

A Dynamic Credit Assessment Algorithm For Cross-Border E-Commerce Based On An Improved Temporal Graph Convolutional Network

Huan Li^{1*} and Zhiyang Zhang²

¹ School of Digital Bussiness, Zhejiang Business College, Hangzhou 310053, China

² School of Ocean Engineering, Jiangsu Ocean University, Lianyungang 222000, China

* Corresponding author. E-mail: 00791@zjbc.edu.cn

Received: July. 14, 2026; Accepted: Aug. 31, 2026

Cross-border e-commerce requires dynamic, heterogeneous credit evaluation, but static models fail to capture temporal and network interactions. This paper proposes an Improved Temporal Graph Convolutional Network (ITGCN) for dynamic credit assessment. ITGCN constructs a dynamic heterogeneous graph integrating sellers, buyers, logistics, and payment nodes, with a relationship-aware graph convolution operator and a temporal decay attention mechanism to model heterogeneous dependencies. A multi-scale time series module captures both short-term fluctuations and long-term credit trends, while a cross-domain feature alignment strategy handles inconsistencies in multi-source data granularities. The experimental results on two public fraud detection datasets show that the ITGCN achieves an AUC of 0.938 (D1) and 0.934 (D2), outperforming CARE-GNN by 1.8 per cent and 2.1 per cent, and an F1 score of 0.876 and 0.871, representing improvements of 2.2 per cent and 2.6 per cent. Ablation and robustness tests confirm its effectiveness and resilience. ITGCN offers a reliable solution for cross-border credit assessment.

Keywords: time series graph convolutional network; dynamic credit evaluation; cross-border e-commerce; heterogeneous graph; multi-scale time series modeling

© The Author(s). This is an open-access article distributed under the terms of the [Creative Commons Attribution License \(CC BY 4.0\)](https://creativecommons.org/licenses/by/4.0/), which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are cited.

http://dx.doi.org/10.6180/jase.202701_36.033

1. Introduction

Cross-border e-commerce, a major digital trade form, has grown rapidly, China's 2025 imports and exports reached 2.84 trillion yuan. As a result, credit risk has increased significantly [1], and extended logistics and varied settlement periods have led to fraud, default and false logistics [2–4]. A problem is how to determine a person's credit risk at the same time.

Credit assessment anticipates the future operating capability of borrowers and partners. Early statistical models, such as logistic regression and linear discriminant analysis [5]. Although interpretable, they assume linear relationships among features and sample independence, and therefore fail to capture the risk propagation among related entities [6]. Later, non-parametric Gradient Boosting Deci-

sion Tree (GBDT) [7] significantly improves the prediction accuracy, but it still uses static cross-sectional features and fails to consider the temporal changes in credit status or interactive information among related parties that could enhance prediction.

Graph Neural Networks (GNNs) have introduced a new approach to credit assessment by learning node representations through message passing and effectively incorporating neighbour information and complex dependencies [8]. Entities and relations in credit scenarios form graphs, and the task is to perform node classification or link prediction [9]. Early works have adopted static architectures such as GCN and GAT [10, 11]; for example, Wang et al. [12] used GCN on merchant graphs for corporate risk, and Shao et al. [13] proposed GNN-EADD for e-commerce anomaly

detection.

In order to capture temporal dynamics, researchers have combined Recurrent Neural Network (RNN) or temporal convolution with GNN to propose a series of Temporal Graph Neural Network models. Among them, TGCN first uses GCN to extract spatial dependencies, and then captures temporal evolution through a Gated Recurrent Unit [14]. Wu et al. [15] proposed Graph WaveNet combined with an adaptive adjacency matrix and dilated convolution. Feng et al. [16] used the spatio-temporal attention mechanism to achieve long-term dependent explicit modeling. In the field of financial credit evaluation, Temporal-aware Graph Neural Network jointly models user credit dynamics through three parts: static graph coding, short-term graph coding, and a long-term timing module based on Long Short-Term Memory [17]. Temporal Relational Graph Convolutional Networks integrates financial knowledge graphs and uses temporal relational reasoning for risk prediction [18].

Although TGNNs have shown good results in dynamic credit risk assessment, the current models have three main deficiencies: they use discrete-time snapshots and thus lose detailed information on continuous changes; they do not adequately address the heterogeneity of multiple node types and relationships in cross-border data; and they lack an effective cross-domain temporal alignment mechanism for data sources with different time resolutions.

In view of the above shortcomings, this paper proposes an ITGCN, which is specially designed for cross-border e-commerce dynamic credit evaluation tasks. The first is a unified framework that combines four techniques for cross-border e-commerce credit assessment, along with customised decay-rate and credit-history mechanisms.

2. Theoretical basis

2.1. Dynamic heterogeneous graph

In the cross-border e-commerce scene, the participants include sellers, buyers, logistics providers, and payment intermediaries. We define a dynamic heterogeneous graph $G^{(t)} = (v, \varepsilon^{(t)}, x^{(t)})$, where v is the set of nodes; $\varepsilon^{(t)}$ is the set of edges at time t , and the types of edges include transaction, payment, logistics, and evaluation. $x^{(t)} = \{x_v^{(t)} \in R^{d_v} \mid v \in V\}$ is the node feature matrix, and the feature dimension d_v of different node types can be different [19].

2.2. Time series graph convolution network

TGCN combines graph convolution with gated recurrent unit. For each time step t , the graph convolution layer spatially aggregates the current graph structure $G^{(t)}$ and

node feature $x^{(t)}$ to obtain the hidden state $H^{(t)}$ [20]. The propagation rules of the single-layer GCN are shown in formula (1):

$$H^{(t)} = \sigma \left(\tilde{D}^{-\frac{1}{2}} \tilde{A} \tilde{D}^{-\frac{1}{2}} X^{(t)} W \right) \quad (1)$$

In the formula, $\tilde{A} = A + I$ is the adjacency matrix added to the self-ring, and I is the unit matrix; $\tilde{D}$ is the degree matrix; W is a learnable weight matrix; σ is a nonlinear activation function. Subsequently, the GRU uses $H^{(t)}$ as input to update the timing hidden state $S^{(t)}$, as shown in Formula (2):

$$S^{(t)} = \text{GRU} \left(S^{(t-1)}, H^{(t)} \right) \quad (2)$$

2.3. Problem definition

Given the dynamic heterogeneous graph sequence $\{G^{(1)}, G^{(2)}, \dots, G^{(T)}\}$ in the historical T time steps and the historical characteristics and edge information of each node, the goal of this paper is to predict the credit default probability $y_v \in [0, 1]$ of all seller nodes at $T + 1$ in the future. In the public data set used in this paper, there is no direct labeling of credit default by the seller; therefore, we define credit default operationalization as the seller is marked as at least one fraudulent transaction in the prediction window. Credit default is defined as a seller's bad behavior in the forecast window, such as overdue delivery, poor quality, or false transactions [21]. At the experimental level, transaction-level fraud labels are used as observable proxy indicators. This challenge is structured as a temporal binary categorization problem, as illustrated by Equation (3):

$$\hat{y}_v^{(T+1)} = f_{\Theta} \left(\left\{ G^{(t)}, x^{(t)} \right\}_{t=1}^T, v \right) \quad (3)$$

In the formula, f_{Θ} is the mapping function to be learned with parameter Θ .

2.4. Improved temporal graph convolution network model

2.4.1. General framework

The four modules in the ITGCN framework are: (a) dynamic heterogeneous graph construction; (b) improved temporal graph convolution with relationship-aware propagation and temporal decay attention; (c) multi-scale temporal feature extraction and cross-domain alignment; and (d) credit prediction. The Framework is trained end-to-endly (Fig. 1).

Note: The concrete node types, edge types, and feature domains vary across datasets.

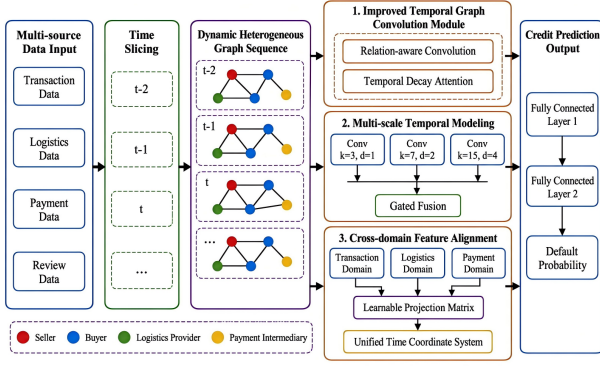


Fig. 1. ITGCN overall framework.

The above framework assumes that transaction records are aggregated into daily graph snapshots, transaction-level fraud labels can serve as seller-level credit default proxies, and baseline GNN models are evaluated on comparable graph representations.

2.4.2. Dynamic heterogeneous graph construction

For each time window t , we convert the records in the original data into a heterogeneous graph. The specific steps include: Node extraction; Feature encoding; Edge construction; Time slice [22].

2.5. Improved Temporal Graph Convolution Module

2.5.1. Relationship-aware graph convolution

The main difference is that our RGC acts on an evolving graph sequence and is jointly optimized with temporal components. To this end, a Relation-aware Graph Convolution (RGC) operator is designed [23]. For the node v of type, its propagation rule in the l th layer is Formula (4):

$$h_v^{(l+1)} = \text{ReLU} \left(\sum_{r \in R} \sum_{u \in N_r(v)} \frac{1}{c_{v,u}} W_r^{(l)} h_u^{(l)} + W_{\varphi(v)}^{(l)} h_v^{(l)} \right) \quad (4)$$

In the formula, R is a set of edge types; $N_r(v)$ represents the set of neighbor nodes connected to v through type r ; $c_{v,u} = |N_r(v)|$ is the degree of node v within relation r , serving as the normalization constant; $W_r^{(l)}$ is a learnable weight matrix exclusive to the edge type r ; $W_{\varphi(v)}^{(l)}$ is the self-loop transformation matrix of the node type $\varphi(v)$; $h_v^{(l)}$ represents the hidden vector of the l th layer node v ; r index edge type. Let $H^{(t,l)}$ is the node representation matrix of the l th layer of the t th time step, then the relation-aware convolution can be expressed as Formula (5):

$$H^{(t,l+1)} = \text{ReLU} \left(\sum_{r \in R} \tilde{A}_r H^{(t,l)} W_r^{(l)} + H^{(t,l)} W_{\text{self}}^{(l)} \right) \quad (5)$$

In the formula, $\tilde{A}_r$ is the normalized adjacency matrix of the relation r ; $W_{\text{self}}^{(l)}$ is the self-loop weight matrix, which is set to the mean value of each $W_r^{(l)}$.

2.5.2. Temporal Decay Attention

We introduce a temporal decaying attention mechanism to perform a weighted regrouping before inputting the convolution output $H^{(t)}$ of each moment graph into the GRU [24]. For the current prediction time T and the historical time t , the time decay factor $\delta_{T,t} = \exp(-\beta \cdot (T - t))$ is defined, where $\beta = \text{softplus}(b)$ with learnable b (initialized so that $\beta = 0.5$), ensuring $\beta > 0$ throughout training. Then use Formula (6) to calculate the attention weight:

$$\alpha_{T,t} = \frac{\exp(\delta_{T,t} \cdot \text{score}(H^{(T)}, H^{(t)}))}{\sum_{s=1}^T \exp(\delta_{T,s} \cdot \text{score}(H^{(T)}, H^{(s)}))} \quad (6)$$

In the formula, a is a learnable vector. The weighted historical representation $\tilde{H}^{(t)} = \alpha_{T,t} H^{(t)}$ replaces the original $H^{(t)}$ input GRU.

2.6. Multi-scale time series modeling and cross-domain feature alignment

2.6.1. Multi-scale temporal feature extraction

Cross-border e-commerce credit is driven by short-term emergencies and long-term trends. In order to capture both at the same time, a multi-scale temporal convolution module is designed [25]. After GRU, the temporal hidden state sequence $S^{(1:T)} \in R^{T \times d_h}$ of each node is input into three parallel one-dimensional convolution layers respectively. The convolution kernel sizes are $k_1 = 3, k_2 = 7$, and $k_3 = 15$, respectively, and the corresponding expansion rates are set to 1, 2, and 4, respectively, so that the receptive field covers 3, 13, and 57 time steps, respectively. The outputs of the three convolutional layers are $S_{\text{short}}^{(T)}$, and $S_{\text{long}}^{(T)}$, all of which are d -dimensional vectors. Then adaptive fusion is performed through the gated fusion unit, as shown in Formula (7):

$$g_s = \text{sigmoid} \left(U_s S_{\text{short}}^{(T)} + U_m S_{\text{mid}}^{(T)} + U_l S_{\text{long}}^{(T)} + b_g \right) \\ S_{\text{fused}} = \sum_{s \in \{\text{short, mid, long}\}} g_s \odot S_s^{(T)} \quad (7)$$

In the formula, U_s, U_m, U_l are a learnable gated weight matrix; b_g is the bias vector; $\odot$ denotes the product of elements.

2.6.2. Cross-domain feature alignment

The cross-domain alignment module is designed to project the features of each domain to a unified time coordinate system [26]. Suppose that for a node v , there is a characteristic

Table 1. ITGCN hyperparameter settings

Category	Hyperparameter	Value	Category	Hyperparameter	Value
Graph Convolution	Number of GCN layers	2	Multi-scale Convolution	Kernel sizes	$k_1 = 3, k_2 = 7, k_3 = 15$
	Hidden dimension	128		Dilation rates	$d_1 = 1, d_2 = 2, d_3 = 4$
	Number of edge types	$\mathbf{R}$ (Determined by the dataset, D1:4, D2:1)		Input / Output channels	$d_s = 256$
	Weight matrices	$\mathbf{R}^{128 \times 128}$, independent		Padding mode	same padding
GRU	Self-loop weight	Mean of all $W_r^{(l)}$	Training	Batch size	64
	Hidden dimension	256		Initial learning rate	$10^{-3} \times 0.9$
	Number of layers	1		Learning rate decay	every 20 epochs
Temporal Decay	Decay rate	Initialized as 0.5		Adam parameters	$\beta_1 = 0.9, \beta_2 = 0.999, \epsilon = 10^{-8}$
Attention	LeakyReLU negative slope	0.2	Loss	Early stopping patience	50 epochs
Cross-domain Alignment	Number of domains	Configured by dataset (D1: 2 domains, D2: 1 domains)		L2 regularization coefficient	10^{-5}
	Number of time interval bins	5		Positive class weight	D1: 25.0, D2: 49.0
	Projection matrix	$\mathbf{R}^{128 \times d_s}$	Environment	Random seed	42

$z_{v,d}^{(t)}$ from domain d at time t . A set of time-aware projection matrices $P_{d,\tau}$ is learned for each domain d , where τ is the time interval from the last update. The time intervals are binned at $\tau \in \{0, [1, 3), [3, 7), [7, 15), [15, \infty)\}$ days, with bin boundaries chosen to capture short-, medium-, and long-term gaps. The projection matrices $P_{d,\tau}$ are shared across nodes within the same domain and time bin. After projection, the alignment feature $\tilde{z}_{v,d}^{(t)} = P_{d,\tau(v,d,t)} z_{v,d}^{(t)}$ is obtained. Then the attention mechanism is used to aggregate the features of each domain, as shown in Formula (8):

$$z_{v,\text{align}}^{(t)} = \sum_{d \in D} \frac{\exp(q^T \tilde{z}_{v,d}^{(t)})}{\sum_{d \in D} \exp(q^T \tilde{z}_{v,d}^{(t)})} \tilde{z}_{v,d}^{(t)} \quad (8)$$

In the formula, q is a learning query vector. Finally, $z_{v,\text{align}}^{(t)}$ and S_{fused} are stitched together as a comprehensive representation of node v at time t .

2.7. Credit prediction and model training

The composite representation is input into a two-layer fully connected network to obtain the probability of default, as shown in formula (9):

$$\hat{y}_v^{(T+1)} = \text{sigmoid} \left(W_2 \cdot \text{ReLU} \left(W_1 \left[S_{\text{fused}}^{(v)} \parallel z_{v,\text{align}}^{(T)} \right] + b_1 \right) + b_2 \right) \quad (9)$$

In the formula, W_1, W_2, b_1, b_2 are trainable parameters. The training loss uses a weighted two-class cross entropy, as shown in Formula (10):

$$L = -\frac{1}{N} \sum_{v=1}^N [\lambda y_v \log \hat{y}_v + (1 - y_v) \log (1 - \hat{y}_v)] \quad (10)$$

In the formula, N is the total number of seller nodes in the training set; y_v is the true default label; λ is computed on the original training set before undersampling, reflecting the natural class distribution.

2.8. Experimental settings

2.8.1. Data set introduction

Dataset D1 (<https://www.kaggle.com/datasets/dylanxie1/cross-border-transaction-fraud-detection-dataset>) is a public cross-border transaction fraud detection dataset with transaction, user, device/IP, and fraud information. Dataset D2 (<https://www.kaggle.com/datasets/umuttuygurr/e-commerce-fraud-detection-dataset>) is an e-commerce fraud detection dataset with global transaction records and fraud labels. Both datasets are simulated public datasets used as benchmarks because real cross-border credit data is unavailable.

Transaction-level fraud labels must be aggregated to the seller level for credit default prediction, and each seller serves as the base unit. For a seller v , if any transaction in the forecast window W is identified as fraudulent, the default label y_v is set to 1; otherwise, it is 0. Aggregation is ordered by timestamp and will not contain future data. The data are chronologically divided into training, validation and test sets (70%, 10%, 20%), and label aggregation within each split is limited to the historical transactions of that time period.

2.8.2. Data preprocessing

Data preprocessing steps: (1) Missing values: numerical features with $< 30\%$ missing were median-filled; $> 30\%$ were deleted; missing categorical values formed a new category. (2) Outliers: Z-score truncation at the 1% and 99% quantiles. (3) Feature engineering: rolling-window user statistics; seller complaint rate, average transaction interval, and late-night transaction ratio. (4) Time alignment: transaction timestamps discretized into equal time windows. (5) Graph construction: heterogeneous graphs built by time-window slices; isolated or all-zero nodes removed. (6) Label balance: fraud rates were 3%-5% (D1) and 2% (D2); negative instances were undersampled to below 1:5 for training only, while λ used the original distribution to avoid majority-class bias.

Table 2. Performance comparison of each model on D1 and D2 datasets

Model	D1-AUC	D1-F1	D1-Recall	D1-Accuracy	D2-AUC	D2-F1	D2-Recall	D2-Accuracy	p (D1)	p (D2)
XGBoost	0.808 ± 0.005	0.735 ± 0.006	0.704 ± 0.008	0.781 ± 0.004	0.802 ± 0.006	0.728 ± 0.007	0.695 ± 0.009	0.775 ± 0.005	< 0.001**	< 0.001**
LSTM	0.839 ± 0.004	0.763 ± 0.005	0.737 ± 0.007	0.814 ± 0.004	0.835 ± 0.005	0.756 ± 0.006	0.731 ± 0.008	0.810 ± 0.005	< 0.001**	< 0.001**
GCN	0.858 ± 0.004	0.780 ± 0.005	0.754 ± 0.006	0.833 ± 0.003	0.853 ± 0.005	0.773 ± 0.006	0.746 ± 0.007	0.829 ± 0.004	< 0.001**	< 0.001**
TGCN	0.876 ± 0.003	0.794 ± 0.004	0.770 ± 0.005	0.852 ± 0.003	0.871 ± 0.004	0.789 ± 0.005	0.764 ± 0.006	0.846 ± 0.004	< 0.001**	< 0.001**
STGCN	0.891 ± 0.003	0.811 ± 0.004	0.788 ± 0.005	0.868 ± 0.003	0.886 ± 0.004	0.805 ± 0.005	0.781 ± 0.006	0.862 ± 0.003	< 0.001**	< 0.001**
GraphWaveNet	0.915 ± 0.003	0.845 ± 0.004	0.823 ± 0.005	0.893 ± 0.003	0.909 ± 0.004	0.836 ± 0.005	0.812 ± 0.006	0.887 ± 0.003	0.008*	0.012*
TemGNN	0.905 ± 0.004	0.831 ± 0.005	0.826 ± 0.006	0.882 ± 0.004	0.899 ± 0.005	0.823 ± 0.006	0.818 ± 0.007	0.876 ± 0.004	< 0.001**	< 0.001**
TRGCN	0.912 ± 0.003	0.838 ± 0.004	0.822 ± 0.005	0.890 ± 0.003	0.912 ± 0.004	0.844 ± 0.005	0.827 ± 0.006	0.891 ± 0.003	0.004*	0.006*
R-GCN	0.902 ± 0.004	0.824 ± 0.005	0.808 ± 0.006	0.880 ± 0.004	0.897 ± 0.005	0.817 ± 0.006	0.801 ± 0.007	0.873 ± 0.004	< 0.001**	< 0.001**
HGT	0.918 ± 0.003	0.848 ± 0.004	0.830 ± 0.005	0.896 ± 0.003	0.913 ± 0.004	0.841 ± 0.005	0.823 ± 0.006	0.890 ± 0.003	0.006*	0.008*
CARE-GNN	0.921 ± 0.004	0.857 ± 0.005	0.851 ± 0.006	0.901 ± 0.004	0.915 ± 0.005	0.849 ± 0.006	0.842 ± 0.007	0.893 ± 0.004	0.010*	0.014*
ITGCN	0.938 ± 0.003	0.876 ± 0.004	0.864 ± 0.005	0.920 ± 0.003	0.934 ± 0.004	0.871 ± 0.005	0.858 ± 0.006	0.916 ± 0.003	-	-

Note: ** $p < 0.001$; * $p < 0.05$.

2.8.3. Baseline model and evaluation index

In order to verify the effectiveness of ITGCN, the following baseline models are selected for comparison: XGBoost [27], LSTM [28], GCN [10], TGCN [28], STGCN [29], GraphWaveNet [16], TemGNN [17], TRGCN [18], R-GCN [30], HGT [31], CARE-GNN [32].

The evaluation indexes were PR-AUC, AUC, F1-score, Recall, and Accuracy. All indicators are calculated on the test set, and each experiment is repeated 5 times to take the mean.

2.8.4. Super parameter settings

The hyperparameters of ITGCN are set as shown in Table 1. Every hyperparameter is identified via grid search across the validation data, and the exploration scope follows previous studies. All experiments are implemented under the framework of PyTorch 1.12 and PyTorch Geometric 2.2, using NVIDIA A100 GPU (40GB video memory).

3. Experimental results and analysis

3.1. Overall Performance Comparison

As shown in Table 2, ITGCN outperforms all the baselines on both datasets. On D1, it achieves AUC 0.938 and F1 0.876; on D2, AUC 0.934 and F1 0.871. Non-graph models are significantly worse and do not perform well. Dynamic models outperform static GCN, and those that incorporate relational/knowledge graph modeling are better, but still lower than ITGCN. D1 show stronger fraud detection ability.

In addition, in terms of PR-AUC, ITGCN achieves 0.612 on D1 and 0.586 on D2, outperforming the strongest baseline GraphWaveNet by 8.7% and 7.3%. For Precision at 1% Recall, ITGCN reaches 0.385 on D1 and 0.362 on D2, compared with 0.347 and 0.334 for GraphWaveNet, representing improvements of 10.9% and 8.4%. These results further confirm that ITGCN's advantages are not solely driven by ROC-AUC but hold under more stringent metrics for imbalanced classification.

Graph-based models significantly outperform non-graph baselines, and relational information is required for this improvement. ITGCN has outperformed both homogeneous and heterogeneous baselines consistently, and combines the advantages of relationship-aware propagation with temporal modelling. Quantitatively, it has improved the AUC by 2.4%-2.5% over the best baseline and is statistically and practically significant.

3.2. Ablation experiment

In order to verify the contribution of each improved module of ITGCN, the following variants are designed: (1) ITGCN w/o RGC; (2) ITGCN w/o TDA; (3) ITGCN w/o Multi-scale; (4) ITGCN w/o Align; (5) ITGCN w/o all; (6) ITGCN w/o RGC & Multi-scale.

The ablation experiments in Table 3 show that the removal of RGC reduces AUC and F1 by 1.8% and 2.5%, respectively, confirming that relational awareness is crucial to the semantic distinction of heterogeneous graphs. Removing TDA led to a decrease in AUC by 1.2% and a decrease in F1 by 1.6%, indicating that time-attenuated attention can use recent information more accurately; the removal of multi-scale modules had the largest decrease, highlighting the key role of capturing short-term and long-term models. The removal of the alignment module reduces AUC and F1 by 1.4% and 2.0%, indicating that cross-domain feature alignment alleviates the problem of time granularity inconsistency. The variant without all modules drops to AUC 0.876 and F1 0.794, indicating that each module contributes independently. The multi-scale module has the largest increase and is required to capture both short-term and long-term features.

3.3. Timing dynamic analysis

In order to evaluate the performance of the model under different prediction time windows, we set the prediction windows to 3 days, 7 days, 14 days, and 30 days, respectively, and tested the AUC and F1 of each model under these four windows. The results are shown in Table 4. We

Table 3. Ablation experimental results on the D1 dataset

Model Variant	AUC	F1	Recall	Accuracy
ITGCN w/o RGC	0.921 ± 0.003	0.854 ± 0.004	0.841 ± 0.005	0.899 ± 0.003
ITGCN w/o TDA	0.927 ± 0.003	0.862 ± 0.004	0.850 ± 0.005	0.906 ± 0.003
ITGCN w/o Multi-scale	0.918 ± 0.004	0.851 ± 0.005	0.837 ± 0.006	0.897 ± 0.004
ITGCN w/o Align	0.925 ± 0.003	0.858 ± 0.004	0.846 ± 0.005	0.903 ± 0.003
ITGCN w/o RGC & Multi-scale	0.905 ± 0.004	0.836 ± 0.005	0.823 ± 0.006	0.883 ± 0.004
ITGCN w/o all (TGCN)	0.876 ± 0.003	0.794 ± 0.004	0.770 ± 0.005	0.852 ± 0.003
Complete ITGCN	0.938 ± 0.003	0.876 ± 0.004	0.864 ± 0.005	0.920 ± 0.003

Table 4. Comparison of model performance under different prediction windows

Model	Prediction Window (days)	AUC	F1	Positive Rate (Test Set) (%)
ITGCN	3	0.961 ± 0.003	0.905 ± 0.004	5.08
	7	0.952 ± 0.003	0.892 ± 0.004	8.36
	14	0.938 ± 0.003	0.876 ± 0.004	13.36
	30	0.912 ± 0.004	0.847 ± 0.005	20.90
GraphWaveNet	3	0.931 ± 0.004	0.858 ± 0.005	5.08
	7	0.921 ± 0.004	0.847 ± 0.005	8.36
	14	0.912 ± 0.004	0.838 ± 0.005	13.36
	30	0.864 ± 0.005	0.789 ± 0.006	20.90
TGCN	3	0.934 ± 0.003	0.864 ± 0.004	5.08
	7	0.926 ± 0.003	0.854 ± 0.004	8.36
	14	0.915 ± 0.003	0.845 ± 0.004	13.36
	30	0.873 ± 0.004	0.801 ± 0.005	20.90

Table 5. AUC of each model under different feature missing ratios

Feature Missing Ratio	ITGCN	GraphWaveNet	TGCN
0%	0.938 ± 0.003	0.915 ± 0.003	0.876 ± 0.003
5%	0.934 ± 0.003	0.909 ± 0.003	0.868 ± 0.004
10%	0.929 ± 0.003	0.897 ± 0.004	0.856 ± 0.004
15%	0.923 ± 0.004	0.883 ± 0.004	0.841 ± 0.004
20%	0.916 ± 0.004	0.867 ± 0.004	0.825 ± 0.005
25%	0.906 ± 0.004	0.854 ± 0.005	0.808 ± 0.005
30%	0.894 ± 0.005	0.838 ± 0.005	0.770 ± 0.006

Table 6. Generalization results of time distribution migration

Model	Recent AUC	Medium-term AUC	Long-term AUC	Degradation (Recent $\rightarrow$ Long-term)
ITGCN	0.938 ± 0.003	0.925 ± 0.003	0.904 ± 0.004	3.6%
TRGCN	0.912 ± 0.004	0.890 ± 0.004	0.857 ± 0.005	6.0%
GraphWaveNet	0.915 ± 0.003	0.894 ± 0.004	0.863 ± 0.004	5.7%
TGCN	0.876 ± 0.003	0.848 ± 0.004	0.814 ± 0.005	7.1%

also evaluated the sensitivity of ITGCN to λ by varying it from $0.5\times$ to $2\times$ of the default value; AUC fluctuated within ± 0.003 , indicating that the model is robust to the exact choice of this hyperparameter.

Table 4 shows that ITGCN consistently outperforms all baselines in prediction windows. It has better long-term trend capture and robustness to window extension. Performance degrades with longer windows, as expected. ITGCN has a relatively small drop in AUC compared with TGCN, TRGCN and GraphWaveNet, and its multi-scale module can effectively learn long-term credit changes. ITGCN has a 30-day AUC that is still higher than TGCN’s 3-day AUC and shows better long-term performance than the original GNN.

3.4. Robustness and generalization performance analysis

To verify ITGCN’s deployment reliability, we evaluate its robustness and generalization from three perspectives.

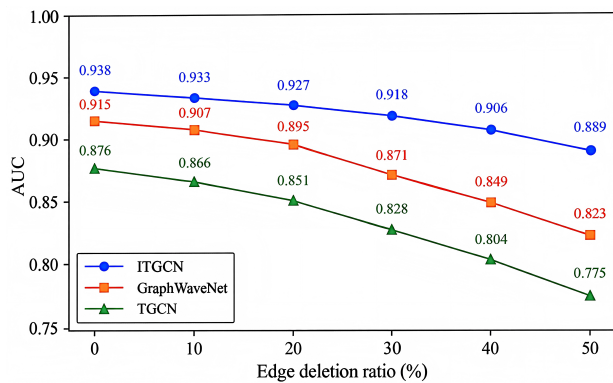
(1) Data sparsity robustness, Fig. 2 shows the effect of edge deletion ratio on AUC of each model.

(2) Missing value robustness, Table 5 shows the effect of the proportion of missing features on AUC.

As shown in Table 5, when the deletion rate was 30%, the AUC of ITGCN remained at 0.894, while GraphWaveNet

Table 7. Comparison of efficiency of each model

Model	Params (M)	Training time per epoch (s)	Inference time (ms/sample)	Epochs to converge
XGBoost	-	12	0.3	-
LSTM	1.2	36	0.8	33
GCN	1.8	58	1.1	29
TGCN	2.5	92	1.5	39
STGCN	3.2	120	1.8	43
GraphWaveNet	3.9	134	2.0	45
TemGNN	4.2	148	2.1	47
TRGCN	5.1	206	2.8	53
ITGCN	4.8	158	2.3	48

**Fig. 2.** Data sparsity robustness curve

decreased to 0.838 and TGCN decreased to 0.775. The redundancy of different domain features in the cross-domain alignment module makes up for some missing information. (3) Generalization of time distribution shift: Table 6 is the generalization result of time distribution migration.

Table 6 shows that the AUC of ITGCN on the long-term subset is 0.904, which is 3.6% lower than that in the near future, while TGCN decreased by 7.1%, TRGCN decreased by 6.0%, and GraphWaveNet decreased by 5.7%.

In summary, the superior robustness of ITGCN across all three scenarios stems from the complementary nature of its four modules: RGC provides redundant relational paths, TDA prioritizes recent reliable signals, Multi-scale captures stable long-term patterns, and Align compensates for missing features via cross-domain redundancy.

3.5. Efficiency analysis

All experiments were carried out in the same hardware environment. Table 7 is the result of efficiency analysis.

As shown in Table 7, the parameter of ITGCN is 4.8M; it is slightly smaller than that of TRGCN and slightly larger than that of GraphWaveNet. The training time per round is 158 seconds; although it is longer than that of TGCN, it is shorter than that of TRGCN, so it is within an accept-

able range. Inference time per sample is 2.3 ms; therefore, ITGCN can meet the real-time credit assessment demand of less than 100ms. It converges in 48 epochs, the same as TemGNN and GraphWaveNet. Inference time is 2.3ms/sample for the forward pass; including dynamic graph construction and neighbour sampling, the total delay is 12 – 18 ms/ sample and well below the 100 ms limit.

4. Conclusions

Aiming at the problem of dynamic credit evaluation of cross-border e-commerce, this paper proposes an algorithm ITGCN based on improved temporal graph convolution network. Extensive experiments on two public fraud detection datasets show that ITGCN achieves an AUC of 0.938 and 0.934 on D1 and D2, respectively. The F1 score reached 0.876 and 0.871. The ablation experiment verified that each module had a positive contribution, and the multi-scale component had the largest gain, with an AUC increase of 2.1%. Robustness analysis shows that ITGCN still maintains an AUC of more than 0.889 under the condition of 50% edge deletion, 0.894 under the condition of 30% feature deletion, and only 3.6% attenuation under the condition of time distribution deviation, which is consistently better than all baseline models. Efficiency analysis shows that ITGCN has 4.8M parameters and 2.3 ms / sample inference time, which achieves a good performance calculation trade-off.

The shortcomings of the research are as follows: the data set used in the experiment is simulated data, which is different from the real platform data in terms of noise distribution and abnormal pattern diversity, and the performance of the model in the actual deployment needs to be further verified. In the future, a self-supervised pre-training mechanism can be introduced to learn node representation on unlabeled transaction data to alleviate the problem of label sparseness.

Acknowledgements

There is no funding in this manuscript. The authors have no relevant financial or non-financial interests to disclose.

References

- [1] G. A. of Customs of the People's Republic of China, (2026) "China's Cross-Border E-Commerce Imports and Exports in 2025": URL: http://dzs.customs.gov.cn/customs/2026-06/16/article_2026061616113889190.html.
- [2] S. Zandi, K. Korangi, M. Óskarsdóttir, C. Mues, and C. Bravo, (2025) "Attention-based dynamic multilayer graph neural networks for loan default prediction" **European Journal of Operational Research** 321(2): 586–599. DOI: [10.1016/j.ejor.2024.09.025](https://doi.org/10.1016/j.ejor.2024.09.025). URL: <https://doi.org/10.1016/j.ejor.2024.09.025>.
- [3] Y. Qi, Y. Liu, Y. Tang, X. Chen, X. Zheng, Q. He, and X. Ao. "Dynamic Graph Learning with Static Relations for Credit Risk Assessment". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. 39. 12. 2025, 13133–13141. DOI: [10.1609/aaai.v39i12.33433](https://doi.org/10.1609/aaai.v39i12.33433). URL: <https://doi.org/10.1609/aaai.v39i12.33433>.
- [4] D. Yu and B. Xiang, (2025) "Customized integrated decision model for CBEC enterprise credit evaluation: The fusion of multi-source features and machine learning" **Electronic Markets** 35(1): 45. DOI: [10.1007/s12525-025-00793-9](https://doi.org/10.1007/s12525-025-00793-9). URL: <https://doi.org/10.1007/s12525-025-00793-9>.
- [5] Y. Jiang and P. Du, (2025) "Cross-border e-commerce credit risk early warning model based on deep learning" **International Journal of Reasoning-based Intelligent Systems** 17(10): 1–9. DOI: [10.1504/IJRIS.2025.148174](https://doi.org/10.1504/IJRIS.2025.148174). URL: <https://doi.org/10.1504/IJRIS.2025.148174>.
- [6] D. Yu and A. Fang, (2026) "Achieving credit risk prediction framework for Chinese CBECs: a hybrid CNN-BiLSTM-AM approach" **Electronic Commerce Research** 26: 2681–2704. DOI: [10.1007/s10660-025-10025-z](https://doi.org/10.1007/s10660-025-10025-z). URL: <https://doi.org/10.1007/s10660-025-10025-z>.
- [7] S. Lessmann, B. Baesens, H.-V. Seow, and L. C. Thomas, (2015) "Benchmarking state-of-the-art classification algorithms for credit scoring: An update of research" **European Journal of Operational Research** 247(1): 124–136. DOI: [10.1016/j.ejor.2015.05.030](https://doi.org/10.1016/j.ejor.2015.05.030). URL: <https://doi.org/10.1016/j.ejor.2015.05.030>.
- [8] Y. Liu, S. Liu, and Y. Lu, (2025) "Supply chain financial risk assessment: A modified graph attention neural network" **Finance Research Letters** 86: 108285. DOI: [10.1016/j.frl.2025.108285](https://doi.org/10.1016/j.frl.2025.108285). URL: <https://doi.org/10.1016/j.frl.2025.108285>.
- [9] H. Kim, J. Choi, and J. J. Whang. "Dynamic Relation-Attentive Graph Neural Networks for Fraud Detection". In: *2023 IEEE International Conference on Data Mining Workshops (ICDMW)*. 2023, 1092–1096. DOI: [10.1109/ICDMW60847.2023.00143](https://doi.org/10.1109/ICDMW60847.2023.00143). URL: <https://doi.org/10.1109/ICDMW60847.2023.00143>.
- [10] T. N. Kipf and M. Welling, (2017) "Semi-Supervised Classification with Graph Convolutional Networks" **arXiv preprint**: DOI: [10.48550/arXiv.1609.02907](https://arxiv.org/abs/1609.02907). URL: <https://arxiv.org/abs/1609.02907>.
- [11] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio, (2018) "Graph Attention Networks" **arXiv preprint**: DOI: [10.48550/arXiv.1710.10903](https://arxiv.org/abs/1710.10903). URL: <https://arxiv.org/abs/1710.10903>.
- [12] J. Wang, G. Liu, X. Xu, and X. Xing, (2024) "Credit risk prediction for small and medium enterprises utilizing adjacent enterprise data and a relational graph attention network" **Journal of Management Science and Engineering** 9(2): 177–192. DOI: [10.1016/j.jmse.2023.11.005](https://doi.org/10.1016/j.jmse.2023.11.005). URL: <https://doi.org/10.1016/j.jmse.2023.11.005>.
- [13] Z. Shao, X. Wang, E. Ji, S. Chen, and J. Wang, (2025) "GNN-EADD: Graph Neural Network-Based E-Commerce Anomaly Detection via Dual-Stage Learning" **IEEE Access** 13: 8963–8976. DOI: [10.1109/ACCESS.2025.3526239](https://doi.org/10.1109/ACCESS.2025.3526239). URL: <https://doi.org/10.1109/ACCESS.2025.3526239>.
- [14] L. Song, Z. Ma, Z. Li, and X. Shang, (2024) "Review on Temporal Graph Neural Networks for Financial Risk Prediction" **Journal of Software** 35(8): 3897–3922. DOI: [10.13328/j.cnki.jos.007087](https://doi.org/10.13328/j.cnki.jos.007087). URL: <https://doi.org/10.13328/j.cnki.jos.007087>.
- [15] Z. Wu, S. Pan, G. Long, J. Jiang, and C. Zhang. "Graph WaveNet for Deep Spatial-Temporal Graph Modeling". In: *Proceedings of the 28th International Joint Conference on Artificial Intelligence (IJCAI)*. 2019, 1907–1913. DOI: [10.24963/ijcai.2019/264](https://doi.org/10.24963/ijcai.2019/264). URL: <https://doi.org/10.24963/ijcai.2019/264>.
- [16] L. Feng, Z. Bai, and Y. Wang, (2025) "Dual-channel E-commerce fraud detection method integrating user behavior and review relationships" **Journal of Zhejiang University (Engineering Science)** 59(10): 2164–2174.

- DOI: 10.3785/j.issn.1008-973X.2025.10.017. URL: <https://doi.org/10.3785/j.issn.1008-973X.2025.10.017>.
- [17] D. Wang, Z. Zhang, J. Zhou, P. Cui, J. Fang, Q. Jia, Y. Fang, and Y. Qi. "Temporal-Aware Graph Neural Network for Credit Risk Prediction". In: *Proceedings of the 2021 SIAM International Conference on Data Mining (SDM)*. 2021, 702–710. DOI: 10.1137/1.9781611976700.79. URL: <https://doi.org/10.1137/1.9781611976700.79>.
- [18] B. P. Jeyaraman, B. T. Dai, and Y. Fang, (2024) "Temporal Relational Graph Convolutional Network Approach to Financial Performance Prediction" **Machine Learning and Knowledge Extraction** 6(4): 2303–2320. DOI: 10.3390/make6040113. URL: <https://doi.org/10.3390/make6040113>.
- [19] X. Wang, W. Lin, J. Zhao, and Z. Liu, (2026) "AdaFuse: memory-augmented adaptive fusion for few-shot fraud detection in heterogeneous graphs" **Annals of Operations Research** 361(1): 437–463. DOI: 10.1007/s10479-026-07043-x. URL: <https://doi.org/10.1007/s10479-026-07043-x>.
- [20] K. Bentaleb, M. B. Houad, and M. Mestari, (2024) "A Graph Neural Network for Comparing the Propagation of Stock Market Shocks" **IEEE Access** 12: 158611–158626. DOI: 10.1109/ACCESS.2024.3487766. URL: <https://doi.org/10.1109/ACCESS.2024.3487766>.
- [21] X. Zhang, Y. Tu, M. H. Haney, and H. Cheng, (2024) "Buyers' Negative Ratings and Textual Comments on eBay: Reasons for Posting Ratings and Factors in Denouncing Sellers" **Journal of Theoretical and Applied Electronic Commerce Research** 19(3): 1717–1733. DOI: 10.3390/jtaer19030084. URL: <https://doi.org/10.3390/jtaer19030084>.
- [22] X. Wang, J. Guo, X. Luo, and H. Yu, (2024) "DyHDGE: Dynamic heterogeneous transaction graph embedding for safety-centric fraud detection in financial scenarios" **Journal of Safety Science and Resilience** 5(4): 486–497. DOI: 10.1016/j.jnlssr.2024.05.005. URL: <https://doi.org/10.1016/j.jnlssr.2024.05.005>.
- [23] X. Mo, R. Tang, and H. Liu, (2023) "A relation-aware heterogeneous graph convolutional network for relationship prediction" **Information Sciences** 623: 311–323. DOI: 10.1016/j.ins.2022.12.059. URL: <https://doi.org/10.1016/j.ins.2022.12.059>.
- [24] F. Wang and E. Damiani, (2026) "Time-aware and transition-semantic graph neural networks for interpretable predictive business process monitoring" **Expert Systems with Applications** 301: 130320. DOI: 10.1016/j.eswa.2025.130320. URL: <https://doi.org/10.1016/j.eswa.2025.130320>.
- [25] Z. Zhang, X. Ao, C. J. Tessone, G. Liu, M. Zhou, and R. Mao, (2025) "Multiplex graph fusion network with reinforcement structure learning for fraud detection in online e-commerce platforms" **Expert Systems with Applications** 262: 125598. DOI: 10.1016/j.eswa.2024.125598. URL: <https://doi.org/10.1016/j.eswa.2024.125598>.
- [26] R. Chen, Y. Xie, M. Liao, J. Hu, and X. Liu, (2025) "Identifying Fraudulent Users in E-commerce Applications through Spatiotemporal Fusion and Selective Aggregation" **ACM Transactions on Privacy and Security** 29(1): 1–27. DOI: 10.1145/3772076. URL: <https://doi.org/10.1145/3772076>.
- [27] N. Suhadolnik, J. Ueyama, and S. Da Silva, (2023) "Machine Learning for Enhanced Credit Risk Assessment: An Empirical Approach" **Journal of Risk and Financial Management** 16(12): 496. DOI: 10.3390/jrfm16120496. URL: <https://doi.org/10.3390/jrfm16120496>.
- [28] L. Zhao, Y. Song, C. Zhang, Y. Liu, P. Wang, T. Lin, M. Deng, and H. Li, (2020) "T-GCN: A Temporal Graph Convolutional Network for Traffic Prediction" **IEEE Transactions on Intelligent Transportation Systems** 21(9): 3848–3858. DOI: 10.1109/TITS.2019.2935152. URL: <https://doi.org/10.1109/TITS.2019.2935152>.
- [29] B. Yu, H. Yin, and Z. Zhu. "Spatio-Temporal Graph Convolutional Networks: A Deep Learning Framework for Traffic Forecasting". In: *Proceedings of the 27th International Joint Conference on Artificial Intelligence (IJCAI)*. 2018, 3634–3640. DOI: 10.24963/ijcai.2018/505. URL: <https://doi.org/10.24963/ijcai.2018/505>.
- [30] M. Schlichtkrull, T. N. Kipf, P. Bloem, R. van den Berg, I. Titov, and M. Welling. "Modeling Relational Data with Graph Convolutional Networks". In: *The Semantic Web - ESWC 2018*. 10843. Lecture Notes in Computer Science. 2018, 593–607. DOI: 10.1007/978-3-319-93417-4_38. URL: https://doi.org/10.1007/978-3-319-93417-4_38.
- [31] Z. Hu, Y. Dong, K. Wang, and Y. Sun. "Heterogeneous Graph Transformer". In: *Proceedings of The Web Conference 2020 (WWW '20)*. 2020, 2704–2710. DOI: 10.1145/3366423.3380027. URL: <https://doi.org/10.1145/3366423.3380027>.

- [32] Y. Dou, Z. Liu, L. Sun, Y. Deng, H. Peng, and P. S. Yu. “Enhancing Graph Neural Network-based Fraud Detectors against Camouflaged Fraudsters”. In: *Proceedings of the 29th ACM International Conference on Information and Knowledge Management (CIKM)*. 2020, 315–324. DOI: [10.1145/3340531.3411903](https://doi.org/10.1145/3340531.3411903). URL: <https://doi.org/10.1145/3340531.3411903>.