IG-DP: Graph Neural Network-based Default Prediction Method for Imbalanced Financial Datasets

Changhai Wang, Jiaxi Ren, Min Huang, Qing Liu*, and Yaoli Xu

***Abstract*—Predicting financial defaults plays a vital role in reducing financial risks for credit businesses. A prominent trend in this area is the incorporation of borrowers' social profiles into predictive models using graph neural networks. However, the significant imbalance between default and normal users in labeled financial datasets poses challenges for training effective prediction models, often leading to overfitting. A financial default prediction model named IG-DP is proposed in this paper, which is designed to use unlabeled background nodes to enhance prediction performance. First, imbalanced labeled samples are used to train an initial graph neural network classifier, and the node embeddings of the unlabeled samples can be obtained with the initial model. Then, background nodes proximal to the labeled samples are selected for pseudo labeling with the similarity selection module based on the embedding vector of the node. Finally, the pseudo-labeled samples are added to the training set to retrain the prediction model. The DGraphFin dataset is used for experimental evaluation, and the $AUC-ROC$ and $F1-measure$ are chosen as evaluation metrics. The experimental results demonstrate that IG-DP significantly outperforms other methods. Meanwhile, ablation experiments confirm the effectiveness of both the similarity selection module and the retraining loss calculation method for fusion confidence.**

***Index Terms*—Default prediction, Graph neural network, Samples imbalanced issue, Similarity measurement.**

I. Introduction

INTERNET financial services have become an essential part of people's social lives due to the rapid growth of the economy. Under the internet financial model, borrowing and lending channels have shifted from single, centralized banks and financial institutions to increasingly decentralized peer-to-peer lending platforms [1]. However, there are both opportunities and threats. Credit risk in internet finance is more complex compared to the traditional financial sector [2]. Assessing borrowers' ability to repay debt promptly and accurately can lower financial risks and guarantee the security of online financial transactions [3]. The fundamental function of online financial services is to identify defaulting users. The goal of default prediction is to determine whether a user will be unable to repay in the future. A user will be classified as a default user if they cannot make their repayments on time [4]. It is therefore typically regarded as a binary categorization problem [5].

The financial data is being studied using a number of statistical learning models [6], such as autoregressive integrated moving average model (ARIMA) [7], vector autoregressive model (VAR) [8] and graph neural network (GNN) [9]. GNN is the future trend in this field since it can take social relationship data into account when making predictions [10], [11]. However, the majority of GNN-based classification tasks depend on reasonably balanced datasets [12]. In financial default prediction, there is a class imbalance problem in the model's training data because there are more normal users than default users [9]. The class imbalance problem leads to a model bias towards the majority class, neglecting the minority class during training. As a result, directly applying GNN to the default prediction task often fails to yield satisfactory outcomes [2].

Typically, there are two approaches to addressing this problem. The first involves synthesizing minority class samples using oversampling methods to enhance training data [13]. However, the newly synthesized samples cannot properly represent the minority classes and lack interpretability. The second approach involves adjusting the class weights, where the model assigns higher weights to the minority class to give it more attention [14]. Since neither of these two methods effectively expands the classification boundaries, they lead to overfitting issues. However, compared to other imbalanced classification problems, default prediction benefits from a large number of unlabeled background samples. These samples, registered as financial company users but not having made any loans, provide valuable background information for predicting defaults. This paper proposes a default prediction method on the imbalanced graph (IG-DP), which leverages these background samples to predict default risk. The main contributions of this work are as follows.

- The graph neural network is adopted for default prediction to take into account borrower's social profiles.
- A default prediction framework that takes advantage of unlabeled background samples for the class imbalance problem is proposed.
- The efficacy of this approach is validated utilizing the publicly available dataset, and the impacts of various

Manuscript received September 19, 2024; revised February 6, 2025. This work is supported by the National Natural Science Foundation of China (No.62102002 and No.62102372), the Key Science and Technology Program of Henan Province (No.232102210078 and No.242102210106) and the Key Scientific Research Projects of Henan Higher School (No.25B520013).

Changhai Wang is a associate professor of Software Engineering College, Zhengzhou University of Light Industry, Zhengzhou, 450000, China.(e-mail: chw@zzuli.edu.cn)

Jiaxi Ren is a postgraduate student of Software Engineering College, Zhengzhou University of Light Industry, Zhengzhou, 450000, China.(e-mail:1750334102@qq.com)

Min Huang is a professor of Software Engineering College, Zhengzhou University of Light Industry, Zhengzhou, 450000, China.(e-mail: huangmin@zzuli.edu.cn)

Qing Liu is a associate professor of Electronics and Information Engineering, West Anhui University, Lu'an, Anhui, 237012, China.(*corresponding author to provide e-mail:clyqig2008@126.com)

Yaoli Xu is a lecturer of Software Engineering College, Zhengzhou University of Light Industry, Zhengzhou, 450000, China.(e-mail: yaolixu@zzuli.edu.cn)

modules are investigated through ablation experiments.

This paper is organized as follows. Firstly, a brief review of the related work is introduced in Section II. Following this, the proposed prediction method is described in Section III. Next, the performance evaluations, including experimental design and comparison results, are shown in Section IV. Lastly, future work and overall summary are discussed in Section V.

II. Related works

A. Financial default prediction

The issue of financial default prediction has been widely studied and is typically regarded as a classification or regression task based on machine learning. Using machine learning methods, researchers are starting to forecast default probabilities with the advent of big data [15].

Machine learning methods use different classifiers, such as decision trees and neural networks, to make predictions by extracting key features that represent each user [16], [17]. Dai et al. [16] utilized reinforcement learning to decide whether to sample the data. Wang et al. [18] proposed an adaptive classification boundary adjustment method and used a multi-objective evolution mechanism for ensemble creation. Then, graph-based methods were used to enrich user information through neighbor nodes [19].

Financial default prediction tasks also leverage graph neural networks and their various variants, including graph convolutional network (GCN) [2], graph sample and aggregate (GraphSAGE) [20] and graph attention network (GAT) [21]. These networks use graph topology and node features to learn node representations [22]. Wang et al. [2] proposed a graph-preserving graph neural network with learning function to jointly learn the low-order structure in the original graph and the high-order structure in the graph-based multi-view. Cheng et al. [23] merged credit behavior and network structure data, builds recursive and self-attention mechanisms, and accelerates the risk prediction process. In addition, some methods combine time series models with convolutional neural networks (CNN) [24]. Yang et al. [25] utilized the long-term time model of long short-term memory recurrent neural network (LSTM) to obtain short- and long-term structure information.

None of these methods fully utilize the vast amounts of unlabeled node information in graph data, which are crucial for estimating the credit risk of each user.

B. Class imbalance problem

There is a class imbalance problem, as some categories have very few samples while others have a large number of samples [26]. In real-world applications, such as credit card fraud detection [27], disease identification [28], traffic control [29], credit score [30] and emotion recognition [31], imbalanced data is frequently encountered.

Re-weighting and re-sampling are common solutions for addressing this issue. The re-weighting approach assigns greater weight to minority class samples when calculating the loss, which makes the model more focused on the minority class [14], [32]. A solution to the issue of decision boundary shift brought on by topological imbalance was suggested by Chen et al. [14] to reweight the distance between each labeled node and its class boundary. Menon et al. [32] put forward two long-tail logit adjustment methods that offer adaptable control over the relative value labels' share of the overall loss. Undersampling and oversampling are two types of resampling methods [33]. Reducing the quantity of samples from the majority class through undersampling puts the class distribution into balance [34]. Cui et al. [35] utilized a hybrid sampling method that undersamples the majority class and oversamples the minority class to achieve a balance of samples from different classes. Through the use of more minority class samples, oversampling methods adjust the distribution of classes [13], [36]. Zhao et al. [13] generated synthetic minority nodes by interpolating two minority class nodes. A pre-trained edge predictor determines the connectivity of synthesized nodes between the generated nodes and the neighbors of the two source minor nodes. However, this approach cannot be efficiently extended to the minority class, as the created nodes only depend on minority class nodes. Park et al. [36] designed the diversity of these minority classes by mixing some minority class nodes from other classes and synthesizing new minority class nodes with their one-hop neighbors. If the mixing ratio is not adjusted correctly, synthetic nodes created by a subjectively designed mix of a few nodes with other nodes might not accurately represent the state of the underlying data, which could damage the results. Chang et al. [37] proposed a modified cluster-based over-sampling (MCS) method to tackle the class imbalance problems, which duplicates minority examples until the imbalanced situations are improved in order to select representative minority class examples.

The majority of present methods either repeat a significant number of specific samples or increase their weights, both of which lead to overfitting problems and impair the performance of classifiers based on graph structure, such as GNN.

C. Self-training

Self-training is a widely used method in semi-supervised learning, often applied to node classification tasks [38]. A lack of sufficient accessible labels can result in a decrease in GNN performance. The classifier is initialized using some labeled data through the self-training method. Once trained, it can predict high-confidence unlabeled samples, which are then added to the labeled data to retrain the classifier [39]. This process adresses the shortcomings of GNN caused by the lack of labels.

The selection of high-confidence unlabeled samples is critical to the effectiveness of self-training methods [38]. Jiao et al. [40] utilized natural neighbors to assist ensemble classifiers grown more effectively. When calculating the confidence between samples, Wang et al. [41] included geometric distance and data distribution as factors. Yang et al. [25] proposed a new graph with homogeneous and heterogeneous edges combining labeled and unlabeled data. A common problem with these approaches is that the learning process can become biased due to the label noise introduced by the pseudo-labels generated through predictions [16]. If reliable unlabeled samples are not efficiently selected, it is easy to produce inaccurate predictions on unlabeled data, which in turn reduces classifier performance. Wang et al.

[12] combined graph structure learning and graph neural networks to generate negative pseudo-labels for unlabeled data with low prediction confidence. These pseudo-labeled samples were trained alongside a limited number of labeled samples. But training this way takes a long time.

Our method integrates the concept of self-training, selects high-confidence samples, reduces noise, shortens training time, and generates pseudo-labels for default samples using graph node embeddings and similarity selection.

III. Method

A. Problem definition

We define $\boldsymbol{G} = (\boldsymbol{V}, \boldsymbol{A}, \boldsymbol{X})$ as the graph containing all users and their relationships, $\boldsymbol{V}$ is the user node set, $|\boldsymbol{V}| = n$ is the total number of nodes, $\boldsymbol{X} = \{\boldsymbol{x}_i\} \in \mathbb{R}^{n \times m}$ is the node features matrix corresponding to $\boldsymbol{V}$, m is the number of features, and $\boldsymbol{A} = \{a_{ij}\} \in \mathbb{R}^{n \times n}$ is the adjacency matrix of graph $\boldsymbol{G}$. In the default prediction task, each node in the graph represents a user, and the node features represent individual information that the user provided, such as registration time, gender, age, and income, etc. The relationship between users i and j is represented by edge a_{ij} in the graph, and it can be found by looking up the user's common connections, emergency contacts, etc. The definition makes it clear that matrix $\boldsymbol{A}$ is a sparse matrix. In the subsequent sections, $\xi(v)$ represents the set of nodes connected to node v.

Two labels defined in the financial default prediction task are default users and normal users, denoted by labels l_1 and l_2, respectively. The default user refer to one who has A loan record and defaulted on repayments, while a normal user repays on time. $|l_i|$ is used to represent the number of nodes in the i-th category. Since the number of defaulting users in credit companies is much smaller than that of normal users, $|l_1| << |l_2|$, which leads to the class imbalance problem.

Just a few nodes in the node set $\boldsymbol{V}$ have labels because many users have registered but have not taken out loans from financial institutions. These nodes and the corresponding features are denoted by $\boldsymbol{V}_F$ and $\boldsymbol{X}_F$, respectively. The rest unlabeled nodes are represented by $\boldsymbol{V}_U$ and $\boldsymbol{X}_U$. y_i is the label for $\forall \boldsymbol{x}_i \in \boldsymbol{X}_F$, and $\boldsymbol{Y}_F$ is the label set that corresponds to $\boldsymbol{X}_F$. The financial default prediction task aims to train a prediction model, which can predict if a user will default in the future based on samples that already exist. When a prediction model is trained directly from the class-imbalanced node set $\boldsymbol{V}_F$, the predictions of this model will be skewed in favor of the majority class, which will produce subpar default predictions. This paper is dedicated to train a prediction model f based on the graph neural network, which can avoid the impact of sample imbalance problem and improve the model's classification ability for minority nodes by using background nodes $\boldsymbol{V}_U$.

B. Method overview

This section briefly introduces the proposed IG-DP, with an overview shown in Fig.1.

Given a graph consisting of all users and their relationships, a binary classification GNN prediction model f_0 is first trained with the labeled sample $(\boldsymbol{X}_F, \boldsymbol{Y}_F)$. Then, $\boldsymbol{X}$ is classified with f_0, and $\boldsymbol{V}_{F1}$ and $\boldsymbol{V}_{F2}$ represent the node sets categorized as l_1 and l_2 in $\boldsymbol{V}_F$, respectively. $\boldsymbol{V}_{U1}$ and $\boldsymbol{V}_{U2}$ represent the node sets classified as l_1 and l_2 in $\boldsymbol{V}_U$. The normalized probability vector $\boldsymbol{h}_i$ represents the classification result for $\boldsymbol{x}_i$ in $\boldsymbol{X}$. As it is a binary-classification problem, the dimension of $\boldsymbol{h}_i$ is 2. The classification confidence of sample i, denoted as c_i , which can be credibly assessed with $\boldsymbol{h}_i$. This model also regards the input vector of the final output layer as the embedding vector of the node, in addition to calculating the confidence of the node. The embedding vector can be used as the representation of the node since it combines its initial features and topological structure. $\boldsymbol{U}_{F1}$, $\boldsymbol{U}_{F2}$, $\boldsymbol{U}_{U1}$and $\boldsymbol{U}_{U2}$ are the corresponding embedding vector sets for $\boldsymbol{V}_{F1}$, $\boldsymbol{V}_{F2}$, $\boldsymbol{V}_{U1}$ and $\boldsymbol{V}_{U2}$, respectively.

Some of the background nodes need to be selected in order to enhance the effect of initial prediction model on default users. First, calculating the mean of vectors in $\boldsymbol{U}_{F1}$ and denoted as $\boldsymbol{u}_0$. Then, determining the distance in vector space for $\forall \boldsymbol{u} \in \boldsymbol{U}_{U1}$ between $\boldsymbol{u}$ and $\boldsymbol{u}_0$. These nodes close to $\boldsymbol{u}_0$ are recorded as $\boldsymbol{V}_a$. The classification result l_1 is defined as the label of the node set, and the set composed of the confidence of these nodes is marked $\boldsymbol{C}_a$. Both the original feature set $\boldsymbol{X}_a$ and label set $\boldsymbol{Y}_a$ corresponding to $\boldsymbol{V}_a$ are added to the original training data. Finally, the classifier is retrained with $(\boldsymbol{X}_F \cup \boldsymbol{X}_a, \boldsymbol{Y}_F \cup \boldsymbol{Y}_a)$ and the associated confidence set $\boldsymbol{C}_a$. The final prediction model can be obtained by repeating these steps numerous times. The process of IG-DP will be introduced as Algorithm 1.

Algorithm 1: The process of the IG-DP

Input: $\boldsymbol{G} = (\boldsymbol{V}, \boldsymbol{A}, \boldsymbol{X})$, imbalanced training set $(\boldsymbol{X}_F, \boldsymbol{Y}_F)$, feature matrix for unlabeled nodes $\boldsymbol{X}_U$;
Output: Unbiased GNN classifier f_T;

```
Train the initial GNN classifier f_0 with (X_F, Y_F);
for each epoch t = 1, 2 ··· do
    Classify all nodes with f_{t-1}, and get the classification confidence set C;
    Get the node embedding set U_{F1} and U_{U1};
    Calculate the center of U_{F1}, denoted as u_0;
    Initialize ordered set X_a = ∅, Y_a = ∅, C_a = ∅;
    for ∀u ∈ U_{U1} do
        Calculate the distance between u and u_0, denoted as d;
        if d < θ then
            X_a = X_a ∪ x_u, Y_a = Y_a ∪ l_1, C_a = C_a ∪ c_u;
        end
    end
    Train the new GNN classifier f_t with (X_F ∪ X_a, Y_F ∪ Y_a) and C_a;
end
return GNN classifier f_T;
```

The implementation details of each part will be introduced in the following sections.

C. Graph Neural Networks

The GNN is utilized as a classifier for the prediction framework of this paper in order to make use of connections among registered users. It can obtain the embedding vector

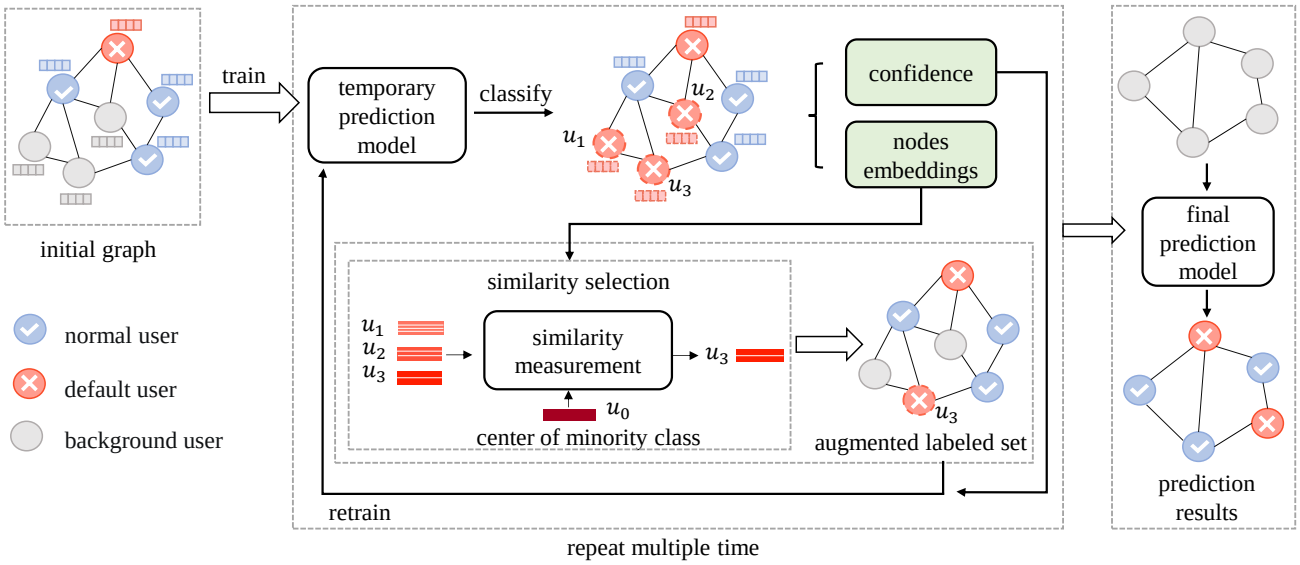

Fig. 1: The overview of the IG-DP. The input data is a graph with limited labeled nodes and massive unlabeled nodes. First, a temporary prediction model is trained with the input data, and the trained model is used to classify the input data to obtain the confidence and embedding vector for each node. Then, the nodes embeddings are input to the similarity selection module to pseudo-label some default users. Finally, the augmented labeled set is applied to retrain the temporary prediction model. After training multiple times, the final prediction model is obtained.

of a node which combines topological relationship by aggregating the features of the neighboring nodes. The GNN is shown in Fig.2, and the general form of feature aggregation is defined as Formula 1.

$$\boldsymbol{u}_v^k = \sigma(\boldsymbol{W}^k \cdot [\boldsymbol{u}_v^{k-1} || (\sum_{v' \in \xi(v)} u_{v'}^{k-1}) \Big/ |\xi(v)|]), \quad (1)$$

where $\boldsymbol{u}_v^k$ is the embedding of node v after k-hop aggregation, and $\boldsymbol{u}_v^0 = \boldsymbol{x}_v$. $\xi(v)$ is the set of nodes connected to node v, and $|\xi(v)|$ is node number in $\xi(v)$. $\boldsymbol{W}^k$ is the trainable matrix, $||$ is the vector connection operation, and σ is the sigmoid() activation function. Feature aggregation of neighboring nodes with k-hop distance can be achieved by Formula 1. After feature aggregation, the vector $\boldsymbol{u}^k$ is mapped as a sample label through a fully connected layer with an output degree of 2, shown as

$$\widehat{\boldsymbol{h}} = \sigma(\boldsymbol{W}_o \boldsymbol{u}^k), \quad (2)$$

where $\widehat{\boldsymbol{h}}$ is the model output and $\boldsymbol{W}_o$ is the trainable weight matrix for output layer. When the model is initialized, the trainable parameters are initialized using a normal distribution. The gradient descent algorithm is used to update these parameters in the training process. The cross-entropy loss function is used in this paper to calculate the prediction loss, shown as

$$L(\boldsymbol{h}_i, \widehat{\boldsymbol{h}}_i) = -\sum_{j=1}^{2} \boldsymbol{h}_{ij} \ln(\widehat{\boldsymbol{h}}_{ij}), \quad (3)$$

where $\boldsymbol{h}_i$ is the one-hot vector generated based on the label of sample $\boldsymbol{x}_i$, $\widehat{\boldsymbol{h}}_i$ is model output. After multiple iterations, the optimal parameters are obtained, and the corresponding prediction model is denoted by f_0.

D. Similarity selection

When the classifier f_0 is utilized directly for default prediction, the results will be biased towards normal users due to the imbalanced training data. The following step is enhancing the initial prediction model with massive unlabeled nodes. The core idea is to use the initial model to classify unlabeled nodes, pseudo-label some nodes, and add them to the training

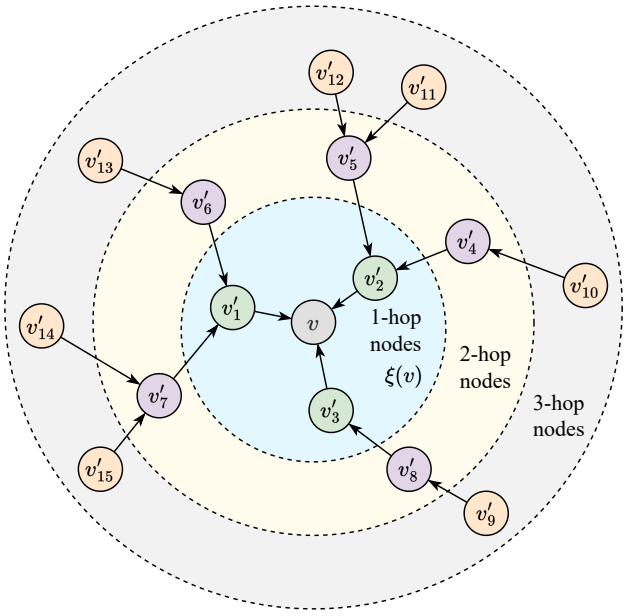

Fig. 2: The aggregation process of nodes in GNN. v is the node that needs to obtain the embedding vector. $\xi(v)$ is the set of nodes connected to node v. In one-hop aggregation, the features of one-hop nodes v_1', v_2' and v_3' are added to v. The features of v, v_4' and v_5' are also added to v_2' as they are neighbors of v_2'. In two-hop aggregation, the enhanced features of v_1', v_2' and v_3' are added to v, which makes node v incorporate the characteristics of nodes v_4' to v_8'. Similarly, the features of nodes v_9' to v_{15}' are fused to v in three-hop aggregation.

set. Since nodes with higher similarity to known default users are more likely to be default users, we constructed a similarity selection module to choose pseudo-label nodes. The nodes that are close to minority class centers in the embedding space will be added to the training set. It should be known that the embedded vector instead of the original features is utilized to calculate the similarity, which has fused the features of neighbor nodes. In other words, this embedding vector has taken into account the information of the borrower's friends.

First, all the samples are classified with the predictive model f_0, shown as

$$\boldsymbol{U}_{F1}, \boldsymbol{U}_{U1}, \boldsymbol{H} = f_0(\boldsymbol{X}), \tag{4}$$

where $\boldsymbol{U}_{F1}$ is the embedding vector set that $\boldsymbol{X}_F$ classified as l_1, $\boldsymbol{U}_{U1}$ is the embedding vector set that $\boldsymbol{X}_U$ classified as l_1, and $\boldsymbol{H}$ is the weight vector set of all sample classification results.

Then, the mean of the embedding vector for default users is calculated through $\boldsymbol{U}_{F1}$, shown as

$$\boldsymbol{u}_0 = \frac{1}{|\boldsymbol{U}_{F1}|} \sum_{\boldsymbol{u} \in \boldsymbol{U}_{F1}} \boldsymbol{u}, \tag{5}$$

where $|\boldsymbol{U}_{F1}|$ is the number of nodes in set $\boldsymbol{U}_{F1}$.

Finally, nodes in $\boldsymbol{U}_{F1}$ whose distance to the center $\boldsymbol{u}_0$ is less than the threshold θ are filtered out, and the corresponding original feature vectors are added to the training set, shown as Formula 6 and 7.

$$(\boldsymbol{X}_a, \boldsymbol{Y}_a) = \{(\boldsymbol{x}_u, l_1) | d(\boldsymbol{u}, \boldsymbol{u}_0) < \theta, \boldsymbol{u} \in \boldsymbol{U}_{U1}\} \tag{6}$$

$$d(\boldsymbol{u}, \boldsymbol{u}_0) = \sqrt{\sum_{i=1}^{m'} (u_i - u_{0i})^2} \tag{7}$$

In these formulas, $\boldsymbol{x}_u$ is the original feature vector corresponding to the embedded vector $\boldsymbol{u}$, $d(\boldsymbol{u}, \boldsymbol{u}_0)$ is the calculated Euler distance between the two vectors, u_i is the i-th element of the vector $\boldsymbol{u}$. θ is determined according to the difference between the number of normal and default users in the labeled data, so as to ensure that the number of the two types of users in the new training set is balanced. The filtered $(\boldsymbol{X}_a, \boldsymbol{Y}_a)$ will be added to the training set for retraining the prediction model.

E. Model retraining

The prediction model needs to be retrained after pseudo-labeled training nodes are selected via similarity selection. It is no guarantee that the labels of nodes newly added to the training set are correct, so the confidence of each sample must be considered. Here, a loss function with confidence is proposed for model training.

First, the classification confidence of each sample in $(\boldsymbol{X}_a, \boldsymbol{Y}_a)$ is calculated utilizing the set of classification weight vectors $\boldsymbol{H}$. For $\forall \boldsymbol{x}_i \in \boldsymbol{X}_a$, the confidence is calculated with

$$c_i = \max(RELU(\boldsymbol{h}_i - \tau)), \tag{8}$$

where $\boldsymbol{h}_i$ is the classification weight vector corresponding to sample i, and $\tau \in [0, 1]$ is the hyperparameter threshold for controlling the confidence of pseudo-labeled samples. The calculated confidence will increase as the threshold decreases, and the influence of the sample on the total loss will also increase. Experiments will be conducted in subsequent sections to determine how different thresholds affect the prediction results. After obtaining the confidence of the pseudo-labeled samples, the loss function for retraining is shown as

$$L_{re} = \sum_{i \in \boldsymbol{V}_F} L(\boldsymbol{h}_i, \widehat{\boldsymbol{h}}_i) + \sum_{j \in \boldsymbol{V}_a} c_j L(\boldsymbol{h}_j, \widehat{\boldsymbol{h}}_j), \tag{9}$$

where $\boldsymbol{L}$ is the cross-entropy function in Formula 3. The loss of labeled samples make up the first half of the loss function, while the loss of pseudo-labeled samples make up the second. It is feasible to control the effect of pseudo-labeled sample loss on the total loss by adjusting τ in Formula 8.

TABLE I: The statistics of the DGraphFin

Nodes	Edges	Classes	Normal users	Default users	Background users	Node features
3,700,550	4,300,999	3	1,210,092	15,509	2,474,949	17

IV. Experiment

A. Experimental design

1) Dataset: Although many datasets for financial default prediction are available, fewer public datasets contain social information. The DGraphFin dataset [42] is used to validate our method. It is a large-scale, real-time dynamic financial dataset with 4.3 million dynamic edges and over 3.7 million nodes. In the dataset, the nodes represent users, and an edge is formed between two users if one is added as an emergency contact by another. Users without any overdue repayments are classified as normal users, while those with at least one overdue repayment are labeled as default users. This dataset is widely used in fields such as fraud detection [20], node classification [43]. An overview of the dataset is shown in Fig.3.

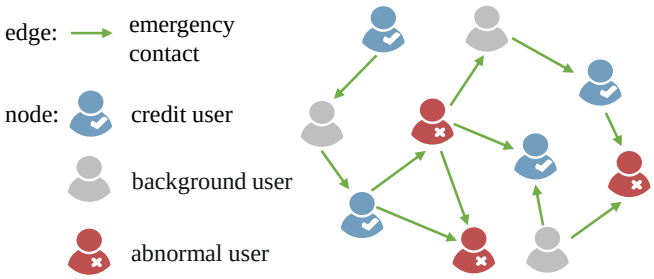

Fig. 3: The overview of DGraphFin. The registered users and their connections constitute the graph.

The dataset provides initial features for each node after desensitization, including gender, age, registration time, repayment date, and so on. In DGraphFin, there are over 2 million background nodes, with a 1:100 ratio of default to normal users. Background nodes are users who have provided their personal profiles but have not taken part in any lending transactions. Table I presents the statistical data for the dataset. In large-scale data scenarios, appropriately processing of background nodes can effectively enhance both data storage capacity and model effectiveness [16].

TABLE II: Experimental hyperparameter settings

Activation function	Optimizer	Learning rate	Dropout	Layers	Batch size	Hidden dimension
Relu	Adam	1e-2	0.3	3	1024	128, 128, 32

2) Experimental setup: The DGraphFin dataset has been randomly split the nodes into training, testing, and validation sets with a ratio of 70:15:15. Before constructing the graph neural network, additional features derived from edges are extracted, and a total of 56 features are utilized for model construction.

In the experiment, the PyTorch and PyTorch Geometric framework were used to build this model. Table II lists the hyperparameters used for model training. Original 17 features are used as the model inputs. Three graph aggregation layers and matching ReLu activation functions make up the graph model. In the three layers, there are 128, 128 and 32 hidden dimensions, respectively. We used the mean of the ten results for the final comparison of the experimental data.

When evaluating the performance of the model, typical metrics like accuracy may not be sufficient since the dataset is imbalanced [32]. Therefore, the area under the receiver operating characteristic curve ($AUC-ROC$) [44] and the $F1-measure$ [22] were used to evaluate the prediction effects.

- The $AUC-ROC$ is a metric used to assess the performance of the classifier. It is determined by the area under the receiver operating characteristic curve, where TPR is plotted on the y-axis and FPR on the x-axis. The definitions of TPR and FPR are as follows.

$$TPR = \frac{TP}{TP+FN} \tag{10}$$

$$FPR = \frac{FP}{TN+FP} \tag{11}$$

In these formulas, TP, FP, TN, and FN are true positive, false positive, true negative, and false negative, respectively. The greater $AUC-ROC$ represents better performance. The $AUC-ROC$ measures the overall performance of the model across various thresholds.

- $F1-measure$ evaluates the quality of the algorithm based on recall and precision, which is equal to the reconciled average of the recall and precision. It effectively evaluates the performance of the model when there is a significant difference in the number of positive and negative samples. The calculation formula is as follows:

$$F1-measure = \frac{2TP}{2TP+FN+FP}. \tag{12}$$

The greater $F1-measure$ represents better performance, according to the metric definition.

B. Experimental results

We compared IG-DP with existing classical methods. These methods are mainly divided into two categories: Unsupervised methods include deepwalk [39], deep graph infomax (DGI) [45]. Supervised methods include multilayer perceptron (MLP) [46], GCN [8], GAT [21], SAGE [20], scalable inception graph neural networks (SIGN) [43], unified message passaging model (UniMP) [47] and GEARSage.

TABLE III: Results of different methods.

Method	AUC -ROC			F1-measure		
	Average	Best	Worst	Average	Best	Worst
DeepWalk	0.6978	0.6986	0.6966	0.0832	0.0846	0.0823
DGI	0.7092	0.7099	0.7086	0.0912	0.0934	0.0893
MLP	0.7208	0.7244	0.7176	0.1046	0.1063	0.1021
GCN	0.7378	0.7404	0.7351	0.1121	0.1145	0.1102
GAT	0.7678	0.7792	0.7653	0.1174	0.1188	0.1162
SAGE	0.7767	0.7788	0.7749	0.1212	0.1230	0.1201
SIGN	0.7820	0.7835	0.7798	0.1268	0.1279	0.1293
UniMP	0.7827	0.7852	0.7813	0.1324	0.1341	0.1311
GEARSage	0.8460	0.8463	0.8458	0.1366	0.1361	0.1370
IG-DP	**0.8676**	**0.8690**	**0.8657**	**0.1426**	**0.1437**	**0.1417**

- DeepWalk: An online representation learning method for graph node embedding. It learns a representation that encodes structural regularities by using local information from truncated random walks as input.
- DGI: A method for learning node representations with graphs. It relies on local mutual information maximization across the patch representations of the graph, gained by graph convolutional architectures.
- MLP: A common feedforward neural network with two layers. Features are passed to the next layer through a forward layer and activation function.
- GCN: A graph-based convolutional neural network. It aggregates node features and leverages the adjacency relationship of nodes to perform information transfer and learn node feature representation.
- GAT: A graph neural network model that aggregates information and learns feature representation of nodes using an attention mechanism.
- SAGE: An inductive learning framework that generates embeddings via learning a function that samples and aggregates features from the local neighborhood of a node.
- SIGN: A scalable graph learning framework that avoids the necessity for graph sampling by employing graph convolutional filters of various sizes, which can be efficiently precomputed and rapidly training and inference.
- UniMP: A model that incorporates feature and label propagation at both the training and inference stages. The Graph Transformer network is utilized as the prediction model, which takes feature embedding and label embedding as input for propagation. Meanwhile, it introduces a shielded label prediction strategy.
- GEARSage: An improved method of GraphSAGE. Different from GraphSAGE, it extracted the features of edges.

The experimental results of different methods are shown in Table III. A comparison of the data in Table III clearly indicates that IG-DP outperforms all other methods. Deepwalk and DGI are unsupervised learning methods with the worst experimental results. It shows that supervised methods outperform unsupervised methods when there is labeled

TABLE IV: Results of different pseudo-labeled sample selection methods.

Method	AUC-ROC			F1-measure		
	Average	Best	Worst	Average	Best	Worst
H-selection	0.8528	0.8532	0.8523	0.1403	0.1408	0.1397
R-selection	0.8504	0.8507	0.8499	0.1390	0.1393	0.1387
L-selection	0.8458	0.8472	0.8420	0.1310	0.1348	0.1298
IG-DP	**0.8676**	**0.8690**	**0.8657**	**0.1426**	**0.1437**	**0.1417**

samples. In supervised models, the worst performance comes from MLP, which only considers node features and ignores their connections. In the other graph-based methods, the connections between the nodes are taken into account, with GEARSage achieving the best performance on both evaluation metrics. However, IG-DP improves these two metrics by 2.6% and 4.5%, respectively. The main reason is that these traditional methods focus on optimizing the propagation of features for their models. They neglect the imbalance in the number of samples used during training. Due to this issue, the prediction effects of the model tend toward normal nodes, which lowers the overall prediction outcomes. IG-DP generates pseudo labels for background nodes, and nodes with higher confidence are selected to expand the original dataset. Background node information was added to the propagation process of the GNN model, and the experimental results were greatly improved.

C. Ablation experiment

1) Impact of similarity selection module: One of the key modules of IG-DP is similarity selection, which is used to select retraining samples from pseudo-labeled background nodes. This section primarily focuses on the impact of different pseudo-labeled sample selection methods under the proposed prediction framework. Three pseudo-labeling methods are selected for comparison.

- H-selection: A traditional GNN-based self-training method. Difference from IG-DP, this method adds pseudo-labeled samples with high confidence that are predicted to be default users to the retraining set.
- R-selection: A GNN-based random sample selection method. This method randomly selects some pseudo-labeled samples that are predicted to be default users and adds them to the retraining set, ensuring a balanced number of samples.
- L-selection: Difference from H-selection, this method adds pseudo-labeled samples with low confidence that are predicted to be default users to the retraining set.

The prediction results of different methods are presented in Table IV. Our method outperforms H-selection, R-selection and A-selection with the metrics of $AUC-ROC$ and $F1-measure$. It improved the two metrics by 1.7% and 1.6%, respectively. The following are the primary reasons. Since the initial classifier is affected by the imbalanced labeled data set, the retraining samples directly selected by the initial prediction model also rely on the imbalanced labeled data set. It makes the retraining samples not expand the classification boundary of the default samples, which in turn leads to the updated model low robust when facing unknown samples. L-selection and R-selection include a large number of noisy samples, resulting in a higher error rate for pseudo-labels. The similarity selection module proposed in this paper uses spatial distance to filter samples again based on the classification results, reducing the impact of imbalanced labeled data on the selected results. Therefore, the method proposed in this paper delivers optimal performance. It also confirms that choosing retraining samples that do not rely on imbalanced labeled datasets plays a key role in improving the prediction performance of the updated model.

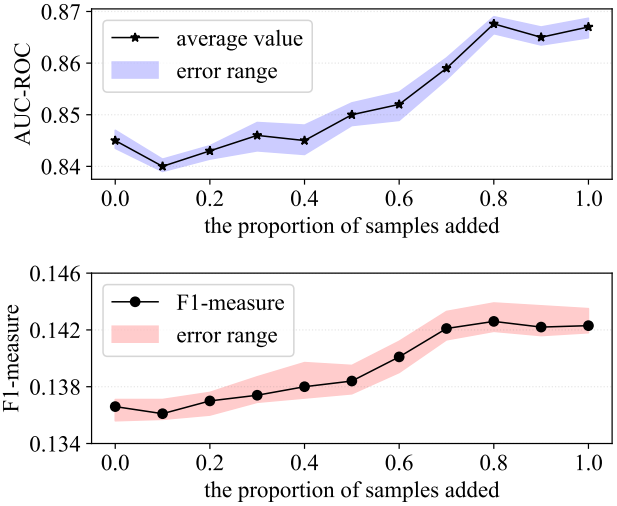

Fig. 4: The impact of added sample proportions

To further explore the impact of the selected sample number on the final prediction results, Fig.4 illustrates the changes in the predictive indicator $AUC-ROC$ and $F1-measure$ as the number of selected pseudo-labeled samples increases. The solid line represents the predicted mean, and the shaded area is the prediction error. The x-axis represents the ratio of the difference between the number of default users and normal users in the training set. The ratio is 0 indicates that pseudo-labeled samples are not selected with similarity selection module, $\boldsymbol{X}_a$ is empty. When the ratio is 1, the number of normal and default users in the retraining data is equal. These two figures demonstrate that the prediction effect first fall and then increases as the proportion of samples increases. The primary reasons are analyzed as follows.

When the ratio is low, the nodes that are extremely near to the central vectors are selected and added to the retraining data. At this stage, the number of default users has increased, but the prediction boundary of the prediction model has not been effectively expanded, which has exacerbated the overfitting problem and the prediction effect has somewhat decreased. As the proportion of new samples increases, a large number of correctly labeled samples whose distribution is different from the initial prediction model are added to the retraining data. The prediction effect is evidently improved since these samples are essential in expanding the classification boundary of the model. This analysis further highlights that identifying samples with accurate predictions but distributions different from the original default users is essential for enhancing prediction performance.

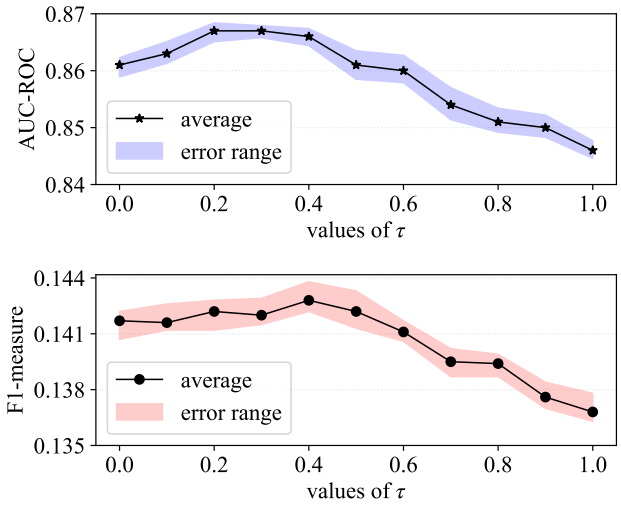

Fig. 5: The impact of parameter τ

2) The impact of confidence threshold: A novel loss function incorporating the confidence of pseudo-labeled samples is specifically designed for model retraining in this study. The parameter τ within the loss function plays a crucial role as it determines the weight pseudo-labeled samples contribute to the overall loss. Specifically, when τ is smaller, the pseudo-labeled samples have a higher impact on the training process as their contribution to the loss is proportionally greater. Conversely, as τ increases, the contribution of pseudo-labeled samples to the total loss diminishes, thereby reducing their influence on the model's training process. The relationship between τ and model performance, in terms of $AUC-ROC$ and $F1-measure$, is systematically evaluated and visualized in Fig.5.

The experimental results demonstrate that model performance first improves and then deteriorates as τ increases. The $AUC-ROC$ and $F1-measure$ reach their optimal values when τ lies within the range of 0.2 to 0.4, suggesting a balance between leveraging high-confidence pseudo-labeled samples and avoiding adverse impacts from mislabeled data. When τ is too small, despite pseudo-labeled samples contributing significantly to the loss, the model's performance does not improve consistently. This is attributed to the presence of mislabeled nodes among the pseudo-labeled samples, which negatively impact the training process. The high impact of these erroneous samples can degrade the model's predictive capabilities, highlighting the need for refined confidence evaluation mechanisms.

On the other hand, as τ increases beyond the optimal range, the loss contribution of pseudo-labeled samples diminishes, reducing their influence on the retrained model. While this may initially reduce the negative impact of misclassified samples, it also limits the positive contribution of genuinely high-confidence pseudo-labeled nodes. When τ becomes excessively large, the influence of pseudo-labeled samples on the retraining process becomes negligible, and the model performance converges to that of the initial model trained on the labeled dataset only. This result underscores the importance of appropriately tuning τ to balance the trade-off between incorporating pseudo-labeled samples and mitigating the risks associated with mislabeled data.

V. Conclusion

Default prediction is a critical research topic in the current financial domain. The primary challenge in this area is the overfitting issue caused by the imbalance in the number of labeled samples. This paper introduces IG-DP, a default prediction method based on graph neural networks, which leverages a large number of unlabeled background samples to improve prediction performance and address the class imbalance issue. The publicly available dataset DGraphFin is utilized to evaluate the performance of this model, and the ablation experiments are conducted to examine the effects of model parameters.

Several potential directions for future research could further improve the proposed method. First, in this paper, background nodes are pseudo-labeled using the similarity selection module. This method assumes that the node embedding vector of default user has one cluster in the vector space. It could expand to several clusters and explore the effect of cluster size. Second, the IG-DP relies on the expanded dataset when retraining the model. However, in real-world economic scenarios, fully retraining the model may be impractical. Thus, another promising direction is to develop an online update mechanism based on graph neural networks.

References

[1] C. Wei, K. Sohn, C. Mellina, A. Yuille, and F. Yang, "Crest: A class-rebalancing self-training framework for imbalanced semi-supervised learning," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 10 857–10 866.

[2] D. Wang, Z. Zhang, Y. Zhao, K. Huang, Y. Kang, and J. Zhou, "Financial default prediction via motif-preserving graph neural network with curriculum learning," in *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2023, pp. 2233–2242.

[3] M. J. Korkoman and M. Abdullah, "Evolutionary algorithms based on oversampling techniques for enhancing the imbalanced credit card fraud detection," *Journal of Intelligent & Fuzzy Systems*, no. Preprint, pp. 1–13, 2023.

[4] Q. Liu, Y. Luo, S. Wu, Z. Zhang, X. Yue, H. Jin, and L. Wang, "Rmt-net: Reject-aware multi-task network for modeling missing-not-at-random data in financial credit scoring," *IEEE Transactions on Knowledge and Data Engineering*, vol. 35, no. 7, pp. 7427–7439, 2022.

[5] Q. Liu, Z. Liu, H. Zhang, Y. Chen, and J. Zhu, "Mining cross features for financial credit risk assessment," in *Proceedings of the 30th ACM international conference on information & knowledge management*, 2021, pp. 1069–1078.

[6] H. Li and W. Wu, "Loan default predictability with explainable machine learning," *Finance Research Letters*, vol. 60, p. 104867, 2024.

[7] U. M. Sirisha, M. C. Belavagi, and G. Attigeri, "Profit prediction using arima, sarima and lstm models in time series forecasting: A comparison," *IEEE Access*, vol. 10, pp. 124 715–124 727, 2022.

[8] L. Deng and Y. Zhao, "Investment lag, financially constraints and company value—evidence from china," *Emerging Markets Finance and Trade*, vol. 58, no. 11, pp. 3034–3047, 2022.

[9] A. Singh, A. Gupta, H. Wadhwa, S. Asthana, and A. Arora, "Temporal debiasing using adversarial loss based gnn architecture for crypto fraud detection," in *2021 20th IEEE International Conference on Machine Learning and Applications (ICMLA)*. IEEE, 2021, pp. 391–396.

[10] Y. Teng and K. Yang, "Research on enhanced multi-head self-attention social recommendation algorithm based on graph neural network." *IAENG International Journal of Computer Science*, vol. 51, no. 7, pp. 754–764, 2024.

[11] H. Hu, D. Yang, and Y. Zhang, "Dprec: Social recommendation based on dynamic user preferences." *IAENG International Journal of Computer Science*, vol. 50, no. 3, pp. 980–987, 2023.

[12] Y. Wang, Y. Huang, Q. Wang, C. Zhao, Z. Zhang, and J. Chen, "Graph-based self-training for semi-supervised deep similarity learning," *Sensors*, vol. 23, no. 8, p. 3944, 2023.

[13] T. Zhao, X. Zhang, and S. Wang, "Graphsmote: Imbalanced node classification on graphs with graph neural networks," in *Proceedings of the 14th ACM international conference on web search and data mining*, 2021, pp. 833–841.

[14] D. Chen, Y. Lin, G. Zhao, X. Ren, P. Li, J. Zhou, and X. Sun, "Topology-imbalance learning for semi-supervised node classification," *Advances in Neural Information Processing Systems*, vol. 34, pp. 29 885–29 897, 2021.

[15] R. Blanco, E. Fernández-Ortiz, M. García-Posada, and S. Mayordomo, "A new estimation of default probabilities based on non-performing loans," *Finance Research Letters*, vol. 62, p. 105149, 2024.

[16] E. Dai, C. Aggarwal, and S. Wang, "Nrgnn: Learning a label noise resistant graph neural network on sparsely and noisily labeled graphs," in *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, 2021, pp. 227–236.

[17] Y. Ma, P. Zhang, S. Duan, and T. Zhang, "Credit default prediction of chinese real estate listed companies based on explainable machine learning," *Finance Research Letters*, vol. 58, p. 104305, 2023.

[18] X. Wang, H. Liu, C. Shi, and C. Yang, "Be confident! towards trustworthy graph neural networks via confidence calibration," *Advances in Neural Information Processing Systems*, vol. 34, pp. 23 768–23 779, 2021.

[19] Y. Chen, J. You, J. He, Y. Lin, Y. Peng, C. Wu, and Y. Zhu, "Spgnn: Learning structure and position information from graphs," *Neural Networks*, vol. 161, pp. 505–514, 2023.

[20] W. Hamilton, Z. Ying, and J. Leskovec, "Inductive representation learning on large graphs," *Advances in neural information processing systems*, vol. 30, 2017.

[21] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Lio, and Y. Bengio, "Graph attention networks," *arXiv preprint arXiv:1710.10903*, 2017.

[22] T.-T. Wong, "Linear approximation of f-measure for the performance evaluation of classification algorithms on imbalanced data sets," *IEEE Transactions on Knowledge and Data Engineering*, vol. 34, no. 2, pp. 753–763, 2020.

[23] D. Cheng, Z. Niu, and L. Zhang, "Delinquent events prediction in temporal networked-guarantee loans," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 34, no. 4, pp. 1692–1704, 2020.

[24] B. Peng, Y. Ding, Q. Xia, and Y. Yang, "Recurrent neural networks integrate multiple graph operators for spatial time series prediction," *Applied Intelligence*, vol. 53, no. 21, pp. 26 067–26 078, 2023.

[25] H. Yang, X. Yan, X. Dai, Y. Chen, and J. Cheng, "Self-enhanced gnn: Improving graph neural networks using model outputs," in *2021 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2021, pp. 1–8.

[26] Y. Duan, X. Liu, A. Jatowt, H.-t. Yu, S. Lynden, K.-S. Kim, and A. Matono, "Sorag: Synthetic data over-sampling strategy on multi-label graphs," *Remote Sensing*, vol. 14, no. 18, p. 4479, 2022.

[27] S. Jiang, R. Dong, J. Wang, and M. Xia, "Credit card fraud detection based on unsupervised attentional anomaly detection network," *Systems*, vol. 11, no. 6, p. 305, 2023.

[28] H. Zhang, R. Song, L. Wang, L. Zhang, D. Wang, C. Wang, and W. Zhang, "Classification of brain disorders in rs-fmri via local-to-global graph neural networks," *IEEE Transactions on Medical Imaging*, vol. 42, no. 2, pp. 444–425, 2022.

[29] T. S. Jepsen, C. S. Jensen, and T. D. Nielsen, "Relational fusion networks: Graph convolutional networks for road networks," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 1, pp. 418–429, 2020.

[30] R.-C. Chen *et al.*, "Using deep learning to predict user rating on imbalance classification data." *IAENG International Journal of Computer Science*, vol. 46, no. 1, pp. 109–117, 2019.

[31] P. Wanda and H. J. Jie, "Deepsentiment: Finding malicious sentiment in online social network based on dynamic deep learning." *IAENG International Journal of Computer Science*, vol. 46, no. 4, pp. 616–627, 2019.

[32] A. K. Menon, S. Jayasumana, A. S. Rawat, H. Jain, A. Veit, and S. Kumar, "Long-tail learning via logit adjustment," *arXiv preprint arXiv:2007.07314*, 2020.

[33] W. Chong, N. Chen, and C. Fang, "Rbsp-boosting: A shapley value-based resampling approach for imbalanced data classification," *Intelligent Data Analysis*, vol. 26, no. 6, pp. 1579–1595, 2022.

[34] P. Soltanzadeh, M. R. Feizi-Derakhshi, and M. Hashemzadeh, "Addressing the class-imbalance and class-overlap problems by a metaheuristic-based under-sampling approach," *Pattern Recognition*, vol. 143, p. 109721, 2023.

[35] C. Cui, J. Wang, W. Wei, and J. Liang, "Hybrid sampling-based contrastive learning for imbalanced node classification," *International Journal of Machine Learning and Cybernetics*, vol. 14, no. 3, pp. 989–1001, 2023.

[36] J. Park, J. Song, and E. Yang, "Graphens: Neighbor-aware ego network synthesis for class-imbalanced node classification," in *The Tenth International Conference on Learning Representations, ICLR 2022*. International Conference on Learning Representations (ICLR), 2022.

[37] J.-R. Chang, L.-S. Chen, and L.-W. Lin, "A novel cluster based over-sampling approach for classifying imbalanced sentiment data," *IAENG International Journal of Computer Science*, vol. 48, no. 4, pp. 1118–1128, 2021.

[38] M. Chen, J. Dou, Y. Fan, and Y. Song, "Robust semi-supervised classification for imbalanced and incomplete data," *Journal of Intelligent & Fuzzy Systems*, vol. 45, no. 2, pp. 2781–2797, 2023.

[39] S. Zhao and J. Li, "A semi-supervised self-training method based on density peaks and natural neighbors," *Journal of Ambient Intelligence and Humanized Computing*, vol. 12, pp. 2939–2953, 2021.

[40] J. Jiao, H. Li, and J. Lin, "Self-training reinforced adversarial adaptation for machine fault diagnosis," *IEEE Transactions on Industrial Electronics*, vol. 70, no. 11, pp. 11 649–11 658, 2022.

[41] J. Wang, Y. Wu, S. Li, and F. Nie, "A self-training algorithm based on the two-stage data editing method with mass-based dissimilarity," *Neural Networks*, vol. 168, pp. 431–449, 2023.

[42] X. Huang, Y. Yang, Y. Wang, C. Wang, Z. Zhang, J. Xu, L. Chen, and M. Vazirgiannis, "Dgraph: A large-scale financial dataset for graph anomaly detection," *Advances in Neural Information Processing Systems*, vol. 35, pp. 22 765–22 777, 2022.

[43] F. Frasca, E. Rossi, D. Eynard, B. Chamberlain, M. Bronstein, and F. Monti, "Sign: Scalable inception graph neural networks," *arXiv preprint arXiv:2004.11198*, 2020.

[44] P. A. Jaskowiak, I. G. Costa, and R. J. Campello, "The area under the roc curve as a measure of clustering quality," *Data Mining and Knowledge Discovery*, vol. 36, no. 3, pp. 1219–1245, 2022.

[45] P. Veličković, W. Fedus, W. L. Hamilton, P. Liò, Y. Bengio, and R. D. Hjelm, "Deep graph infomax," *arXiv preprint arXiv:1809.10341*, 2018.

[46] H. Taud and J.-F. Mas, "Multilayer perceptron (mlp)," *Geomatic approaches for modeling land change scenarios*, pp. 451–455, 2018.

[47] Y. Shi, Z. Huang, S. Feng, H. Zhong, W. Wang, and Y. Sun, "Masked label prediction: Unified message passing model for semi-supervised classification," *arXiv preprint arXiv:2009.03509*, 2020.