Dynamic Graph Learning with Static Relations for Credit Risk Assessment

Qi Yuan[1], Yang Liu[1*], Yateng Tang[2], Xinhuan Chen[2], Xuehao Zheng[2], Qing He[1], Xiang Ao[1, 3*]

[1]University of Chinese Academy of Sciences, CAS
[2]Tencent Weixin Group [3]Institute of Intelligent Computing Technology, Suzhou, CAS
yuanqi15@mails.ucas.ac.cn, liuyang520ict@gmail.com, heq2002@hotmail.com, aoxiang@ucas.ac.cn,
{fredyttang, holidaychen, xuehaozheng}@tencent.com

Abstract

Credit risk assessment has increasingly become a prominent research field due to the dramatically increased incidents of financial default. Traditional graph-based methods have been developed to detect defaulters within user-merchant commercial payment networks. However, these methods face challenges in detecting complex risks, primarily due to their neglect of user-to-user fund transfer interactions and the underutilization of temporal information. In this paper, we propose a novel framework named Dynamic Graph Neural Network with Static Relations (DGNN-SR) for credit risk assessment, which can encode the dynamic transaction graph and the static fund transfer graph simultaneously. To fully harness the temporal information, DGNN-SR employs a multi-view time encoder to explore the semantics of both relative and absolute time. To enhance the dynamic representations with static relations, we devise an adaptive re-weighting strategy to incorporate the static relations into the dynamic representations of time encoder, which extracts more discriminative features for risk assessment. Extensive experiments on two real-world business datasets demonstrate that our proposed method achieves a 0.85% - 2.5% improvement over existing SOTA methods.

Introduction

With 2.3 trillion cashless transactions worldwide forecasted for 2027 by Capgemini[1] compared to 1.3 trillion in 2023, the future of payments is undoubtedly digital. Inclusive finance, while facilitating digital payments, also reduces the cost of default, leading to a substantial increase in the default rate. Following 2023 with 38 billion dollars in losses due to online default (the forecast is 91 billion by 2028), it is evident that assessing credit risk on online platforms becomes a crucial task to mitigate the losses resulting from defaults.

Credit risk assessment, which aims to evaluate the probability that a user will default on a payment, plays an important role for online payment platforms (Li 2019; Wang et al. 2020; Moscato, Picariello, and Sperlí 2021; Zhu et al. 2021; Kanaparthi 2023). Due to privacy restrictions, online platforms lack access to credit reporting provided by banks or the government, which is the basis for traditional methods (Crook, Edelman, and Thomas 2007; Chen, Ribeiro, and Chen 2016). The challenge of this task lies in how to assess the credit risk via consumer records, such as purchase payments and fund transfers, which form a hybrid transaction graph encompassing both user-merchant transaction interactions and user-user fund transfer interactions. For instance, if a user makes multiple purchases from a merchant within a short period and subsequently receives a transfer, it could indicate a potential cash-out behavior, resulting in a poor credit record for this user. Therefore, considering user-merchant and user-user interactions simultaneously, credit risk signals are more likely to be detected.

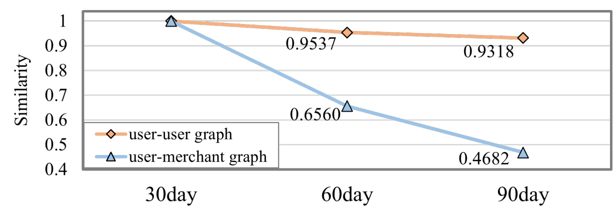


Figure 1: Neighbor set similarity relative to the 30-day neighbor set over different time windows. The user-user graph shows high stability with minor variations in similarity. In contrast, the user-merchant graph exhibits a sharp decline in similarity as the time window increases.

Existing works widely adopt graphs to model financial interactions between users and merchants (Liu et al. 2018; Sukharev et al. 2020; Zhong et al. 2020; Yang et al. 2021). An edge is constructed between a user-merchant pair if the user completed a payment with the merchant, along with the timestamp of the payment. In this way, the user-merchant graph reflects commercial payment behaviors and is a continuous-time dynamic graph. Similarly, an edge is connected when a user transfers funds to another user. Different from commercial payment, we observe that user-user fund transfer behaviors indicate a more static relationship between users. As shown in Figure 1, user-user interactions maintain a high similarity across different time windows, suggesting stable relationships between users that are relatively unaffected by time. This implies that users usually transfer funds with a small, fixed set of friends and the times-

[*]Corresponding authors.

[1]https://www.capgemini.com/insights/research-library/payments-top-trends-2024/

tamps do not include much extra information. Consequently, the user-user interactions are modeled by a static graph in this work.

The challenges for designing dynamic graph neural networks to assess credit risk lie in two aspects. Firstly, some default users can only be detected by considering both user-merchant and user-user interactions simultaneously, such as the cash-out behaviors we have discussed before. However, existing GNN-based solutions struggle to detect complex risks, as they are unable to effectively integrate dynamic and static graphs. Secondly, existing dynamic graph representation methods only consider relative time information in the temporal dimension (Xu et al. 2020; Cong et al. 2023; Yu et al. 2023), which is appropriate for tasks that rely solely on temporal proximity such as the temporal link prediction task. However, for the task of credit risk assessment, an exclusive focus on relative time can result in significant information loss. For example, people tend to go out shopping and spend more during holidays and festivals. Absolute time is crucial for modeling such kind of behaviors and relative time loses the holiday information.

To overcome the above challenges, we propose a novel framework named Dynamic Graph Neural Network with Static Relations (DGNN-SR) for credit risk assessment, which can encode the dynamic user-merchant graph with the static user-user graph simultaneously. The contributions of this work are listed as follows:

- We propose a novel framework, DGNN-SR, which integrates user-merchant payments modeled as a dynamic graph with user-user fund transfers modeled as a static graph. This fused representation enables a more comprehensive understanding of user behavior for credit risk assessment.
- To fully leverage temporal information, we introduce a multi-view time encoder combined with a temporal attention mechanism.
- Combining with different backbones, our framework consistently achieves superior performance on two real-world business datasets. Experiments comparing different kinds of baselines further verify the robustness of the new framework.

Preliminaries

Notations

The hybrid transaction graph consists of user-merchant commercial payment interactions and user-user fund transfer interactions. The user-merchant interactions are constructed as a continuous-time dynamic graph G_D. A continuous-time dynamic graph can be represented as a sequence of non-decreasing chronological interactions $G_D = \{(u_1^s, u_1^d, t_1'), (u_2^s, u_2^d, t_2'), \cdots\}$ with $0 \leq t_1' \leq t_2' \leq \cdots$, where $u_i^s, u_i^d \in V_D$ denote the source node and destination node of the i-th interaction at timestamp t_i', V_D denotes the node set of dynamic graph. Each node $u \in V_D$ can be associated with node feature $\boldsymbol{x}_u \in \mathbb{R}^{d_{V_D}}$ and timestamp t, and each interaction (u, v, t') has link feature $\boldsymbol{e}_{u,v}^{t'} \in \mathbb{R}^{d_{E_D}}$. d_{V_D} and d_{E_D} denote the dimensions of the node feature and edge feature in the dynamic graph, respectively.

The user-user interactions are formulated as a static graph $G_S = \{V_S, E_S\}$, where V_S is the node set and E_S is the edge set. Each node $v \in V_S$ can be associated with node feature $\boldsymbol{x}_v \in \mathbb{R}^{d_{V_S}}$, where d_{V_S} denotes the dimension of the node feature in G_S.

Problem Definition

The objective of credit risk assessment is to predict the probability that a user will commit a financial default based on a hybrid transaction graph. Given a timestamp t and the historical hybrid transaction graph $< G_D, G_S >$ before t, the task is to learn a time-aware representation $\boldsymbol{h}_u$ for each user u and predict whether u is a defaulter or not. All users to be assessed are collected as a user set U. Each user $u \in U$ is assigned a label $y_u \in \{0, 1\}$ to signify whether u is a defaulter ($y_u = 1$) or not. Note that the user set U is a subset of $V_D \cap V_S$.

Methodology

To utilize both dynamic commercial payment graph and static fund transfer graph for credit risk assessment, as illustrated in Figure 2, the proposed DGNN-SR framework has two branches: the dynamic branch and the static branch. In the dynamic branch, we model user-merchant interactions as a continuous-time dynamic graph wherein each edge and node is associated with a timestamp. These timestamps are encoded via a multi-view time encoder, encompassing a relative time encoder, absolute time encoder, and time semantics encoder . The encoded time embeddings are fed into both the dynamic backbone and the temporal attention module. The temporal attention module is proposed as a replacement for the conventional self-attention module in dynamic backbones. In the static branch, the user-user payment interactions are structured as a static graph, and user embeddings are computed using static backbones. The user embeddings from both the dynamic and static graphs are input into re-weighting units, and the re-weighted embeddings of the same user are summated for classification.

Multi-view Time Encoder

Most existing dynamic graph learning methods focus on utilizing relative time, resulting in the underutilization of time information. To address this issue, we design a multi-view time encoder(MTE) to fully utilize time information.

Firstly, based on the assumption that the relative timespan uncovers more crucial temporal information, following TGAT (Xu et al. 2020), the periodic temporal pattern of time interval $(t - t')$ is encoded via Eq.(1),

$$\boldsymbol{z}_{t,t'}^{re} = \sqrt{\frac{1}{d_{T_{re}}}} \{\cos(\omega_i(t - t')), \sin(\omega_i(t - t'))\}_{i=1}^{d_{T_{re}}} \quad (1)$$

where $\omega_1, \cdots, \omega_{d_{T_{re}}}$ are trainable parameters. $d_{T_{re}}$ is the dimension of relative time encoding.

However, relying solely on relative time information leads to information loss, as relative time can only represent temporal proximity. Absolute timestamps, on the other

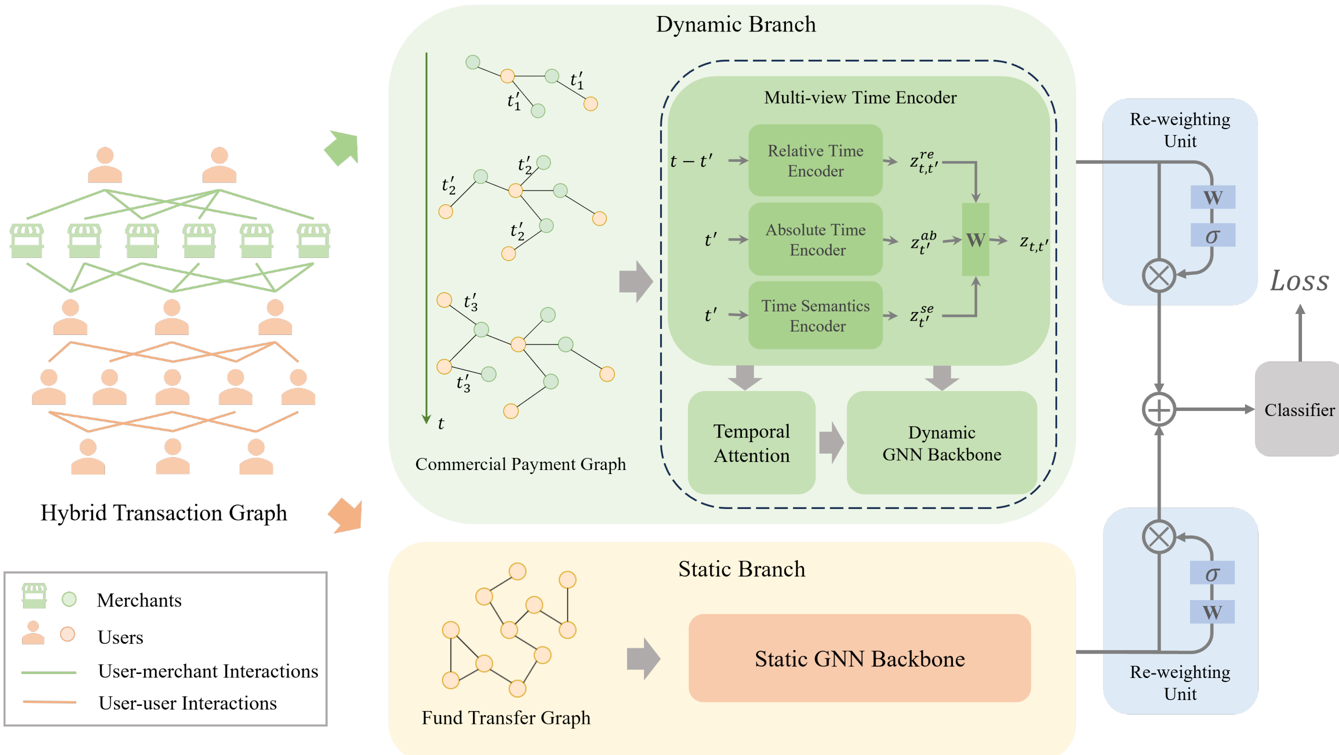


Figure 2: Overview of DGNN-SR. The Hybrid Transaction Graph contains user-merchant interactions and user-user interactions. User-merchant interactions form a continuous-time dynamic graph with timestamp-encoded edges and nodes, while user-user interactions constitute a static graph. A multi-view time encoder processes the temporal information, and the resulting embeddings are input into temporal attention and dynamic backbones. The user embeddings from both graphs are re-weighted and combined for classification.

hand, encompass semantic information not captured by relative timestamps, including fixed periodic information (e.g., month, day, weekday, hour) and occurrences of holidays and special events. Similar to relative time encoding, we design a trainable absolute time encoder and encode absolute timestamp t' as Eq.(2),

$$\boldsymbol{z}_{t'}^{ab} = \sqrt{\frac{1}{d_{T_{ab}}}}\{\cos(\omega_i' t'), \sin(\omega_i' t')\}_{i=1}^{d_{T_{ab}}} \tag{2}$$

where $\omega_1', \cdots, \omega_{d_{T_{ab}}}'$ are trainable parameters. $d_{T_{ab}}$ is the dimension of absolute time encoding.

In addition, to explicitly exploit the semantic information of absolute time and better represent the periodicity, a sine-cosine transformation-based explicit time embedding is further proposed with different granularities, namely month, day, week, and hour. For each granularity with period T, the absolute transaction time t' is encoded as Eq.(3).

$$\phi(t', T) = (\cos(\frac{2\pi t'}{T}), \sin(\frac{2\pi t'}{T})) \tag{3}$$

To further utilize the semantic information, the timestamp is also encoded as binary features to denote occurrences of holidays and special events. All the periodic features and binary features are concatenated to form the semantic feature $\boldsymbol{x}_{se} \in \mathbb{R}^{d_{se}}$, which is subsequently encoded as Eq.(4),

$$\boldsymbol{z}_{t'}^{se} = \mathbf{W_{se}} \boldsymbol{x}_{se} \tag{4}$$

where $W_{se} \in \mathbb{R}^{d_{T_{se}} \times d_{se}}$ is the weight matrix, $d_{T_{se}}$ is the dimension of semantic encoding.

Finally, we obtain the encoded time embedding via Eq.(5),

$$\mathrm{MTE}(t, t') = \boldsymbol{z_{t,t'}} = \mathbf{W_{t'}}(\boldsymbol{z}_{t,t'}^{re} || \boldsymbol{z}_{t'}^{ab} || \boldsymbol{z}_{t'}^{se}) \tag{5}$$

where $||$ denotes the concatenation operation, $W_{t'} \in \mathbb{R}^{d_t \times (d_{T_{re}} + d_{T_{ab}} + d_{T_{se}})}$ is the weight matrix, d_t is the dimension of multi-view time encoding.

Temporal Attention

Numerous DGNNs have employed attention mechanisms to adaptively aggregate neighbor information. An attention function can be characterized as mapping a query and a set of key-value pairs to an output, wherein the query, keys, values, and outputs are all vector representations. The output is computed as a weighted sum of the values, with the weight assigned to each value being determined by the dot product of the query and all the corresponding keys, as shown in Eq.(6),

$$\mathrm{Attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \mathrm{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^{\top}}{\sqrt{d_k}}\right)\mathbf{V} \tag{6}$$

where $\mathbf{Q}, \mathbf{K}, \mathbf{V}$ represent the query matrix, key matrix, and value matrix, respectively.

Nonetheless, attention-based DGNNs primarily consider time embedding as a component of edge features or node features, thereby overlooking the crucial influence of temporal information on the attention mechanism. Consequently, we propose the incorporation of temporal information into the attention mechanism, with the aim of more effectively capturing time-related relationships. Similar to temporal attention in language models (Rosin and Radinsky 2022), given $\mathbf{Q}, \mathbf{K}, \mathbf{V}, \mathbf{T}$ representing the query matrix, key matrix, value matrix, and time matrix, we define temporal attention as Eq.(7).

$$\text{T-Attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V}, \mathbf{T}) = \text{softmax}\left(\frac{\mathbf{Q}\frac{\mathbf{T}^\top \mathbf{T}}{||\mathbf{T}||}\mathbf{K}^\top}{\sqrt{d_k}}\right)\mathbf{V} \tag{7}$$

The $\mathbf{Q}, \mathbf{K}, \mathbf{V}, \mathbf{T}$ are obtained by linear projections as Eq.(8),

$$\begin{aligned} \mathbf{Q} = \mathbf{W}_Q\mathbf{X}_Q, \mathbf{K} = \mathbf{W}_K\mathbf{X}_K, \\ \mathbf{V} = \mathbf{W}_V\mathbf{X}_V, \mathbf{T} = \mathbf{W}_T\mathbf{X}_T \end{aligned} \tag{8}$$

where $\mathbf{W}_Q, \mathbf{W}_K, \mathbf{W}_V, \mathbf{W}_T$ are weight matrices. In DGNNs, the central node acts as a query node, and its neighbors and edges act as keys. For a query node q with the timestamp t_q and its neighbors $N_k = \{k_i\}_{i=1}^{|N_k|}, k_i \in V_D$ associated with edges $E_k = \{(q, k_i, t'_i)\}_{i=1}^{|N_k|}$, the query features $\mathbf{X}_Q$, key features $\mathbf{X}_K$, value features $\mathbf{X}_V$ and time features $\mathbf{X}_T$ are calculated by Eq.(9),

$$\begin{aligned} &\mathbf{X}_Q = \boldsymbol{x}_q || \boldsymbol{z}_{t_q,t_q} \\ &\mathbf{X}_K = \left[\boldsymbol{r}_K^{(1)}, \boldsymbol{r}_K^{(2)}, \cdots, \boldsymbol{r}_K^{(|N_k|)}\right]^\top, \boldsymbol{r}_K^{(i)} = \boldsymbol{x}_{k_i} || \boldsymbol{e}_{q,k_i}^{t'_i} || \boldsymbol{z}_{t_q,t'_i} \\ &\mathbf{X}_V = \mathbf{X}_K \\ &\mathbf{X}_T = \left[\boldsymbol{r}_T^{(1)}, \boldsymbol{r}_T^{(2)}, \cdots, \boldsymbol{r}_T^{(|N_k|)}\right]^\top, \boldsymbol{r}_T^{(i)} = \boldsymbol{z}_{t_q,t'_i} \end{aligned} \tag{9}$$

where $|N_k|$ is the number of key neighbors, $\boldsymbol{x}_u$ denotes the node feature of u, $\boldsymbol{e}_{u,v}^{t'}$ denotes the edge feature of edge (u, v, t'), $\boldsymbol{z}_{t,t'}$ denotes the time embedding calculating by multi-view time encoder. Intuitively, by computing the autocorrelation of the time factor, the attention weights are henceforth conditioned on the time variable, implying that they are time-dependent.

Adaptive Re-weighting

We formulate user-user interaction data as static graphs and employ graph neural networks to learn user representations. For each user $u \in U$ under assessment, a representation vector $\boldsymbol{h}_u^s$ is derived from the static graph, and another representation vector $\boldsymbol{h}_u^d$ is obtained from the dynamic graph, formulated in Eq.(10),

$$\begin{aligned} \boldsymbol{h}_u^s = f^S(G_S(u)) \\ \boldsymbol{h}_u^d = f^D(G_D(u)) \end{aligned} \tag{10}$$

where f^S is a static GNN, and f^D is a dynamic GNN that contains the multi-view time encoder and temporal attention mechanism.

In order to incorporate the static relations, we introduce a weighted self-gated method for re-weighting the two vectors and subsequently fusing them together. The re-weighting unit is expressed by the element-wise product of the input and the sigmoid activation as Eq.(11),

$$\kappa(\boldsymbol{h}) = \boldsymbol{h} \odot \sigma(\mathbf{W}_g\boldsymbol{h} + \boldsymbol{b}_g) \tag{11}$$

where $\boldsymbol{h} \in \mathbb{R}^d$ is d-dimension input vector, $\mathbf{W}_g$ is $N \times N$ weighting matrix, $\boldsymbol{b}_g \in \mathbb{R}^d$ is d-dimensional bias vector, $\sigma(\cdot)$ represents the element-wise sigmoid function, and $\odot$ represents element-wise products. The embedding of users is obtained as Eq.(12).

$$\boldsymbol{h}_u = \kappa(\boldsymbol{h}_u^s) + \kappa(\boldsymbol{h}_u^d) \tag{12}$$

The self-gating mechanism is introduced to dynamically assign importance to static and dynamic representations for each user.

Subsequently, each user embedding is input into a Multi-Layer Perceptron(MLP) classifier, ultimately enabling the prediction of the user's credit default, as shown in Eq.(13).

$$\hat{y}_u = \text{MLP}(\boldsymbol{h}_u) \tag{13}$$

We perform model training by minimizing the Binary Cross-Entropy Loss as Eq. (14), with N as the number of training samples.

$$\mathcal{L} = -\frac{1}{N}\sum_{u \in U}\left[y_u \text{log}(\hat{y}_u) + (1 - y_u)\text{log}(1 - \hat{y}_u)\right] \tag{14}$$

Experiments

In this section, we present and discuss the experimental results obtained from real-world datasets to illustrate the effectiveness of DGNN-SR, with the aim of answering the following research questions:

- **RQ1**: Does our proposed SyGRA model surpass both static and dynamic baseline models in performance?
- **RQ2**: How do the multi-view time encoder (MTE) and the temporal attention mechanism contribute to the performance enhancement of DGNNs?
- **RQ3**: How do different combination and fusion strategies affect the performance of integrating static and dynamic GNNs?

Experimental Setup

Datasets. We collect two real-world datasets[2] called D1 and D2 from Tencent Mobile Payment, ensuring full compliance with security and privacy policies. Each dataset encompasses two distinct components: (1) user-merchant payment interaction data and (2) user-user payment interaction data. The user-merchant payment interaction data is constructed as a continuous-time dynamic graph with users and

[2]The datasets in this paper were properly sampled only for testing purposes and do not imply any commercial information. All users' private information is removed from the dataset. Besides, the experiment was conducted locally on Tencent's server by formal employees who strictly followed data protection regulations.

Table 1: Statistics of datasets

Datasets	Components	Users	Nodes	Edges	Node Features	Edge Features	%Pos. Rate
D1	User-merchant Graph User-user Graph	458,739	916,903 15,030,759	24,405,055 48,117,469	1,968 1,187	313 -	7.97%
D2	User-merchant Graph User-user Graph	1,010,535	2,797,404 40,725,504	96,244,594 119,549,291	1,992 585	307 -	8.55%

merchants as nodes and interactions between users and merchants as edges. On the other hand, the user-user payment interaction data is built as a static graph with users as nodes and interactions between users as edges.

The statistics of datasets are exhibited in Table 1. The two datasets are split chronologically into training, validation, and test sets according to user timestamps. The respective ratios for training, validation, and test sets in Dataset D1 are approximately 12:1:1, while those for Dataset D2 are approximately 20:1:1.

Baselines. We choose six baselines for comparison, and also as the backbone of our proposed method, including three static GNNs: GCN (Kipf and Welling 2016), GAT (Veličković et al. 2017) and GraphSAGE (Hamilton, Ying, and Leskovec 2017), and three state-of-the-art continuous-time dynamic graph representation learning methods: TGAT (Xu et al. 2020), GraphMixer (Cong et al. 2023), and DyGFormer (Yu et al. 2023). Note that the static GNNs and Dynamic GNNs are employed explicitly for modeling user-user graphs and user-merchant graphs, respectively, with different node features.

Implementation Details. All experiments are conducted on a server with an A100 GPU. Our DGNN-SR model is implemented using Pytorch and Pytorch Geometric frameworks. For optimization, we employ the AdamW Optimizer with a learning rate of 1e-4. In terms of parameters, we utilize a 2-layer MLP classifier with a hidden dimension of 256. The binary features in the time semantic encoder contain two items: *is_Weekend* and *is_Holiday*. The batch size is set to 8000, and all models are trained for 100 epochs, incorporating an early stopping strategy with a patience of 5. We conduct the experiments five times with seeds ranging from 0 to 4 and report the average scores with deviations.

Metrics. We adopt three metrics widely used in real-world scenarios to evaluate model performance: **AUC**, **Recall@10%**, **TP@10%**. AUC, or Area Under the Receiver Operating Characteristic (ROC) Curve, serves as a comprehensive measure of the model's discriminative capability across all thresholds.

Recall@10% represents the proportion of actual positive instances that are accurately identified within the top 10% of the ranked predictions. It is worth noting that the top 10% of ranked predictions refers to the top 10% of suspicious users, whose default probability ranks within the top 10% of all users.

Similarly, TP@10% represents the number of True Positive instances within the top 10% of the ranked predictions. In the task of financial risk assessment, positive samples, denoting users who have defaulted, typically constitute a minority. As such, we employ not only the AUC to evaluate the overall discriminative capacity of the models but also adopt Recall@10% and TP@10% as a measure to assess the quality of the highest-ranked predictions.

Experiment Results

Performance Comparison. To answer RQ1, we compare the proposed DGNN-SR with various baselines and their combinations on two real-world datasets from Tencent Mobile Payment. The AUC, Recall@10%, and TP@10% scores are reported in Table 2.

Firstly, compared with the baselines, our proposed models consistently surpass their static and dynamic backbones in terms of performance across the two datasets. In addition, DGNN-SR also outperforms the simple combination of their dynamic and static backbone in the vast majority of cases, which demonstrates the effectiveness of the proposed method.

To provide a more detailed analysis, we compare the performance of the static and dynamic baselines, respectively. Among the static GNNs, GraphSAGE outperforms the other two baselines across most evaluation metrics, while GCN consistently exhibits the weakest performance, which may be due to the fact that GCN neglects important central node features. The performance of GAT is comparable to that of GraphSAGE. So, we choose GraphSAGE as the static backbone in the following experiments. In the context of dynamic GNNs, the three baseline models display varying strengths and weaknesses when applied to the two datasets, with no single model exhibiting a definitive advantage. DyGFormer performs worst on the AUC metric in D1 but outperforms the other two baselines in D2.

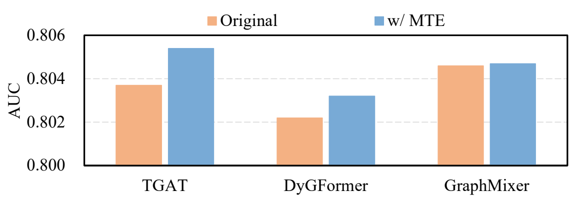
We further combine the dynamic GNNs with static GNNs by employing a simple concatenation operation to fuse dynamic and static embeddings, thereby establishing baseline models for comparison. Based on the obtained results, it can be substantiated that the simple combination of the user-user graph and the user-merchant graph can result in a significant enhancement in performance.

Ablation Study. To answer RQ2 and investigate the effectiveness of the multi-view time encoder, we designed experiments to compare the performance of the original DGNNs with the multi-view time encoder enhanced DGNNs. The results are plotted in Fig.3. For TGAT and DyGFormer, we utilize the trainable time encoder as Eq.(1) and Eq.(2) to encode relative time and absolute time, respectively. For GraphMixer, we follow the design in the original paper (Cong et al. 2023) using a fixed time encoder $\phi(t) = \cos(\boldsymbol{\omega} t)$, which utilizes features $\boldsymbol{\omega} = \{\alpha^{-(i-1)/\beta}\}_{i=1}^{d}$ to en-

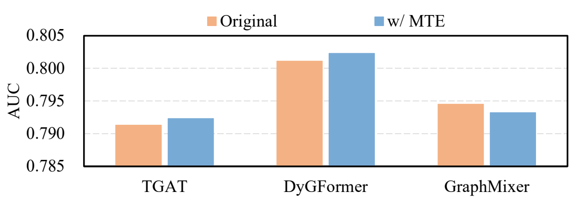
Table 2: Performance comparison on real-world datasets

Type	Model	D1 AUC	D1 Recall@10%	D1 TP@10%	D2 AUC	D2 Recall@10%	D2 TP@10%
Static	GCN	0.7601 ± 0.0009	0.3406 ± 0.0052	1355.2 ± 20.7	0.8135 ± 0.0036	0.4043 ± 0.0056	2100.6 ± 29.0
	GAT	0.7943 ± 0.0010	0.3861 ± 0.0046	1536.2 ± 18.1	0.8459 ± 0.0009	0.4544 ± 0.0027	2361.0 ± 13.8
	GraphSage	0.7970 ± 0.0009	0.3833 ± 0.0023	1525.0 ± 09.1	0.8468 ± 0.0018	0.4572 ± 0.0037	2375.8 ± 19.4
Dynamic	TGAT	0.8037 ± 0.0011	0.3812 ± 0.0027	1516.8 ± 10.8	0.7913 ± 0.0006	0.3706 ± 0.0018	1925.4 ± 09.3
	DyGFormer	0.8022 ± 0.0012	0.3793 ± 0.0034	1509.2 ± 13.6	0.8011 ± 0.0013	0.3781 ± 0.0022	1964.8 ± 11.3
	GraphMixer	0.8046 ± 0.0006	0.3790 ± 0.0021	1508.2 ± 08.4	0.7945 ± 0.0010	0.3716 ± 0.0018	1930.6 ± 09.2
Dynamic w/ static	TGAT w/ s	0.8178 ± 0.0009	0.4044 ± 0.0023	1609.0 ± 09.2	0.8373 ± 0.0004	0.4468 ± 0.0035	2321.8 ± 17.9
	DyGFormer w/ s	0.8185 ± 0.0009	0.4090 ± 0.0048	1627.6 ± 18.9	0.8546 ± 0.0013	0.4687 ± 0.0027	2435.4 ± 14.2
	GraphMixer w/ s	0.8200 ± 0.0006	0.4086 ± 0.0013	1625.8 ± 5.0	0.8536 ± 0.0008	0.4667 ± 0.0030	2425.2 ± 15.4
DGNN-SR	TGAT-SR	0.8184 ± 0.0007	0.4028 ± 0.0017	1602.6 ± 06.7	0.8383 ± 0.0007	0.4507 ± 0.0036	2341.6 ± 18.6
	DyGFormer-SR	**0.8215 ± 0.0010**	**0.4107 ± 0.0024**	**1634.2 ± 09.7**	**0.8553 ± 0.0016**	0.4663 ± 0.0029	2422.6 ± 15.3
	GraphMixer-SR	0.8209 ± 0.0008	0.4093 ± 0.0039	1628.6 ± 15.6	0.8547 ± 0.0017	**0.4694 ± 0.0034**	**2439.0 ± 17.5**

*Dynamic w/ static represents refers to the application of a simple concatenation operation to fuse dynamic and static embeddings. Unless otherwise specified, GraphSAGE is employed as the static backbone in our experiments.



(a) AUC comparison on D1



(b) AUC comparison on D2

Figure 3: AUC of original dynamic GNNs and MTE enhanced dynamic GNNs

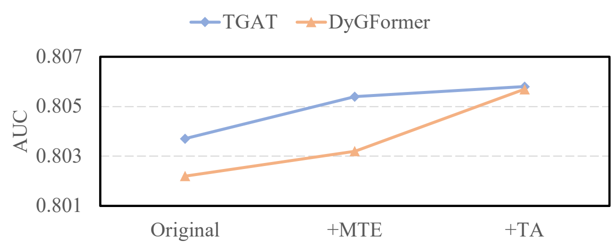


Figure 4: AUC enhancement achieved through the incorporation of Multi-view Time Encoding (MTE) and Temporal Attention (TA) mechanisms.

code related time and absolute time into d-dimensional vectors. α and β are hyperparameters and be set to $\alpha = \beta = 10$.

As shown in Fig.3, GraphMixer with the fixed time encoder shows no performance improvement (in D1) or even performance degradation (in D2). This observation can be attributed to the fact that employing the same fixed time encoder for both relative and absolute times may potentially confuse the model. However, methods that use trainable time encoding, i.e., TGAT and DyGFormer, both benefit from multi-view time encoders on both datasets. This verifies the effectiveness of multi-view time encoders with trainable time encoders.

With regard to temporal attention, the results depicted in Fig.4 illustrate that the temporal attention mechanism further enhances the performance of dynamic GNNs based on the multi-view time encoder. The temporal attention mechanism is inherently compatible with Transformer-based methods such as DyGFormer.

Impact of Combinations and Fusion Strategies. To answer RQ3, we further examine the impact of different combination and fusion strategies. Table 3 presents the AUC for different combinations via concatenation and re-weighting on D1. Recall@10% and TP@10% show a similar trend and are omitted for space reasons.

The results show that GraphSAGE always outperforms the other two static GNNs, no matter which dynamic GNN is combined with it. Interestingly, although GCN exhibits the lowest performance as a standalone model, its effectiveness does not consistently remain inferior to other static GNNs when integrated with dynamic GNNs. Regarding dynamic GNNs, the results indicate that TGAT exhibits the least favorable performance, while DyGFormer demonstrates a level of effectiveness comparable to that of GraphMixer when integrated with static GNNs.

Upon evaluating the diverse strategies, it is evident that the re-weighting approach surpasses concatenation in the vast majority of instances. The sole exception arises in the case of DyGFormer combined with GCN. With this exception, the re-weighting strategy demonstrates remarkable performance enhancement in other DyGFormer-based combinations when compared to the concatenation strategy.

Table 3: AUC comparison of various combinations between concatenation and re-weighting strategies

	GCN			GAT			GraphSage		
	Concatenation	Re-weighting	Imp.	Concatenation	Re-weighting	Imp.	Concatenation	Re-weighting	Imp.
TGAT	0.8070 ± 0.0012	0.8082 ± 0.0012	0.12%	0.8049 ± 0.0011	0.8052 ± 0.0019	0.03%	0.8178 ± 0.0009	0.8181 ± 0.0006	0.03%
DyGFormer	0.8136 ± 0.0012	0.8134 ± 0.0005	-0.02%	0.8176 ± 0.0008	0.8197 ± 0.0009	0.21%	0.8185 ± 0.0009	0.8207 ± 0.0005	0.22%
GraphMixer	0.8136 ± 0.0007	0.8155 ± 0.0008	0.19%	0.8168 ± 0.0013	0.8178 ± 0.0011	0.10%	0.8200 ± 0.0005	0.8201 ± 0.0002	0.01%

Related Work

Dynamic Graph Representation Learning

Dynamic graph representation learning has been extensively explored in recent years (Kazemi et al. 2020; Skarding, Gabrys, and Musial 2021; Xue et al. 2022). Existing methods of dynamic graph learning can be classified into two categories: models based on the discrete-time dynamic graph (DTDG) and models based on the continuous-time dynamic graph (CTDG). DTDG methods treat the dynamic graphs as a sequence of snapshots sampled at regularly-spaced times and fuse information extracted from different snapshots (Goyal, Chhetri, and Canedo 2020; Pareja et al. 2020; Sankar et al. 2020; You, Du, and Leskovec 2022). However, these methods frequently experience information loss due to the inherent limitations of time discretization, which may fail to capture crucial interactions. To address these issues, there has been a growing interest in developing CTDG models that treat dynamic graph data as link streams and directly learn node representations from continuous interactions. Recent researchers have designed CTDG methods based on self-attention mechanisms (Xu et al. 2020; Wang et al. 2021a; Yu et al. 2023), memory networks (Kumar, Zhang, and Leskovec 2019; Trivedi et al. 2019; Rossi et al. 2020; Souza et al. 2022), temporal random walks (Wang et al. 2021b; Jin, Li, and Pan 2022), and sequential models (Wang et al. 2021a; Cong et al. 2023; Yu et al. 2023). Note that the CTDGs are more informative than DTDGs, and CTDG methods are more expressive than DTDG methods (Souza et al. 2022; Gao and Ribeiro 2022).

Existing representation learning methods of CTDG solely take into account relative time information in the temporal dimension, which can result in significant information loss for credit risk assessment. This is primarily due to its incapacity to represent the semantic information inherent in the timestamp. In our work, we employ both relative and absolute time. Moreover, we explicitly exploit the semantic information by encoding periodicity and occurrences of holidays and special events, thereby improving the performance of credit risk assessment.

Credit Risk Assessment

Credit risk assessment is a significant focus in the banking and finance industry, and researchers have explored various machine learning (ML) techniques to address this issue. Individual ML methods, such as Random Forest (RF) (Moscato, Picariello, and Sperlí 2021), Naive Bayes, Logistic Regression, Decision Tree (Aniceto, Barboza, and Kimura 2020), and K-Nearest Neighbor (KNN) Classifier (Wang et al. 2020), XGBoost (Li 2019), Support Vector Machine (SVM) with different kernels (Putri, Fatekurohman, and Tirta 2021), and artificial neural networks with optimization algorithms (Sharifi et al. 2021), have been employed to predict credit risk. These methods have achieved varying levels of accuracy, with some outperforming others. In addition to individual ML techniques, hybrid models combining multiple ML methods have also been proposed for credit risk assessment (Machado and Karray 2022; Lappas and Yannacopoulos 2021; Chi et al. 2019; Kanaparthi 2023). These hybrid models often combine supervised and unsupervised learning techniques, expert knowledge, and various neural network types.

Furthermore, graph representation learning has been employed in various business and management applications, such as anti-spam advertisements, malicious account identification, and fraud detection (Li et al. 2019; Liu et al. 2018, 2020; Li et al. 2022; Huang et al. 2022; Gong et al. 2023; Gong and Sun 2024). Pioneering works have incorporated Graph Neural Networks (GNNs) methods into credit risk assessment and analysis by establishing graph structures based on specific relationships (Cheng et al. 2019; Sukharev et al. 2020; Zhong et al. 2020; Yang et al. 2021; Lee, Lee, and Sohn 2021; Liu et al. 2021; Shi et al. 2024). These approaches have demonstrated improved performance and interpretability compared to traditional classification tools.

Existing works widely adopt static graphs to model the financial interactions between users and merchants, neglecting the temporal information on the interactions. In addition, the financial interactions between users are infrequently incorporated into credit risk assessment models. In our work, we model user-merchant interactions as a continuous dynamic graph and integrate user-user interactions, represented as a static graph, with the dynamic graph to augment the performance and stability of credit risk assessment.

Conclusion

In this paper, we introduced a novel framework for credit risk assessment, DGNN-SR, which seamlessly integrates various dynamic and static backbones. To enhance performance in the CRA task, user-merchant payment interactions are modeled as a continuous-time dynamic graph. The DGNN-SR framework employs a multi-view time encoder to exploit the semantics of both relative and absolute time dimensions. A temporal attention mechanism is incorporated to capture time-related relationships more effectively. Additionally, a dynamic re-weighting strategy is developed to augment dynamic representations with static relations derived from user-user payment interactions. Comprehensive experiments validate the overall effectiveness of the proposed DGNN-SR framework and highlight performance disparities across different backbone structures.

Acknowledgements

The research work is supported by National Key R&D Plan No. 2022YFC3303302, the National Natural Science Foundation of China under Grant No. 62476263, 62406307, U2436209. This work is also supported by the Tencent Rhino-Bird Focused Research Program. Xiang Ao is also supported by Beijing Nova Program 20230484430 and the Innovation Funding of ICT, CAS under Grant No. E461060. Yang Liu is also supported by the China Postdoctoral Science Foundation under Grant No. 2023M743567 and the Postdoctoral Fellowship Program of CPSF under Grant No. GZB20240761. We would like to thank the anonymous reviewers for their valuable comments, and Zhiyu Guo, Yangwu Zhao and Chen Shen for their insightful discussions.

References

Aniceto, M. C.; Barboza, F.; and Kimura, H. 2020. Machine learning predictivity applied to consumer creditworthiness. *Future Business Journal*, 6(1): 37.

Chen, N.; Ribeiro, B.; and Chen, A. 2016. Financial credit risk assessment: a recent review. *Artificial Intelligence Review*, 45: 1–23.

Cheng, D.; Tu, Y.; Ma, Z.-W.; Niu, Z.; and Zhang, L. 2019. Risk Assessment for Networked-guarantee Loans Using High-order Graph Attention Representation. In *IJCAI*, 5822–5828.

Chi, G.; Uddin, M. S.; Abedin, M. Z.; and Yuan, K. 2019. Hybrid model for credit risk prediction: An application of neural network approaches. *International Journal on Artificial Intelligence Tools*, 28(05): 1950017.

Cong, W.; Zhang, S.; Kang, J.; Yuan, B.; Wu, H.; Zhou, X.; Tong, H.; and Mahdavi, M. 2023. Do We Really Need Complicated Model Architectures For Temporal Networks? In *International Conference on Learning Representations (ICLR)*.

Crook, J. N.; Edelman, D. B.; and Thomas, L. C. 2007. Recent developments in consumer credit risk assessment. *European Journal of Operational Research*, 183(3): 1447–1465.

Gao, J.; and Ribeiro, B. 2022. On the equivalence between temporal and static equivariant graph representations. In *International Conference on Machine Learning*, 7052–7076. PMLR.

Gong, Z.; and Sun, Y. 2024. An Energy-centric Framework for Category-free Out-of-distribution Node Detection in Graphs. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 908–919.

Gong, Z.; Wang, G.; Sun, Y.; Liu, Q.; Ning, Y.; Xiong, H.; and Peng, J. 2023. Beyond Homophily: Robust Graph Anomaly Detection via Neural Sparsification. In *IJCAI*, 2104–2113.

Goyal, P.; Chhetri, S. R.; and Canedo, A. 2020. dyngraph2vec: Capturing network dynamics using dynamic graph representation learning. *Knowledge-Based Systems*, 187: 104816.

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

Huang, M.; Liu, Y.; Ao, X.; Li, K.; Chi, J.; Feng, J.; Yang, H.; and He, Q. 2022. Auc-oriented graph neural network for fraud detection. In *Proceedings of the ACM web conference 2022*, 1311–1321.

Jin, M.; Li, Y.-F.; and Pan, S. 2022. Neural temporal walks: Motif-aware representation learning on continuous-time dynamic graphs. *Advances in Neural Information Processing Systems*, 35: 19874–19886.

Kanaparthi, V. 2023. Credit Risk Prediction using Ensemble Machine Learning Algorithms. In *2023 International Conference on Inventive Computation Technologies (ICICT)*, 41–47. IEEE.

Kazemi, S. M.; Goel, R.; Jain, K.; Kobyzev, I.; Sethi, A.; Forsyth, P.; and Poupart, P. 2020. Representation learning for dynamic graphs: A survey. *Journal of Machine Learning Research*, 21(70): 1–73.

Kipf, T. N.; and Welling, M. 2016. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*.

Kumar, S.; Zhang, X.; and Leskovec, J. 2019. Predicting dynamic embedding trajectory in temporal interaction networks. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, 1269–1278.

Lappas, P. Z.; and Yannacopoulos, A. N. 2021. A machine learning approach combining expert knowledge with genetic algorithms in feature selection for credit risk assessment. *Applied Soft Computing*, 107: 107391.

Lee, J. W.; Lee, W. K.; and Sohn, S. Y. 2021. Graph convolutional network-based credit default prediction utilizing three types of virtual distances among borrowers. *Expert Systems with Applications*, 168: 114411.

Li, A.; Qin, Z.; Liu, R.; Yang, Y.; and Li, D. 2019. Spam review detection with graph convolutional networks. In *Proceedings of the 28th ACM international conference on information and knowledge management*, 2703–2711.

Li, K.; Liu, Y.; Ao, X.; Chi, J.; Feng, J.; Yang, H.; and He, Q. 2022. Reliable representations make a stronger defender: Unsupervised structure refinement for robust gnn. In *Proceedings of the 28th ACM SIGKDD conference on knowledge discovery and data mining*, 925–935.

Li, Y. 2019. Credit risk prediction based on machine learning methods. In *2019 14th international conference on computer science & education (ICCSE)*, 1011–1013. IEEE.

Liu, Y.; Ao, X.; Qin, Z.; Chi, J.; Feng, J.; Yang, H.; and He, Q. 2021. Pick and choose: a GNN-based imbalanced learning approach for fraud detection. In *Proceedings of the web conference 2021*, 3168–3177.

Liu, Z.; Chen, C.; Yang, X.; Zhou, J.; Li, X.; and Song, L. 2018. Heterogeneous graph neural networks for malicious account detection. In *Proceedings of the 27th ACM international conference on information and knowledge management*, 2077–2085.

Liu, Z.; Dou, Y.; Yu, P. S.; Deng, Y.; and Peng, H. 2020. Alleviating the inconsistency problem of applying graph neural network to fraud detection. In *Proceedings of the 43rd international ACM SIGIR conference on research and development in information retrieval*, 1569–1572.

Machado, M. R.; and Karray, S. 2022. Assessing credit risk of commercial customers using hybrid machine learning algorithms. *Expert Systems with Applications*, 200: 116889.

Moscato, V.; Picariello, A.; and Sperlí, G. 2021. A benchmark of machine learning approaches for credit score prediction. *Expert Systems with Applications*, 165: 113986.

Pareja, A.; Domeniconi, G.; Chen, J.; Ma, T.; Suzumura, T.; Kanezashi, H.; Kaler, T.; Schardl, T.; and Leiserson, C. 2020. Evolvegcn: Evolving graph convolutional networks for dynamic graphs. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, 5363–5370.

Putri, N.; Fatekurohman, M.; and Tirta, I. 2021. Credit risk analysis using support vector machines algorithm. In *Journal of Physics: Conference Series*, volume 1836, 012039. IOP Publishing.

Rosin, G. D.; and Radinsky, K. 2022. Temporal attention for language models. *arXiv preprint arXiv:2202.02093*.

Rossi, E.; Chamberlain, B.; Frasca, F.; Eynard, D.; Monti, F.; and Bronstein, M. 2020. Temporal graph networks for deep learning on dynamic graphs. *arXiv preprint arXiv:2006.10637*.

Sankar, A.; Wu, Y.; Gou, L.; Zhang, W.; and Yang, H. 2020. Dysat: Deep neural representation learning on dynamic graphs via self-attention networks. In *Proceedings of the 13th international conference on web search and data mining*, 519–527.

Sharifi, P.; Jain, V.; Arab Poshtkohi, M.; Seyyedi, E.; and Aghapour, V. 2021. Banks credit risk prediction with optimized ANN based on improved owl search algorithm. *Mathematical Problems in Engineering*, 2021(1): 8458501.

Shi, Y.; Qu, Y.; Chen, Z.; Mi, Y.; and Wang, Y. 2024. Improved credit risk prediction based on an integrated graph representation learning approach with graph transformation. *European Journal of Operational Research*, 315(2): 786–801.

Skarding, J.; Gabrys, B.; and Musial, K. 2021. Foundations and modeling of dynamic networks using dynamic graph neural networks: A survey. *iEEE Access*, 9: 79143–79168.

Souza, A.; Mesquita, D.; Kaski, S.; and Garg, V. 2022. Provably expressive temporal graph networks. *Advances in neural information processing systems*, 35: 32257–32269.

Sukharev, I.; Shumovskaia, V.; Fedyanin, K.; Panov, M.; and Berestnev, D. 2020. Ews-gcn: Edge weight-shared graph convolutional network for transactional banking data. In *2020 IEEE International Conference on Data Mining (ICDM)*, 1268–1273. IEEE.

Trivedi, R.; Farajtabar, M.; Biswal, P.; and Zha, H. 2019. Dyrep: Learning representations over dynamic graphs. In *International conference on learning representations*.

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

Wang, X.; Lyu, D.; Li, M.; Xia, Y.; Yang, Q.; Wang, X.; Wang, X.; Cui, P.; Yang, Y.; Sun, B.; et al. 2021a. Apan: Asynchronous propagation attention network for real-time temporal graph embedding. In *Proceedings of the 2021 international conference on management of data*, 2628–2638.

Wang, Y.; Chang, Y.-Y.; Liu, Y.; Leskovec, J.; and Li, P. 2021b. Inductive Representation Learning in Temporal Networks via Causal Anonymous Walks. In *International Conference on Learning Representations (ICLR)*.

Wang, Y.; Zhang, Y.; Lu, Y.; and Yu, X. 2020. A Comparative Assessment of Credit Risk Model Based on Machine Learning——a case study of bank loan data. *Procedia Computer Science*, 174: 141–149.

Xu, D.; Ruan, C.; Korpeoglu, E.; Kumar, S.; and Achan, K. 2020. Inductive representation learning on temporal graphs. In *International Conference on Learning Representations (ICLR)*.

Xue, G.; Zhong, M.; Li, J.; Chen, J.; Zhai, C.; and Kong, R. 2022. Dynamic network embedding survey. *Neurocomputing*, 472: 212–223.

Yang, S.; Zhang, Z.; Zhou, J.; Wang, Y.; Sun, W.; Zhong, X.; Fang, Y.; Yu, Q.; and Qi, Y. 2021. Financial risk analysis for SMEs with graph-based supply chain mining. In *Proceedings of the Twenty-Ninth International Conference on International Joint Conferences on Artificial Intelligence*, 4661–4667.

You, J.; Du, T.; and Leskovec, J. 2022. ROLAND: graph learning framework for dynamic graphs. In *Proceedings of the 28th ACM SIGKDD conference on knowledge discovery and data mining*, 2358–2366.

Yu, L.; Sun, L.; Du, B.; and Lv, W. 2023. Towards better dynamic graph learning: New architecture and unified library. *Advances in Neural Information Processing Systems*, 36: 67686–67700.

Zhong, Q.; Liu, Y.; Ao, X.; Hu, B.; Feng, J.; Tang, J.; and He, Q. 2020. Financial defaulter detection on online credit payment via multi-view attributed heterogeneous information network. In *Proceedings of the web conference 2020*, 785–795.

Zhu, X.; Ao, X.; Qin, Z.; Chang, Y.; Liu, Y.; He, Q.; and Li, J. 2021. Intelligent financial fraud detection practices in post-pandemic era. *The Innovation*, 2(4).