN model.

5.1. Datasets

In our experiments, we validate our proposed LG-GNN by testing it on seven publicly available real-world datasets.

5.1.1. Lending club datasets

Lending Club, a prominent global credit lending company, provides a rare, publicly available real-world dataset.[1] This dataset, known as the Lending Club dataset,[2] includes four primary features: Fico score, debt-to-income ratio, loan amount, and employment length. Notably, the debt-to-income ratio is computed as $\frac{MonthlyIncome}{LoanAmount}$, signifying that it is a derived variable from the loan amount. To explore distinctions between default and non-default samples, we present the four features for defaulters and non-defaulters in the dataset across various years in Table 1. We have derived three separate subdatasets, Lending1, Lending2, and Lending3, from the 2013, 2014, and 2015 Lending Club datasets, respectively.

[1] https://www.lendingclub.com/

[2] https://www.kaggle.com/wordsforthewise/lending-club

Table 2
Performance comparison on Lending Club datasets, evaluated by AUC (%) and KS (%). Average values are also listed.

Type	Approach	Lending1		Lending2		Lending3		Average	
		AUC	KS	AUC	KS	AUC	KS	AUC	KS
Baseline	LR	59.94	13.53	60.52	15.81	61.67	17.43	60.71	15.59
	MLP	59.96	13.77	60.50	15.72	61.54	17.53	60.67	15.67
	XGB	59.93	13.62	60.54	15.69	61.34	17.18	60.60	15.50
Semi-Supervised Learning	ST+LR	60.18	14.02	60.49	15.73	61.87	17.86	60.85	15.87
	ST+MLP	60.13	13.89	60.54	15.89	61.71	17.78	60.79	15.85
	ST+XGB	60.19	14.11	60.51	15.86	61.83	17.89	60.84	15.95
	SS-GMM	60.09	13.75	60.55	15.78	62.06	18.24	60.90	15.92
Counterfactual Learning	IPS+LR	60.21	13.45	60.36	15.52	61.53	17.59	60.70	15.52
	IPS+MLP	60.25	14.58	60.44	16.00	61.59	17.63	60.76	16.07
	DR+LR	60.18	13.69	60.32	15.36	61.37	17.41	60.62	15.49
	DR+MLP	60.22	14.34	60.48	15.72	61.68	17.72	60.79	15.93
	DRJL+LR	60.28	14.53	60.41	15.52	61.42	17.58	60.70	15.88
	DRJL+MLP	60.25	14.59	60.54	16.10	61.73	17.81	60.84	16.17
	ACL+LR	60.31	14.46	60.49	15.75	61.79	17.88	60.86	16.03
	ACL+MLP	60.34	14.63	60.59	15.86	61.85	17.94	60.93	16.14
	SRDO+LR	60.24	14.03	60.56	15.74	61.58	17.59	60.79	15.79
	SRDO+MLP	60.13	13.59	60.39	15.82	61.47	17.47	60.66	15.63
Multi-Task Learning	Cross-Stitch	60.33	14.51	60.76	16.54	62.17	18.56	61.09	16.54
	MMOE	60.24	14.63	60.62	16.17	62.09	18.31	60.98	16.37
	PLE	60.32	14.41	60.71	16.45	61.97	18.04	61.00	16.30
	RMT-Net	60.61	15.35	61.02	17.08	62.48	18.97	61.37	17.13
Ours	LG-GNN	**61.38***	**25.17***	**62.27***	**25.42***	**62.92***	**23.52***	**62.19***	**24.70***

* Denotes statistically significant improvement, measured by t-test with p-value < 0.01, over the second-best approach on each dataset.

Table 3
Performance comparison on Lending Club datasets, evaluated by Acc. (%) and F1 (%). Average values are also listed.

Type	Approach	Lending1		Lending2		Lending3		Average	
		Acc.	F1	Acc.	F1	Acc.	F1	Acc.	F1
Baseline	LR	55.60	51.82	56.46	57.70	57.40	52.73	56.49	54.08
	MLP	55.80	55.36	56.90	59.23	57.55	59.60	56.75	58.06
	XGB	55.83	52.34	56.25	59.20	57.96	58.31	56.68	56.62
Semi-Supervised Learning	ST+LR	56.01	53.21	57.33	58.92	58.23	58.73	57.19	56.95
	ST+MLP	55.87	52.95	57.21	59.33	57.94	57.42	57.01	56.57
	ST+XGB	56.14	54.08	57.27	57.85	58.07	59.18	57.16	57.04
	SS-GMM	55.63	52.73	57.45	59.76	57.71	57.89	56.93	56.79
Counterfactual Learning	IPS+LR	56.27	56.64	56.78	59.73	57.34	60.23	56.80	58.87
	IPS+MLP	55.74	55.18	56.92	60.22	58.62	61.15	57.09	58.85
	DR+LR	55.92	55.92	56.57	60.65	56.98	59.77	56.49	58.78
	DR+MLP	56.42	54.75	56.62	60.17	57.77	60.06	56.94	58.33
	DRJL+LR	56.15	56.01	56.83	61.24	57.05	59.39	56.68	58.88
	DRJL+MLP	56.03	55.88	56.71	60.41	56.87	59.92	56.54	58.74
	ACL+LR	55.73	53.97	56.55	59.81	57.23	61.64	56.50	58.47
	ACL+MLP	56.36	56.45	56.94	59.98	58.11	60.88	57.14	59.10
	SRDO+LR	55.68	54.23	56.61	60.37	56.75	60.51	56.35	58.37
	SRDO+MLP	55.86	55.07	56.75	60.79	57.93	60.02	56.85	58.63
Multi-Task Learning	Cross-Stitch	56.75	57.11	57.14	61.93	58.39	62.05	57.43	60.36
	MMOE	56.63	57.25	57.29	61.82	57.93	62.20	57.28	60.42
	PLE	56.55	57.37	57.22	62.04	58.21	62.13	57.33	60.51
	RMT-Net	56.76	57.49	57.35	62.29	**58.96**	62.22	57.69	60.67
Ours	LG-GNN	**56.80***	**57.68***	**57.50***	**62.37***	58.90	**62.25***	**57.73***	**60.77***

* Denotes statistically significant improvement, measured by t-test with p-value < 0.01, over the second-best approach on each dataset.

5.1.2. Home datasets

Home Credit[3] is a consumer finance company in Eastern Europe and Asia, providing financial services and offering the Home dataset,[4] a real-world credit scoring dataset. In default detection, the quantity of features in a dataset significantly impacts the distinction between defaulters and non-defaulters. More features create more pronounced differences in their distributions, facilitating easier differentiation. To model diverse feature distributions, we generated two datasets: Home1, which includes all features from the Home dataset, and Home2, which uses only 50% of the features. The two datasets allow us to assess how varying feature quantities affect default detection model performance.

5.1.3. PPD datasets

Ppdai[5] is a consumer financial technology brand offering short-term lending services to individuals and businesses. It also provides open access to a rich dataset of personal credit histories, the PPD[6] dataset. Similarly, we generated two datasets to capture diverse feature distributions: PPD1, encompassing all features from the PPD dataset,

[3] https://www.homecredit.net/
[4] https://www.kaggle.com/c/home-credit-default-risk
[5] https://www.ppdai.com/
[6] https://www.heywhale.com/home/competition/56cd5f02b89b5bd026cb39c9/content/1

and PPD2, utilizing only 50% of the features. The two datasets permit us to evaluate how varying feature quantities impact the performance of default detection models.

Overall, there are seven datasets: Lending1, Lending2, Lending3, Home1, Home2, PPD1, and PPD2. Additionally, We have removed samples with missing values from all datasets, and we randomly allocated 80% for training, 10% for validation, and 10% for testing in each dataset.

5.2. Experimental setting and implementation

In our LG-GNN, we empirically set the following hyperparameters for different datasets: a learning rate of 0.0005 and an input feature embedding dimensionality of 8. For the GNN part, we utilized two multi-heads and configured the hidden layer dimensionality as [128, 64]. The batch size was set to 1000 for LendingClub and Home datasets and 2000 for PPD datasets. The training lasted for 1000 epochs, with each epoch completing in approximately 0.0527 s for LendingClub, 0.0284 s for Home, and 0.1240 s for PPD datasets. We employed the Adam optimization algorithm to update network parameters for effective convergence. Other methods' hyperparameters were optimized based on validation set performance. Early stopping, guided by validation set performance, was applied for all methods. Each method underwent ten runs, randomly selecting a balanced dataset with the specified batch size. Reported results represent average values from these runs. Experiments were conducted on a GeForce RTX 4090 GPU server with 24 GB of memory, powered by an Intel Core 13700KF CPU, running Windows 11. Programming was done in Python 3.9, using PyTorch version 2.0.0 for developing and implementing deep learning algorithms.

5.3. Metrics

We incorporated two additional evaluation metrics, accuracy (Acc.) and F1-score (F1), alongside the commonly used metrics for default detection research and prediction, namely the area under the receiver operating characteristic curve (AUC) and the Kolmogorov–Smirnov statistic (KS).

1. AUC is an indicator that evaluates the classification performance of positive and negative classes and can be used to describe the model's overall performance.
2. KS evaluates the model's risk discrimination ability by measuring the difference between the cumulative distribution of true positive rate and false positive rate and can be used to determine the threshold for approving loan applications.
3. Accuracy reflects the proportion of correctly classified instances among all instances, providing a comprehensive measure of model performance.
4. F1-score balances precision and recall, offering insight into the model's overall accuracy in predicting positive instances while considering false positives and negatives.

5.4. Compared approaches

To assess the performance of LG-GNN, we compared four categories of methods across seven datasets. These categories encompass baseline, semi-supervised learning methods, counterfactual learning methods, and multi-task learning methods. The specific methods are as follows:

- Baselines. This category includes three commonly employed credit assessment classification methods: Logistic Regression (LR), Multilayer Perceptron (MLP), and XGBoost (XGB).
- Semi-supervised Learning Methods: In this category, we primarily focus on methods that incorporate Self-Training (ST) (Maldonado and Paredes, 2010) with LR, MLP, and XGB. Additionally, we will conduct a comparative analysis with the widely adopted semi-supervised rejection inference method, SS-GMM (Mancisidor et al., 2020).
- Counterfactual Approaches: Here, our primary emphasis is on methods that integrate IPS (Schnabel et al., 2016; Swaminathan and Joachims, 2015), DR (Jiang and Li, 2016), DRJL (Wang et al., 2019), ACL (Xu et al., 2020), and SRDO (Shen et al., 2020) with LR and MLP.
- Multi-Task Approaches: For multi-task approaches, we will consider Cross-Stitch (Misra et al., 2016), MMOE (Ma et al., 2018a), PLE (Tang et al., 2020) and RMT-Net (Liu et al., 2022).

This structured evaluation will provide a comprehensive understanding of our experimental results compared to these diverse categories of methods, aiding in a thorough assessment of our proposed LG-GNN.

5.5. Result analysis

First, Table 2 and Table 3 illustrates the performance across LendingClub1, LendingClub2, and LendingClub3 datasets. Notably, employing semi-supervised learning, counterfactual learning, multi-task learning, and LG-GNN yields substantial enhancements over baseline models. On the LendingClub1 dataset, LG-GNN relatively improves baselines by 84.53% and improves the second-best compared approach by 63.97%, evaluated by KS. On the LendingClub2 dataset, LG-GNN exhibits a notable 61.50% improvement over baselines, outperforming the second-best method by 48.83%, according to the KS. Similarly, on the LendingClub3 dataset, LG-GNN achieves a significant 35.33% improvement over baselines, surpassing the second-best method by 23.99%, according to the KS. Based on the KS metric, LG-GNN showcases an outstanding 58.46% enhancement over the baseline model, outperforming the second-best method by 44.18%, highlighting its efficacy in improving feature distinctions and addressing context inconsistency. Additionally, across the LendingClub datasets, LG-GNN showed 2.52%, 12.36%, and 1.93% improvements in the AUC, F-score, and ACC metrics, respectively, compared to the baseline models. Compared to the second-best method, LG-GNN showed enhancements of 1.34%, 0.16%, and 0.08% in the AUC, F-score, and ACC metrics, respectively.

Second, Table 4 and Table 5 displays a performance comparison across the Home1, Home2, PPD1, and PPD2 datasets. Like the LendingClub datasets, employing semi-supervised learning, counterfactual learning, multi-task learning, and LG-GNN yields substantial improvements over baseline models, reinforcing the effectiveness of these methods in default detection. LG-GNN consistently outperforms its counterparts. Notably, on the Home1 dataset, LG-GNN exhibits an impressive 48.33% improvement over baselines, surpassing the second-best method by 19.57%, according to the KS. On the Home2 dataset, LG-GNN showcases a remarkable 415.19% improvement, outperforming the second-best method by 29.02%, according to the KS. For the PPD1 dataset, LG-GNN achieves a substantial 98.36% improvement over baselines, surpassing the second-best method by 25.15%, according to the KS. On the PPD2 dataset, LG-GNN achieves a noteworthy 186.84% improvement compared to baselines, surpassing the second-best method by 33.03%, according to the KS. In summary, LG-GNN exhibits an outstanding 133.46% enhancement over the baseline model based on the KS metric, outperforming the second-best method by 26.51%. Moreover, the LG-GNN model demonstrated notable enhancements of 18.05%, 25.76%, and 9.62% in the AUC, F-score, and ACC metrics, respectively, when compared to the baseline models. Compared to the second-best method, the LG-GNN model showed enhancements of 1.73%, 1.18%, and 1.44% in the AUC, F-score, and ACC metrics, respectively. These results underscore LG-GNN's ability to

Table 4
Performance comparison on Home datasets and PPD datasets, evaluated by AUC (%) and KS (%). Average values are also listed.

Type	Approach	Home1		Home2		PPD1		PPD2		Average	
Type	Approach	AUC	KS	AUC	KS	AUC	KS	AUC	KS	AUC	KS
Baseline	LR	68.05	26.46	54.87	7.42	62.83	19.88	58.80	13.41	61.14	16.79
	MLP	67.72	25.82	55.17	7.63	60.80	16.54	58.63	12.75	60.58	15.69
	XGB	67.81	26.07	55.60	8.12	63.21	20.93	59.32	14.19	61.49	17.33
Semi-Supervised Learning	ST+LR	67.88	27.71	66.12	23.96	66.22	23.90	64.57	21.11	66.20	24.17
	ST+MLP	67.93	27.56	66.37	24.40	64.35	21.28	64.87	21.40	65.88	23.66
	ST+XGB	67.77	27.23	66.60	24.58	66.58	24.71	64.79	21.22	66.44	24.44
	SS-GMM	68.59	27.71	66.21	23.61	67.12	25.93	64.50	20.96	66.61	24.55
Counterfactual Learning	IPS+LR	67.24	24.99	66.26	24.43	69.20	28.43	63.73	19.98	66.61	24.46
	IPS+MLP	67.93	25.87	66.37	24.59	69.88	29.32	63.87	20.26	67.01	25.01
	DR+LR	67.66	25.23	65.72	24.14	69.66	29.64	63.49	19.95	66.63	24.74
	DR+MLP	67.92	25.93	65.87	24.20	69.43	29.39	64.26	21.19	66.87	25.18
	DRJL+LR	68.42	27.11	66.67	24.61	69.31	29.10	64.20	21.27	67.15	25.52
	DRJL+MLP	68.68	27.97	66.30	24.42	69.57	29.71	64.59	21.58	67.29	25.92
	ACL+LR	67.81	25.09	65.97	23.56	68.49	27.54	62.86	19.11	66.28	23.83
	ACL+MLP	67.91	25.40	66.59	24.23	69.30	27.69	63.69	19.77	66.87	24.27
	SRDO+LR	68.26	26.13	66.14	24.53	69.47	28.60	63.41	19.74	66.82	24.75
	SRDO+MLP	67.58	25.49	65.98	24.20	69.62	28.49	63.66	19.90	66.71	24.52
Multi-Task Learning	Cross-Stitch	66.96	23.81	64.33	21.96	66.71	26.63	60.93	17.06	64.73	22.37
	MMOE	63.55	22.49	56.31	14.39	66.09	24.53	56.17	10.54	60.53	17.99
	PLE	63.90	21.51	57.63	15.70	65.89	24.61	54.62	9.16	60.51	17.75
	RMT-Net	71.99	32.40	71.03	30.84	71.00	30.30	69.44	29.00	70.87	30.64
Ours	LG-GNN	**72.67***	**38.74***	**72.26***	**39.79***	**71.87***	**37.92***	**71.57***	**38.58***	**72.09***	**38.76***

* Denotes statistically significant improvement, measured by t-test with p-value < 0.01, over the second-best approach on each dataset.

Table 5
Performance comparison on Home datasets and PPD datasets, evaluated by Acc. (%) and F1 (%). Average values are also listed.

Type	Approach	Home1		Home2		PPD1		PPD2		Average	
		Acc.	F1	Acc.	F1	Acc.	F1	Acc.	F1	Acc.	F1
Baseline	LR	63.70	46.05	60.60	60.47	49.95	15.38	52.05	35.50	56.58	39.35
	MLP	65.10	63.03	60.30	55.65	62.00	57.24	58.85	50.53	61.56	56.61
	XGB	64.67	61.82	63.75	57.32	61.63	61.25	60.47	61.38	62.63	60.44
Semi-Supervised Learning	ST+LR	64.32	62.35	62.31	62.05	63.29	61.86	63.12	60.79	63.26	61.76
	ST+MLP	64.98	62.11	61.48	62.45	63.95	62.73	61.89	60.12	63.08	61.85
	ST+XGB	63.75	62.77	62.97	62.33	64.17	60.92	62.76	61.25	63.41	61.82
	SS-GMM	64.21	61.98	63.12	61.88	63.72	62.11	61.33	61.07	63.10	61.76
Counterfactual Learning	IPS+LR	63.48	62.64	61.89	63.10	63.83	61.09	61.02	61.91	62.56	62.19
	IPS+MLP	64.89	62.19	62.04	63.67	64.59	62.45	62.58	62.38	63.53	62.67
	DR+LR	63.51	62.53	61.68	61.93	64.27	62.37	62.39	60.59	62.96	61.86
	DR+MLP	64.76	61.76	63.55	63.40	64.88	61.68	63.81	61.94	64.25	62.20
	DRJL+LR	63.43	62.92	61.71	61.24	63.41	61.32	61.65	61.09	62.55	61.64
	DRJL+MLP	63.95	61.67	62.22	62.87	63.62	62.64	61.21	60.44	62.75	61.91
	ACL+LR	64.53	62.26	63.06	62.61	64.78	60.58	60.88	60.27	63.31	61.43
	ACL+MLP	63.22	62.49	63.54	62.01	64.35	62.98	62.11	61.73	63.31	62.30
	SRDO+LR	64.01	62.03	60.89	61.36	63.05	62.05	62.34	61.40	62.57	61.71
	SRDO+MLP	64.63	62.84	63.66	63.18	64.47	60.29	63.72	61.12	64.12	61.86
Multi-Task Learning	Cross-Stitch	63.87	62.08	60.67	63.01	63.18	60.77	63.59	60.91	62.83	61.69
	MMOE	61.23	60.17	61.50	60.99	64.94	61.54	60.94	60.67	62.15	60.84
	PLE	61.95	60.25	60.52	61.57	63.37	62.82	62.25	62.16	62.02	61.70
	RMT-Net	65.87	62.97	63.83	63.73	65.11	66.72	65.64	65.78	65.11	64.80
Ours	LG-GNN	**66.00***	**63.21***	**64.50***	**64.16***	**66.15***	**67.66***	**67.55***	**67.23***	**66.05***	**65.57***

* Denotes statistically significant improvement, measured by t-test with p-value < 0.01, over the second-best approach on each dataset.

deliver significant performance gains in datasets with fewer features, as observed in Home2 and PPD2, emphasizing its proficiency in improving feature distinctions and mitigating context inconsistency, especially in scenarios with minimal feature variations and substantial contextual homogeneity.

These experimental results validate the effectiveness of our proposed LG-GNN method.

5.6. Ablation study

In this section, we present the overall experimental results and the corresponding analysis of the ablation study on the LG-GNN. We systematically removed each component of the LG-GNN, evaluating its performance to gain a comprehensive understanding of the functions and impacts of each module. Detailed experimental results are provided in Table 6.

5.6.1. Features embedding

The notable impact of Feature Embedding (FE) is evident in the performance contrast between lines 1 and 2. Compared to a model utilizing only a default/non-default classifier, FE enriches the model with more informative feature representations, resulting in a 9.58% increase in AUC and a 52.66% increase in KS.

5.6.2. Loan applicant relation graph construction

The significant effect of graph construction (GC) is highlighted in the performance contrast between lines 2 and 3. In contrast to FE alone, GC considers relationships between different types of loan applicants,

Table 6
The average performance (%) of LG-GNN across seven datasets, considering different components of LG-GNN.

Row Number	Components					Performance	
	FE	GC	LIF	GIF	CL	AUC	KS
1	–	–	–	–	–	60.95	16.28
2	✓	–	–	–	–	66.80	24.85
3	✓	✓	–	–	–	66.90	30.20
4	✓	✓	✓	–	–	67.24	31.76
5	✓	✓	✓	✓	–	67.35	31.80
6	✓	✓	✓	–	✓	67.53	31.88
7	✓	✓	✓	✓	✓	**67.85**	**32.73**

leading to a 0.16% improvement in AUC and a 21.54% improvement in KS.

5.6.3. Local information filtering

The substantial impact of local information filtering (LIF) is emphasized in the performance contrast between lines 3 and 4. Compared to using FE and GC alone, LIF aids the feature learning process by reducing noise interference from first-order neighbors in the context inconsistency scenario, resulting in a 0.51% improvement in AUC and a 5.16% improvement in KS.

5.6.4. Global information filtering

The significant effect of global information filtering (GIF) is highlighted in the performance contrast between lines 4 and 5. Compared to LIF with GC, GIF enhances the feature learning process by capturing features from distant neighbors blocked by context inconsistency, leading to a 0.17% improvement in AUC and a 0.11% improvement in KS for the model (GLIF with GC).

5.6.5. Default/non-default contrastive learning

The impact of employing contrastive learning (CL) is evident in the contrasts between lines 4 and 6, as well as lines 5 and 7. CL combined with LIF and GC brings a 0.44% improvement in AUC and a 0.36% improvement in KS compared to LIF with GC alone. Similarly, CL combined with GLIF and GC yields a 0.74% improvement in AUC and a 2.95% improvement in KS compared to GLIF with GC. The above improvements underscore the importance of focusing on the similarities and dissimilarities among loan applicants during the feature learning process.

In summary, optimal performance is attained through the simultaneous adoption of FE, GC, LIF, GIF, and CL, which are combined to form our LG-GNN and demonstrate the reasonableness of our model.

5.7. Hyper-parameter study

This section delves into a detailed examination of the hyperparameters in our proposed LG-GNN model. First, we explore the impact of graph edge and node selection parameters, namely, p, K_{local}, and K_{global}, on model performance. Second, we investigate the influence of loss balancing parameters η on model performance. Third, we analyze how the difference between the target node embedding dimension d and the layer number T of the classifier affects model performance.

5.7.1. Edge and neighbor filtering

In our model, we use three parameters for edge and node filtration. The first parameter, p, sets the threshold for establishing edges in the loan-applicant relation graph. The second and third parameters, K_{local} and K_{global}, are associated with local and global networks, respectively, and are used to filter neighbors of target nodes. Fig. 3 shows LG-GNN's test set performance across varying filtration thresholds. We adjusted p within the range of 0.7 to 0.99 for optimal validation set results. The results emphasize the critical role of the edge filtration threshold p in LG-GNN's performance across datasets. A small p leads to densely connected edges, hampering the aggregation of target nodes based on neighbor node weights. Proper p selection is crucial to filter out noise in the graph.

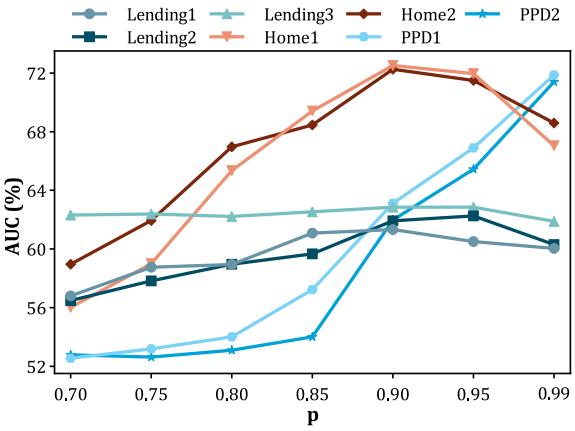


Fig. 3. Experimental results of LG-GNN with different edge filtering parameter.

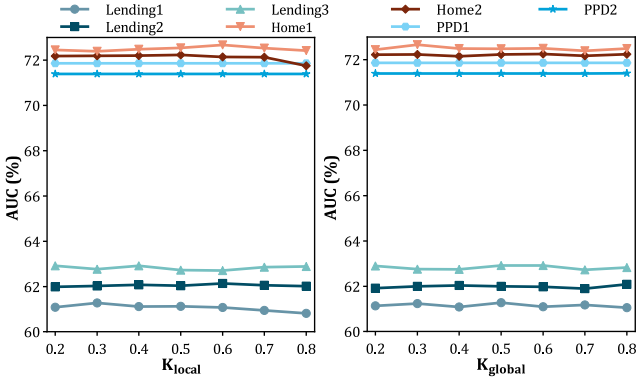


Fig. 4. Experimental results of LG-GNN with varying hyper-parameters: (1) the left part shows the impact of the local filter parameter K_{local}; (2) the right part shows the impact of the global filter parameter K_{global}.

Moreover, K_{local} and K_{global} vary from 0.2 to 0.8. Fig. 4 illustrates LG-GNN's performance on the test set as the filtration thresholds vary. Within a certain range, variations in K_{local} and K_{global} exhibit stable model performance across all datasets.

5.7.2. Loss balancing

It is essential to consider the impact of the loss balance parameter, η, on the performance of the LG-GNN model. We recommend tuning this hyperparameter based on a validation set to achieve optimal performance. By exploring the best performance on the validation set, we adjusted η in the range of 0.0 to 0.9. Fig. 5 illustrates the relatively stable performance of LG-GNN across varying values of η, as measured by both AUC and KS metrics. These limited fluctuations indicate that cross-entropy loss is dominant during LG-GNN training. In classification tasks, contrastive loss aims to improve feature representation learning, while cross-entropy loss directly measures the distributional difference between samples. Consequently, cross-entropy loss often plays a dominant role as a supervisory signal during training, guiding the model toward accurate classification. Contrastive loss, on the other hand, operates in conjunction with cross-entropy loss to refine the model's representation space. Fig. 6 shows that increasing η initially improves the performance of LG-GNN, but after a certain point, a further increase in η leads to a decline in performance. Smaller η values help LG-GNN to focus more on feature representation learning, while excessively large η values can cause overfitting. To highlight the importance of contrastive

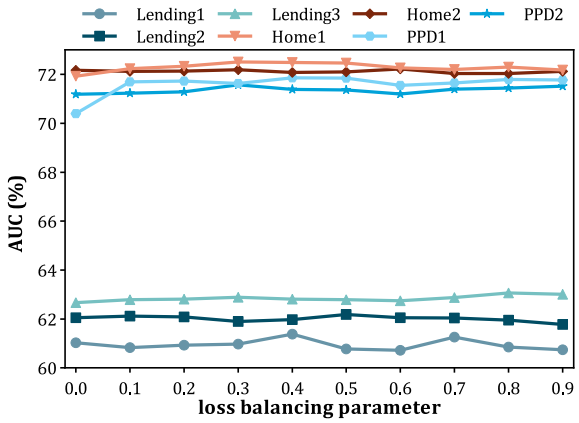


Fig. 5. Experimental results of LG-GNN with different loss balancing parameter, evaluated by AUC.

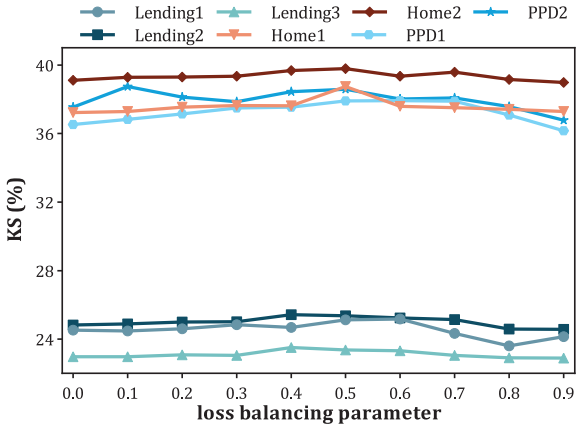


Fig. 6. Experimental results of LG-GNN with different loss balancing parameter, evaluated by KS.

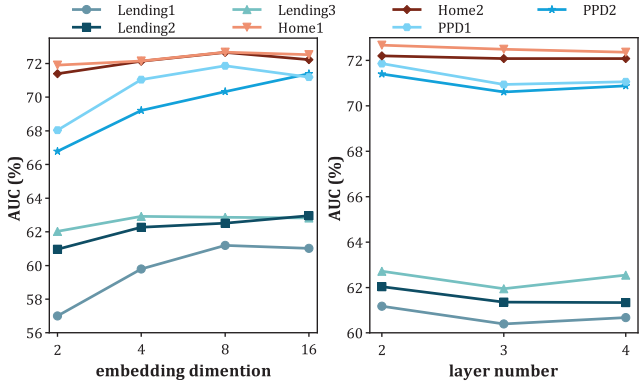


Fig. 7. Experimental results of LG-GNN with varying hyper-parameters: (1) the left part shows the impact of the embedding dimension d; (2) the right part shows the impact of the classification layer number t.

loss, we extracted data from Figs. 5 and 6 and organized it in Tables 7 and 8. These tables compare the performance of LG-GNN using $\mathcal{L}_2$ alone ($\eta = 0$) versus combined $\mathcal{L}_1$ and $\mathcal{L}_2$ (η lies within the range of 0 to 1). It is observed that the addition of contrastive learning enhances the model's performance.

5.7.3. Embedding dimension and classification layer

In this study, our primary focus is enhancing feature embeddings by incorporating local and global representations of target nodes. These embeddings are constructed based on the initial node embeddings. To find the ideal dimension, we experimented with dimensions 2, 4, 8, and 16 on a validation set, and the results are shown in Fig. 7. Our findings demonstrate an optimal dimension for different datasets, leading to the best model performance. However, if the dimension is too small, it may miss crucial target information, and if it is too large, it can introduce redundancy and reduce classification performance.

Furthermore, the performance of LG-GNN exhibits minimal sensitivity to the classification hierarchy level (T). To determine the optimal configuration, we conducted experiments with T values ranging from 2 to 4, selecting the best-performing setup on the validation set. For simplicity, we chose $T = 2$ and present the corresponding test set results in Fig. 7. Our experimental results reveal that the choice of T has no significant impact on LG-GNN's performance; it remains consistent within a certain range, ensuring its applicability across all datasets.

5.8. Complexity analysis

We theoretically analyze the space complexity of LG-GNN by focusing on the sizes of learnable variables and layer outputs. We omit variables with relatively small sizes, such as bias terms.

For the embedding layer, the size of each node's embedding vector is d_{dk}, resulting in a total space complexity of $O\left(N \times d_{dk}\right)$. During local and global information filtering, the constructed graph size is $O\left(N^2\right)$, leading to a space complexity of $O\left(N^2\right)$. In the output layers of local and global information fusion, each node's output size is $d_{h_l^2}$. Since we set the number of heads as 2, the corresponding space complexity is $O\left(N \times d_{h_l^2}\right)$. The space complexity of default/non-default contrastive learning is $O(1)$ as it does not vary with the number of nodes N. For default/non-default classification, each node has an output vector size of 2, resulting in a space complexity of $O(N \times 2)$. In summary, the total space complexity of LG-GNN is $O\left(N \times d_{dk} + N^2 + N \times d_{h_l^2} + N \times 2\right)$.

6. Discussion

Based on graph representation learning, we propose the LG-GNN method, which effectively addresses the default detection task. LG-GNN first captures the latent relations between samples by constructing and learning a graph neural network. It then employs contrastive learning to amplify the differences in sample features. As a result, LG-GNN demonstrates improved average performance compared to the most competitive default detection methods. The experimental results yield two implications: (1) Latent relations exist among loan applicants in personal credit, violating the assumption of independent and identically distributed samples required by most machine learning models. Therefore, it is crucial to select appropriate baseline models based on data features and sample distribution characteristics and design advanced models. This model can also be effectively applied to problems beyond default detection if it satisfies the basic assumptions and has suitable data structures. (2) The latent relations among loan applicants significantly impact the default detection of individual loan applicants. Even implicit relationships extracted from pre-loan features can be considered supplementary information for pre-loan features in individual default detection.

7. Conclusion

In recent years, researchers and practitioners have extensively explored the essential explicit attributes of instances to enhance personal default detection. However, latent relations among instances have often been overlooked, rendering instances no longer independent of personal default detection. Motivated by those, we propose a novel approach called the Local and Global Information-Aware Graph Neural Network (LG-GNN) for default detection. Unlike existing methods, LG-GNN adopts a holistic strategy by aggregating filtered information from local and global perspectives, enabling LG-GNN to address context

Table 7
Average AUC (%) of LG-GNN across seven datasets, considering different loss combinations for LG-GNN.

Dataset	Lending1	Lending2	Lending3	Home1	Home2	PPD1	PPD2	Average
$\mathcal{L}_2$	61.03	62.05	62.67	71.93	72.17	70.4	71.19	67.35
$\mathcal{L}_1+\mathcal{L}_2$	61.38	62.27	62.92	72.67	72.26	71.87	71.57	67.85
Improvement	0.57%	0.35%	0.40%	1.03%	0.12%	2.09%	0.53%	0.74%

Table 8
Average KS (%) of LG-GNN across seven datasets, considering different loss combinations for LG-GNN.

Dataset	Lending1	Lending2	Lending3	Home1	Home2	PPD1	PPD2	Average
$\mathcal{L}_2$	23.80	24.91	23.14	37.67	39.40	36.82	36.84	31.80
$\mathcal{L}_1+\mathcal{L}_2$	25.17	25.42	23.52	38.74	39.79	37.92	38.58	32.73
Improvement	5.75%	2.04%	1.63%	2.85%	0.99%	2.99%	4.72%	2.95%

inconsistencies in the applicants' relation graph while leveraging contrastive representation learning to enhance feature diversity. According to empirical experiments conducted on seven datasets with varying settings, LG-GNN demonstrates a substantial improvement over several state-of-the-art approaches from different perspectives. Future research directions will emphasize capturing latent relations among loan applicants using the graph structure and developing effective information mining strategies for personal default detection.

CRediT authorship contribution statement

Yi Liu: Writing – original draft, Supervision, Methodology, Investigation, Funding acquisition, Formal analysis, Conceptualization. **Xuan Wang:** Writing – original draft, Visualization, Validation, Software, Formal analysis, Data curation, Conceptualization. **Tao Meng:** Writing – review & editing, Validation, Software, Resources, Methodology, Investigation, Funding acquisition. **Wei Ai:** Writing – review & editing, Supervision, Resources, Project administration, Funding acquisition. **Keqin Li:** Writing – review & editing, Supervision, Project administration, Investigation.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgments

The authors deepest gratitude goes to the anonymous reviewers and AE for their careful work and thoughtful suggestions that have helped improve this paper substantially. This work is supported by National Natural Science Foundation of China (Grant No. 61802444); the Changsha Natural Science Foundation, China (Grant No. kq2202294), the Research Foundation of Education Bureau of Hunan Province of China (No. 22B0275); the Research on Local Community Structure Detection Algorithms in Complex Networks, China (Grant No. 2020YJ009).

References

Altman, E.I., 1968. Financial ratios, discriminant analysis and the prediction of corporate bankruptcy. J. Finance 23 (4), 589–609.
Brody, S., Alon, U., Yahav, E., 2021. How attentive are graph attention networks? arXiv preprint arXiv:2105.14491.
Bueff, A.C., Cytryński, M., Calabrese, R., Jones, M., Roberts, J., Moore, J., Brown, I., 2022. Machine learning interpretability for a stress scenario generation in credit scoring based on counterfactuals. Expert Syst. Appl. 202, 117271.
Chapelle, O., Manavoglu, E., Rosales, R., 2014. Simple and scalable response prediction for display advertising. ACM Trans. Intell. Syst. Technol. 5 (4), 1–34.
Chen, T., Kornblith, S., Norouzi, M., Hinton, G., 2020. A simple framework for contrastive learning of visual representations. In: International Conference on Machine Learning. PMLR, pp. 1597–1607.
Crook, J., 2002. Credit scoring and its applications. J. Oper. Res. Soc. 52, 997–1006.
Dahooie, J.H., Hajiagha, S.H.R., Farazmehr, S., Zavadskas, E.K., Antucheviciene, J., 2021. A novel dynamic credit risk evaluation method using data envelopment analysis with common weights and combination of multi-attribute decision-making methods. Comput. Oper. Res. 129, 105223.
Ding, Y., Robinson, N., Tong, C., Zeng, Q., Guan, C., 2023. LGGNet: Learning from local-global-graph representations for brain–computer interface. IEEE Trans. Neural Netw. Learn. Syst..
Gao, T., Yao, X., Chen, D., 2021. Simcse: Simple contrastive learning of sentence embeddings. arXiv preprint arXiv:2104.08821.
Garg, V., Jegelka, S., Jaakkola, T., 2020. Generalization and representational limits of graph neural networks. In: International Conference on Machine Learning. PMLR, pp. 3419–3430.
Gunel, B., Du, J., Conneau, A., Stoyanov, V., 2020. Supervised contrastive learning for pre-trained language model fine-tuning. arXiv preprint arXiv:2011.01403.
Guo, Z., Ao, X., He, Q., 2023. Transductive semi-supervised metric network for reject inference in credit scoring. IEEE Trans. Comput. Soc. Syst..
Guo, J., Huang, K., Yi, X., Zhang, R., 2022. Learning disentangled graph convolutional networks locally and globally. IEEE Trans. Neural Netw. Learn. Syst..
Hadsell, R., Chopra, S., LeCun, Y., 2006. Dimensionality reduction by learning an invariant mapping. In: 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR'06, Vol. 2, IEEE, pp. 1735–1742.
Hamilton, W., Ying, Z., Leskovec, J., 2017. Inductive representation learning on large graphs. Adv. Neural Inf. Process. Syst. 30.
Hjelm, R.D., Fedorov, A., Lavoie-Marchildon, S., Grewal, K., Bachman, P., Trischler, A., Bengio, Y., 2018. Learning deep representations by mutual information estimation and maximization. arXiv preprint arXiv:1808.06670.
Holte, R.C., 1993. Very simple classification rules perform well on most commonly used datasets. Mach. Learn. 11, 63–90.
Jiang, N., Li, L., 2016. Doubly robust off-policy value evaluation for reinforcement learning. In: International Conference on Machine Learning. PMLR, pp. 652–661.
Jones, S., 2017. Corporate bankruptcy prediction: a high dimensional analysis. Rev. Account. Stud. 22, 1366–1422.
Khosla, P., Teterwak, P., Wang, C., Sarna, A., Tian, Y., Isola, P., Maschinot, A., Liu, C., Krishnan, D., 2020. Supervised contrastive learning. Adv. Neural Inf. Process. Syst. 33, 18661–18673.
Lakhan, A., Mohammed, M.A., Ibrahim, D.A., Kadry, S., Abdulkareem, K.H., 2022. ITS based on deep graph convolutional fraud detection network blockchain-enabled fog-cloud. IEEE Trans. Intell. Transp. Syst..
Li, Z., Xu, Q., Ma, L., Fang, Z., Wang, Y., He, W., Du, Q., 2023a. Supervised contrastive learning-based unsupervised domain adaptation for hyperspectral image classification. IEEE Trans. Geosci. Remote Sens..
Li, P., Yu, H., Luo, X., Wu, J., 2023b. LGM-GNN: A local and global aware memory-based graph neural network for fraud detection. IEEE Trans. Big Data.
Liang, D., Lu, C.-C., Tsai, C.-F., Shih, G.-A., 2016. Financial ratios and corporate governance indicators in bankruptcy prediction: A comprehensive study. European J. Oper. Res. 252 (2), 561–572.
Liu, H., Hussain, F., Tan, C.L., Dash, M., 2002. Discretization: An enabling technique. Data Min. Knowl. Discov. 6, 393–423.
Liu, Q., Luo, Y., Wu, S., Zhang, Z., Yue, X., Jin, H., Wang, L., 2022. RMT-net: Reject-aware multi-task network for modeling missing-not-at-random data in financial credit scoring. IEEE Trans. Knowl. Data Eng..
Ma, J., Zhao, Z., Yi, X., Chen, J., Hong, L., Chi, E.H., 2018a. Modeling task relationships in multi-task learning with multi-gate mixture-of-experts. In: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. pp. 1930–1939.
Ma, L., Zhao, X., Zhou, Z., Liu, Y., 2018b. A new aspect on P2P online lending default prediction using meta-level phone usage data in China. Decis. Support Syst. 111, 60–71.

Maldonado, S., Paredes, G., 2010. A semi-supervised approach for reject inference in credit scoring using SVMs. In: Advances in Data Mining. Applications and Theoretical Aspects: 10th Industrial Conference, ICDM 2010, Berlin, Germany, July 12-14, 2010. Proceedings 10. Springer, pp. 558–571.

Mancisidor, R.A., Kampffmeyer, M., Aas, K., Jenssen, R., 2020. Deep generative models for reject inference in credit scoring. Knowl.-Based Syst. 196, 105758.

Misra, I., Shrivastava, A., Gupta, A., Hebert, M., 2016. Cross-stitch networks for multi-task learning. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 3994–4003.

Ohlson, J.A., 1980. Financial ratios and the probabilistic prediction of bankruptcy. J. Account. Res. 109–131.

Scarselli, F., Gori, M., Tsoi, A.C., Hagenbuchner, M., Monfardini, G., 2008. The graph neural network model. IEEE Trans. Neural Netw. 20 (1), 61–80.

Schnabel, T., Swaminathan, A., Singh, A., Chandak, N., Joachims, T., 2016. Recommendations as treatments: Debiasing learning and evaluation. In: International Conference on Machine Learning. PMLR, pp. 1670–1679.

Shen, Z., Cui, P., Zhang, T., Kunag, K., 2020. Stable learning via sample reweighting. In: Proceedings of the AAAI Conference on Artificial Intelligence. Vol. 34, (04), pp. 5692–5699.

Shen, X., Liu, X., Hu, X., Zhang, D., Song, S., 2022. Contrastive learning of subject-invariant eeg representations for cross-subject emotion recognition. IEEE Trans. Affect. Comput..

Shi, Y., Qu, Y., Chen, Z., Mi, Y., Wang, Y., 2024. Improved credit risk prediction based on an integrated graph representation learning approach with graph transformation. European J. Oper. Res. 315 (2), 786–801.

Sohn, S.Y., Kim, D.H., Yoon, J.H., 2016. Technology credit scoring model with fuzzy logistic regression. Appl. Soft Comput. 43, 150–158.

Sukharev, I., Shumovskaia, V., Fedyanin, K., Panov, M., Berestnev, D., 2020. EWS-GCN: Edge weight-shared graph convolutional network for transactional banking data. In: 2020 IEEE International Conference on Data Mining. ICDM, IEEE, pp. 1268–1273.

Swaminathan, A., Joachims, T., 2015. The self-normalized estimator for counterfactual learning. Adv. Neural Inf. Process. Syst. 28.

Tang, H., Liu, J., Zhao, M., Gong, X., 2020. Progressive layered extraction (ple): A novel multi-task learning (mtl) model for personalized recommendations. In: Proceedings of the 14th ACM Conference on Recommender Systems. pp. 269–278.

Veličković, P., Cucurull, G., Casanova, A., Romero, A., Lio, P., Bengio, Y., 2017. Graph attention networks. arXiv preprint arXiv:1710.10903.

Wang, X., Yang, S., Zhang, J., Wang, M., Zhang, J., Yang, W., Huang, J., Han, X., 2022. Transformer-based unsupervised contrastive learning for histopathological image classification. Med. Image Anal. 81, 102559.

Wang, X., Zhang, R., Sun, Y., Qi, J., 2019. Doubly robust joint learning for recommendation on data missing not at random. In: International Conference on Machine Learning. PMLR, pp. 6638–6647.

West, D., 2000. Neural network credit scoring models. Comput. Oper. Res. 27 (11–12), 1131–1152.

Wu, B., Chao, K.-M., Li, Y., 2024. Heterogeneous graph neural networks for fraud detection and explanation in supply chain finance. Inf. Syst. 121, 102335.

Xiao, J., Zhou, X., Zhong, Y., Xie, L., Gu, X., Liu, D., 2020. Cost-sensitive semi-supervised selective ensemble model for customer credit scoring. Knowl.-Based Syst. 189, 105118.

Xiong, T., Wang, S., Mayers, A., Monga, E., 2013. Personal bankruptcy prediction by mining credit card data. Expert Syst. Appl. 40 (2), 665–676.

Xu, D., Ruan, C., Korpeoglu, E., Kumar, S., Achan, K., 2020. Adversarial counterfactual learning and evaluation for recommender system. Adv. Neural Inf. Process. Syst. 33, 13515–13526.

Yang, K., Zhang, T., Alhuzali, H., Ananiadou, S., 2023. Cluster-level contrastive learning for emotion recognition in conversations. IEEE Trans. Affect. Comput..

Ying, Z., Bourgeois, D., You, J., Zitnik, M., Leskovec, J., 2019. Gnnexplainer: Generating explanations for graph neural networks. Adv. Neural Inf. Process. Syst. 32.

You, Y., Chen, T., Sui, Y., Chen, T., Wang, Z., Shen, Y., 2020. Graph contrastive learning with augmentations. Adv. Neural Inf. Process. Syst. 33, 5812–5823.

Yuan, H., Tang, J., Hu, X., Ji, S., 2020. Xgnn: Towards model-level explanations of graph neural networks. In: Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. pp. 430–438.

Yuan, H., Yu, H., Wang, J., Li, K., Ji, S., 2021. On explainability of graph neural networks via subgraph explorations. In: International Conference on Machine Learning. PMLR, pp. 12241–12252.

Zhang, X., Yu, L., Yin, H., Lai, K.K., 2022. Integrating data augmentation and hybrid feature selection for small sample credit risk assessment with high dimensionality. Comput. Oper. Res. 146, 105937.

Zhu, Y., Xu, Y., Yu, F., Liu, Q., Wu, S., Wang, L., 2021. Graph contrastive learning with adaptive augmentation. In: Proceedings of the Web Conference 2021. pp. 2069–2080.

Zhu, J., Yan, Y., Zhao, L., Heimann, M., Akoglu, L., Koutra, D., 2020. Beyond homophily in graph neural networks: Current limitations and effective designs. Adv. Neural Inf. Process. Syst. 33, 7793–7804.