Neural Computing and Applications (2024) 36:12149–12162
https://doi.org/10.1007/s00521-024-09695-x

ORIGINAL ARTICLE

Multi-view GCN for loan default risk prediction

Zihao Li[1] · Yakun Chen[1] · Xianzhi Wang[1] · Lina Yao[2] · Guandong Xu[1]

Received: 19 September 2023 / Accepted: 25 March 2024 / Published online: 19 April 2024

Abstract
As a significant application of machine learning in financial scenarios, loan default risk prediction aims to evaluate the client's default probability. However, most existing deep learning solutions treat each application as an independent individual, neglecting the explicit connections among different application records. Besides, these attempts suffer from the problem of missing data and imbalanced distribution (i.e., the default records are small samples against all the applications). We believe similar records could provide some auxiliary signals, which are of critical importance to alleviate the data missing issue and facilitate data argumentation. To this end, we propose multi-view loan application graphs, dubbed MLAGs. By evaluating the similarity between the records, a loan application graph can be constructed. Furthermore, we arrange different similarity thresholds to organize various graph structures for multi-graph constructions; thus, a variety of representations can be generated via information propagation and aggregation for small sample argumentation. Consequently, the imbalanced data distribution and missing values issues can be alleviated effectively. We conduct experiments on three public datasets from real-world home credit and P2P lending platforms, which show that MGCN outperforms both conventional and deep learning models. Ablation studies also illustrated the validity of each module design.

Keywords Loan default prediction · Graph neural network · Multi-view graphs · Heterogeneous and unbalanced data

1 Introduction

Credit risk, a borrower failing to make required payments on the designated time, will cause greater uncertainty and instability for the development of the banking sector. Hence, it is significant and necessary to assess the client's repayment willingness and abilities according to his or her detailed demographic information and historical loan records (e.g., annual income, housing status, historical loan records, and family members' information) before authorizing a loan. Recently, the released datasets, e.g., Lending Club,[1] Home Credit Default Risk,[2] Prosper,[3] attracted widespread attention from academia and industry for credit default risk prediction [1–3].

Conventional credit assessment heavily relies on rules of thumb and the professional knowledge of seasoned experts. For instance, according to the instruction of '5C principle' [4], we could manually evaluate the default risk of borrowers on five aspects: *character*, *capital*, *capacity*, *collateral* and *conditions*, then decide whether the loan application could be approved. However, this process is time-consuming and labor-intensive.

To circumvent this issue, some machine learning methods, e.g., Logistic Regression, Decision Tree, Random Forest, XGBoost, etc., have been proposed and achieved satisfactory results for credit assessment [5–7]. However,

✉ Zihao Li
zihao.li@student.uts.edu.au

Yakun Chen
yakun.chen@student.uts.edu.au

Xianzhi Wang
xianzhi.wang@uts.edu.au

Lina Yao
lina.yao@unsw.edu.au

Guandong Xu
guandong.xu@uts.edu.au

[1] School of Computer Science, University of Technology Sydney, Broadway, Sydney, NSW 2007, Australia

[2] School of Computer Science and Engineering, University of New South Wales, Garden, Sydney, NSW 2052, Australia

[1] www.lendingclub.com.

[2] https://www.kaggle.com/c/home-credit-default-risk.

[3] https://www.prosper.com/.

the performance of machine learning methods heavily relies on manual feature engineering, which is a tedious, error-prone process, and also requires fertile domain knowledge. For instance, the Home Credit Default Risk dataset includes hundreds of attributes for each loan application. It is rather laborious to design features manually. Furthermore, feature engineering work is problem-dependent and domain-specified. As the raw data or tasks change, the hand-craft features require to be adjusted correspondingly, which significantly affects the transferability, reuse, and robustness of models. Furthermore, it is intractable for most machine learning techniques to extract effective patterns from big, multi-modal heterogeneous data [8]. The loan default relevant datasets contain both numerical/continuous and category data, e.g., gender, age, education, income, etc. The complicated data structure obstructs the wide application of machine learning methods.

Attribute to the powerful capability and flexibility of feature extraction and representation, deep learning could alleviate the above issues and it has been proven successful in multiple domains, such as computer versions, natural language processing, games, and recommendation systems [8]. However, existing deep learning methods for loan default risk prediction remain two key challenges unaddressed:

- *Missing values and imbalanced data distribution* Due to information sensitivity, privacy requirements, and trust issues, users' information may be incomplete or even non-existent. For instance, the percentage of records containing missing values exceeds 50% in the Home Credit Default Risk dataset. In addition, the default samples only account for a small part of the most of datasets. As shown in Fig. 1, as the default records (negative) are small samples in practical scenarios, the data distribution is imbalanced, which may lead to the unequal cost of misclassification errors and scarce accuracy of prediction [9, 10]. In general, the missing values and skewed data distribution pose a huge challenge for default risk prediction.
- *Explicit connections between records* Deep learning methods focus more on record representation learning while neglecting the explicit connection relationships between records. We believe the similar information encapsulated in the records could also facilitate the feature representation and final prediction [3, 11, 12].

As shown in Fig. 2, the record in the table can be regarded as a node; thus, a loan application graph could be constructed via the similarity among each record. Through node classification, the loan default risk problem could be solved effectively. To this end, we propose a multi-view graph convolution network (MGCN) for loan default prediction. First, we endeavor to construct multi-view loan application graphs via similarity calculation for small sample augmentation and skew distribution adjustment. Besides, based on graph convolution, similar loan application records can be aggregated as auxiliary information; thus, the model's robustness to missing values could be improved. We make the following contributions in this paper:

- We devise multi-view loan application graphs (MLAGs) via the similarity between application records. By adjusting the thresholds of similarity, we could control the sparsity of graphs for multi-view graph construction. Then, the data augmentation strategy is applied to small samples to balance the data distribution.
- We propose a multi-view graph convolution network (MGCN) for loan default prediction. By information aggregation and propagation, the similar record information encapsulated in the graph structures can be introduced as auxiliary information to alleviate the issue of missing values.
- We conducted experiments on three real-world open loan default risk prediction datasets. The experimental results show a significant performance improvement in our approach over state-of-the-art methods [11–13].

The rest of the paper is organized as follows. Section 2 overviews existing methods for loan default risk prediction.

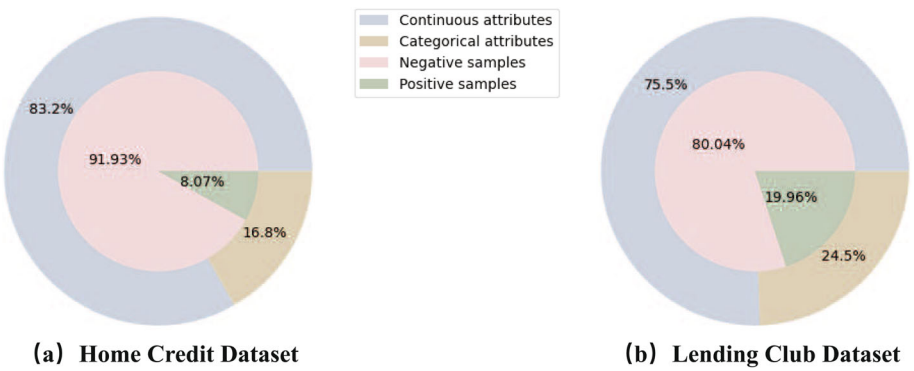

Fig. 1 Percentage of categorical and continuous attributes, positive and negative samples account for all the records on the Home Credit Default Risk Dataset and Lending Club Dataset

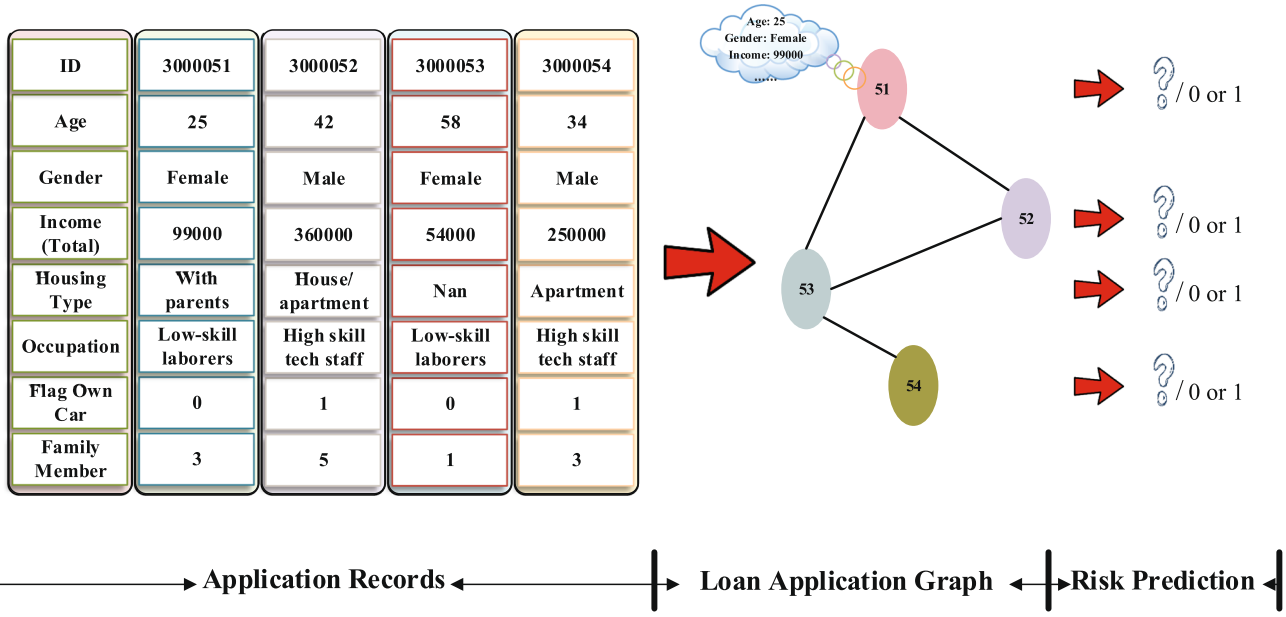

Fig. 2 A toy example of loan graph. Each loan application record (ID) can be represented as a node. Based on the similarity, we could add edges between nodes

Section 3 introduces the problem statement briefly. Section 4 presents the technical details of our multi-view graph convolution network for loan default risk prediction. Section 5 reports our experiments on three real-world datasets, and Sect. 6 concludes this paper.

2 Relate work

2.1 Machine learning methods

In the early stage, tremendous efforts have been devoted to developing qualitative and quantitative loan default risk assessment for loan default prediction [14]. Based on the credit scoring systems and the rule of thumb, the credit score of borrowers could be obtained for loan application authorize [15–18]. For instance, Serrano-Cinca et al. [19] pointed out that the loan purpose, annual income, and current housing situation have a strong correlation with credit score assessment. However, it is challenging to acquire precise loan default risk evaluation results due to the various risks and the complexities of dependencies between numerous influencing factors. Hence, the machine learning methods, e.g., tree-based classifiers [5, 20], support vector machines (SVM) [7, 21–24], neural networks (NN) [21, 25, 26], ensemble-based methods [6, 7], hybrid approaches [24, 25, 27, 28], and others [29–31], have been proposed for loan default risk prediction.

Yang [32] presented an incremental kernel learning method for credit scoring. With this approach, the scoring model can be adjusted via an online update procedure. Huang et al. [33] proposed a two-stage genetic programming (2SGP) method, which can incorporate the IF-THEN rules with neural network training for credit score prediction. Lee et al. [25] integrated neural networks with a traditional analysis approach and proposed a two-stage hybrid model. First, LDA was applied for predictor feature selection. Then, the selected features and predicted scores were further fed into a neural network to improve the accuracy of prediction results. Moreover, some scholars also argued that neural networks consistently perform better and more robustness than the multiple discriminant analysis models and machine methods including K-nearest neighbor, logistic regression, classification trees, discriminant analysis, and naive bayes, for loan default risk classification [34, 35].

2.2 Deep learning methods

Although conventional machine learning methods achieve promising performance for credit default risk prediction, these models heavily rely on the quality of feature engineering, i.e., the results for the same dataset can vary enormously even with a slight change of selected features [1]. In addition, it is non-trivial for such models to handle complicated data due to the limited feature representation capacity. To resolve these issues, deep learning methods

were developed recently. Babaev et al. applied RNN for credit loan application task [36]. Kvammea et al. applied a convolutional neural network for mortgage default prediction [37]. Tan et al. developed a deep learning approach for charge-off and prepayment risk prediction on P2P lending dataset [2]. To be specific, the authors first generated the risk grads based on loan status, survival time, and loan term. Then, the deep learning model transformed the hierarchical grades prediction into multiple binary classification subtasks for risk prediction. Considering the interpretability of deep learning methods, Liu et al. proposed a novel automatic feature crossing method called DNN2LR, which applied a deep neural network for feature extraction. Then, the extracted features were further fed into a Logistic Regression model for loan credit risk assessment [38].

2.3 Graph-based methods

As the loan application records may exist implicit connections among them, graph-based methods are devoted to loan default prediction. Experiments further demonstrated the effectiveness of graph-based features [39]. Zhong et al. proposed an attributed heterogeneous information network for credit risk prediction. First, user behaviors were adopted for node representation learning via the meta-path method. Then, the attention mechanism was applied for node classification [40]. Guo et al. pointed out that existing works mainly focus on feature interactions and ignore sample relations. Hence, the authors constructed a multiplex graph and applied a graph neural network to learn enhanced representation for each sample [13]. Considering the complexity of financial scenarios, Hu et al. developed an Attributed Multiplex Graph (AMG) to model various relations and the rich attributes of nodes and edges simultaneously. Through local structure and multiplex relations modeling, the AMG achieved the state-of-the-art for users' credit risk prediction on a large-scale real-world dataset [3]. Although the above works proposed graph neural networks for loan default prediction, they mainly rely on users' social relations for graph construction. In the real scenario, the social relationships between users may be unknown as privacy protection limitations. Thus, it is intractable to construct a graph based on existing methods. In addition, the above methods model the heterogeneous attributes of nodes and edges with a single graph, which is insufficient to represent such complicated information. Moreover, the imbalanced data distribution prevents the model from achieving optimal results. Consequently, in our work, we consider both the heterogeneity of attributes and graph structures and construct multi-graphs for small sample augmentation and loan application.

3 Problem formulation

Based on the raw application records, multi-view loan application graphs (MLAGs) will be constructed as follows.

3.1 Loan application graph (LAG)

Given a set of loan application records $\mathcal{D}$, we denote an application record as r and $r \in \mathcal{D}$. In LAG, the loan application record r will be represented by a node v. And the correspondent attributes of v are denoted as x_v. Let $\mathcal{X}^{\mathcal{V}} = \{x^v \mid v \in \mathcal{V}\}$ as the set of attributes, which can be further classified as categorical attributes $\mathcal{X}^{\mathcal{V}}{}_{\text{Ca}}$ (e.g., gender, education, occupation, etc.) and continuous attributes $\mathcal{X}^{\mathcal{V}}{}_{\text{Co}}$ (e.g., age, income, etc.) based on the containing number of unique values. Specifically, in this paper, we define the attribute whose containing number of unique values is less than the threshold as a categorical attribute, otherwise, it belongs to the continuous attribute. It can be formalized as follows:

$$\begin{cases} x \in \mathcal{X}^{\text{Ca}} & \text{if } \#(x) < \eta N \\ x \in \mathcal{X}^{\text{Co}} & \text{others} \end{cases} \tag{1}$$

where $\mathcal{X}^{\text{Ca}}, \mathcal{X}^{\text{Co}}$ are categorical attributes set and continuous attributes set, respectively. N is the number of total records in this dataset. η is a hyperparameter, which controls the granularity of attributes modeling, i.e., if the η is bigger, more attributes will encode to category embedding.. We set $\eta = 10^{-4}$. $\#(\cdot)$ is the counting operation. Hence, for attribute x, if the number of containing values is less than threshold ηN, we categorize x as a categorical attribute, otherwise, x belongs to a continuous attribute.

Assuming each attribute includes m categorical attributes and n continuous attributes. We, thereby, calculate the similarity between any of the two nodes. If the similarity between node v_i and node v_j is over the threshold δ, we add an edge e_{ij} between them. Consequently, a loan application graph (LAG) $\mathcal{G} = \{\mathcal{V}, \mathcal{E}, \mathcal{X}^{\mathcal{V}}\}$ could be constructed, where $\mathcal{V}$ and $\mathcal{E}$ represent the set of nodes and edges in the LAG, respectively.

3.2 Multi-view loan application graphs (MLAGs)

As described in section 3.1, changing the similarity threshold δ, we could adjust the structure of LAG dynamically. Therefore, given a series of $\delta \in \Delta, \Delta = \{\delta \mid 0 < \delta < 1\}$, multi LAGs can be constructed. We denote the $\mathbf{G} = \{\mathcal{G}_1, \mathcal{G}_2, \ldots\}$ as the multi-view loan application graphs (MLAGs). The different LAG allows us to dynamically select and aggregate information from

similar records in a flexible fashion. Besides, the MLAGs will also be utilized for small sample argumentation.

3.3 Loan credit default risk prediction

Given $\mathcal{D} = \{r, y_r\}$, where $y_r \in \{0, 1\}$ is the label of loan default and r is users' correspondent application information and historical records. Based on the application record r, the task of Loan credit default risk prediction aims to identify the likelihood that the user will default or not.

4 Methodology

In this section, we first give an overall illustration of our proposed MGCN model. Then, we introduce each module of MGCN specifically. Figure 3 illustrates the MGCN structure which contains four parts:

- *MLAGs construction* Based on the Sects. 3.1 and 3.2, each loan application record r can be regarded as a node v. Therefore, by calculating the similarity between any of two nodes and adjusting the thresholds δ, we could construct the MLAGs for loan default risk prediction.
- *Node embedding and graph convolution* We concatenate the category attributes embedding and continuous attributes embedding for application record (i.e., node) representation generation. Then, attributed to the information propagation and aggregation from similar neighbors, a graph convolution with a gate mechanism is proposed to alleviate the issue of missing values.
- *Data augmentation* As the data distribution is imbalanced in a practical scenario, for each negative (small) sample, we collect all the representations from MLAGs, while for each positive sample, we fuse all the representations under different LAGs with a soft attention mechanism to generate a single representation for data distribution re-balance.
- *Prediction layer* After obtaining loan application representation updated by the graph convolution layer, we devise a prediction layer (a feed-forward network) for loan default risk prediction.

4.1 Multi-view loan application graphs construction

As aforementioned described, there are two category attributes for each loan application record. Hence, we calculate the similarity for the two kinds of attributes respectively for loan application graph construction. To be specific, for categorical attributes x^{Ca}, we calculate the Jaccard similarity between any of two records, which can be formalized as follows:

$$\mathrm{Sim}_{\mathrm{Ca}}\left(x_i^{\mathrm{Ca}}, x_j^{\mathrm{Ca}}\right) = \frac{x_i^{\mathrm{Ca}} \cap x_j^{\mathrm{Ca}}}{x_i^{\mathrm{Ca}} \cup x_j^{\mathrm{Ca}}} \tag{2}$$

where $x_i^{\mathrm{Ca}}, x_j^{\mathrm{Ca}}$ are categorical attributes values of record r_i and r_j, respectively. Hence, the numerator counts the number of attributes with the same values for records r_i and r_j. And the denominator counts the total number of attribute values from r_i and r_j. n is the total number of attributes.

For continuous attributes, we utilize Euclidean distance to measure the similarity between each record.

$$\mathrm{Sim}_{\mathrm{Co}}\left(x_i^{\mathrm{Co}}, x_j^{\mathrm{Co}}\right) = 1 - \sum_{k=1}^{n}\left(x_{i,k}^{\mathrm{Co}} - x_{j,k}^{\mathrm{Co}}\right)^2 \tag{3}$$

where $x_{i,k}^{\mathrm{Co}}, x_{j,k}^{\mathrm{Co}}$ are k-th continuous attribute values (undergo normalization operation) of records r_i and r_j, respectively.

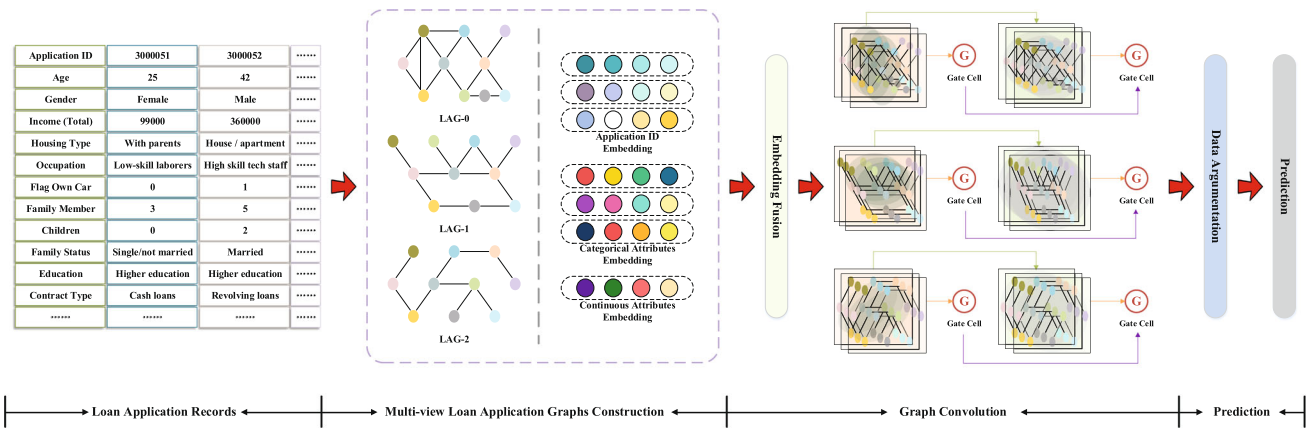

Fig. 3 Architecture of MGCN. It includes four main parts, which are arranged from left to right: (1) Multi-view Loan Application Graph Construction; (2) Graph Convolution Layer; (3) Data Argumentation Layer; and (4) Prediction Layer

Thus, the final similarity between records r_i and r_j can be defined as follows:

$$\begin{aligned}\mathrm{Sim}(r_i, r_j) =& \alpha \cdot \mathrm{Sim}_{\mathrm{Ca}}\left(x_i^{\mathrm{Ca}}, x_j^{\mathrm{Ca}}\right) + (1-\alpha) \\ & \cdot \mathrm{Sim}^{\mathrm{Co}}\left(x_i^{\mathrm{Co}}, x_j^{\mathrm{Co}}\right)\end{aligned} \tag{4}$$

where α is a hyperparameter to balance the similarity between categorical attributes and continuous attributes. As shown in Fig. 1, as the categorical attributes account for the major part of all the attributes and it is sufficient to measure the similarity between any of two nodes via categorical attributes, thus, we only calculate the similarity of categorical attributes to reduce computation complexity of MLAGs construction in our paper.

Again, we further define a hyperparameter δ, if $\mathrm{Sim}(r_i, r_j) > \delta$, we add an edge between node v_i and v_j. Hence, the weight of edge $e_{i,j}$ is $\mathrm{Sim}(r_i, r_j)$. Adjusting the threshold δ, we could control the sparsity of graph structure.

4.2 Embedding layer

4.2.1 Categorical attributes embedding

Firstly, we project the categorical attributes of each record to a latent space through an embedding table $\mathbf{E} \in \mathbb{R}^{M \times d}$. Here, M is the total number of unique values for all the categorical attributes, and d is the embedding dimension. The embedding operation is implemented as follows,

$$\mathbf{x}_{\mathbf{v}}^{\mathbf{Ca}} = \mathrm{Onehot}(x_v^{\mathrm{Ca}})\mathbf{E} \tag{5}$$

where $\mathrm{Onehot}(\cdot)$ is one-hot operation, which encodes the original categorical attributes $x_{\mathrm{v}}^{\mathrm{Ca}}$ of node v to an one-hot vector. $\mathrm{Onehot}(x_{\mathrm{v}}^{\mathrm{Ca}}) \in \mathbb{R}^{m \times M}$, m is the number of categorical attributes for each record.

4.2.2 Continuous attributes embedding

For each continuous attribute, we adopt $\mathrm{Norm}(\cdot)$ operation for normalization, which will further undergo a linear projection matrix for representation initialization.

$$\mathbf{x}_{\mathbf{v}}^{\mathbf{Co}} = \mathrm{Norm}(\mathbf{x}_{\mathbf{v}}^{\mathbf{Co}})\mathbf{W} \tag{6}$$

where $\mathbf{W} \in \mathbb{R}^{n \times n}$ is a learnable matrix.

4.2.3 Node representation

We flatten the categorical attributes embedding and concatenate the continuous attribute representation to construct the node representation $\mathbf{x}_{\mathbf{v}}$. $\mathbf{x}_{\mathbf{v}} \in \mathbf{X}$, and $\mathbf{X} \in \mathbb{R}^{|V| \times (m \times d + n)}$.

$$\mathbf{x}^{\mathbf{v}} = \mathop{\Big\|}_{k=0}^{m} \mathbf{x}_{\mathbf{v}}^{\mathbf{Ca_k}} \,\|\, \mathbf{x}_{\mathbf{v}}^{\mathbf{Co}} \tag{7}$$

where $\mathbf{x}_{\mathbf{v}}^{\mathbf{Ca_k}} \in \mathbb{R}^d$ is k-th categorical attribute embedding of node v. $x_v^{Co} \in \mathbb{R}^n$ is the continuous attributes embedding. $\|$ is the concatenation operation.

4.3 Graph convolution layer

We utilize graph convolution on each LAG for information propagation and aggregation [41]. Thus, similar records can be introduced as auxiliary information to alleviate the missing value issue. However, the deeper graph convolution operation will incur the over-smoothing [42] (also known as information loss [43] problem). Consequently, we propose to aggregate the information from the shallow graph convolution layer via a gate mechanism for node representation updating. For LAG - g, the node representation can be formulated as:

$$\begin{aligned}\mathbf{X}_{\mathbf{g}}^{(\mathbf{l})} &= \mathrm{ReLU}(\tilde{\mathbf{D}}_{\mathbf{g}}^{-\frac{1}{2}}\tilde{\mathbf{A}}_{\mathbf{g}}\tilde{\mathbf{D}}_{\mathbf{g}}^{-\frac{1}{2}}\mathbf{X}_{\mathbf{g}}^{(\mathbf{l}-\mathbf{1})}\mathbf{W}^{(\mathbf{l}-\mathbf{1})}) \\ \mathbf{H}_{\mathbf{g}}^{(\mathbf{l})} &= \alpha\mathbf{X}_{\mathbf{g}}^{(\mathbf{l})} + (1-\alpha)\mathbf{H}_{\mathbf{g}}^{(\mathbf{l}-\mathbf{1})} \\ \alpha &= \mathrm{Diag}(\sigma(\mathbf{X}_{\mathbf{g}}^{(\mathbf{l})}\mathbf{W}_{\mathbf{X}} \cdot \mathbf{H}_{\mathbf{g}}^{(\mathbf{l}-\mathbf{1})}\mathbf{W}_{\mathbf{H}}))\end{aligned} \tag{8}$$

where $\mathbf{X}_{\mathbf{g}}^{(\mathbf{l})}$ is the node representation after l-th graph convolution layer with regard to g. $\mathcal{A}_g \in \mathbb{R}^{N \times N}$ is the adjacency matrix of graph g, $\tilde{\mathbf{A}}_{\mathbf{g}} = \mathbf{A}_{\mathbf{g}} + I$, and $\tilde{\mathbf{D}}_{\mathbf{g}_{\mathbf{ii}}} = \sum_j \tilde{\mathbf{A}}_{\mathbf{g}_{\mathbf{ij}}}$. $\tilde{\mathbf{D}}_{\mathbf{g}}$ is a degree matrix of $\tilde{\mathbf{A}}_{\mathbf{g}}$; $\mathbf{W}^{\mathbf{l}-\mathbf{1}}, \mathbf{W}_{\mathbf{X}}, \mathbf{W}_{\mathbf{H}}$ are learnable matrices; $\mathrm{ReLU}(\cdot)$ and $\sigma(\cdot)$ (i.e., $\mathrm{sigmoid}(\cdot)$) are activation functions; $\mathrm{Diag}(\cdot)$ means obtain diagonal values. In our paper, we initialize $\mathbf{X}_{\mathbf{g}}^{(\mathbf{0})}, \mathbf{H}_{\mathbf{g}}{}^{(0)}$ as the node embedding generated by Sect. 4.2.3 and obtain the last update $\mathbf{H}_{\mathbf{g}}^{\mathbf{L}}$ as the graph convolution output of g.

4.4 Data augmentation layer

As the default records (i.e., negative samples) are small samples in practical scenarios, the data distribution is skewed. Hence, we carry out data augmentation on small samples to balance the data distribution. To be specific, for negative samples, we collect all the representations from different LAGs, while so as to the positive sample, we mix the different representations from LAGs with a soft attention mechanism as below.

$$\begin{aligned}\mathbf{h}^{\mathbf{p}} &= \sum_{g=1}^{s} \alpha_g(\mathbf{h}_{\mathbf{g}}^{\mathbf{p}}\mathbf{W}_{\mathbf{p}}) \\ \alpha_g &= \frac{\mathbf{x}^{\mathbf{p}}\mathbf{W}_{\mathbf{p}} \cdot \mathbf{h}_{\mathbf{g}}^{\mathbf{p}}\mathbf{W}_{\mathbf{p}}}{\sum_{g=1}^{s} \mathbf{x}^{\mathbf{p}}\mathbf{W}_{\mathbf{p}} \cdot \mathbf{h}_{\mathbf{g}}^{\mathbf{p}}\mathbf{W}_{\mathbf{p}}}\end{aligned} \tag{9}$$

where $\mathbf{x^p}$ is the positive sample's embedding; $\mathbf{h_g^p}$ is the positive sample's representation with regard to LAG - g after graph convolution derived from Eq. (8). $\mathbf{W_p}$ is a learnable matrix. s is the number of LAGs.

4.5 Prediction layer

Note that each loan application record r can be recognized as a node v in the MLAGs. Therefore, the loan default risk prediction can be transformed into the node classification task. Specifically, we implement the feed-forward network (FFN) and with *softmax* function for node classification, which is formulated as follows:

$$\begin{aligned} \hat{\mathbf{y}} &= \mathrm{ReLU}(\mathbf{W_2} \cdot \mathrm{ReLU}(\mathbf{W_1} \cdot \mathbf{h^L} + \mathbf{b_1}) + \mathbf{b_2} \\ \hat{\mathbf{y}} &= \mathrm{softmax}(\hat{\mathbf{y}}) \end{aligned} \tag{10}$$

where $\mathbf{h^L}$ is the record representation after data augmentation. $\hat{\mathbf{y}}$ is the probabilities of predicted labels. $\mathbf{W_1}, \mathbf{W_2}, \mathbf{b_1}, \mathbf{b_2}$ are the trainable weight matrices and the bias vectors, respectively.

4.6 Model learning

The cross-entropy with L2 regularization is adopted as the loss function to train the MGCN in an end-to-end mode. Formally, we define the loss function over all the loan application records $\mathcal{D}$ as:

$$\mathcal{L}(\mathbf{\Theta}) = -\sum_{<r,y>\in\mathcal{D}} y \log(\hat{\mathbf{y}}) - \lambda ||\mathbf{\Theta}||_2^2 \tag{11}$$

where y is the ground-truth of loan risk prediction, $\mathbf{\Theta}$ is the trainable parameters of our proposed model, and λ is a regularized hyperparameter.

5 Experiments

In this section, we investigate the effectiveness of our designed model. We conduct extensive experiments on three real-world public datasets. First, we describe the experimental settings including the datasets, methods in comparison, evaluation metrics, and experimental setup. Then, we discuss the performance comparison and results from the model analysis. Additionally, we perform ablation tests to demonstrate the effectiveness of every component in our model. We aim to answer the following research questions:

- *RQ1:* Does our model outperform the state-of-the-art loan default risk prediction methods on the real-world datasets?
- *RQ2:* How do the key components (e.g., gate mechanism in graph convolution layer and multi-view loan application graphs) and hyperparameters of our model benefit the prediction?
- *RQ3:* How about sampling strategies (e.g., up and downsampling, SMOTE method) to MGCN performance improvement compared with our data augmentation?

Table 1 Statistical information of datasets. Label-1 means the loan default samples

Statistic index	Home credit	Lending club	PPD
# Total	307,511	2,260,701	30,000
# Label-1	24,825	268,559	2,198
# Label-0	282,686	1,076,751	27,802
# Continuous Attributes	288	113	208
# Categorical Attributes	58	38	20
Ratio of missing values	50%	100%	21%

5.1 Dataset

We evaluate our method performance on three public datasets. Table 1 summarizes the basic statistics of these datasets. Observing Table 1, we could find that (1) the ratio of loan default samples to others is up to 1 : 12.6, the data distribution is skewed significantly; (2) the number of continuous attributes is about 2.97 to 10.4 times that of categorical attributes; (3) a large number of missing values, e.g., for Lending Club dataset, all the attributes contain missing values.

- *Home credit default risk*[4] primarily to lending those people with little or no credit history. Due to incomplete or non-existent credit histories, financial institutions are required to evaluate various alternative data for a positive and safe borrowing service. Hence, the application records are always complex, multi-source, and heterogeneous. This dataset covers various information about applicants, e.g., family information, income and expenditure, credit records, loan records, repayment history, etc.
- *Lending Club*[5] is a P2P lending dataset, which contains millions of loan records with 151 attributes from 2007 to 2018. All the attributes in raw data have missing values except *id*. Following previous work [5], we remove meaningless attributes (e.g., URL, member_id), and the attributes with more than 30% missing values. Afterward, we fill in the missing values with zeros.

[4] https://www.kaggle.com/c/home-credit-default-risk/overview.

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

- *PPD*[6] is a public loan default prediction dataset released from a risk control algorithm competition at Heywhale community. Each record comprises 208 continuous attributes and 20 categorical attributes, which cover the basic information of lenders, the login information and modify records.

5.2 Experimental settings

5.2.1 Evaluation metrics

Following previous works [3, 38, 40], we select Recall, F1 score, and AUC (i.e., Area Under the ROC Curve) to evaluate the performance of each model on the above three public datasets.

AUC signifies the probability that the positive item sample's score is higher than the negative item sample's score, a larger AUC indicates a better performance. Compared with precision, recall, and F1 value, AUC trends to reveal the rate of false positives (also known as false alarms). Considering the cost-sensitive and imbalanced distribution in the default risk prediction task, it may be more reasonable to calculate AUC for model evaluation.

5.2.2 Compared methods

We compare our model with multiple representative and competitive methods from four categories: machine learning methods (e.g., Logistic Regression, Decision Tree, Random Forest, XGBoost), deep learning methods (e.g., DNN, CNN, Wide & Deep Neural Network), unsupervised graph embedding methods (e.g., DeepWalk, Node2vec), and GCN-based methods (e.g., GCN, GraphSage, GAT, SGC, GCNII).

- *Logistic regression* [38]: a simple and efficient linear model for binary classification.
- *Decision tree* [20]: a nonparametric supervised learning algorithm. It has a hierarchical tree structure, classifying samples following an ordering of attributes.
- *Random forest* [7]: a bagging method that trains a multitude of decision trees and determines the final reason via majority voting.
- *XGBoost* [5]: a decision-tree-based ensemble machine learning algorithm that uses a gradient boosting framework.
- *DNN* [3]: a neural network with fully connected layers. We choose ReLU as the activation function and cross-entropy loss for model optimization.
- *CNN* [37]: a neural network that feeds the concatenation of the representation of categorical data (obtained via convolution operations) and numeric attributes to fully connected layers for predictions.
- *Wide and deep neural network* [44]: a neural network that includes a deep module and a wide module. The wide module applies a linear model to improve the sparsity of categorical features and robustness of the model via cross-product feature transformations. For the deep module, a feed-forward neural network is applied for numeric features to improve the memory ability of the network.
- *DeepWalk* [45]: references language model with random walk for unsupervised node representation learning.
- *Node2vec* [46]: an unsupervised learning method, which applies a random walk for node representation.
- *GCN* [41]: a basic graph convolution neural network for node classification.
- *GraphSage* [3]: a general inductive framework that generates embedding by sampling and aggregating features from nodes' local neighborhoods.
- *GAT* [3]: applies the attention mechanism for feature aggregation.
- *SGC* [47]: a lightweight GCN, which removes nonlinear activation functions and collapses weight matrices between consecutive layers.
- *GCNII* [48] is an extension of the vanilla GCN, which applies initial residual and identity mapping techniques to alleviate the over-smoothing problem.

5.2.3 Implementation

For a fair comparison, we set the learning rate $= 0.001$, regularizer=$1e-5$, batch size$= 500$, embedding size $d = 8$, and select Adam as the optimizer for all deep learning models. Concretely, we set the number of graph convolution layers as 4. The size of the output of graph convolution is kept the same as the input size, i.e.,$d \times m + n$. The size of FFN layers is $d \times m + n$ and 2. In addition, we set the inverse of regularization strength as $1e-4$ for logistic regression; the max depth is 5 for the decision tree; the number of trees for ensemble methods is 1000. For all the datasets, we extract 10% samples as a testing set, the left as the training set.

5.3 Performance comparison (RQ1)

Table 2 demonstrates the main results of all compared methods on three datasets. The major findings from the experimental results can be summarized as follows:

[6] https://www.heywhale.com/home/competition/56cd5f02b89b5bd026cb39c9/content/1.

Table 2 Experiment results on three datasets

Dataset	Home credit			Lending club			PPD		
	Recall	F1	AUC	Recall	F1	AUC	Recall	F1	AUC
LR	0.50163	0.48288	0.75361	0.54456	0.54161	0.69017	0.50000	0.48052	0.59130
DT	0.50075	0.48089	0.72718	0.53269	0.51864	0.69925	0.51874	0.52024	0.66042
RF	0.50115	0.48172	0.75074	0.52870	0.50943	0.70380	0.50000	0.48052	0.69286
XGB	0.60400	0.55801	0.77267	0.65707	0.58513	0.71616	**0.64745**	0.55218	0.69180
DNN	0.50000	0.47929	0.77183	0.50000	0.44317	0.70448	0.50000	0.48047	0.71577
CNN	0.50000	0.47922	0.77457	0.50000	0.44317	0.72961	0.50000	0.48050	0.71384
Wide & Deep	0.50000	0.47922	0.75824	0.50000	0.44317	0.70207	0.50000	0.48049	0.71109
DeepWalk	0.50475	0.49004	0.73620	0.50904	0.46885	0.61531	0.49820	0.47962	0.57000
Node2vec	0.50364	0.50370	0.61629	OOM	OOM	OOM	0.50168	0.48470	0.57227
GCN	0.50205	0.48348	0.74254	0.88461	0.87307	0.96684	0.50000	0.48051	0.66027
GraphSage	0.50084	0.48531	0.77232	0.73630	0.73955	0.90904	0.50000	0.48051	0.59451
GAT	0.50000	0.47920	0.76235	0.79781	0.83054	0.96094	0.50000	0.48051	0.61799
SGC	0.52008	0.51906	0.72078	0.87306	0.87731	0.89735	0.49982	0.48042	0.63664
GCNII	0.50000	0.47892	0.73059	0.86322	0.85776	0.96147	0.50000	0.48052	0.66953
MGCN	**0.61886**	**0.61811**	**0.78455**	**0.90588**	**0.87849**	**0.97546**	0.63109	**0.55987**	**0.71927**

We highlight the best performance and underline the sub-optimal results from the baselines for each comparison. OOM means Out of Memory

We can observe that our model MGCN outperforms all the baselines, especially on the Lending Club dataset (38% and 36% increases on recall and AUC respectively, compared with XGBoost). Specifically, MGCN achieves a remarkable improvement (up to 15.28%) over the strongest baselines in terms of recall, which demonstrates the effectiveness of the multi-view loan application graphs and the data augmentation module we proposed.

Compared with other machine learning and deep learning methods, graph convolution-based models do not have a significant superiority except for the Lending Club dataset. Compared with other datasets, the number of attributes on the Lending club is rather small and the user's information is also limited. Hence, we believe information aggregation from neighbors will be more beneficial for the scenarios in that the raw information is limited. Additionally, DeepWalk and Node2vec are the worst-performing methods among all the models, which illustrates that the graph information obtained by the unsupervised learning method does not work well for node classification. Furthermore, deep learning methods do not achieve significant improvements on all datasets; while, ensemble models, e.g., Random Forest and XGBoost, are still strong baselines and achieve comparable results.

For neural networks, Recall and F1 are virtually indistinguishable on all the datasets. Attribute to the imbalanced data distribution and missing values, it is difficult for all the baselines to achieve satisfactory results on all the datasets.

5.4 In-depth analysis of MGCN (RQ2)

5.4.1 Impact of embedding size

As mentioned above, the categorical attributes should be projected into a high dimensional latent space as embedding first. Thus, we put the lens on the impact of different embedding sizes ($d = 5, 8, 16, 32$) on our model. As shown in Fig. 4, when the embedding size is changed, there is no significant fluctuation for AUC on the three datasets; while, Recall varies more. In general, when setting the embedding size as eight, our model achieves promising performance. We argue a moderate embedding size could provide sufficient information capacity for the model while simultaneously avoiding the over-fitting problem.

5.4.2 Impact of gate mechanism and graph convolution layers

In order to verify the effectiveness of the gate mechanism in MGCN, we set the number of graph convolution layers in the range of $\{1, 2, 3, 4\}$ to evaluate the model's performance. In addition, we replace the gate mechanism with mean pooling for the ablation experiment. The other hyperparameters and model structure remain the same as described above. We show the performance comparison results in Fig. 5. It is obvious that the performance declines significantly when we replace the gate mechanism as mean pooling, indicating the usefulness of the gate mechanism

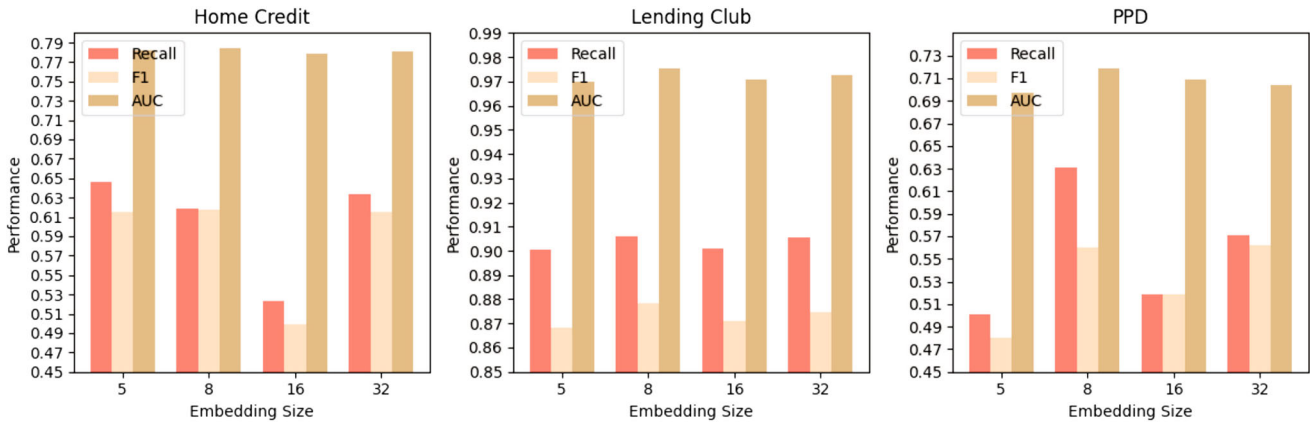

Fig. 4 Effect of embedding size $\{5, 8, 16, 32\}$ of our model on three public datasets

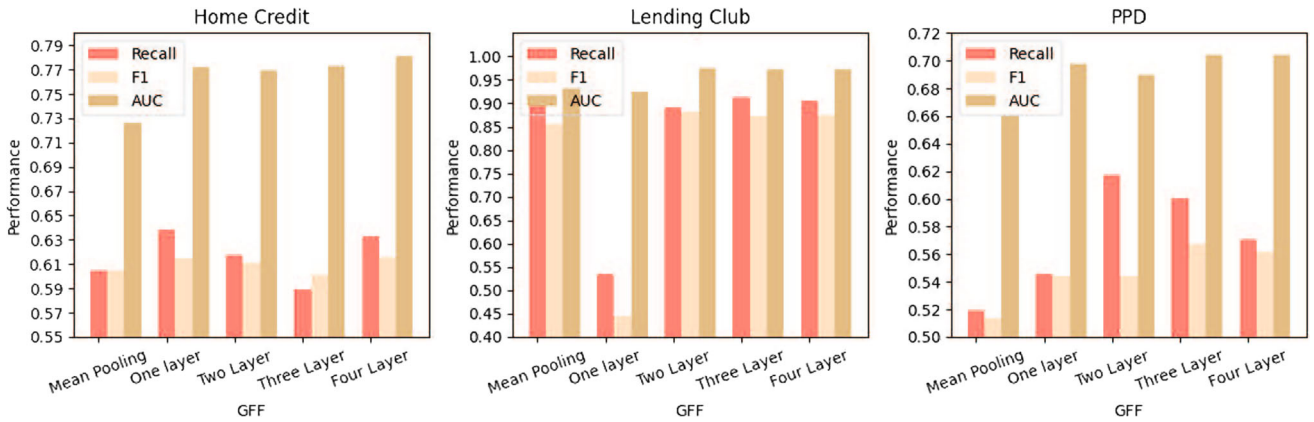

Fig. 5 Performance of MGCN with different numbers of graph convolution layers and mean pooling for feature aggregation

for information aggregation. Besides, as we increase the layer of graph convolution, the AUC also rises, which demonstrates the effectiveness of high-level semantic information for loan default detection.

5.4.3 Impact of shallow and depth features

As mentioned above, the deep layer contains rich semantic features and neighbor information. Hence, the feature representation from deep layers may facilitate the prediction of those samples whose neighbors are limited. In contrast, for the nodes which contain many neighbors, we assume the shallow layer's feature representation is sufficient for classification. Consequently, we devise the gate mechanism for hierarchical feature aggregation and alleviate the over-smoothing problem simultaneously. In this experiment, we dive into analyzing the effectiveness of the gate mechanism. Specifically, we first remove the gate mechanism and keep the other hyperparameters and model structure the same as described above. Then, applying one-hop layer feature representation (one graph convolution layer), multi-hops layers feature representation (four graph convolution layers), and mix-hops feature representation (four-hops feature representation for those nodes where the number of neighbors is less than 90% of the total nodes and one-hop feature representation for the remaining nodes) for loan default risk prediction. In addition, we also resort to one LAGs with the best result among all MLAGs to eliminate the effects of data augmentation for fair performance comparison. Results are shown in Table 3. Overall, our model achieves the best results on three datasets, which verifies the effectiveness of our proposed gate-based feature aggregation module. In addition, we can also observe that the mix-hop convolution achieves sub-optimal results on AUC, which confirms our aforementioned assumption that both deep and shallow features are equally important for default risk prediction.

5.4.4 Impact of multi-view loan application graphs

As mentioned in 4.1, the distribution between positive and negative samples is imbalanced. Thus, we construct multi-view loan application graphs for data augmentation. Specifically, by adjusting the similarity threshold between

Table 3 Impact of shallow layer, depth layer and mix layer representation on loan default prediction

Dataset	Metric	One-hop	Four-hops	Mix-hops	LAG
Home credit	Recall	0.50021	0.50205	0.50000	**0.60345**
	F1	0.48143	0.48348	0.47920	**0.60887**
	AUC	0.75145	0.74254	0.75691	**0.76321**
Lending Club	Recall	**0.88627**	0.88461	0.87716	0.88192
	F1	0.85839	**0.87307**	0.86405	0.86210
	AUC	0.96447	0.96684	0.96914	**0.97117**
PPD	Recall	0.50000	0.50000	0.50000	**0.63109**
	F1	0.48051	0.48052	0.48051	**0.55987**
	AUC	0.68627	0.67561	0.69666	**0.70650**

We highlight and underline the best and sub-optimal results

two nodes, we could change the graph structure and construct multiple loan application graphs, i.e., multi-view loan application graphs (MLAGs). Thus, the various node representations can also be yielded under different LAGs. Then, we collect the small sample representation from multi-view loan application graphs for model training. This process can be regarded as small sample augmentation. In the experiment, we set the δ in the range of $\{0.05, 0.1, 0.15, 0.2, 0.25\}$. Hence, five LAGs are constructed, and we consider different numbers of LAGs simultaneously for small sample augmentation and loan default prediction. As mentioned above, the other hyperparameters and model structures remain the same. The performance is presented in Fig. 6. Overall, the performance of MGCN improved as the number of LAGs increases. Especially for the Lending Club dataset, when we increase the number of LAGs from one to five, the AUC rises from 0.70432 to 0.97546 (38.5% improvement). However, for the Home Credit Default Risk and PPD

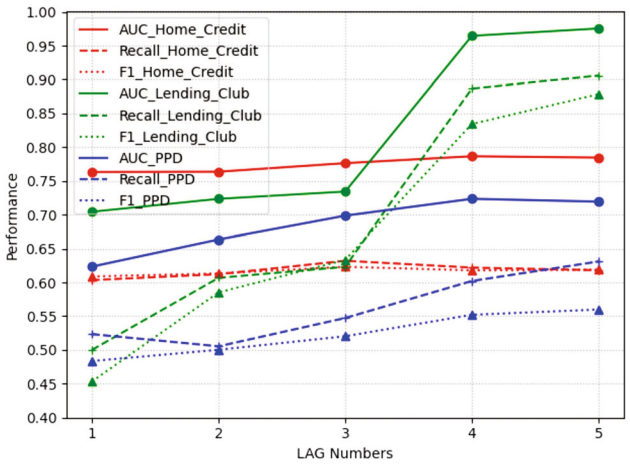

Fig. 6 Performance of MGCN with different number of LAGs 1, 2, 3, 4, 5 on three public dataset

datasets, when the number of LAGs is over 3, the performance does not improve notably. We believe this is because the size of the Home Credit Risk dataset and PPD dataset is relatively smaller than Lending Club (2, 260, 701 vs. 307, 511 for Home Credit Risk and 30, 000 for PDD, ref. Table 1 for details.). Consequently, a moderate graph number is sufficient to rebalance the data distribution. In general, we believe the data argumentation with multi-view graphs can mitigate the distribution imbalanced issue to some extent, but we should carefully adjust the number of graphs to maximize performance improvement with limited computational costs.

5.5 The effects of sampling strategies (RQ3)

As analyzed in section 5.4.4, the defaulter records are small samples in a realistic scenario, and the data distribution is also imbalanced. Following previous works, we, therefore, investigated the impact of the sampling strategy on the performance of our model. To avoid the effects incurred by data augmentation of MLAGs, we only consider one LAG for model training. Upsampling, downsampling, and SMOTE strategies are devised on train data. We define ξ as the ratio of minority class to majority class after resampling. Other hyperparameters and model structures keep the same as described above. As shown in Table 4, the sampling strategies do not contribute to performance improvement, in turn, SMOTE will lead to a decline in AUC. However, compared with Table 3, multi-view graphs data augmentation is more effective for performance improvement.

6 Conclusion

In this paper, we investigated the research of loan default risk prediction in a realistic scenario. By elaborately analyzing the characteristics of loan application records, we propose a graph convolution-based model MGCN to solve the problem. First, considering numerous missing values in raw data, we devise to construct loan application graphs. Consequently, similar loan application records could be aggregated as auxiliary information to improve the model's performance and robustness to the missing values. Besides, to alleviate the effect of imbalanced distribution, multi-view loan application graphs are proposed for data augmentation. Furthermore, we devise a novel gate mechanism to fuse the deep and shallow features generated by the graph convolution and alleviate the over-smoothing issues. Extensive experiments are conducted on three real-world loan default prediction datasets with large numbers of feature fields. The results demonstrate the effectiveness of multi-view loan application graphs and the gate

Table 4 Performance of different sampling strategies on three datasets. We highlight the best performance for each comparison

Dataset	Metric	Ratio	0.25	0.50	0.75	1.0
Home credit	Recall	Upsample	0.59243	0.50000	0.69988	0.69991
		Downsample	0.50000	0.65075	0.69579	**0.70285**
		SMOTE	0.50000	0.51350	0.51731	0.50000
	F1	Upsample	0.60301	0.47919	0.56838	0.53515
		Downsample	0.47918	**0.61846**	0.56568	0.55149
		SMOTE	0.47919	0.50833	0.51517	0.47918
	AUC	Upsample	**0.77493**	0.76185	0.76959	0.76677
		Downsample	0.77315	0.77237	0.77090	0.77001
		SMOTE	0.76647	0.76454	0.76104	0.75882
Lending club	Recall	Upsample	0.87896	0.89726	0.50000	0.84846
		Downsample	0.86974	**0.90453**	0.90392	0.86328
		SMOTE	0.88160	0.89483	0.86206	0.86465
	F1	Upsample	0.87215	0.86568	0.44467	0.82742
		Downsample	0.87435	**0.87443**	0.86412	0.82533
		SMOTE	0.87005	0.87554	0.85927	0.83774
	AUC	Upsample	0.97022	0.97083	0.94174	0.94174
		Downsample	0.97095	**0.97279**	0.96957	0.94516
		SMOTE	0.97053	0.97052	0.96381	0.94901
PDD	Recall	Upsample	0.50000	0.61156	**0.64837**	0.63997
		Downsample	0.50000	0.50000	0.50000	0.50000
		SMOTE	0.50000	0.51929	0.52793	0.50810
	F1	Upsample	0.48051	**0.57569**	0.49455	0.54587
		Downsample	0.48050	0.48048	0.48051	0.48051
		SMOTE	0.48050	0.51982	0.53223	0.50087
	AUC	Upsample	**0.71375**	0.71306	0.70454	0.70286
		Downsample	0.68783	0.69684	0.68317	0.65616
		SMOTE	0.70387	0.68739	0.52785	0.68045

mechanism. Although we achieve promising results for default risk prediction in this work, we fuse the attribute embedding with a simple concatenation operation. Further, we will explore the implicit relationships between any of two attributes and fuse the attributes embedding in a more sophisticated way.

Author contributions All authors contributed to the study conception and design. Material preparation, data collection and analysis were performed by ZL, YC, and XW; the first draft of the manuscript was written by ZL and XW and all authors commented on previous versions of the manuscript. All authors read and approved the final manuscript.

Funding Open Access funding enabled and organized by CAUL and its Member Institutions. Not applicable.

Data availability All datasets are available and can be accessed through the links posted in our paper.

Declarations

Conflict of interest The authors have no relevant financial or non-financial interests to disclose.

Ethical approval Not applicable.

References

1. Hajek P, Michalak K (2013) Feature selection in corporate credit rating prediction. Knowl-Based Syst 51:72–84

2. Tan F, Hou X, Zhang J, Wei Z, Yan Z (2018) A deep learning approach to competing risks representation in peer-to-peer lending. IEEE Trans Neural Netw Learn Syst 30(5):1565–1574
3. Hu B, Zhang Z, Zhou J, Fang J, Jia Q, Fang Y, Yu Q, Qi Y (2020) Loan default analysis with multiplex graph learning. In: Proceedings of the 29th ACM international conference on information & knowledge management, pp 2525–2532
4. Abrahams CR, Zhang M (2008) Fair lending compliance: intelligence and implications for credit risk management. Wiley, New York
5. Aleksandrova Y (2021) Comparing performance of machine learning algorithms for default risk prediction in peer to peer lending. TEM J 10(1):133–143
6. Malekipirbazari M, Aksakalli V (2015) Risk assessment in social lending via random forests. Expert Syst Appl 42(10):4621–4631
7. Bhattacharyya S, Jha S, Tharakunnel K, Westland JC (2011) Data mining for credit card fraud: a comparative study. Decis Support Syst 50(3):602–613
8. LeCun Y, Bengio Y, Hinton G (2015) Deep learning. nature 521(7553):436–444
9. Zhou F, Qi X, Xiao C, Wang J (2021) Metarisk: semi-supervised few-shot operational risk classification in banking industry. Inf Sci 552:1–16
10. Chi J, Zeng G, Zhong Q, Liang T, Feng J, Ao X, Tang J (2020) Learning to undersampling for class imbalanced credit risk forecasting. In: 2020 IEEE international conference on data mining (ICDM), IEEE, pp 72–81
11. Wang D, Zhang Z, Zhou J, Cui P, Fang J, Jia Q, Fang Y, Qi Y (2021) Temporal-aware graph neural network for credit risk prediction. In: Proceedings of the 2021 SIAM international conference on data mining (SDM), SIAM, pp 702–710
12. Wei S, Lv J, Guo Y, Yang Q, Chen X, Zhao Y, Li Q, Zhuang F, Kou G (2024) Combining intra-risk and contagion risk for enterprise bankruptcy prediction using graph neural networks. Inform Sci 659:120081
13. Guo X, Quan Y, Zhao H, Yao Q, Li Y, Tu W (2021) Tabgnn: Multiplex graph neural network for tabular data prediction. arXiv:2108.09127
14. Zhao H, Ge Y, Liu Q, Wang G, Chen E, Zhang H (2017) P2p lending survey: platforms, recent advances and prospects. ACM Trans Intell Syst Technol 8(6):1–28
15. Hand DJ, Henley WE (1997) Statistical classification methods in consumer credit scoring: a review. J R Stat Soc A Stat Soc 160(3):523–541
16. Crook JN, Edelman DB, Thomas LC (2007) Recent developments in consumer credit risk assessment. Eur J Oper Res 183(3):1447–1465
17. Stein RM (2005) The relationship between default prediction and lending profits: integrating roc analysis and loan pricing. J Bank Financ 29(5):1213–1236
18. Piramuthu S (1999) Financial credit-risk evaluation with neural and neurofuzzy systems. Eur J Oper Res 112(2):310–321
19. Serrano-Cinca C, Gutiérrez-Nieto B, López-Palacios L (2015) Determinants of default in p2p lending. PLoS ONE 10(10):0139427
20. Sohn SY, Kim JW (2012) Decision tree-based technology credit scoring for start-up firms: Korean case. Expert Syst Appl 39(4):4007–4012
21. Huang Z, Chen H, Hsu C-J, Chen W-H, Wu S (2004) Credit rating analysis with support vector machines and neural networks: a market comparative study. Decis Support Syst 37(4):543–558
22. Kim HS, Sohn SY (2010) Support vector machines for default prediction of smes based on technology credit. Eur J Oper Res 201(3):838–846
23. Wang Y, Wang S, Lai KK (2005) A new fuzzy support vector machine to evaluate credit risk. IEEE Trans Fuzzy Syst 13(6):820–831
24. Huang C-L, Chen M-C, Wang C-J (2007) Credit scoring with a data mining approach based on support vector machines. Expert Syst Appl 33(4):847–856
25. Lee T-S, Chiu C-C, Lu C-J, Chen I-F (2002) Credit scoring using the hybrid neural discriminant technique. Expert Syst Appl 23(3):245–254
26. Ravisankar P, Ravi V, Rao GR, Bose I (2011) Detection of financial statement fraud and feature selection using data mining techniques. Decis Support Syst 50(2):491–500
27. Wu T-C, Hsu M-F (2012) Credit risk assessment and decision making by a fusion approach. Knowl-Based Syst 35:102–110
28. Chi B-W, Hsu C-C (2012) A hybrid approach to integrate genetic algorithm into dual scoring model in enhancing the performance of credit scoring model. Expert Syst Appl 39(3):2650–2661
29. Ong C-S, Huang J-J, Tzeng G-H (2005) Building credit scoring models using genetic programming. Expert Syst Appl 29(1):41–47
30. Quah JT, Sriganesh M (2008) Real-time credit card fraud detection using computational intelligence. Expert Syst Appl 35(4):1721–1732
31. Bellotti T, Crook J (2013) Forecasting and stress testing credit card default using dynamic models. Int J Forecast 29(4):563–574
32. Yang Y (2007) Adaptive credit scoring with kernel learning methods. Eur J Oper Res 183(3):1521–1536
33. Huang J-J, Tzeng G-H, Ong C-S (2006) Two-stage genetic programming (2sgp) for the credit scoring model. Appl Math Comput 174(2):1039–1053
34. Malhotra R, Malhotra DK (2003) Evaluating consumer loans using neural networks. Omega 31(2):83–96
35. Yeh I-C, Lien C-H (2009) The comparisons of data mining techniques for the predictive accuracy of probability of default of credit card clients. Expert Syst Appl 36(2):2473–2480
36. Babaev D, Savchenko M, Tuzhilin A, Umerenkov D (2019) Et-rnn: applying deep learning to credit loan applications. In: Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining, pp 2183–2190
37. Kvamme H, Sellereite N, Aas K, Sjursen S (2018) Predicting mortgage default using convolutional neural networks. Expert Syst Appl 102:207–217
38. Liu Q, Liu Z, Zhang H, Chen Y, Zhu J (2021) Mining cross features for financial credit risk assessment. In: Proceedings of the 30th ACM international conference on information & knowledge management, pp 1069–1078
39. Cui L, Bai L, Wang Y, Bai X, Zhang Z, Hancock ER (2016) P2p lending analysis using the most relevant graph-based features. In: Joint IAPR international workshops on statistical techniques in pattern recognition (SPR) and structural and syntactic pattern recognition (SSPR), Springer, pp 3–14
40. Zhong Q, Liu Y, Ao X, Hu B, Feng J, Tang J, He Q (2020) Financial defaulter detection on online credit payment via multi-view attributed heterogeneous information network. In: Proceedings of The Web conference 2020, pp 785–795
41. Kipf TN, Welling M (2016) Semi-supervised classification with graph convolutional networks. arXiv:1609.02907
42. Li Q, Han Z, Wu X-M (2018) Deeper insights into graph convolutional networks for semi-supervised learning. In: Thirty-second AAAI conference on artificial intelligence
43. Oono K, Suzuki T (2019) Graph neural networks exponentially lose expressive power for node classification. arXiv:1905.10947
44. Cheng H-T, Koc L, Harmsen J, Shaked T, Chandra T, Aradhye H, Anderson G, Corrado G, Chai W, Ispir M, et al (2016) Wide & deep learning for recommender systems. In: 1st Workshop on deep learning for recommender systems, pp 7–10

45. Perozzi B, Al-Rfou R, Skiena S (2014) Deepwalk: online learning of social representations. In: Proceedings of the 20th ACM SIGKDD international conference on knowledge discovery and data mining, pp 701–710
46. Grover A, Leskovec J (2016) node2vec: Scalable feature learning for networks. In: Proceedings of the 22nd ACM sigkdd international conference on knowledge discovery and data mining, pp 855–864
47. Wu F, Souza A, Zhang T, Fifty C, Yu T, Weinberger K (2019) Simplifying graph convolutional networks. In: International conference on machine learning, PMLR, pp 6861–6871
48. Chen M, Wei Z, Huang Z, Ding B, Li Y (2020) Simple and deep graph convolutional networks. In: International conference on machine learning, PMLR, pp 1725–1735