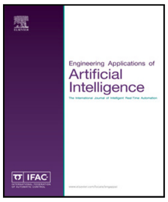
Contents lists available at ScienceDirect

Engineering Applications of Artificial Intelligence

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



WIGNN: An adaptive graph-structured reasoning model for credit default prediction

Zhipeng Yan [a], Hanwen Qu [a], Chen Chen [a], Xiaoyi Lv [b,*], Enguang Zuo [c,*], Kui Wang [d], Xulun Cai [d]

[a] *College of Computer Science and Technology, Xinjiang University, Urumqi, 830046, Xinjiang, China*
[b] *College of Software, Xinjiang University, Urumqi, 830046, Xinjiang, China*
[c] *College of Intelligent Science and Technology (Future Technology), Xinjiang University, Urumqi, 830046, Xinjiang, China*
[d] *Xinjiang Uygur Autonomous Region Rural Credit Cooperatives Union, Urumqi, 830002, Xinjiang, China*

ARTICLE INFO

Keywords:
Credit default
Graph neural networks
Weighted connectivity
Data imbalance

ABSTRACT

In credit default prediction, the main challenge is handling complex data structures and addressing data class imbalance. Given class imbalance and multi-dimensional data, general models find it difficult to fully explore the deep interdependencies within the data and the interaction effects between local and global. To overcome these challenges, this study proposes a Weighted Imbalanced Graph Neural Network (WIGNN) model that integrates adaptive graph structure inference with differential weight connectivity strategy, and the model solves the existing problems from the perspective of differential weight connectivity and graph balancing. Here, the weight connection uses the Gaussian kernel function to refine calculations and an adaptive percentile method to adjust sparsity, improving the understanding and efficiency of mining data connections. The weighted graph generated by this method can reflect the interaction between nodes and improve the model's ability to analyse complex data structures. Based on this weighted graph, the graph imbalance module adopts a reinforcement learning-driven neighbour sampling strategy to adjust the sampling threshold automatically, optimizes the node embedding through message aggregation, and combines with a cost-sensitive matrix to improve classification accuracy and cost-effectiveness of the model on diverse credit datasets. We applied the WIGNN model to six real and class-imbalanced credit datasets, comparing it with 11 mainstream credit default prediction models. Evaluated using metrics Area Under the Curve (AUC), Geometric Mean (G-mean), and Accuracy. The results show that WIGNN significantly outperforms other models in handling class imbalance and graph sparsity, demonstrating its potential in financial credit applications.

1. Introduction

In the era of big data, the financial sector faces an unprecedented flood of information, in which a complex situation of challenges and opportunities lies. Particularly in the field of credit, financial credit, a vital pillar of the market economy and a core component of the modern financial system, has far-reaching implications for supply chain maintenance, securities investment, bank loan assessment and even the formulation of government economic policies. The Basel Committee's Core Principles for Effective Banking Supervision in 1997 made it clear that credit risk is an essential challenge that financial institutions must face and is directly related to the soundness and safety of financial markets (Basel Committee on Banking Supervision, 2006). Credit default forecasting becomes an adequate protection mechanism to balance the growing demand for credit and the potential risks of credit operations, which is the key to maintaining financial stability and promoting sustainable development.

Credit default forecasting is assessing the likelihood that a borrower will not be able to repay a debt on time in the future. It helps financial institutions reduce credit risk by analysing the borrower's economic situation, credit history and behavioural patterns. Methods in this field fall into two main categories: statistics-based and machine learning-based. Statistical-based credit default aims to use classical statistical methods to identify critical factors that are closely related to default prediction. For credit default based on statistics, Durand first introduced discriminant analysis in credit evaluation in 1941 (Durand, 1941), taking borrower information as a variable to identify whether it is a non-performing loan. However, discriminant analysis requires strict statistical assumptions, and the linearity and independence between assumed variables are challenging to meet in practise. In the 1950s, Bill

* Corresponding authors.
E-mail addresses: 107552203955@stu.xju.edu.cn (Z. Yan), quhanwen@stu.xju.edu.cn (H. Qu), chen_chen@stu.xju.edu.cn (C. Chen), lvxiaoyi@xju.edu.cn (X. Lv), zeg@xju.edu.cn (E. Zuo), 107552103632@stu.xju.edu.cn (K. Wang), 107552203968@stu.xju.edu.cn (X. Cai).

https://doi.org/10.1016/j.engappai.2024.109597
Received 30 May 2024; Received in revised form 22 September 2024; Accepted 30 October 2024
Available online 11 November 2024

Fair and Earl Isaac developed the Fair Isaac Corporation (FICO) credit scoring system, which converts the probability of corporate default into a continuous variable through a logistic regression model. Its transparent logic and strong interpretability have become a mainstream credit scoring tool. However, the model assumes a linear relationship between variables and may face the problem of multicollinearity, which limits its accuracy in dealing with complex variable relationships. In 2023, Yang et al. (2023)used neural networks with L_1 and L_2 regularization for feature selection, effectively reducing the data dimensionality but difficult to interpret.

Despite the continuous improvement of traditional credit models, there are still limitations in dealing with deep data and non-traditional credit information. Graph Neural Networks (GNNs) show great potential in these areas. GNNs can reveal complex dependencies between entities and mine deep information from the data structure instead of relying on feature attributes as previous models did. Currently, GNNs have been successful in several application areas, such as community trust assessment, recommender systems, and traffic flow prediction, proving the possibility of their wide application. For example, the Graph Attention network with Cost-sensitive Boosting (GAT-COBO) model improves minority class identification in telecom fraud detection by combining graph attention mechanism and cost-sensitive learning (Hu et al., 2024). Adaptive Graphs with Self-Supervised Learning (AGSSL) improves data representation by self-supervised learning and graph structure optimization for traffic prediction (Wang et al., 2024). Liu et al. use heterogeneous GNNs to identify malicious accounts (Maron et al., 2019). However, research on applying graph neural networks in credit is still relatively limited. The direct use of GNNs for classification tasks has not been as effective as expected, mainly attributed to the complex data structure of credit, which requires a careful adaptation of the model structure to suit specific data characteristics. Credit data covers complex interactions between individuals and organizations and carries rich unstructured information, which requires the model to have deep parsing capabilities for the data. Simply applying GNN makes it challenging to uncover the data's deep interdependencies and local–global interactions. Therefore, optimizing the GNN structure through inference becomes crucial, aiming to improve the completeness of the GNN's representation of the raw data, enabling it to identify potential features better. At the core of the inference lies the enhanced depth of the model's understanding of the data structure and the ability to propose strategies from multiple perspectives, such as node weights, connection strengths, and topology, especially when dealing with class-imbalanced data.

Therefore, in this paper, to deal with complex data structures and data class imbalance problems, we propose highly explanatory credit default prediction based on adaptive graph structure inference, which we refer to as WIGNN. WIGNN is divided into a weighting module and a graph imbalance learning module. Firstly, structured tabular data is converted to graph structure, and the Gaussian kernel function is used to convert the Euclidean distances between data points into similarity scores, which are used as the weights of node connections. Secondly, the model also controls the connection sparsity of the graph through an adaptive threshold adjustment mechanism to optimize the graph's expressiveness. In addition, to cope with the class imbalance problem, the model introduces a graph imbalance learning module, employs the neighbour sampling technique of reinforcement learning to balance the disparity in the number of positive and negative class samples, and combines with an automated cost-sensitive matrix to solve the model bias problem. In brief, this paper has the following highlights:

1. A nominal WIGNN model is proposed, which is dedicated to solving the problem of complex data structure and data class imbalance, and this method explores the differentiated weight connection and coping graph imbalance strategy for the first time in the field of credit defaults, which opens up a new way to improve the accuracy of credit decision-making.
2. We designed a differentiated weighted connection module that employs a Gaussian kernel for precise weight computation, effectively reducing feature oversmoothing. An adaptive threshold adjusts graph sparsity dynamically, ensuring structural stability and connection precision.
3. This study proposes a dual reduction strategy to balance node counts and adjust minority class weights. By leveraging reinforcement learning-based neighbourhood sampling and cost-sensitive matrix optimization, it effectively resolves graph imbalance in credit scenarios, significantly enhancing model performance on imbalanced datasets.
4. The performance of the WIGNN model is evaluated by comparing the performance of 11 mainstream credit default models on several real, class-imbalanced credit datasets using AUC, G-mean and other metrics. The results confirm that the WIGNN model has significant advantages in achieving differential weight connections and stabilizing graph sparsity. It outperforms other reference models in terms of comprehensive performance.

The remainder of this paper is organized as follows: Section 2 reviews the development of credit default prediction models and related work on Graph Neural Networks in financial application scenarios. Section 3 first introduces the framework of our proposed method, WIGNN, and then provides detailed descriptions of the weighted connectivity and graph imbalance modules. Section 4 outlines the experimental setup and evaluation criteria and comprehensively discusses the experimental results. Section 5 presents the study's conclusions and discusses its limitations.

2. Related works

2.1. Credit default prediction research model

The credit default risk prediction field has a long and extensive history of research. Initially, classical statistical methods were used to identify critical factors closely related to default prediction: univariate analysis of financial ratios was introduced by William Beaver (Beaver, 1966), which used single financial ratios, such as working capital flows over liabilities and net profit over total assets, to predict corporate financial distress. Subsequently, Edward I. Altman's (Altman, 1968) Z-score multivariate model marked an innovation in financial ratio analysis in predicting corporate bankruptcy by creating a Z-score model that accurately distinguishes between bankrupt and non-bankrupt firms. Given that the Logit regression model handles corporate default prediction using logistic functions, it is superior to the treatment of multivariate discriminant models. Following this, Altman and Sabato (2007) developed a Logit model for small and medium-sized enterprise (SME) default prediction that exceeded the prediction accuracy of the generalized model by 30%. In addition, non-parametric statistical methods such as Linear Discriminant Analysis (LDA) (Barboza et al., 2017; Moscatelli et al., 2020; Stefania et al., 2023), K-Nearest Neighbours (KNN) (Ratnam, 2012) and Naive Bayes (NB) (Bhattacharya et al., 2019; Cao et al., 2022) are widely used in credit scoring, and these methods provide new perspectives and practical tools for solving complex problems in financial forecasting. Among these approaches,Bayesian models offer a robust probabilistic framework for addressing complex dependencies and uncertainty. Giudici (2001) (Giudici, 2001) the first to apply Bayesian network models to credit scoring, utilizing Reversible Jump Markov Chain Monte Carlo (RJ-MCMC) techniques to capture both direct and indirect relationships between variables, reducing model bias and improving predictive accuracy. Agosto et al. (2023) citefurther expanded this approach by integrating multiple ESG data sources, using Bayesian inference to assign weights based on predictive accuracy, optimizing the fairness and interpretability of the model. By combining the S.A.F.E. method, they achieved a balance between sustainability and high predictive performance.

In recent years, machine learning methods have become a core research area for credit default risk prediction. Logistic regression is widely used in credit default prediction because of its good interpretability and ability to understand the key factors affecting default (Dong et al., 2010; Jha et al., 2012; Nikolic et al., 2013). Random forests are critical in credit default prediction due to excellent classification performance and adaptability to large datasets but with weak interpretability (Bahnsen et al., 2013; Dal Pozzolo et al., 2014; Malekipirbazari and Aksakalli, 2015; Nami and Shajari, 2018). Extreme Gradient Boosting (XGBoost) effectively handles complex nonlinear problems through multilevel decision trees and gradient-boosting techniques. Still, it highly depends on the data quality, and data bias significantly affects the results (Wang et al., 2022; Mushava and Murray, 2022). There are also models integrating Synthetic Minority Over-sampling Technique, Long Short-Term Memory networks and Adaptive Boosting, which effectively mitigate the problem of inaccurate prediction in data imbalance scenarios (Shen et al., 2021). In addition, the Particle Swarm Optimization and Extreme Gradient Boosting (PSO-XGBoost) model was applied to assess the risk of loan default (Rao et al., 2023). Support Vector Machines (SVMs) (Cowden et al., 2019; Luo et al., 2020; Sun et al., 2021), Artificial Neural Networks (ANNs) (du Jardin, 2021), and Ensemble Learning (EL) (Sun et al., 2020) have also been widely used in credit risk assessment. All of these techniques have demonstrated significant effectiveness in the credit domain, as shown in the related literature in Table 1. Although previous models have made significant progress in certain classification tasks, they usually fail to dig deeper into the internal dependencies of the data and their local–global interactions. In dealing with category imbalance and multidimensional data, this study employs graph methods to enhance the model's predictive power and improve the depth of data analysis by uncovering potential connections within the data structure. Additionally, it effectively addresses imbalances in the data structure.

2.2. GNN field applications

GNN processes information through graph structures to deeply explore the implicit information properties among entities and understand the complex interactions and dependencies among customers. Higher-order Graph Attention Representation (HGAR) optimizes node importance capture in financial networks by learning to embed network-guaranteed loans using binary roles and higher-order adjacency metrics. Its superiority and usefulness are verified in a natural loan risk control system (Cheng et al., 2019). Quantity and Topology Imbalance-aware Heterogeneous Graph Neural Network (QTIAN-GNN) effectively solves topological imbalance by employing multi-layered label-aware neighbourhood strategies and class-semantic representations and reasonably predicts corporate bankruptcy through non-financial factors (Liu et al., 2023). Similarly, the Graph Attention Network (GAT)–based credit default risk prediction model employs GAT techniques to process user relationship graphs constructed from loan and personal data, extracting critical neighbourhood information (Zhou et al., 2023). K-Nearest Neighbours Graph Neural Network (KNN-GNN) integrates a graph representation learning approach that significantly improves credit risk prediction even when only internal information is exploited by combining an unsupervised graph transformation and a supervised node classification technique to improve credit risk prediction accuracy (Shi et al., 2024). Although current research has provided valuable insights into the deep connections between entities from a graph perspective, the weight connections during the composition process must be differentially treated to refine the strength of edges and improve the graph representation. In addition, the graph balancing strategy can effectively solve the class imbalance problem of credit data from the graph perspective and achieve controllable and adjustable graph sparsity through reasonable adaptive threshold adjustment to optimize the effectiveness of the overall network structure.

In addition, in the field of credit research, the application of graph structures and Graph Neural Networks (GNNs) in handling complex data has gradually gained attention. GNNs exhibit significant advantages in dealing with unstructured features and complex dependencies. However, during the data transformation process, simplified processing often fails to fully account for class imbalance characteristics in the data, leading to graph structures that still reflect the imbalances in the original data. This imbalance persists in the graph data and may affect model performance, causing the model to favour the majority class and overlook critical information from the minority class. The proposed model in this study addresses this by employing differentiated connections and precisely capturing neighbourhood similarity, enhancing the integrity of the graph representation. Additionally, the model addresses data imbalance from both node count and weight perspectives, optimizing the entire data processing workflow and achieving end-to-end optimization, emphasizing the importance of deeply analysing problems from the graph data perspective.

3. Models and methods

3.1. Modelling framework

This study proposes a highly explanatory credit default prediction model based on adaptive graph-structured reasoning, aiming to deepen insights into credit default behaviour through highly explanatory analysis. In Fig. 1, the core architecture incorporates differential weight connectivity and graph imbalance learning. Among them, the weight connection uses the Gaussian kernel mapping as the transformation mechanism to capture node interactions from a high-dimensional perspective and significantly improve the differentiation between data populations. The adaptive thresholding effectively refines the sparsity control of the model. For graph imbalance learning, a reinforcement learning-driven neighbour sampling strategy is adopted to automatically adjust the sampling threshold to balance the category distribution and obtain node embeddings through the message aggregation mechanism of GNN. The node embeddings are optimized in the cost-sensitive matrix training session to ensure the classification accuracy and cost-effectiveness of the model on diverse credit datasets.

Tabular to GNN Module: Credit tabular data consists of multiple features $X_1, X_2, \dots, X_n$ and corresponding labels y. First, by calculating the L2 distance between data points, the similarity between nodes is determined. Next, the distance is transformed into weights through the Radial Basis Function (RBF), which enhances the connection of neighbouring nodes. The connection sparsity is controlled by percentile thresholds. After this process, a graph dataset with similarity weights is obtained, which is suitable for entry into the GNN for subsequent classification tasks.

Figure Unbalanced Module: For the graph data, to address the issue of data imbalance, reinforcement learning-based neighbour sampling is used to balance the number of positive and negative samples through top sampling. By applying mean aggregation, the relationships and features between nodes are effectively captured. A cost-sensitive learning mechanism is implemented, placing particular emphasis on those categories that are scarce but important in the dataset, thus improving the model's predictive performance on these minority classes. The final output embeddings can be used for various downstream tasks such as classification, regression, or other forms of data analysis, reflecting the model's understanding and processing of the entire dataset structure.

3.2. Transfer module

Unstructured data is framed using similarity or distance. A graph can be defined as $G(V, E, X, y)$, where V represents the set of nodes in the graph, E represents the set of edges, X represents the attributes of the nodes, y represents the labels of the nodes, and n represents

Table 1
Summary of cited literature.

Credit model	Methodology	Vintages	Literature
Statistics-based	LDA	2017, 2020, 2023	Barboza et al. (2017), Moscatelli et al. (2020), Stefania et al. (2023)
	KNN	2012	Ratnam (2012)
	NB	2017, 2022	Bhattacharya et al. (2019), Cao et al. (2022)
Machine learning-based	HGAR	2019	Cheng et al. (2019)
	SVM	2019, 2020, 2021	Cowden et al. (2019), Luo et al. (2020), Sun et al. (2021)
	EL	2020	Shen et al. (2021), Sun et al. (2020)
	ANN	2021	du Jardin (2021)
	PSO-XGBoost	2023	Rao et al. (2023)
	GAT	2023	Zhou et al. (2023)
	KNN-GNN	2024	Liu et al. (2023)

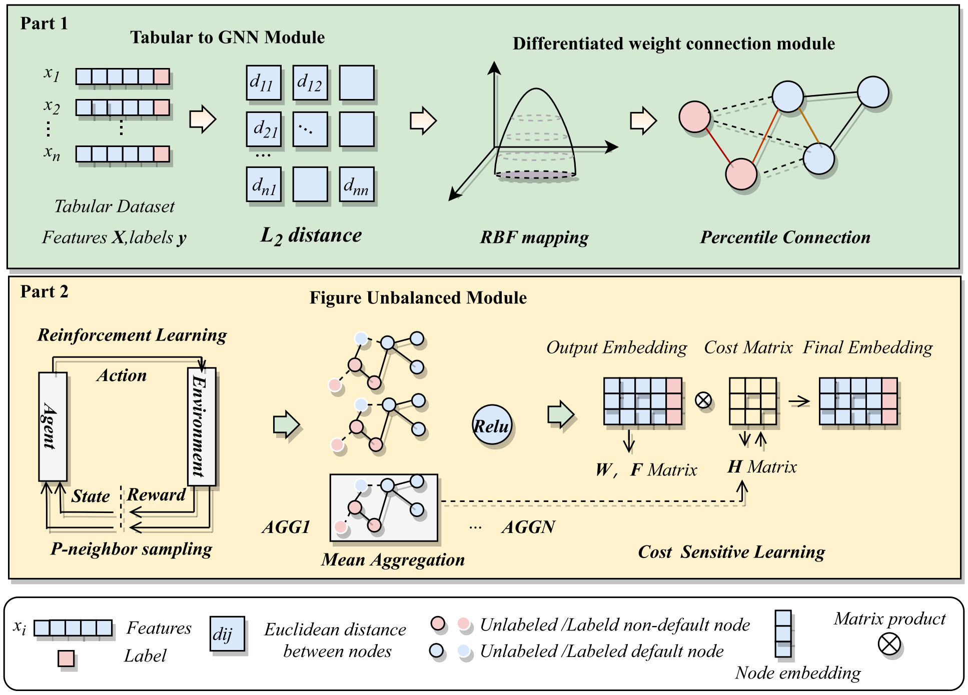


Fig. 1. WIGNN credit default prediction model with adaptive weights and graph imbalance learning.

the total number of nodes in the graph. This study uses a similar graph construction method, where each customer is defined as a graph node V, with features as node attributes X, and the default status is marked with node labels y (1 for default and 0 for non-default). Node connections are determined by comparison with a threshold to generate an unweighted adjacency matrix representing the connection states. The method is unsupervised learning and does not rely on a priori category information. The formula (1) for the edge connection relationship is shown as follows:

$$a_{ij} = I(\text{threshold} - a_{ij}) \tag{1}$$

In the equation, a_{ij} represents the connectivity between the node i and the node j, and I is an indicator function that assigns a connection based on whether the measure of distance or similarity exceeds a certain threshold.

3.3. Differential weight connections

This study selects Euclidean distance as the graph construction method, primarily due to its simplicity, intuitiveness, and high computational efficiency on large-scale datasets. However, this simplified method also introduces some significant limitations. First, Euclidean distance only measures the straight-line distance between data points, making it difficult to effectively capture complex nonlinear relationships or underlying geometric structures in the data. When dealing with complex datasets, this may cause the model to oversimplify the representation of interactions between data points. Moreover, Euclidean distance assumes independence between features, ignoring correlations among them. As a result, graph structures built based on Euclidean distance may experience information loss and bias when expressing the similarity between data points, ultimately affecting the overall performance of the model. Additionally, the Gaussian kernel function can map the Euclidean distance between nodes into nonlinear similarity weights, reducing the excessive smoothing of information caused by the adjacency matrix. By introducing a differentiated weight connection strategy through the Gaussian kernel function, the Euclidean distance between nodes is nonlinearly transformed into similarity weights. This method is particularly suitable for complex or hard-to-handle datasets that are difficult to separate using linear methods, by mapping the data into a higher-dimensional space. Specifically, the Gaussian kernel function, as a mapping mechanism, can lift the data from low-dimensional space to high-dimensional space, constructing a graph structure that reflects complex data interactions. As shown in Fig. 2, the distribution originally measured by Euclidean distance is relatively scattered, but after the Gaussian kernel function mapping, the data points exhibit a tighter clustering trend in the high-dimensional space, and the similarity between nodes is effectively enhanced.

Furthermore, the Gaussian kernel function can map the Euclidean distance between nodes into nonlinear similarity weights, effectively alleviating the oversmoothing phenomenon in the adjacency matrix.

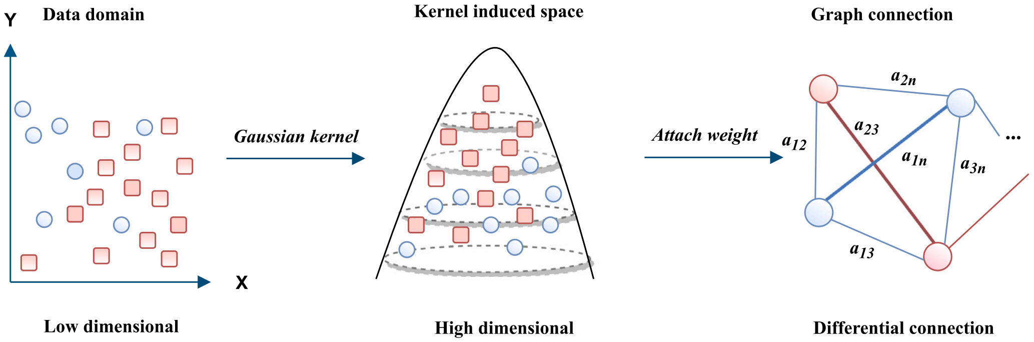


Fig. 2. Gaussian kernel differentiated connection.

Specifically, the Gaussian kernel function assigns higher similarity weights to neighbouring nodes and reduces the connection strength for distant nodes, making the relationships between adjacent nodes closer and those between distant nodes more sparse. This differentiated connection approach avoids global oversmoothing in the adjacency matrix, thereby preserving local differences between nodes. Particularly in imbalanced datasets, it can better distinguish between normal samples and minority default samples.

The Gaussian kernel mapping strategy used in this study maintains the original similarity properties of the data while refining the degree of similarity. The study uses the standard deviation of the actual dataset as the bandwidth ϕ, which allows the mapping process to adapt to the distribution characteristics of the original data. Without altering the intrinsic properties of the data, the mapping process becomes more sensitive to the variability and structural complexity of the data, accurately revealing and strengthening the intrinsic similarity relationships between data points.

Additionally, this study regulates the sparsity of the graph through the combination of the Percentile connection method. Unlike traditional distance-based connection strategies, Percentile connections connect node pairs that meet a certain percentile threshold, ensuring that the graph's sparsity no longer depends on a fixed distance threshold. This method controls sparsity while ensuring the reasonableness of connections, reducing noise interference. By combining with the Percentile strategy, node connections can be adaptively filtered to ensure that high-similarity node connections are retained while low-similarity node connections are effectively removed. This method prevents the graph from becoming overly sparse or overly dense, allowing the constructed graph to better reflect the true distribution of the data.

3.3.1. Calculation of similarity scores

In this model, node feature conversion aims to make the node features of the same category in the high-dimensional space closer and the node features of different categories more distant. For this purpose, a fully connected network is used as the node feature converter and combined with a Gaussian kernel function to convert the computed Euclidean distances into similarity scores between nodes. Percentile thresholds are introduced to adaptively control the connection density of the graph when determining whether connections are formed between nodes. The loss of the conversion process is optimized by minimizing the cross-loss entropy loss function, allowing label information to learn to distinguish between different classes of node features.

Specifically, the input embedding $\mathbf{h}_v$ of the features of the node v transformed by the fully connected layer can be expressed by the following formula (2):

$$\mathbf{h}_v \;= \sigma(\mathbf{W} x_v) \tag{2}$$

σ represents the activation function, which in this case is ReLU, and $\mathbf{W}$ denotes the network weights to be learned. x_v represents the input feature vector of node v.

After calculating the Euclidean distance $D(v, v')$ between node v and its neighbouring node v', the distance is converted to similarity score $\mathrm{Sim}(v, v')$ by the Gaussian kernel function. This process is used to evaluate the similarity between nodes, the specific transformation formula (3)–(4):

$$D(v, v') = \|\mathbf{h}_v - \mathbf{h}_{v'}\|_2 \tag{3}$$

$$\mathrm{Sim}(v, v') = \exp\left(-\frac{D(v, v')}{2\phi^2}\right) \tag{4}$$

Here is the bandwidth parameter of the ϕ Gaussian kernel function, which controls the decay rate of this similarity.

An adaptive mechanism is employed to define the connections between nodes by dynamically selecting different percentile thresholds, allowing the model to adjust adaptively based on the specific distribution and density of the dataset. This improves the graph structure's quality and the information's accuracy. To this end, an adaptive percentile threshold is introduced to select edges where the score exceeds a certain percentile to construct the graph. The specific formula (5) is as follows:

$$T = \{(v, v') \mid \mathrm{Sim}(v, v') > \mathrm{Percentile}(S, \theta)\} \tag{5}$$

S represents the set of similarity scores between all node pairs, quantifying the similarity between them. θ denotes the percentile parameter used to define the threshold, such that only node pairs with similarity scores exceeding this percentile are selected to form edges in the graph.

The process defines L_{sim} to minimize the difference between the transformed node embeddings and the similarity scores used in graph construction. Thus, our optimization objective formula (6) :

$$L_{\mathrm{total}} = L_{\mathrm{trans}} + \lambda L_{\mathrm{sim}} \tag{6}$$

L_{trans} refers to the conventional cross-entropy loss based on label information, utilized to train the fully connected layer; L_{sim} is a loss based on similarity scores, ensuring that the graph structure produced by the model aligns with the expected similarity scores; and λ is a parameter that balances these two loss components. In this manner, the model concurrently considers the transformation of node features and the construction of graph structure during training, thereby establishing meaningful distinctions between different categories.

3.4. Graph imbalance learning

Credit data exhibit class imbalance characteristics. Despite adopting GNN models, the issue of model bias arising from this problem remains inadequately addressed. Class imbalance manifests as quantitative and topological imbalances after converting tables into graphs. The former refers to a disparity in the number of instances per class. At the same time, the latter pertains to the uneven presence of nodes from different classes within the graph structure, resulting in a non-uniform distribution of node connections. This non-uniformity can lead to situations

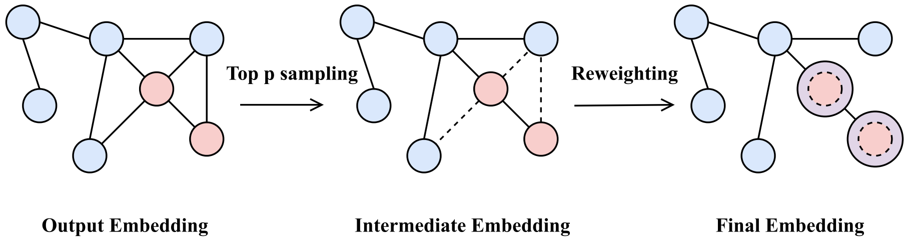


Fig. 3. Graph imbalance learning strategy.

where majority class nodes surround minority class nodes, receiving neighbour information inconsistent with their own. This can cause the minority signals to be overwhelmed during message passing and aggregation, leading to model bias.

To our knowledge, this is the first instance of employing graph-balancing learning in credit default prediction. This study addresses graph imbalance by applying adaptive neighbour sampling and a cost-sensitive matrix. As illustrated in Fig. 3, the methodology systematically addresses graph imbalance from two aspects: the number of classes and the weight of classes.

3.4.1. Neighbourhood sampling

To cope with the class imbalance problem of GNN models in dealing with this kind of data and to overcome the possible bias of GNN models in dealing with this kind of data, the neighbour sampling strategy of BMAB with added reinforcement learning is introduced, which aims to accurately select neighbour nodes with high similarity to the central node by dynamically balancing the difference in the number of classes sampled. The key of the BMAB algorithm lies in the state, action, reward function, and termination state, i.e., $B(Z, A, R, T)$. The specific steps of the neighbour sampling method are as follows:

Initial and termination states: Neighbour sampling is conducted through the reinforcement learning BMAB (Bernoulli Multi-Armed Bandit) method to select appropriate neighbours. Within the reinforcement learning framework, we define the state representation Z as the current status of the model, which is typically based on a metric of average similarity $S_{avg}^{(e)}$ between nodes. The formula (7) for calculating the average similarity $S_{avg}^{(e)}$ is as follows:

$$S_{avg}^{(e)} = \frac{\sum_{v' \in \mathcal{N}_{\min}^{(e)}} \mathrm{Sim}(v, v')^{(e)}}{|\mathcal{E}(e)|} \tag{7}$$

Here, $E(e)$ denotes the edges of the training set in the first e cycle, and $sim(v, v')^{(e)}$ denotes the similarity between node v and its neighbour v' in the first e cycle. The state is defined as $Z = \{\mathrm{z}_1, \mathrm{z}_2\}$. Here, z_1 is the average node similarity under the current period, which is more significant than the average node similarity under the previous period, and vice versa, which defaults to z_2.

Adjustment of movement and sampling criteria: In the current state of Z, the reinforcement learning model chooses action A to modify the sampling threshold p. Action A can either raise or lower the value of p. The mapping function defines the action. The association between the action A and the state Z is defined by the mapping function $M(Z)$, as shown in formula (8) :

$$A = M(Z) = \begin{cases} \pi + \delta, & \mathrm{z}_2 \\ \pi - \delta, & \mathrm{z}_1 \end{cases} \tag{8}$$

where δ is the step size for adjusting the sampling probability p.

Incentives: The incentive mechanism $R(\mathrm{p}, A)^{(\mathrm{e})}$ is defined based on the average similarity between two consecutive cycles. The specific formula is as follows:

$$R(p, A)^{(e)} = \begin{cases} +1, & S_{avg}^{(e-1)} \le S_{avg}^{(e)} \\ -1, & S_{avg}^{(e-1)} > S_{avg}^{(e)} \end{cases} \tag{9}$$

Here, the model is encouraged to increase the average similarity between nodes, which is achieved by appropriately adjusting the sampling probability p.

Termination conditions and thresholds are fixed: When the cumulative change in reward is less than the preset sampling threshold in consecutive n cycles, the optimal sampling threshold p is found by default, and the reinforcement learning process is terminated. The formula (10) is as follows:

$$\left| \sum_{e=-n}^{e} R(p, A)^{(e)} \right| \le \text{sampling-threshold}, \text{ where } n \ge 20 \tag{10}$$

After the sampling threshold p is fixed, it can be used as the next GNN model training.

3.4.2. Cost sensitivity matrix

Cost-sensitive learning in the dichotomous problem of credit default is concerned with the classification of default and non-default classes. The reason for using cost-sensitive matrices is that incorrectly judging a defaulting customer as a non-defaulting underreporting problem often differs from the cost of misreporting a non-defaulting customer as a defaulting one.

In general, the cost matrix **M** can be defined in the credit binary classification problem. The formula (11) is:

$$\mathbf{M} = \begin{pmatrix} M_{00} & M_{01} \\ M_{10} & M_{11} \end{pmatrix} \tag{11}$$

The cost-sensitive matrix M is dynamically generated during model training to address class imbalance and optimize performance based on misclassification costs. Using the German dataset, we explain the matrix elements and their roles in the model.

In the German dataset, the initial values of the M matrix are either randomly generated or set through a default configuration, and are subsequently adjusted during the training process to better reflect the costs of various classification errors. Since this is a binary classification problem in financial credit, there are only four elements in the matrix: M_{00}, M_{01}, M_{10}, and M_{11}. After training, the resulting cost matrix contains the following elements:

M_{00} (value 1.6528): This represents the cost when a non-defaulting customer is correctly classified as non-defaulting. Although correct classification incurs a cost, it is significantly lower than the cost of misclassification, as correct classification does not directly impact business decisions or financial risk.

M_{01} (value 2.0548): This represents the cost when a defaulting customer is misclassified as non-defaulting. This is the highest value in the matrix, which aligns with expectations in practical applications. In a financial credit system, misclassifying a defaulting customer as non-defaulting could lead to severe economic losses and credit risk. Therefore, the model must prioritize minimizing this type of misclassification.

M_{10} (value 0.9884): This represents the cost when a non-defaulting customer is misclassified as defaulting. Although such misclassification may harm customer experience (e.g., an unnecessary rejection of a credit application), it poses less business risk compared to M_{01}, and thus the cost is relatively lower.

M_{11} (value 1.4974): This represents the cost when a defaulting customer is correctly classified as defaulting. The cost of correctly classifying defaulting customers is lower than M_{01}, but still higher than for non-defaulting customers. This is because correctly classifying defaulting customers requires a model with strong discriminative capabilities, while also accounting for potential risks under uncertainty.

In summary, this matrix reflects the model's sensitivity to different types of misclassifications when dealing with imbalanced datasets. Specifically, the high weight of M_{01} indicates that the model aims to reduce the occurrence of misclassifying defaulting customers as non-defaulting. This type of misclassification poses significant risks in financial contexts, as it may lead to uncollectible loans or increased default rates. Thus, the value of M_{01} is designed to be higher than other elements, indicating that the model needs to focus on minimizing this error during training to ensure robustness in real-world business scenarios. In contrast, the weight of M_{10} is lower because misclassifying non-defaulting customers as defaulting primarily affects customer experience, but does not impose direct financial risks to the institution. Therefore, the model tolerates a higher misclassification rate in this case. By setting the cost-sensitive matrix in this way, the model effectively guides the learning process to maximize overall accuracy while prioritizing the reduction of high-cost misclassifications, especially when handling critical classifications like defaulting customers.

Specific steps after neighbour sampling: after completing neighbour sampling, the new graph obtained is input into the cost-sensitive learner (CSL). First, the node embedding $\mathbf{h}_v$ is obtained by the average aggregation of GNNs. The specific formula (12) is as follows:

$$\mathbf{h}_v = \sigma(\mathbf{W} \cdot \mathrm{MEAN}(\{\mathbf{h}_u, \forall u \in \mathcal{N}(v)\})) \tag{12}$$

σ is the activation function, $\mathbf{W}$ is the to-be-learned weights, and MEAN is the aggregation function that averages the node and its neighbour embeddings. It is the aggregation function that averages the node and its neighbour embeddings.

The traditional classification operation in unbalanced graph learning results are not very satisfactory. Based on this, cost-sensitive learning is introduced in the model, and the output embedding of the GNN is transformed to be cost-sensitive. For a given node v, the probability that it belongs to the category k, $\mathbf{p}_k(v)$, is calculated by cost-sensitive matrix $\mathbf{M}$ transformation for a given node v, the probability that it belongs to the category k, $\mathbf{p}_k(v)$, is calculated by cost-sensitive matrix $\mathbf{M}$ transformation. The calculation formula (13) is as follows:

$$\mathbf{p}_k(v) = \mathrm{Softmax}(\mathbf{M}_{v,k} \cdot \mathbf{z}_v[k]) = \frac{\exp(mathbf{M}_{v,k} \cdot \mathbf{z}_v[k])}{\sum_K \exp(\mathbf{M}_{v,k} \cdot \mathbf{z}_v[k])} \tag{13}$$

Here $\mathbf{z}_v[k]$ is the k-th dimensional embedding of node v in the last layer of GNN output, and $\mathbf{M}_{v,k}$ is the corresponding element in the cost matrix $\mathbf{M}$.

In the model, to ensure the reasonableness of the values of the elements in the cost matrix $\mathbf{M}$, an automatic optimization mechanism is used, which considers the data distribution matrix $\mathbf{H}$, the scatter matrix of the node embedding $\mathbf{W}$, and the confusion matrix $\mathbf{F}$. The combined effect of these matrices is adjusted using η and combined by the Hadamard product to form the target cost matrix loss $\mathbf{T}$. The loss function $L_{\mathrm{cost}}(\mathbf{M}, \mathbf{T})$ is used to optimize the cost matrix $\mathbf{M}$. The formulas mentioned above are (14)–(15).

$$\mathbf{T} = \eta \cdot \mathbf{H} \odot \mathbf{W} \odot \mathbf{F}. \tag{14}$$

$$L_{\mathrm{cost}}(\mathbf{M}, \mathbf{T}) = \|\mathbf{T} - \mathbf{M}\|_2^2 + \mathrm{Eval}(\theta, \mathbf{M}). \tag{15}$$

Here, T is the target cost matrix calculated based on the data distribution, embedding scatter, and confusion mean, θ represents model parameters, and $\mathrm{Eval}(\theta, \mathbf{M})$ is the validation error under the current parameters.

3.5. WIGNN module selection and reasoning

In the WIGNN model, the selection of various techniques reflects a deep analysis of the complexity and imbalance inherent in credit data. The key to the model's performance lies in effectively representing graph data and addressing the imbalance in the graph structure. During the process of transforming tabular data into a graph structure, Euclidean distance and Gaussian kernel mapping are used to convert distances into similarity scores as weights. This reduces the oversmoothing caused by highly correlated features (such as financial features), avoids overly tight connections, and improves the model's ability to capture nonlinear data. Subsequent experiments, as shown in Section 4.4.1, demonstrate the rationale behind the design of the differentiated weighted connection module.

In dealing with the inherent imbalance in credit data (where the number of defaulting customers is much smaller than that of non-defaulting customers), this imbalance manifests in the graph as uneven node counts and topology. The reason for selecting reinforcement learning-based neighbour sampling techniques and the cost-sensitive matrix method is that these methods address both quantity and weight aspects. Through dynamic sampling, they balance the distribution of samples and strengthen the focus on minority classes, addressing common issues in credit default prediction. Gaussian kernel mapping enhances the integrity of the transformation from tabular to graph data, while subsequent techniques solve the key challenges in financial credit default prediction. Together, they complement each other and jointly improve the effectiveness of WIGNN.

Furthermore, the selection of Euclidean distance for graph conversion and Gaussian kernel mapping is elaborated in the graph structure comparison section:

In the process of converting tabular data to graph data, selecting an appropriate distance metric is crucial. Euclidean distance is commonly used for initial graph conversion due to its simplicity and broad applicability, especially for various tabular datasets. However, when dealing with datasets containing highly correlated financial features, such as the risk dataset, the simplified representation of Euclidean distance may lead to overly dense connections in the adjacency matrix at certain high threshold ranges, thus reducing its ability to effectively distinguish between different connection strengths. To address this issue, the Gaussian kernel function can be employed to map the Euclidean distance output into similarity scores. This method not only enhances the differentiation of values but also mitigates the oversmoothing of the adjacency matrix. By considering the local structural relationships between points, the Gaussian kernel function can more precisely capture the interactions between nodes, thereby improving the expressiveness of the graph and the performance of the model.

4. Experiment

4.1. Datasets

In the field of credit assessment, a variety of public datasets are widely used for research and model development, and this time, five public datasets and one private dataset were applied to validate the experimental study.

4.1.1. Public financial datasets

The following five public datasets were selected, all derived from the official Kaggle website:

German: The dataset (Hofmann, 1994) consists of 1000 loan applicant entries, with 300 default samples. Each record includes 20 features/attributes defined by Professor Hofmann to assess applicants' credit risk, categorized as "good" or "bad".

The Irish Dummy Banks (IDB): A fictional dataset (Ferozi, 2019) simulating operations similar to the Lending Club in Ireland. It includes 12,537 loan entries, with 1406 defaults.

Table 2(a)
Raw data for the risk component.

Customer number	Detection indicators (partial)						The rest
	Return on net assets	Profitability of main business	Net margin from main operations	Captial product cumulative	Negative equity debt ratio	Net assets	
**994185	0.0697	0.0489	0.0462	0.0063	0.3714	76927653.47	……
**011025	0.142	0.761	0.2036	0.056	0.9182	11734112.5	……
**035309	0.0843	0.1452	0.6844	0.0881	0.6189	191378182.4	……
**139433	0.0538	0.0035	0.0055	0.2764	0.8567	96789226.09	……

Table 2(b)
Descriptive analysis of risk data characteristics.

Diagnostic property	Average value	Average value standard error	Median	Standard misalignment
Return on net assets	0.04	0.14	0.07	5.58
Profitability of main business	0.26	0.03	0.17	1.02
Net margin from main operations	−0.23	0.1	0.04	3.35
Cash ratios from main operations	1.13	0.13	0.99	4.67
Profit growth rate	−4.98	4.19	−0.16	119.71
Sales growth rate	1321156.85	1321154.09	0.06	32656834.07
Total asset growth rate	4.29	2.59	0.1	75.29
Accounts receivable growth rate	-11	15.95	0.05	407.46
Rate of capital accumulation	20.92	22.02	0.06	638.94
Accounts receivable turnover	9.03	72.96	3.83	2691.44
Average turnover days for accounts receivable	221.66	38.92	49.21	1255.83
Inventory turnover	272.69	127.09	1.13	4862.77
Average inventory turnover days	2753.27	1150.31	102.63	35473.7
Current asset turnover ratio	1.21	0.08	0.53	3.24
Total asset turnover	1.12	0.12	0.47	4.97
Quick ratio	258.57	242.36	0.66	9892.44
Current ratio	265.51	242.48	1.26	9897.01
Interest coverage multiples	599928.1	600325.1	4.19	8884000
Operating cash current liabilities ratio	−206.37	208.31	−0.06	8504.97
Gearing	0.59	0.01	0.61	0.32
Long-term asset suitability ratio	410.48	161.6	1.83	6466.19
Equity ratio	0.17	3.07	1.39	126.39
Net assets	4.5E+08	64185526	31714021	2.65E+09
Net cash flows from operating activities	−1E+08	22624549	−1404275	9.33E+08
Net profit	24351299	5670868	1642094	2.34E+08
Net assets/year-end loan balance	7.94	1.65	1.83	55.91

Lending Club: Data (Ye, 2021) from 2007 to 2015, comprising 9578 records, with 1168 defaults. It details borrowers' personal information, loan amount, purpose, term, interest rate, credit score, employment, and income.

Taiwan: Focuses on repayment records of Taiwanese credit card users since 2005, with 30,000 entries, including 6636 defaults. The dataset (Yeh, 2016) includes attributes like gender, education, age, marital status, and credit card default status.

customer_data (CD): Contains about 1125 entries, with 225 defaults. It (Praveen, 2020) holds transaction and demographic data related to customers, distinguishing between risky and non-risky clients of specific banking products.

4.1.2. Analysis of bank credit datasets

In addition to the publicly available datasets, this study of the risk dataset provided by a regional agricultural credit bank for credit risk assessment is selected for partial presentation due to the excessive number of indicators provided in the dataset. The credit data used in this paper is presented in Table 2(a).

The original dataset has 1702 records with 27 indicators and 26 numeric indicators. The risk dataset consists of 1476 standard samples, 90 overdue samples, 27 normal overdue samples and 109 unknown samples. There are 1593 valid samples after removing the unknown samples to ensure accuracy. In the end, there were 1476 non-defaulted samples and 117 defaulted samples, with a ratio of about 13:1. Due to the high missing rate (86.9%), guaranteed interest rate metrics were removed, and the total number of metrics was reduced to 25. A hot-filling technique handled missing values.

In the descriptive analysis of Table 2(b), the central tendency and dispersion of the 26 variables are shown through average, median, etc., statistical measures, which facilitates an understanding of the distribution and volatility of the data.

Table 3
Confusion matrix.

Actual \Predicted	Positive Prediction (+)	Negative Prediction (−)
Actual Positive (+)	True Positives (TP)	False Negatives (FN)
Actual Negative (−)	False Positives (FP)	True Negatives (TN)

4.2. Evaluation indicators

In the credit domain, correctly evaluating the classification efficacy of a model is particularly critical to understanding and predicting loan default risk. Especially when dealing with class-imbalanced datasets, confusion matrices provide a structured way to summarize the accuracy of model predictions, as shown in Table 3, including True Positives (TP), False Negatives (FN), False Positives (FP), and True Negatives (TN), which help to identify default classes in a study better.

Based on the confusion matrix, the following indicators can be calculated:

- Sensitivity (Sens), the formula (16) is:

$$Sens = \frac{TP}{TP + FN} \tag{16}$$

Sensitivity measures the proportion of all defaults correctly predicted to be defaults.

- Specificity (Spec), the formula (17) is:

$$Spec = \frac{TN}{TN + FP} \tag{17}$$

Specificity measures the proportion of all actual non-default samples correctly predicted non-default.

- The metrics used in this study are AUC, G-mean, and Accuracy (ACC); AUC is the area under the ROC curve; the closer the AUC is to 1, the stronger the classification ability of the model, aiming to distinguish between positive and negative samples. G-mean, the geometric mean of sensitivity and specificity, is mainly used to judge the ability of the balanced model to identify the positive and negative samples in the imbalanced dataset. The specific formulas (18)–(20) for several indicators are as follows:

$$AUC = \frac{Sens + Spec}{2} \tag{18}$$

$$G\text{-}mean = \sqrt{Sens \cdot Spec} \tag{19}$$

$$ACC = \frac{TP + TN}{TP + TN + FP + TN} \tag{20}$$

4.3. Baseline modelling

To compare the effectiveness of the WIGNN technique more fully, We compare WIGNN with 11 baseline methods in Python and weigh five comparative contrasts in entitled weights. These methods cover credit default modelling from traditional machine learning algorithms to the latest graph neural network techniques. LR and SVM are all from the literature (Bequé and Lessmann, 2017). RF out of literature (Saia and Carta, 2016). Deep Neural Network (DNN) out of the literature (Loureiro et al., 2018). Logistic Regression with XGBoost (LR_Xgb) from the literature (Li and Chen, 2020). Graph Sample and Aggregation (GraphSage), Graph Convolutional Networks (GCN), GAT, Class-Adaptive Resampling Enhancement Graph Neural Network (Caregnn), Cost-Sensitive Graph Neural Network (CSGNN) and GAT-COBO are all derived from the literature (Hu et al., 2024).

- **GraphSAGE (2017):** A graph neural network model that aggregates neighbourhood features to generate node representation vectors, capturing local graph structures effectively with scalable and robust generalization capabilities.
- **GAT (2017):** Using the attention mechanism to assign different importance to nodes in the graph improves the learning ability and generalization ability of graph data features.
- **GCN (2017):** By performing convolution operations on graph nodes to capture relationships and features between nodes, it is widely used in prediction and classification tasks.
- **LR_Xgb (2020):** Logistic regression with adjusted class weights handles class imbalance, while XGBoost optimizes hyperparameters.
- **CARE-GNN (2020):** A neighbour sampling strategy driven by reinforcement learning solves the class imbalance in graph data. A dynamic sampling process selects the neighbour node that contributes most to the task.
- **GAT-COBO (2022):** Combining Graph Attention Networks (GAT) with cost-sensitive learning and Adaboost aims to improve the recognition rate for minority class samples.
- **CSGNN (2023):** Combining reinforcement learning neighbour sampling and cost-sensitive matrix to solve graph unbalance problem.

We implemented the proposed method using Pytorch with all models running on Python 3.7.10,1 NVIDIA A100 80 GB GPU, 127 GB RAM, 13-core Intel(R) Xeon(R) Gold 6230R CPU. where the experimental vital parameters are shown in Table 4.

As shown in Fig. 4, a flowchart is provided to visually illustrate the experimental process, helping readers understand the steps and their interrelations.

The experiments divided the training and validation sets and the test set into a ratio of 0.2:0.2:0.6. The AUC, G-mean, and ACC metrics are utilized to assess the performance of different credit default prediction models across multiple datasets. The AUC and G-mean are used for evaluating class-imbalanced data, while ACC is selected due to credit default being a classification problem. Table 5(a) demonstrates the disparities in performance among traditional credit models, generic graphical models, and models specifically designed to address class imbalance across diverse datasets, including those with significant variations in class balance ratios. **Red** indicates the best result, green indicates the second best. Similar notations apply to subsequent tables.

- The traditional models LR, SVM, RF, and DNN perform well in terms of accuracy, which must be inside math mode due to the predominance of majority classes in the data, and the volume leads to a model bias in favour of the majority classes. Regarding classification performance, the AUC and G-mean of the models are poor in the risk and CD datasets, which have small data volumes and high-class imbalances. These models failed to distinguish between positive and negative class samples adequately. They lacked the ability to effectively identify minority classes, resulting in a tendency to ignore minority class samples during analysis. However, the better performance in IDB and Lending Club datasets is due to their well-balanced feature structure and relatively moderate imbalance ratios, which provide sufficient information in terms of data volume to differentiate the samples effectively.
- GAT and GCN show better AUC performance in distinguishing between default and non-default samples for general graph models.GraphSage has a slight advantage over traditional models. However, the improvement in G-mean is not as significant as AUC, suggesting insufficient recognition of minority classes and insufficient mitigation of minority class information being covered by majority class information. The superior performance of traditional models in ACC lies in the fact that graph models are more focused on learning the intrinsic structure of the data and are not solely concerned with accuracy. Graph models can effectively capture complex customer relationships and dependencies, enabling adaptive reasoning, which is difficult to achieve with other models. Although the graph model performs well regarding AUC on the LendingClub and IDB datasets, the ACC and G-mean may not consistently outperform the traditional model. The potential of generalized graph models in such feature-rich larger datasets is yet to be exploited.
- The class-balanced models, LR_Xgb, Caregnn, GAT-COBO, and CS-GNN, significantly outperform the traditional and generic graph models regarding AUC and G-mean metrics. LR_Xgb is the integrated model with a loss function that indirectly alters the sensitivity to a few classes through weighting factors. Its improvement in effectiveness over graph models that address class imbalance is limited, failing to start from the structural characteristics of the data. The graph balancing model Caregnn balances the number of positive and negative classes through the neighbour sampling technique of reinforcement learning. GAT-COBO increases the weights on the minority classes by using the cost-sensitive matrix, which targets the graph characteristics and achieves better results on class imbalance. CSGNN and WIGNN perform excellently on multiple datasets, especially AUC and G-mean, which demonstrate the ability to identify the minority classes in the face of the high imbalance of the small datasets. The reason for the recognition ability of minority classes is that neighbour sampling and cost-sensitive matrices are adopted in the graphs for the number and weight, respectively, to end the problem in terms of the graph structure and balance characteristics and to achieve effective recognition of minority classes and a significant improvement in the overall performance. Compared with CSGNN, WIGNN introduces the weight connection of RBF, which effectively enhances the model's sensitivity to the role of similar

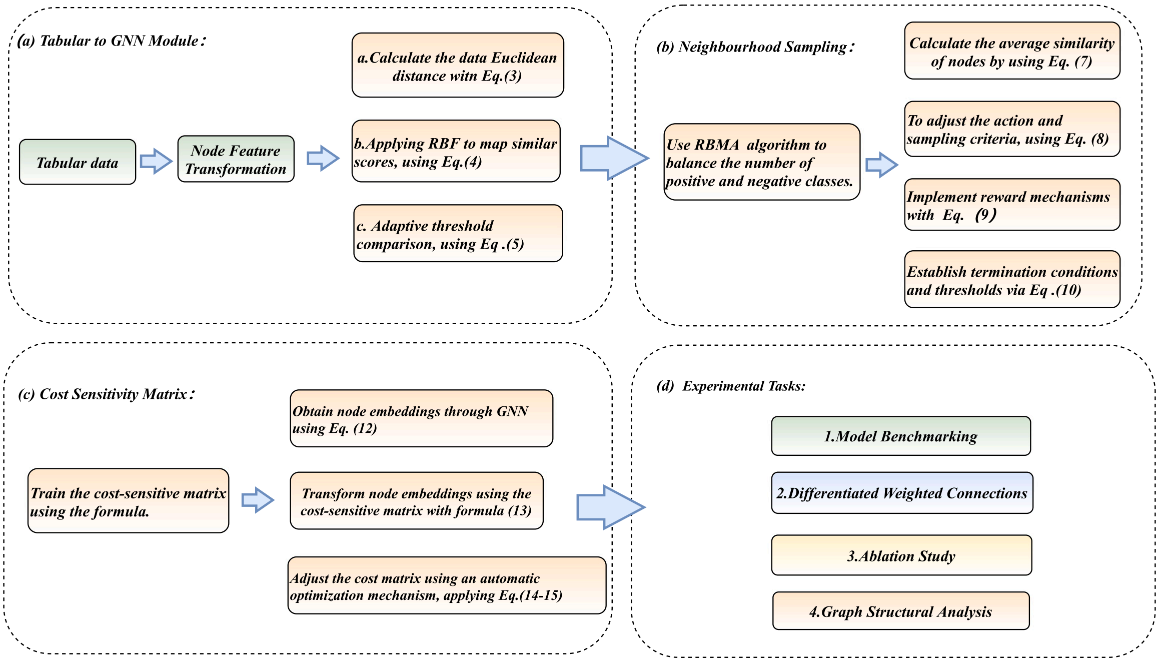

(a) Tabular to GNN Module：
Tabular data
Node Feature Transformation
a.Calculate the data Euclidean distance witn Eq.(3)
b.Applying RBF to map similar scores, using Eq.(4)
c. Adaptive threshold comparison, using Eq .(5)
(b) Neighbourhood Sampling：
Use RBMA algorithm to balance the number of positive and negative classes.
Calculate the average similarity of nodes by using Eq. (7)
To adjust the action and sampling criteria, using Eq. (8)
Implement reward mechanisms with Eq. (9)
Establish termination conditions and thresholds via Eq .(10)
(c) Cost Sensitivity Matrix：
Train the cost-sensitive matrix using the formula.
Obtain node embeddings through GNN using Eq. (12)
Transform node embeddings using the cost-sensitive matrix with formula (13)
Adjust the cost matrix using an automatic optimization mechanism, applying Eq.(14-15)
(d) Experimental Tasks:
1.Model Benchmarking
2.Differentiated Weighted Connections
3.Ablation Study
4.Graph Structural Analysis

Fig. 4. Experimental flow chart.

Table 4
Parameter settings for various datasets.

Datasets	Parameter setting
Risk	num_layers:3, hid_dim:16, lr_Cost:0.001,lr_GNN: 0.001,weight_decay_Cos:t0.02,weight_decay_GNN: 0.02
German	num_layers:3, hid_dim:16, lr_Cost:0.015,lr_GNN: 0.015,weight_decay_Cost: 0.001,weight_decay_GNN: 0.001
CD	num_layers= 3, hid_dim:16, lr_Cost:0.001,lr_GNN: 0.001,weight_decay_Cost: 0.07,weight_decay_GNN: 0.07
IDB	num_layers= 3, hid_dim:24, lr_Cost:0.03,lr_GNN: 0.03,weight_decay_Cost: 0.002,weight_decay_GNN: 0.002
Lending Club	num_layers= 3, hid_dim:24, lr_Cost:0.015,lr_GNN: 0.015,weight_decay_Cost: 0.001,weight_decay_GNN: 0.001
Taiwan	num_layers= 3, hid_dim:12, lr_Cost:0.008,lr_GNN: 0.008,weight_decay_Cost: 0.001,weight_decay_GNN: 0.001

Table 5(a)
Performance comparison of different models across datasets.

Datasets	Method	Traditional credit models				General graph model			Class balance model				Ours
	Metric	LR	SVM	RF	DNN	Graph-Sage	GAT	GCN	LR-Xgb	Care-gnn	GAT-COBO	CS-GNN	WI-GNN
Risk	AUC	0.6344	0.5398	0.6127	0.6122	0.5210	0.7004	0.684	0.5269	0.6416	0.7023	0.9998	0.9977
	G-mean	0	0	0.1142	0.2157	0.4485	0.1960	0.1606	0.3607	0.5520	0.6856	0.9404	0.9909
	ACC	0.9216	0.9265	0.9257	0.9310	0.6433	0.9109	0.9154	0.8437	0.6531	0.6857	0.8934	0.9833
German	AUC	0.8014	0.7845	0.7707	0.8005	0.4896	0.7106	0.7533	0.6764	0.7883	0.7635	0.7821	0.8226
	G-mean	0.6517	0.4603	0.4802	0.7216	0.3094	0.5962	0.6346	0.6585	0.6987	0.6266	0.6547	0.6742
	ACC	0.7625	0.7416	0.7483	0.7600	0.7845	0.7183	0.7389	0.7600	0.6900	0.6627	0.7500	0.7667
CD	AUC	0.5674	0.5768	0.5258	0.4933	0.4796	0.7422	0.5857	0.5285	0.5370	0.5881	0.8994	0.9564
	G-mean	0	0.0022	0.1214	0.4767	0.4698	0.3910	0.2617	0.4140	0.5506	0.3081	0.8507	0.8722
	ACC	0.7822	0.8001	0.7570	0.4000	0.5586	0.7422	0.7748	0.7699	0.5689	0.7615	0.8430	0.8104
IDB	AUC	0.9967	0.9968	0.9985	0.9976	0.5440	0.9506	0.9971	0.9659	0.9128	0.9840	0.9645	0.9931
	G-mean	0.3595	0.9752	0.9592	0.9749	0.5081	0.7787	0.9379	0.9657	0.8416	0.8867	0.7947	0.9292
	ACC	0.9032	0.9075	0.9918	0.9585	0.6485	0.9882	0.9757	0.9808	0.7528	0.9020	0.9293	0.9673
Lending Club	AUC	0.7540	0.9997	0.9999	0.9972	0.4998	0.9936	0.9995	0.9999	0.8739	0.9925	0.9983	0.9994
	G-mean	0.7127	0.9825	0.9999	0.9850	0.3809	0.935	0.9849	0.9999	0.7911	0.9476	0.9799	0.9895
	ACC	0.9044	0.9921	0.9999	0.9932	0.7971	0.9666	0.9904	0.9999	0.7730	0.9470	0.9835	0.9906
Taiwan	AUC	0.7237	0.7157	0.7749	0.7602	0.5117	0.7538	0.7510	0.6998	0.7094	0.7533	0.7451	0.7646
	G-mean	0.4641	0.5154	0.5344	0.6779	0.3691	0.4458	0.4832	0.6680	0.6544	0.6575	0.5865	0.5809
	ACC	0.8095	0.8177	0.8186	0.7916	0.7490	0.8068	0.8111	0.8133	0.6628	0.8064	0.8162	0.8196
Average rank	AUC	6.67	6.17	4.67	5.83	12	6.5	5.33	8.5	8.17	6	5.17	2.83
	G-mean	10.33	8	7	3.17	9.5	9	7.33	3.67	5.17	6.17	5	3.33
	ACC	6.33	4.83	3.82	6.17	9.67	7.33	6	4.67	11.17	9.5	5.5	2.67

Table 5(b)
DeLong Test results comparing WIGNN to LR and CSGNN.

Comparison	Dataset	Z-value	P-value
WIGNN-LR	German	3.51856	0.00043
WIGNN-CSGNN	CD	2.48182	0.01309

nodes and optimizes the message propagation process, especially in sparse environments. The careful adjustment mechanism gives WIGNN excellent results in the face of small datasets with high imbalance or large dataset environments with limited computational resources, highlighting the effectiveness and stability of the proper connection in solving the class imbalance problem.

In summary, although some models outperformed WIGNN in specific datasets, WIGNN still ranked first overall, achieving near-optimal results and demonstrating strong stability. This indicates that WIGNN has a high adaptability when handling graph-structured data, consistently delivering excellent performance, especially in diverse datasets and complex task environments.

4.3.1. Statistical analysis and validation of WIGNN performance

Our experimental analysis examines the stability of WIGNN across multiple credit datasets. Although WIGNN did not achieve the best results on all datasets, it demonstrated stable overall performance across various datasets. The initial AUC analysis revealed that CSGNN performed best on the risk dataset, followed closely by WIGNN. In larger datasets, while WIGNN performed effectively as a graph neural network, it was slightly outperformed by RF, which might be due to WIGNN's limitations in representing large-scale graph structures and its computational resource requirements. Additionally, the DeLong test in this study did not directly compare WIGNN to RF or CSGNN in terms of AUC differences. The detailed DeLong test results are presented in Table 5(b). This test allows for a comparison of the statistical significance of AUC differences between models.

To further assess WIGNN's performance, we focused on the ID and German datasets, where WIGNN achieved the best results. The DeLong test (DeLong et al., 1988) results indicate significant differences in AUC between WIGNN and other models (LR and CSGNN) on the German and CD datasets. In the German dataset, the Z-value between WIGNN and LR was 3.51856, with a P-value of 0.00043, indicating a highly significant difference in AUC, with WIGNN outperforming LR. In the CD dataset, the Z-value between WIGNN and CSGNN was 2.48182, with a P-value of 0.01309, also demonstrating a statistically significant difference, with WIGNN performing better than CSGNN.

Overall, WIGNN exhibited a significant AUC advantage on the German and CD datasets, especially in the German dataset, where its performance relative to LR was particularly strong. This suggests that WIGNN has strong classification capabilities on graph-structured data, effectively enhancing the model's AUC performance. Although WIGNN's performance was slightly below RF in large datasets, the gap was small, and WIGNN remained highly competitive. The DeLong test further confirmed WIGNN's advantage on specific tasks and datasets, particularly in datasets with specific sizes and characteristics, where it demonstrated excellent performance and stability.

We conducted paired t-tests between the WIGNN and the closest models for the remaining four datasets, using five sets of values for each model pair to assess the differences. Although the results do not consistently outperform all models, WIGNN demonstrates a clear advantage in stability, consistently achieving near-optimal or optimal performance across all datasets.

In this study, pairwise t-tests were performed on the remaining four datasets (risk, Lending Club, Taiwan, IDB) to compare WIGNN with closely ranked models (CSGNN, GAT-COBO, DNN), assessing their differences in AUC and ACC metrics. The hypotheses tested are as follows:

- **Null Hypothesis** ($\mathbf{H}_0$): There is no statistically significant difference between WIGNN and the compared models (CSGNN, GAT-COBO, DNN) in terms of AUC and ACC, implying that the mean performance difference between the models is zero.
- **Alternative Hypothesis** ($\mathbf{H}_1$): There exists a statistically significant difference between WIGNN and the compared models in terms of AUC and ACC, meaning that the mean performance difference between the models is non-zero.

The specific paired t-test results can be seen in Table 5(c). Based on the pairwise t-test results, in the **risk** dataset, although WIGNN performed significantly worse than CSGNN in terms of AUC (p = 0.0048), it showed a significant advantage in ACC ($p < 0.0001$). In the **Lending Club** dataset, the difference between WIGNN and GAT-COBO in AUC was not statistically significant ($p = 0.057$), but WIGNN still significantly outperformed GAT-COBO in ACC ($p < 0.0001$). For the **Taiwan** dataset, there was no statistically significant difference between WIGNN and DNN in AUC ($p = 0.228$), but WIGNN demonstrated a clear advantage in ACC ($p < 0.0001$). In the **IDB** dataset, WIGNN significantly outperformed GAT-COBO in both AUC ($p = 0.026$) and ACC ($p < 0.0001$). Overall, WIGNN exhibited stable and significant advantages in ACC, and although it did not demonstrate a significant advantage in AUC across all datasets, its overall performance remained outstanding.

4.4. Weighted connections

4.4.1. Weighted graph structure comparison

The principle of converting tabular data into GNN graph data is to treat the samples in the table as nodes, construct edges between nodes through similarity measures, and map the features in the table to node attributes, thereby realizing the construction of graph structure data. This study focuses on weighted imbalanced graphs, where the structural differences in the graph are reflected by different similarity measures, and the meaning of weights varies. The main difference in the graph structure lies in connectivity.

In this research, we employ three similarity measures — Euclidean distance, cosine similarity, and Mahalanobis distance — to construct graphs. Euclidean distance emphasizes numerical differences based on geometric distance, cosine similarity focuses on the directionality of feature vectors, and Mahalanobis distance considers the correlation between features through the covariance matrix. These methods, under strategies with fixed thresholds or fixed percentiles, result in different connection strengths and structural variations in the graph. Euclidean distance usually generates dense local connections, cosine similarity captures sparse connections in high-dimensional data, while Mahalanobis distance dynamically adjusts weights and accurately captures complex relationships between features. Therefore, different similarity measures significantly affect the graph structure's representation and performance.

Cosine Similarity:This approach can effectively capture directionality in high-dimensional data, its computational complexity is relatively high, especially when the data volume exceeds 10,000. Calculating the angle between vectors requires substantial resources. Given n samples and d features, the cosine similarity for a single node pair has a complexity of $O(d)$, and the overall complexity is $O(n^2 \times d)$. Thus, employing cosine similarity for large datasets (around 10,000 samples) becomes impractical. The cosine similarity formula (21) is as follows:

$$\cos\theta = \frac{\mathbf{x}\cdot\mathbf{y}}{\|\mathbf{x}\|\|\mathbf{y}\|} \tag{21}$$

where x and y represent two nodes. $x \cdot y$ denotes the dot product, and $\|x\|$ and $\|y\|$ are the norms of the vectors.

Mahalanobis Distance:This study explores the use of Mahalanobis distance combined with the covariance matrix as a non-traditional method for graph construction. This method considers not only the

Table 5(c)
Summary of pairwise T-test results for WIGNN vs Other models.

Dataset	Metric	Model comparison	T-Value	*P*-Value	Significance
Risk	AUC	WIGNN vs CSGNN	−5.65	0.0048	Significant
	ACC	WIGNN vs CSGNN	68.14	2.78e−07	Highly significant
Lending Club	AUC	WIGNN vs GAT-COBO	2.65	0.057	Not significant
	ACC	WIGNN vs GAT-COBO	29.42	7.95e−06	Highly significant
Taiwan	AUC	WIGNN vs DNN	1.42	0.228	Not significant
	ACC	WIGNN vs DNN	15.78	9.41e−05	Highly significant
IDB	AUC	WIGNN vs GAT-COBO	3.45	0.026	Significant
	ACC	WIGNN vs GAT-COBO	14.83	0.00012	Highly significant

distance between sample features but also the correlation between them. The Mahalanobis distance formula (21) is as follows:

$$\mathbf{D_M}(\mathbf{x},\mathbf{y}) = \sqrt{(\mathbf{x}-\mathbf{y})^T \boldsymbol{G}^{-1}(\mathbf{x}-\mathbf{y})} \tag{22}$$

$\mathbf{D_M}$ is the Mahalanobis distance between two points $\mathbf{x}$ and $\mathbf{y}$ where x and y are feature vectors of two samples, and G is the feature covariance matrix with G^{-1} as its inverse. The weight of each edge is determined based on the covariance relationship between the node features, giving the graph structure adaptability. Experimental results show that Mahalanobis distance exhibits strong competitiveness in terms of AUC, G-mean, and ACC metrics in small-scale data, significantly improving the expressiveness of graph data. However, the computational complexity of Mahalanobis distance lies in the formula, particularly the inversion of the covariance matrix, making the calculation of the adjacency matrix much more complex than traditional methods. Handling datasets with nearly 10,000 samples is challenging, and there are clear limitations. The complexity is $O(d^3)$.

Euclidean Distance :Euclidean distance is simple to compute, with a complexity of $O(n \times d)$, making it suitable for large-scale datasets. It effectively captures the geometric distance between features and is appropriate for handling simple linear relationships. It is also a suitable choice for converting large datasets into graphs. In this experiment, a weighted graph is employed, so different mapping methods on the basis of Euclidean distance result in different weighted connections, and the differences in graph connections are evident under fixed threshold conditions.

- 1+1/2L: A simple weighting method is added to the Euclidean distance, mainly using smoothing to strengthen the connections between nearby nodes while weakening the connections with distant nodes. The advantage lies in the fact that local neighbour information is crucial for similarity partitioning, resulting in a typically dense graph structure. This method manifests as locally dense but globally sparse in the graph structure. The local connections between nodes are tight, but the overall graph structure often fails to capture complex global relationships. For example, local nodes may be compact, while others are scattered, which could lead to imbalances in graph expressiveness.
- KNN: For each node, a fixed number of connections is established. When features are strong, tight local clusters can form, clearly reflecting the group structure within the dataset. However, the limitation of a fixed number of neighbours manifests as the graph structure struggles to adapt flexibly to differences in the distance distribution between nodes. In cases of high data complexity, KNN may produce incorrect connections, affecting the overall connectivity of the graph, leading to either insufficient connections for certain nodes or excessive connections for certain groups.
- RBF: It enhances the similarity between nodes through nonlinear mapping, resulting in a globally complex and finely weighted graph structure in large datasets. The RBF method is particularly suitable for handling nonlinear feature relationships and capturing latent substructure connections in the data. In the graph structure, this method reflects more detailed global connectivity, not only covering local similarity between nodes but also extending through nonlinear mapping to a global graph, capturing complex interactions between distant nodes. Therefore, in large-scale datasets, the RBF graph structure is more globalized, effectively reflecting the complexity of the data. This method strikes a balance between computational cost and graph expressiveness in large datasets.

Table 6(a)
Fixed thresholds comparison of the effect of different weights under different datasets.

Datasets	Method	CSGNN	1/L2+1	KNN	cosSim	RBF	Mahalanobis
Risk	AUC	0.9998	0.9983	0.9987	0.9945	0.9993	0.9911
	G-mean	0.9404	0.9735	0.9706	0.9977	0.9907	0.9966
	ACC	0.8934	0.9519	0.9467	0.9908	0.9958	0.9937
German	AUC	0.7821	0.7830	0.7560	0.7819	0.7747	0.8562
	G-mean	0.6547	0.6342	0.6196	0.6492	0.6742	0.6233
	ACC	0.7500	0.7367	0.7300	0.6733	0.7667	0.7317
CD	AUC	0.8994	0.9931	0.7966	0.9928	0.9819	0.9833
	G-mean	0.8507	0.9604	0.6857	0.9690	0.9161	0.8388
	ACC	0.8430	0.9422	0.5956	0.9566	0.8726	0.9289
IDB	AUC	0.9645	0.9667	0.9840	0.9362	0.9791	*OOM*
	G-mean	0.7947	0.7419	0.8883	0.5706	0.7996	*OOM*
	ACC	0.9293	0.9276	0.9543	0.9121	0.9371	*OOM*
Lending Club	AUC	0.9983	0.9915	0.9981	0.9964	0.9998	*OOM*
	G-mean	0.9799	0.9494	0.9759	0.9705	0.9955	*OOM*
	ACC	0.9835	0.9490	0.9836	0.9800	0.9944	*OOM*
Taiwan	AUC	0.7451	0.7453	0.7681	0.7616	0.7775	*OOM*
	G-mean	0.5865	0.5770	0.5885	0.5913	0.5937	*OOM*
	ACC	0.8162	0.8156	0.8222	0.8202	0.8227	*OOM*
Average rank	AUC	3	3	3.17	3.67	2.17	*N/A*
	G-mean	3.33	3.83	3.5	2.67	1.67	*N/A*
	ACC	3.5	3.67	3	3.33	1.5	*N/A*

4.4.2. *Differences in weights under fixed thresholds*

Table 6(a), it is shown that CSGNN introduces different weighted connections to improve graph credit default prediction. Different weighting methods reflect different understanding of complex data structures. The weighting strategies show the differences in data capture by different methods. The experiment verifies the impact of weight connections on improving model performance and explores weighting methods suitable for data characteristics to optimize the prediction.

GSGNN, which uses Euclidean distances to form unweighted connections, serves as an experimental baseline control, considering only the effect of the magnitude of the distance between nodes. The comparison also includes other methods for comparing weighted graph structures.

As shown in Fig. 5, compared to CSGNN, the changes in the metrics for different weighting methods at a fixed threshold are concluded:

- The 1/L2 + 1 weight connection emphasizes the node's nearest neighbours. In the small datasets of German and CD, the local neighbourhood information of nodes is critical due to the small amount of data, and the distances and locations between the near

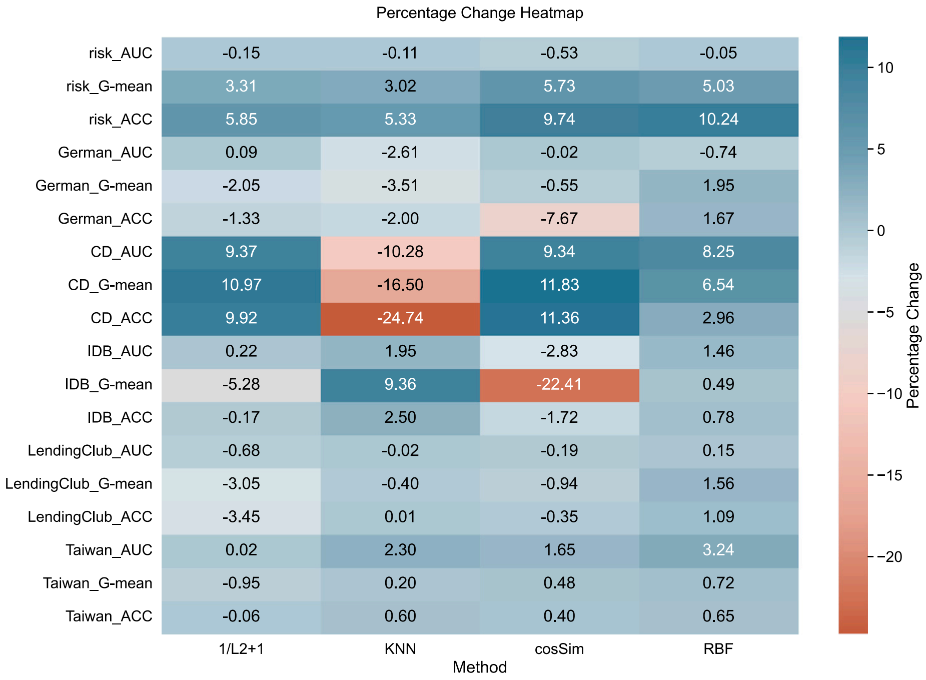


Fig. 5. Heatmap of performance change for connectivity methods vs. CSGNN benchmark.

neighbours are decisive for similarity division. However, when large datasets like Taiwan or Lending Club are available, the structure is complex, and the distances between nodes are widely distributed. Relying on local information alone is insufficient to capture the complex interactions between the data, and there may be an imbalance between the near-neighbour and global information.

- KNN performs well on the risk dataset as its financial features are highly correlated with defaults, forming tight cluster data points in the feature space. This high correlation and good clustering tendency effectively allow KNN to identify close neighbours through distance measures. However, it performs mediocrely on the rest of the small dataset with a high imbalance.
- The cosSim performs well across the datasets. Thinking in terms of data directionality, no comparison with distance is made. Performs well on high-dimensional, big data despite being suboptimal and computationally resource-intensive on medium-sized data.
- The Gaussian kernel approach enhances the segmentation of similarity through RBF mapping, which is suitable for dealing with datasets with complex non-linear relationships between features, such as LendingClub and Taiwan. It can tap into the latent substructural connections and performs best on large datasets with complex features. In addition, RBF can still provide good results on small datasets with high imbalance. Mahalanobis distance performs exceptionally well on small datasets, as it accounts for the intrinsic covariance structure of the data (Ahelegbey et al., 2019; Avdjiev et al., 2019). This allows distance calculations to adapt to the true characteristics of the distribution. Such adaptability enables Mahalanobis distance to accurately capture the dissimilarities between entities, even in smaller datasets, thereby improving the model's ability to identify patterns. However, due to its computational complexity, particularly when inverting the covariance matrix, the process of calculating the adjacency matrix becomes significantly more demanding compared to traditional methods. As the dataset size increases, constructing graph data becomes progressively more difficult.

Table 6(b)
Differences in connections with different weights under fixed sparsity.

Datasets	Method	1/L2+1	KNN	cosSim	RBF	Mahalanobis
Risk	AUC	0.9903	0.9971	0.9992	0.9977	0.9899
	G-mean	0.9994	0.9735	0.9793	0.9909	0.9843
	ACC	0.9660	0.9833	0.9764	0.9728	0.9796
German	AUC	0.8209	0.7799	0.8053	0.8226	0.8250
	G-mean	0.7073	0.6703	0.7136	0.6742	0.7032
	ACC	0.7500	0.7467	0.7654	0.7667	0.7500
CD	AUC	0.9503	0.7467	0.9492	0.9564	0.9796
	G-mean	0.8961	0.6485	0.8448	0.8722	0.9758
	ACC	0.9067	0.5556	0.8933	0.8104	0.9647
IDB	AUC	0.9312	0.9853	0.9332	0.9931	*OOM*
	G-mean	0.6640	0.8416	0.6929	0.9292	*OOM*
	ACC	0.9076	0.9532	0.9107	0.9673	*OOM*
Lending Club	AUC	0.9994	0.9991	0.9995	0.9992	*OOM*
	G-mean	0.9984	0.9846	0.9920	0.9895	*OOM*
	ACC	0.9901	0.9911	0.9925	0.9906	*OOM*
Taiwan	AUC	0.7675	0.7574	0.7697	0.7646	*OOM*
	G-mean	0.5914	0.5964	0.5936	0.5809	*OOM*
	ACC	0.8191	0.8154	0.8211	0.8196	*OOM*
Average Rank	AUC	2.67	3.5	2	1.83	*N/A*
	G-mean	2	3.17	2.33	2.5	*N/A*
	ACC	2.71	2.43	1.43	1.86	*N/A*

Overall, depending on the dataset's size, complexity, and feature variability, the RBF method has a clear advantage with large datasets, with a combined stable performance across datasets.

4.4.3. Differences in weighting methods with fixed sparsity

Under the fixed threshold, both different weight connection methods and connection numbers affect the experimental results. Different connection numbers will directly affect the sparsity of the graph. Therefore, the effects of different weighted connection methods are compared to reflect the ability to capture and exploit the subtle differences in data structure and control the sparsity.Table 6(b) shows the Percentile chosen for different datasets to ensure the number of connections is consistent under different weighted connection methods. The specific settings are 50 per cent for small datasets, 80 per cent for medium

Table 7(a)
WIGNN ablation experiment.

Model	Tabular-to-graph module	Neighbour sampling module	Cost-sensitive matrix module	Weight connection module
WIGNN-1	✓	✓		
WIGNN-2	✓		✓	
WIGNN-3	✓	✓	✓	
WIGNN	✓	✓	✓	✓

datasets, and 99 per cent for large datasets, and the differences in weighting methods are compared horizontally under the same number of connections.

Under fixed connectivity, the WIGNN with RBF-Percentile connections excels in small datasets. On the “IDB” dataset, it achieves an AUC that is 0.78% higher than the second-best method, with G-mean and ACC exceeding by 8.76% and 1.41%, respectively, demonstrating its strong performance in handling minority class issues. In the “Lending Club” and the large “Taiwan” dataset, WIGNN ranks closely behind the top performer, showing robust competitiveness. These results underscore WIGNN’s consistently high performance across diverse datasets with varying class imbalances, sizes, and feature correlations.

Table 7(b)
WIGNN ablation.

Datasets	Method	WIGNN-1	WIGNN-2	WIGNN-3	WIGNN
Risk	AUC	0.9623	0.5759	0.9998	0.9977
	G-mean	0.9301	0.1103	0.9404	0.9909
	ACC	0.8777	0.9195	0.8934	0.9833
German	AUC	0.7613	0.7613	0.7821	0.8226
	G-mean	0.6349	0.6619	0.6547	0.6742
	ACC	0.7483	0.7283	0.7501	0.7667
CD	AUC	0.8989	0.4639	0.8994	0.9564
	G-mean	0.5464	0.1193	0.8507	0.8722
	ACC	0.8341	0.7926	0.8430	0.8104
IDB	AUC	0.9407	0.9569	0.9645	0.9931
	G-mean	0.6528	0.7489	0.7947	0.9292
	ACC	0.9137	0.9168	0.9293	0.9673
Lending Club	AUC	0.9846	0.9951	0.9983	0.9994
	G-mean	0.8970	0.9636	0.9799	0.9895
	ACC	0.9440	0.9727	0.9835	0.9906
Taiwan	AUC	0.7308	0.7437	0.7451	0.7646
	G-mean	0.5727	0.5660	0.5865	0.5809
	ACC	0.8141	0.8096	0.8162	0.8196

4.5. Ablation experiments

This study adopts the Design of Experiments (DoE) method to systematically evaluate various configurations of the WIGNN model, aiming to reveal the specific contribution of each module to the overall performance of the model. This fractional factorial design selectively tests combinations of modules, effectively reducing the number of required experiments while ensuring the exploration of interactions between key modules. The contents are shown in Table 7(a). The module configurations involved in the experiment are as follows:

- **WIGNN-1**: Includes only the neighbour sampling module, used to evaluate its fundamental utility in processing graph data.
- **WIGNN-2**: Includes only the cost-sensitive matrix module, focusing on its effectiveness in handling class imbalance.
- **WIGNN-3**: Combines the neighbour sampling and cost-sensitive matrix modules, aiming to evaluate the performance improvement when the modules work together.
- **WIGNN**: The complete configuration, integrating all modules to demonstrate the model’s full potential. All configurations require the tabular-to-graph conversion module, as this study applies to tabular learning tasks.

The experimental data and evaluation metrics (AUC, Gmean, ACC) are the same as those used in the baseline model comparison, utilizing six datasets.

As shown in the Fig. 6, it is evident that WIGNN-1, which includes only the neighbour sampling module, performs well on some datasets but falls short of the complete model. WIGNN-2, containing only the cost-sensitive matrix module, performs poorly on highly imbalanced datasets like CD, highlighting the limitations of lacking support from other modules. WIGNN-3 combines neighbour sampling and cost-sensitive matrix modules, achieving performance close to the entire model on specific datasets. The complete WIGNN model significantly enhances performance by adding differential weight connections and optimizing sparsity settings, particularly excelling in capturing latent features and enhancing graph representation capabilities, leading to marked performance improvements across multiple datasets. This underscores the importance of integrating modules effectively, demonstrating that combining various modules’ complementary and synergistic effects is critical to achieving efficient and reliable performance in graph neural network design.

The results of this ablation study reveal the impact of different WIGNN model configurations on overall performance. By comparing the performance of each configuration across multiple datasets, we can draw the following conclusions:

As shown in Table 7(b), it is evident that WIGNN-1, which only includes the neighbour sampling module, performs well on certain datasets but falls far short of WIGNN-3 and the full WIGNN model. In comparison to using the cost-sensitive module alone, controlling the class quantity (as seen in WIGNN-3) yields far better results than merely focusing on the minority class, which is especially noticeable in the risk dataset. WIGNN-2, which includes only the cost-sensitive matrix module, performs poorly on highly imbalanced datasets like CD, highlighting the limitations of relying solely on an independent module without the support of others, making it unsuitable for use across multiple datasets. WIGNN-3, which combines the neighbour sampling and cost-sensitive matrix modules, demonstrates performance close to the full model on specific datasets. With a fixed graph data input, the model is highly optimized, and its effectiveness is clearly evident.

The full WIGNN model, by adding differentiated weight connections and optimizing sparsity settings, significantly improves performance, particularly in capturing latent features and enhancing graph representation capabilities, thereby achieving significant performance improvements across multiple datasets. This highlights the importance of effectively integrating all modules and demonstrates that the complementary and synergistic effects of combining these modules are crucial for achieving efficient and reliable performance in the design of graph neural networks. Additionally, in the baseline experiment, a DeLong test was conducted on the ICD dataset to compare WIGNN and WIGNN-3 (CSGNN), further validating the effectiveness of the weight connection module.

4.6. Graph structural analysis

4.6.1. Sensitivity analysis

When considering the stability and applicability of the model, we focus on evaluating its performance under various levels of sparsity, indicated by different numbers of connections. By applying percentile

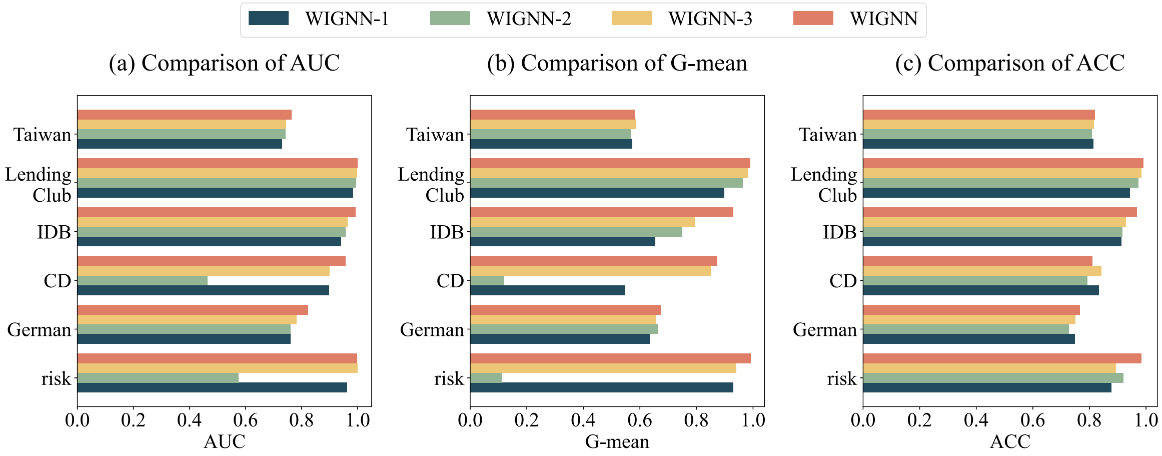


Fig. 6. WIGNN ablation variation graph.

Table 8(a)
Variation of sparsities of WIGNN under different datasets.

Datasets	Method	50%	60%	70%	80%	90%
Risk	AUC	0.9977	0.9977	0.9974	0.9954	0.9903
	G-mean	0.9909	0.9891	0.9824	0.9811	0.9782
	ACC	0.9833	0.9801	0.9781	0.9655	0.9603
German	AUC	0.7681	0.8076	0.7948	0.8226	0.8056
	G-mean	0.5996	0.5760	0.6196	0.6742	0.6705
	ACC	0.7250	0.7217	0.7300	0.7667	0.7600
CD	AUC	0.9564	0.9428	0.9319	0.9317	0.9260
	G-mean	0.8722	0.8339	0.8123	0.7438	0.6726
	ACC	0.8104	0.7585	0.7304	0.6459	0.5659
IDB	AUC	OOM	OOM	OOM	0.9931	0.9925
	G-mean	OOM	OOM	OOM	0.9292	0.9113
	ACC	OOM	OOM	OOM	0.9673	0.9685
Lending Club	AUC	OOM	OOM	0.9994	0.9994	0.9991
	G-mean	OOM	OOM	0.9892	0.9895	0.9824
	ACC	OOM	OOM	0.9908	0.9906	0.9859

thresholds ranging from 50% to 90%, we can observe and analyse the impact of WIGNN on critical performance metrics such as AUC, G-mean, and ACC at different sparsity levels.OOM, Out of Memory in too much computational resources and not enough video memory. Here, RBF-Percentile has the most minor relative computational resources.

Combined with Table 8(a), the model is analysed for sensitive changes at different sparsities. The choice of percentile directly affects the density of data connections, with lower percentiles implying lower connection thresholds and more edges, leading to higher computational resource consumption.

As shown Fig. 7, the sparsity adjustable domain is large for small-sized datasets, and the risk and CD are in percentile increase. Metrics show different decreasing trends. In particular, the ACC in the risk dataset decreases from 0.9833 at the 50% Percentile to 0.9603 at the 90% threshold, indicating that higher sparsity significantly affects the risk and CD datasets in the WIGNN classification accuracy. However, the German dataset shows the best results at a specific sparsity of 80%, suggesting that the critical features of the data have been captured at a particular sparsity. At the same time, noise interference is mitigated to avoid overfitting and computational resource consumption problems.

When dealing with large-scale datasets, “IDB” and “Lending Club” frequently suffer from memory shortages at lower percentiles, where the high number of edges at the lower percentiles creates a large number of edges at lower percentiles creating a large neighbourhood matrix, leading to a rapid increase in memory requirements. Especially in large graphs composed of node classifications, it is inevitable that these adjacency matrices will be accessed frequently. The lack of memory is currently the bottleneck in large-scale graph processing. By adjusting the Percentile design to control the sparsity of connections and optimize the efficiency of the model operation, performance and sparsity are balanced within limited computational resources. This sensitivity analysis helps us deeply understand the model’s performance under different sparsities and provides vital data support for future model optimization.

4.6.2. Stability analysis

Assessing model stability is crucial during practical testing, particularly for imbalanced credit scoring datasets (Chen et al., 2024) where sensitivity analysis is conducted. Similar to those studies, testing model robustness through sparsification of graph structures is essential, especially when faced with variations in training/testing datasets or node loss. Random node and edge deletion (Lou et al., 2022) provide an effective method for preliminary robustness evaluation when calculating more complex metrics is not feasible. Random node deletion simulates data loss within the graph, testing the model’s sensitivity to the loss of local information. Meanwhile, random edge deletion sparsifies the graph structure, allowing us to assess the model’s tolerance to changes in connectivity. This approach mirrors the randomness of real-world data loss, especially in scenarios where it is unknown which nodes or edges are critical, providing a more comprehensive reflection of model stability.

To comprehensively evaluate the stability of the WIGNN model under structural changes, this study simulates potential data loss scenarios through random node and edge deletions. This method allows us to observe and analyse the model’s performance when encountering graph corruption in real-world operations, thereby measuring its stability.

Deletion proportions: In our experiments, the deletion proportions of nodes and edges increased incrementally from 5% to 10%. This range reflects varying degrees of structural damage, from minor to moderate, systematically revealing the model’s responses to different levels of damage. To eliminate the randomness of a single experiment, each deletion proportion test was repeated five times, and the average results were used as the final evaluation metric. In stability experiments, the WIGNN data model employs the 80th percentile as a control for the same level of sparsity across all datasets.

The detailed experimental results are shown in Tables 8(b) and 8(c). The experiment was conducted using the German, risk, and CD datasets to evaluate the stability of the WIGNN model under node deletion scenarios. In the German dataset, there was slight fluctuation in performance metrics, with G-mean and ACC remaining relatively stable. This could be attributed to the relatively low class imbalance, where the impact of missing nodes on model performance is minimal. The AUC metric gradually decreased from 0.8177 to 0.779, indicating that the model retained a high level of discriminatory power even with significant node loss. The fluctuation in AUC suggests the model

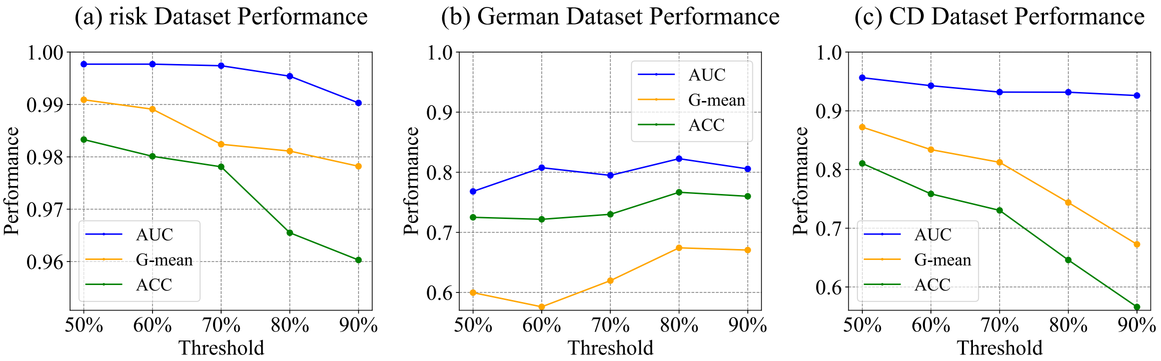


Fig. 7. Stability of WIGNN for small datasets at different sparsities.

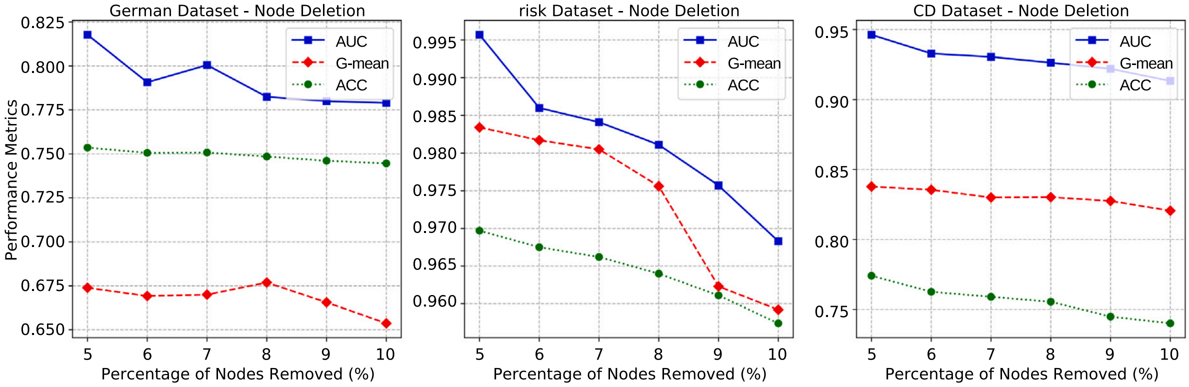


Fig. 8. Stability of WIGNN after proportional random node deletion.

may rely on certain structural nodes, or it may reflect the uneven distribution of nodes within the dataset. This indicates that the model could encounter varying challenges when facing different stages of structural degradation, leading to non-linear performance changes.

In contrast, node deletion had a more pronounced effect on the CD and risk datasets, mainly due to their higher class imbalance. In particular, the risk dataset's extensive financial data features make node information critically important. However, the performance variation across all datasets remained within 3%, demonstrating the WIGNN model's robustness and resistance to moderate levels of graph structural damage and imbalanced data. These variations are depicted in Figs. 8 and 9.

In the edge deletion experiments, the performance degradation was more gradual and generally less volatile than in the node deletion experiments. This may be because edge deletions do not remove entire data points but rather reduce the connections between nodes, resulting in a less direct impact on model performance and a smoother performance decline. This phenomenon further emphasizes the WIGNN model's ability to maintain graph structural integrity, even in the face of significant structural changes. Therefore, across all datasets, WIGNN exhibits acceptable fluctuations within the 5%–10% range of node and edge removal. It is able to maintain a certain degree of stability even in the presence of missing data.

5. Conclusion and future work

In this study, we propose WIGNN, a highly interpretable credit default model for adaptive graph structure inference. By transforming Euclidean distances into similarity scores via Gaussian kernel mapping, WIGNN effectively identifies subtle connections between nodes and data clusters. Additionally, the adaptive threshold adjustment mechanism allows for flexible control of graph sparsity. By integrating the neighbour sampling strategy and a cost-sensitive matrix, WIGNN

Table 8(b)
Performance metrics of WIGNN on different datasets with varying node deletion percentages.

Dataset	Metric	Percentage of nodes deleted					
		5%	6%	7%	8%	9%	10%
German	AUC	0.8177	0.7907	0.8005	0.7824	0.7800	0.7790
	G-mean	0.6739	0.6692	0.6700	0.6769	0.6656	0.6537
	ACC	0.7536	0.7506	0.7508	0.7485	0.7461	0.7446
Risk	AUC	0.9957	0.9860	0.9841	0.9811	0.9757	0.9683
	G-mean	0.9834	0.9817	0.9805	0.9756	0.9623	0.9592
	ACC	0.9697	0.9675	0.9662	0.9640	0.9611	0.9574
CD	AUC	0.9462	0.9328	0.9304	0.9263	0.9219	0.9132
	G-mean	0.8378	0.8355	0.8300	0.8302	0.8275	0.8205
	ACC	0.7741	0.7626	0.7590	0.7555	0.7449	0.7401

Table 8(c)
Performance metrics of WIGNN on different datasets with varying edge deletion percentages.

Dataset	Metric	Edge deletion percentage					
		5%	6%	7%	8%	9%	10%
German	AUC	0.8227	0.8200	0.8159	0.8068	0.8035	0.8001
	G-mean	0.6766	0.6673	0.6652	0.6586	0.6553	0.6520
	ACC	0.7610	0.7580	0.7570	0.7528	0.7505	0.7500
Risk	AUC	0.9940	0.9873	0.9835	0.9810	0.9737	0.9702
	G-mean	0.9872	0.9843	0.9821	0.9780	0.9761	0.9713
	ACC	0.9680	0.9660	0.9627	0.9618	0.9600	0.9580
CD	AUC	0.9376	0.9351	0.9307	0.9275	0.9204	0.9170
	G-mean	0.8239	0.8236	0.8203	0.8174	0.8160	0.8119
	ACC	0.7585	0.7562	0.7510	0.7463	0.7433	0.7390

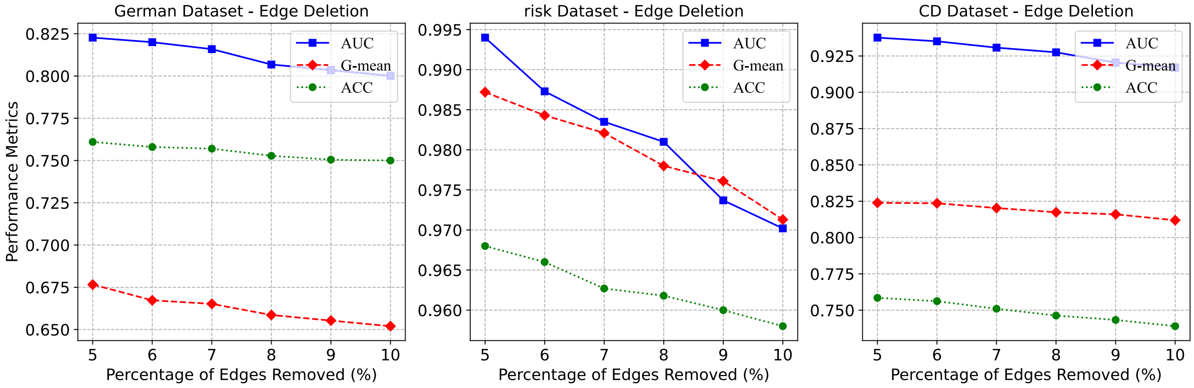


Fig. 9. Stability of WIGNN after proportional random edge deletion.

improves classification accuracy on unbalanced data. Compared to existing methods, WIGNN offers significant advantages and provides a new approach to solving credit default issues through differential weight linkage and graph balancing learning. This study highlights the potential of graph processing methods in addressing credit data diversity and scalability, paving the way for advancements in credit default prediction within complex financial environments. Future work will explore the application of GNNs to large datasets to enhance model robustness across varying dataset sizes and reduce computational demands. Additionally, future research based on existing credit default prediction models should place greater emphasis on fair lending issues. Barocas et al. (2023) assert that algorithmic fairness is crucial in credit assessments, particularly in preventing biases against specific groups. Fuster et al. (2017) discusses the application of machine learning in credit lending and suggests that fairness constraints can mitigate algorithmic bias. Furthermore, Lee and Floridi (2021) emphasizes that, in fintech applications, enhancing model fairness and transparency — especially by controlling disparate impacts using fairness algorithms — can improve credit opportunities for disadvantaged groups.

CRediT authorship contribution statement

Zhipeng Yan: Writing – review & editing, Writing – original draft, Visualization, Software, Methodology. **Hanwen Qu:** Resources, Conceptualization. **Chen Chen:** Investigation. **Xiaoyi Lv:** Funding acquisition, Formal analysis. **Enguang Zuo:** Investigation, Formal analysis, Data curation. **Kui Wang:** Resources, Project administration. **Xulun Cai:** Formal analysis.

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.

Acknowledgements

The authors sincerely appreciate all anonymous reviewers for their worthy comments and suggestions. Our work is supported by the Xinjiang Uygur Autonomous Region Key R&D Program Projects (2022B01005, 2022B01005-3).

Data availability

The data that has been used is confidential.

References

Agosto, Arianna, Silvestrini, Andrea, Nicolini, Marcella, Mattarocci, Gianluca, 2023. Bayesian learning models to measure the relative impact of ESG factors on credit ratings. Int. J. Data Sci. Anal. 16 (3), 205–223. http://dx.doi.org/10.1007/s41060-023-00405-9.

Ahelegbey, Daniel Felix, Giudici, Paolo, Hadji-Misheva, Branka, 2019. Latent factor models for credit scoring in P2P systems. Phys. A 522, 112–121. http://dx.doi.org/10.1016/j.physa.2019.01.130, URL https://www.sciencedirect.com/science/article/pii/S0378437119301372.

Altman, Edward I., 1968. Financial ratios, discriminant analysis and the prediction of corporate bankruptcy. J. Finance 23 (4), 589–609, URL http://www.jstor.org/stable/2978933.

Altman, Edward I., Sabato, Gabriele, 2007. Modelling credit risk for SMEs: Evidence from the US market. Abacus 43 (3), 332–357.

Avdjiev, S., Giudici, P., Spelta, A., 2019. Measuring contagion risk in international banking. Journal of Financial Stability 42, 36–51. http://dx.doi.org/10.1016/j.jfs.2019.05.014, URL https://www.sciencedirect.com/science/article/pii/S1572308919303195. Financial Services Indices, Liquidity and Economic Activity.

Bahnsen, Alejandro Correa, Stojanovic, Aleksandar, Aouada, Djamila, Ottersten, Björn, 2013. Cost sensitive credit card fraud detection using Bayes minimum risk. In: 2013 12th International Conference on Machine Learning and Applications, Vol. 1. pp. 333–338. http://dx.doi.org/10.1109/ICMLA.2013.68.

Barboza, Flavio, Kimura, Herbert, Altman, Edward, 2017. Machine learning models and bankruptcy prediction. Expert Syst. Appl. 83, 405–417. http://dx.doi.org/10.1016/j.eswa.2017.04.006.

Barocas, Solon, Hardt, Moritz, Narayanan, Arvind, 2023. Fairness and Machine Learning: Limitations and Opportunities. MIT Press.

Basel Committee on Banking Supervision, 2006. Core Principles for Effective Banking Supervision. BCBS Publication BCBS 230. Bank for International Settlements.

Beaver, William H., 1966. Financial ratios as predictors of failure. J. Account. Res. 4, 71–111, URL http://www.jstor.org/stable/2490171.

Bequé, Artem, Lessmann, Stefan, 2017. Extreme learning machines for credit scoring: An empirical evaluation. Expert Syst. Appl. 86, 42–53. http://dx.doi.org/10.1016/j.eswa.2017.05.050, URL https://www.sciencedirect.com/science/article/pii/S0957417417303718.

Bhattacharya, Arnab, Wilson, Simon P., Soyer, Refik, 2019. A Bayesian approach to modeling mortgage default and prepayment. European J. Oper. Res. 274 (3), 1112–1124. http://dx.doi.org/10.1016/j.ejor.2018.10.047.

Cao, Yi, Liu, Xiaoquan, Zhai, Jia, Hua, Shan, 2022. A two-stage Bayesian network model for corporate bankruptcy prediction. Int. J. Finance Econ. 27 (1), 455–472.

Chen, Yujia, Calabrese, Raffaella, Martin-Barragan, Belen, 2024. Interpretable machine learning for imbalanced credit scoring datasets. European J. Oper. Res. 312 (1), 357–372. http://dx.doi.org/10.1016/j.ejor.2023.06.036, URL https://www.sciencedirect.com/science/article/pii/S0377221723005088.

Cheng, Dawei, Tu, Yi, Ma, Zhen-Wei, Niu, Zhibin, Zhang, Liqing, 2019. Risk assessment for networked-guarantee loans using high-order graph attention representation. In: IJCAI. pp. 5822–5828.

Cowden, Chad, Fabozzi, Frank J., Nazemi, Abdolreza, 2019. Default prediction of commercial real estate properties using machine learning techniques. J. Portfolio Manag. 45 (7), 55–67.

Dal Pozzolo, Andrea, Caelen, Olivier, Le Borgne, Yann-Aël, Waterschoot, Serge, Bontempi, Gianluca, 2014. Learned lessons in credit card fraud detection from a practitioner perspective. Expert Syst. Appl. 41 (10), 4915–4928. http://dx.doi.org/10.1016/j.eswa.2014.02.026.

DeLong, E.R., DeLong, D.M., Clarke-Pearson, D.L., 1988. Comparing the areas under two or more correlated receiver operating characteristic curves: A nonparametric approach. Biometrics 44 (3), 837–845. http://dx.doi.org/10.2307/2531595.

Dong, Gang, Lai, Kin Keung, Yen, Jerome, 2010. Credit scorecard based on logistic regression with random coefficients. Procedia Comput. Sci. 1 (1), 2463–2468. http://dx.doi.org/10.1016/j.procs.2010.04.278, ICCS 2010.

du Jardin, Philippe, 2021. Forecasting corporate failure using ensemble of self-organizing neural networks. European J. Oper. Res. 288 (3), 869–885. http://dx.doi.org/10.1016/j.ejor.2020.06.020.

Durand, David, 1941. Risk elements in consumer instalment financing. NBER Books, (dura41-1), National Bureau of Economic Research, Inc.

Ferozi, M.N., 2019. Loan data for dummy bank. Kaggle. URL https://www.kaggle.com/datasets/mrferozi/loan-data-for-dummy-bank. (Accessed 17 January 2024).

Fuster, Andreas, Goldsmith-Pinkham, Paul, Ramadorai, Tarun, Walther, Ansgar, 2017. Predictably unequal? The effects of machine learning on credit markets. Regul. Financ. Inst. eJ. URL https://api.semanticscholar.org/CorpusID:49367164.

Giudici, Paolo, 2001. Bayesian data mining, with application to benchmarking and credit scoring. Appl. Stoch. Models Bus. Ind. 17, 69–81, URL https://api.semanticscholar.org/CorpusID:62167891.

Hofmann, H., 1994. Statlog (German credit data). http://dx.doi.org/10.24432/C5NC77, UCI Machine Learning Repository. URL https://archive.ics.uci.edu/ml/datasets/statlog+(german+credit+data). (Accessed 17 January 2024).

Hu, Xinxin, Chen, Haotian, Chen, Hongchang, Liu, Shuxin, Li, Xing, Zhang, Shibo, Wang, Yahui, Xue, Xiangyang, 2024. Cost-sensitive GNN-based imbalanced learning for mobile social network fraud detection. IEEE Trans. Comput. Soc. Syst. 11 (2), 2675–2690. http://dx.doi.org/10.1109/TCSS.2023.3302651.

Jha, Sanjeev, Guillen, Montserrat, Westland, J. Christopher, 2012. Employing transaction aggregation strategy to detect credit card fraud. Expert Syst. Appl. 39 (16), 12650–12657.

Lee, M.S.A., Floridi, L., 2021. Algorithmic fairness in mortgage lending: From absolute conditions to relational trade-offs. Minds Mach. 31, 165–191. http://dx.doi.org/10.1007/s11023-020-09529-4.

Li, Yiheng, Chen, Weidong, 2020. A comparative performance assessment of ensemble learning for credit scoring. Mathematics 8 (10), 1756. http://dx.doi.org/10.3390/math8101756.

Liu, Yucheng, Gao, Zipeng, Liu, Xiangyang, Luo, Pengfei, Yang, Yang, Xiong, Hui, 2023. QTIAH-GNN: Quantity and topology imbalance-aware heterogeneous graph neural network for bankruptcy prediction. In: Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining. pp. 1572–1582.

Lou, Yang, He, Yaodong, Wang, Lin, Chen, Guanrong, 2022. Predicting network controllability robustness: A convolutional neural network approach. IEEE Trans. Cybern. 52 (5), 4052–4063. http://dx.doi.org/10.1109/TCYB.2020.3013251.

Loureiro, A.L.D., Miguéis, V.L., da Silva, Lucas F.M., 2018. Exploring the use of deep neural networks for sales forecasting in fashion retail. Decis. Support Syst. 114, 81–93. http://dx.doi.org/10.1016/j.dss.2018.08.010, URL https://www.sciencedirect.com/science/article/pii/S0167923618301398.

Luo, Jian, Yan, Xin, Tian, Ye, 2020. Unsupervised quadratic surface support vector machine with application to credit risk assessment. European J. Oper. Res. 280 (3), 1008–1017. http://dx.doi.org/10.1016/j.ejor.2019.08.010.

Malekipirbazari, Milad, Aksakalli, Vural, 2015. Risk assessment in social lending via random forests. Expert Syst. Appl. 42 (10), 4621–4631. http://dx.doi.org/10.1016/j.eswa.2015.02.001.

Maron, Haggai, Ben-Hamu, Heli, Serviansky, Hadar, Lipman, Yaron, 2019. Provably powerful graph networks. In: Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., Garnett, R. (Eds.), Advances in Neural Information Processing Systems, Vol. 32. Curran Associates, Inc., URL https://proceedings.neurips.cc/paper_files/paper/2019/file/bb04af0f7ecaee4aae62035497da1387-Paper.pdf.

Moscatelli, Mirko, Parlapiano, Fabio, Narizzano, Simone, Viggiano, Gianluca, 2020. Corporate default forecasting with machine learning. Expert Syst. Appl. 161, 113567. http://dx.doi.org/10.1016/j.eswa.2020.113567.

Mushava, Jonah, Murray, Michael, 2022. A novel xgboost extension for credit scoring class-imbalanced data combining a generalized extreme value link and a modified focal loss function. Expert Syst. Appl. 202, 117233. http://dx.doi.org/10.1016/j.eswa.2022.117233.

Nami, Sanaz, Shajari, Mehdi, 2018. Cost-sensitive payment card fraud detection based on dynamic random forest and k-nearest neighbors. Expert Syst. Appl. 110, 381–392. http://dx.doi.org/10.1016/j.eswa.2018.06.011.

Nikolic, Nebojsa, Zarkic-Joksimovic, Nevenka, Stojanovski, Djordje, Joksimovic, Iva, 2013. The application of brute force logistic regression to corporate credit scoring models: Evidence from Serbian financial statements. Expert Syst. Appl. 40 (15), 5932–5944. http://dx.doi.org/10.1016/j.eswa.2013.05.022.

Praveen, 2020. Credit risk classification dataset. Kaggle. URL https://www.kaggle.com/datasets/praveengovi/credit-risk-classification-dataset. (Accessed 17 September 2024).

Rao, Congjun, Liu, Ying, Goh, Mark, 2023. Credit risk assessment mechanism of personal auto loan based on PSO-XGBoost model. Complex Intell. Syst. 9 (2), 1391–1414.

Ratnam, Venkata, 2012. Credit card fraud detection using anti k-nearest algorithm. International Journal on Computer Science and Engineering 4, 1035.

Saia, Roberto, Carta, Salvatore, 2016. An entropy based algorithm for credit scoring. In: Tjoa, A. Min, Xu, Li Da, Raffai, Maria, Novak, Niina Maarit (Eds.), Research and Practical Issues of Enterprise Information Systems. Springer International Publishing, Cham, pp. 263–276.

Shen, Feng, Zhao, Xingchao, Kou, Gang, Alsaadi, Fawaz E., 2021. A new deep learning ensemble credit risk evaluation model with an improved synthetic minority oversampling technique. Appl. Soft Comput. 98, 106852.

Shi, Yong, Qu, Yi, Chen, Zhensong, Mi, Yunlong, Wang, Yunong, 2024. Improved credit risk prediction based on an integrated graph representation learning approach with graph transformation. European J. Oper. Res. 315 (2), 786–801. http://dx.doi.org/10.1016/j.ejor.2023.12.028.

Stefania, Guzmán-Castillo, Claudia, Garizabalo-Davila, Guillermo, Alvear-Montoya Luis, Gustavo, Gatica, Dario, Rodriguez-Heraz Jaiver, Alfonso, Medina-Tovar Freddy, Tatiana, Andrade-Nieves Sheyla, 2023. Credit risk scoring model based on the discriminant analysis technique. Procedia Comput. Sci. 220, 928–933. http://dx.doi.org/10.1016/j.procs.2023.03.127, The 14th International Conference on Ambient Systems, Networks and Technologies Networks (ANT) and The 6th International Conference on Emerging Data and Industry 4.0 (EDI40).

Sun, Jie, Fujita, Hamido, Zheng, Yujiao, Ai, Wenguo, 2021. Multi-class financial distress prediction based on support vector machines integrated with the decomposition and fusion methods. Inform. Sci. 559, 153–170. http://dx.doi.org/10.1016/j.ins.2021.01.059.

Sun, Jie, Li, Hui, Fujita, Hamido, Fu, Binbin, Ai, Wenguo, 2020. Class-imbalanced dynamic financial distress prediction based on adaboost-SVM ensemble combined with SMOTE and time weighting. Inf. Fusion 54, 128–144. http://dx.doi.org/10.1016/j.inffus.2019.07.006.

Wang, Kui, Li, Meixuan, Cheng, Jingyi, Zhou, Xiaomeng, Li, Gang, 2022. Research on personal credit risk evaluation based on xgboost. Procedia Comput. Sci. 199, 1128–1135. http://dx.doi.org/10.1016/j.procs.2022.01.143, The 8th International Conference on Information Technology and Quantitative Management (ITQM 2020 & 2021): Developing Global Digital Economy after COVID-19.

Wang, Shun, Zhang, Yong, Piao, Xinglin, Lin, Xuanqi, Hu, Yongli, Yin, Baocai, 2024. Data-unbalanced traffic accident prediction via adaptive graph and self-supervised learning. Appl. Soft Comput. 157, 111512. http://dx.doi.org/10.1016/j.asoc.2024.111512.

Yang, Mei, Lim, Ming K., Qu, Yingchi, Li, Xingzhi, Ni, Du, 2023. Deep neural networks with L1 and L2 regularization for high dimensional corporate credit risk prediction. Expert Syst. Appl. 213, 118873. http://dx.doi.org/10.1016/j.eswa.2022.118873.

Ye, X., 2021. A data-driven study on investment strategies for P2P lending platforms. In: 2021 2nd International Conference on E-Commerce and Internet Technology. ECIT, pp. 429–435. http://dx.doi.org/10.1109/ECIT52743.2021.00096.

Yeh, I-Cheng, 2016. Default of credit card clients. http://dx.doi.org/10.24432/C55S3H, UCI Machine Learning Repository. URL https://archive.ics.uci.edu/dataset/350/default+of+credit+card+clients. (Accessed 17 January 2024).

Zhou, Binbin, Jin, Jiayun, Zhou, Hang, Zhou, Xuye, Shi, Longxiang, Ma, Jianhua, Zheng, Zengwei, 2023. Forecasting credit default risk with graph attention networks. Electron. Commer. Res. Appl. 62, 101332. http://dx.doi.org/10.1016/j.elerap.2023.101332.