Check for updates

Intelligent Computing
A SCIENCE PARTNER JOURNAL

RESEARCH ARTICLE

Identifying Financial Risk Contagion with Large Language Models and Temporal Graph Learning

Moyang Liu[1], Letian Zhao[1], Junqi Chen[1], Xiang Li[2*], Shuo Zhang[3], and Zhibin Niu[1*]

[1]School of Intelligence and Computing, Tianjin University, Tianjin, China. [2]China Iron and Steel Research Institute Group, Beijing, China. [3]College of Software, Nankai University, Tianjin, China.

*Address correspondence to: xli@berkeley.edu (X.L.); zniu@tju.edu.cn (Z.N.)

Citation: Liu M, Zhao L, Chen J, Li X, Zhang S, Niu Z. Identifying Financial Risk Contagion with Large Language Models and Temporal Graph Learning. *Intell. Comput.* 2026;5:Article 0292. https://doi.org/10.34133/icomputing.0292

Submitted 6 September 2025
Revised 1 December 2025
Accepted 8 December 2025
Published 6 March 2026

Copyright © 2026 Moyang Liu et al. Exclusive licensee Zhejiang Lab. No claim to original U.S. Government Works. Distributed under a Creative Commons Attribution License (CC BY 4.0).

Fostering economic growth, particularly the development of small and medium-sized enterprises, is central to United Nations Sustainable Development Goal 8. Small and medium-sized enterprises can secure bank loans with the help of guarantors, forming interconnected networked-loans. While essential for access to credit, networked-loans can also amplify risk contagion and thus are vulnerable to epidemic financial crises. Previous research has failed to capture nuanced financial semantics and analyze the mechanisms of risk contagion. We designed FinDoctor to tackle these challenges. FinDoctor leverages large language models to extract context-aware features from unstructured financial text. It then combines temporal graph neural networks with the susceptible, exposed, infectious, recovered epidemiological model to identify financial risk contagion. This enables both accurate prediction and targeted risk mitigation. Evaluated on 2 real-world datasets, FinDoctor achieves state-of-the-art performance, with area under the curve scores of 88% and 89%. We further develop an interactive application that visualizes risk propagation and supports targeted intervention. Our work provides a full-cycle solution to identify financial risk contagion in networked-loans.

Introduction

Effective risk management of networked-loans plays a critical role in safeguarding social welfare and fostering economic resilience by protecting stakeholders from systemic financial crises. To overcome the financing challenges during economic downturns, small and medium-sized enterprises (SMEs) are permitted to involve guarantors, enabling them to meet bank loan evaluation standards. This mechanism has given rise to large-scale, complex, and directed guarantee networks [1], also known as networked-loans.

However, the contagion of risk in networked-loans can be the greatest threat to economic stability. As shown in Fig. 1B, SMEs typically receive the entire loan amount upfront and repay it in regular installments after the loan is granted. If the borrower fails to fulfill its loan obligations, its guarantors are responsible for the borrower's debt repayments [2]. Therefore, as Fig. 1A shows, individual defaults can propagate like viruses through guarantee relationships, amplifying financial instability. Indeed, a few defaults in interconnected financial networks can rapidly escalate into systemic crises [3]. The bankruptcy of Lehman Brothers triggered lasting turmoil in global financial markets [4]. Default stemming from China's Kunming Pan-Asian Non-Ferrous Metals Exchange in 2015 affected companies in 28 provinces, 12 banks, and over 200,000 investors, totaling more than US$5.48 billion. Thus, it is imperative to establish a reliable model that identifies risk contagion in networked-loans, safeguarding financial stability.

Financial risk contagion is characterized by complexity, high infectivity, and partial controllability. To model such dynamics, complex network theory and infectious disease modeling have become pivotal analytical frameworks [5–7]. The global financial crisis exposed a critical flaw in classical econometric models: their inability to account for systemic interdependencies among financial institutions [8]. Graph neural networks (GNNs) have demonstrated considerable advantages in financial risk analysis due to their ability to capture both topological structures and semantic information [9]. Static multiplex GNNs have been used to detect the risk contagion [10]; static GNNs integrated with recurrent neural networks have been used to predict loan defaults in temporal networked-loans [11,12]; and attention-based GNNs have been applied to identify key contagion chains [13]. Epidemic models have been effectively transferred to financial risk analysis to characterize how risk propagates through interconnected networks. Specifically, the susceptible, infectious, recovered model has been leveraged to describe interbank dynamics [14]; the susceptible, infectious, susceptible model has been used to explore the financial contagion [15]; and a susceptible, exposed, infectious, recovered (SEIR) model based on complex networks has been used to analyze the contagion of personal credit risk [16]. While these approaches are insightful and useful, they have certain limitations (L) when applied to networked-loans.

- L1: Most of the previous work models the evolution of networked-loans using discrete-time snapshots. However,

Downloaded from https://spj.science.org on March 09, 2026

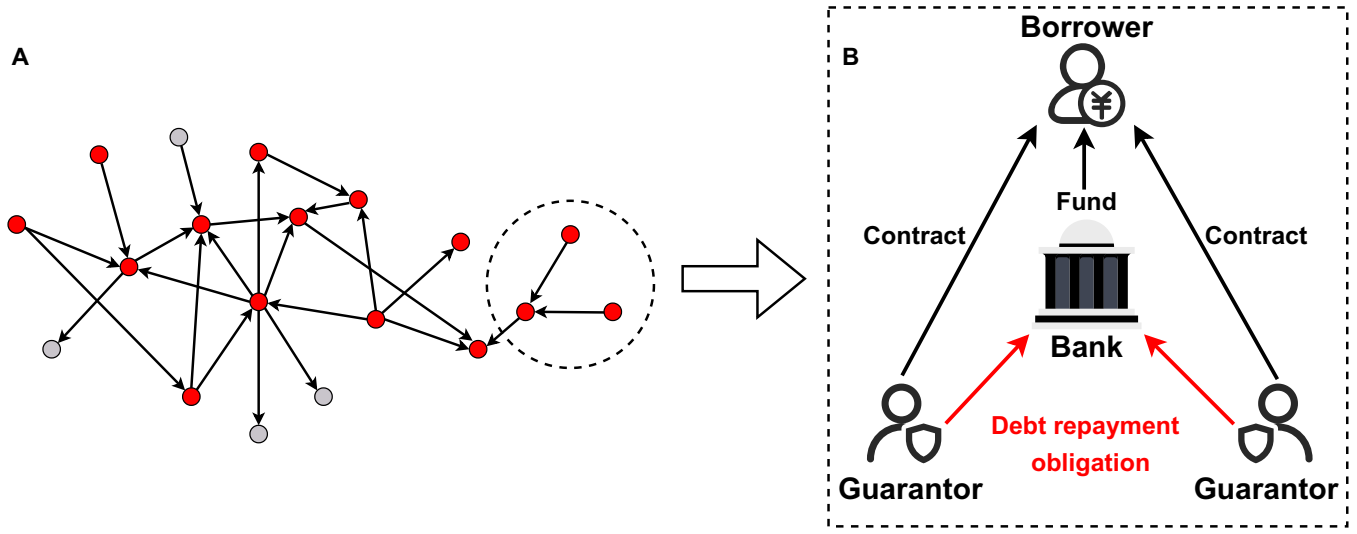

Fig. 1. A practical case of networked-loans. (A) Defaults can propagate along guarantee relationships: Red nodes denote defaulted companies, gray nodes represent healthy companies, and directed edges indicate guarantee relationships. (B) The workflow for guaranteed loan applications.

these discrete time dynamic graph approaches cannot reflect multiple and self-guarantee relationships (multiple edges and self-loops), and the choice of snapshot granularity is often arbitrary and sensitive to results.

- L2: Existing epidemic modeling studies rely on parameterized simulations, neglecting financial network topology and semantic information.
- L3: Previous studies on networked-loans typically derive initial embeddings by extracting structured fields from contracts and naively concatenating them. However, these noncontextual shallow representations fail to capture nuanced financial meanings.

To address these limitations, we present FinDoctor, a model to identify financial risk contagion. By aligning the practical expertise of financial experts with large language models (LLMs), FinDoctor gains augmented deep embeddings of the intricate financial details. Then, leveraging a temporal graph learning model and a sophisticated SEIR epidemic model, we track and predict the evolution of the networked-loans to assess and mitigate contagion risk. In summary, the main contributions (C) are as follows.

- C1: By coupling a continuous time dynamic graph (CTDG) representation with an SEIR epidemic model, our framework explicitly captures the state transitions of financial distress: from susceptible to exposed, infectious, and recovered, enabling interpretable identifications of risk contagion.
- C2: Our method introduces a new paradigm for leveraging LLMs to enhance dynamic GNNs at the text level. The performance is substantially elevated by aligning financial professionals' expertise with LLMs' capabilities.
- C3: We evaluate our method using 2 real-world networked-loans datasets, achieving results that surpass state-of-the-art baselines. In addition, we devise 2 risk mitigation methods to prevent systemic crises, restoration, and isolation, achieving promising results.
- C4: We further develop an interactive system to visualize risk contagion evolution and risk mitigation.

Materials and Methods

Problem formulation

Networked-loan

A guarantee relationship is established when one company guarantees a loan for another. The contract includes loan-level details, such as start and end dates and loan amount, along with firm-level information.

Thus, a guarantee contract i can be present as 2 types of events: edge addition interaction $\delta(t_{\text{start}}) = \left(u, v, t_{\text{start}}, X^{u,v}_{t_{\text{start}}}\right)$ and edge deletion interaction $\delta(t_{\text{end}}) = \left(u, v, t_{\text{end}}, X^{u,v}_{t_{\text{end}}}\right)$. These events signify a guarantee relationship between borrower u and guarantor v starting at t_{start} and ending at t_{end}. Each interaction has an edge feature $X^{u,v}_t \in R^{d_E}$ including the type of event and details of the guarantee contract. Here, d_E denotes the dimensions. Hence, a set of contracts can be represented as a sequence of interactions with timestamps $\{\delta(t_1), \delta(t_2), \ldots\}$. This representation naturally captures the multigraph structure, allowing multiple interactions between 2 nodes. This temporal detail provides a more precise modeling of the duration and timing of guarantee relationships.

Risk contagion modeling via SEIR

Our objective is to forecast the impending SEIR state of each company based on the known states and interactions, which allow us to obtain the basic reproduction number (R_0). After detecting abnormal spikes in R_0, we implement targeted risk mitigation strategies to prevent large-scale financial crises.

R_0 represents the number of secondary infections caused by a single infectious individual in a fully susceptible population during the infectious period [17]. It serves as a threshold in disease dynamics, where an R_0 greater than 1 indicates widespread contagion, while an R_0 less than 1 suggests that the contagion will fade out. In the context of financial risk contagion, $R_0(t)$ quantifies the contagion potential of financial risk at time t.

Risk contagion in financial networks is as intricate as the spread of infectious diseases [18]. Defaults do not occur overnight; they

Downloaded from https://spj.science.org on March 09, 2026

develop through a latent process where risks accumulate within a seemingly stable system, ultimately triggering widespread defaults [19]. To capture this latent process, we collaborate closely with financial experts to modify the classical SEIR model, tailoring it to reflect networked-loans' characteristics. As illustrated in Fig. 2, the risk contagion process in networked-loans involves companies transitioning through 4 states: Susceptible state (S): Borrowers and their guarantors become susceptible after the borrowing behavior occurs. Exposed state (E): The company that contacted the infectious company does not have the ability to infect others. Infectious state (I): The defaulted company's default risk will be transmitted to its guarantors through association. Recovered state (R): Apart from external factors (legal interventions or government aid), an infectious company gets out of risk only when its guarantors assist in repaying the delayed bank loan. Once a recovery state in a financial network is certified by a regulator, it remains immune until the current loan expires [20]. Key parameters include the infection rate (β), which measures how quickly susceptible entities become exposed through contact with infectious companies; the transition rate (σ), which reflects how fast exposed companies become infectious; and the recovery rate (γ), indicating the speed at which infectious companies recover from the defaults.

Fig. 2. Risk contagion process in networked-loans. Here and elsewhere, each color represents a different node status. Gray, susceptible; yellow, exposed; red, infectious; green, recovered.

The well-established SEIR model capitalizes on the following ordinary differential equations to depict the dynamic of the risk contagion:

$$\begin{cases} \dfrac{dS(t)}{dt} = -\dfrac{\beta(t)S(t)I(t)}{N(t)} \\ \dfrac{dE(t)}{dt} = \dfrac{\beta(t)S(t)I(t)}{N(t)} - \sigma(t)E(t) \\ \dfrac{dI(t)}{dt} = \sigma(t)E(t) - \gamma(t)I(t) \\ \dfrac{dR(t)}{dt} = \gamma(t)I(t). \end{cases} \tag{1}$$

Leveraging the calibrated SEIR model, we can estimate the basic reproduction number R_0 at t as follows:

$$R_0(t) = \frac{\beta(t)}{\gamma(t)} \cdot \frac{1}{\sigma(t)}. \tag{2}$$

Methods

Figure 3 illustrates our proposed FinDoctor framework comprising enhancer and encoder modules. The enhancer processes guarantee contracts to capture financial terms and subtle risks, generating semantically enriched embeddings. These embeddings are passed to the encoder, which models the dynamic evolution of risk contagion across the network and predicts the state of nodes in networked-loans. Furthermore, using the SEIR model, we compute R_0, a critical indicator of contagion risk, to inform proactive risk management.

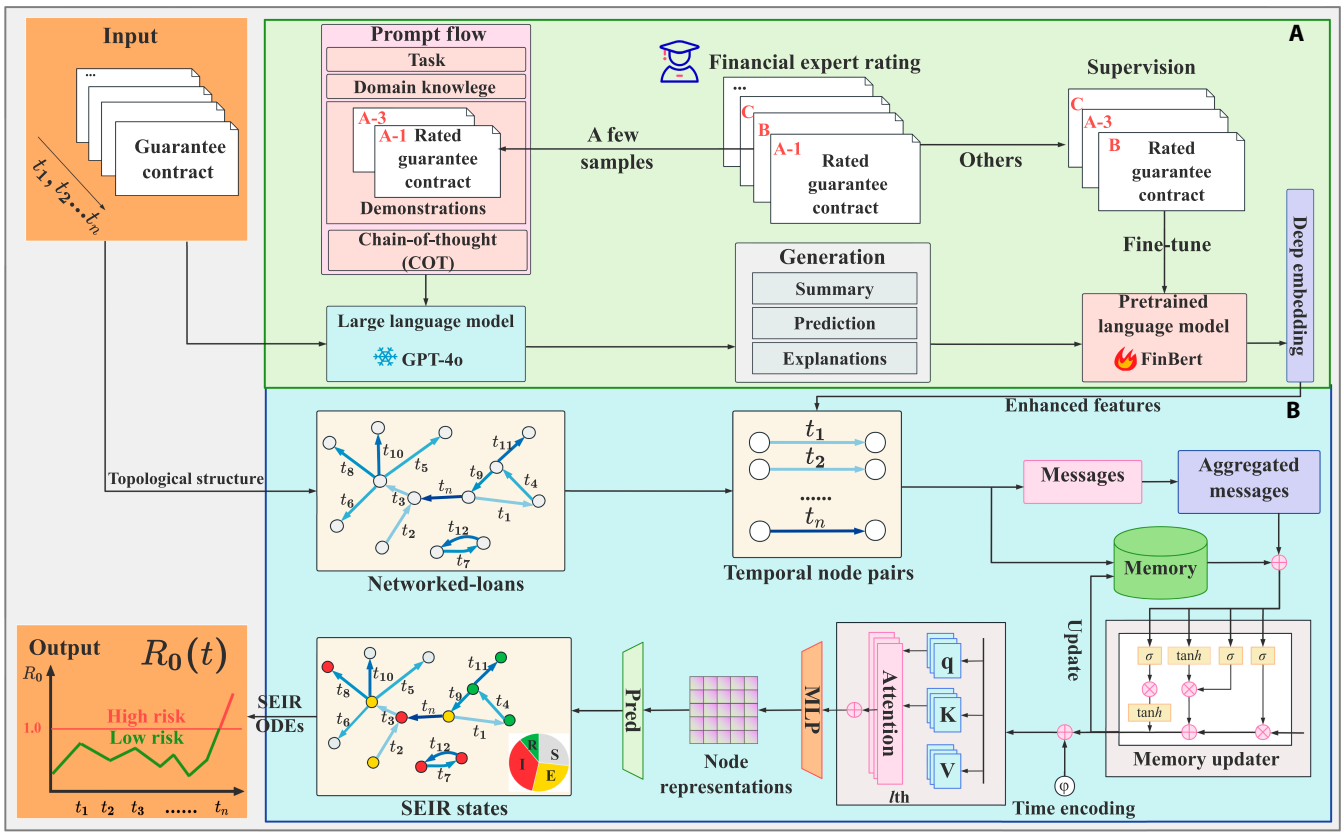

Fig. 3. Overview of our FinDoctor methodology. Its main components are (A) an enhancer module and (B) an encoder module.

Downloaded from https://spj.science.org on March 09, 2026

LLM-based enhancer

To enhance the textual representation for downstream tasks, we propose an LLM-based feature enhancement module that leverages the reasoning power of LLMs and the context-aware representation capabilities of domain-specific pretrained language models (LMs).

Effective feature enhancement requires the introduction of semantically rich signals beyond the original text [21]. Although prior studies demonstrate that combining GNNs with financial expert knowledge enhances credit risk assessment [22], the process of extracting structured information from unstructured documents using expert knowledge is costly, and the results are easily influenced by subjective judgments and changes in the business environment. To overcome this challenge, we propose an enhanced paradigm that combines LLMs and domain-specific LMs. First, powerful LLMs (such as GPT-4o) are used to analyze unstructured guarantee contracts, generating structured supervision signals, including concise summaries, multilevel risk predictions, and natural language explanations. These signals are carefully designed to simulate the judgment style and reasoning logic of professional financial rating personnel. This process essentially transforms the original contract into an interpretable and expert-aligned intermediate representation.

Second, given that decoder-only LLMs are optimized for generation tasks and poorly suited for context-aware representation learning, we use the signals generated by LLMs as auxiliary inputs and perform supervised fine-tuning on the lightweight encoder-only model FinBERT [23], a BERT (Bidirectional Encoder Representations from Transformers) model trained on large-scale financial data. The resulting fixed-dimensional embeddings from the FinBERT encoder are then used as enhanced features for downstream GNNs.

In the following, we provide in-depth descriptions of key steps involved in the LLM-based enhancer module.

Risk labeling with financial experts

We collaborate with financial experts to label a portion of our guarantee contracts following the risk gradient outlined by the Standard & Poor's Short-Term Issuer Credit Rating Criteria.

Weak labeling by LLM

We utilize advanced GPT-4o to assess the guarantee contracts and generate summaries, multiple risk-label predictions, and explanations. We use prompts constructed as shown in Fig. 4.

1. Task: Tasks are assigned, and settings are established.

Task:
You are now a financial expert. This is a guarantee contract **[contract content].** First, please summarize the following guarantee contract. Then, for this guarantee contract, list the possible risk labels (S&P issuer credit ratings), ordered by likelihood, with the most likely label first. Then provide explanations for the predicted risk labels.

Domain Knowledge:
Here are the definitions of the financial terms in the guarantee contract:
corp_owner_type: Type of corporate ownership structure; SME: Indicates whether the company is a small or medium-sized enterprise; Encouragement: Indicates whether the company enjoys policy encouragement or subsidies; Scope: Indicator of the main business category; overdue_times …

Demonstrations:
Contract 1:
- Information: BorrowerID: 8571091902, GuarartorID: 8571051587, corp_owner_type: A, SME: 1, Encouragement: C, overdue_times: 9 …
- summary: The borrower is an SME with a history of 9 overdue incidents. The level of policy encouragement for the borrower is rated as C. The loan amount is $1,000,000 with a term of 12 months. The guarantor company's stock has declined over the past 10 months.
- Ranked Risk labels: **C, D, B**
- Explanations: The borrower is an SME with a history of poor credit behavior and lacks policy encouragement. The loan amount is substantial compared with its average revenue and has a long duration. Additionally, the guarantor company's stock has declined over the past 10 months, indicating financial difficulties …

More Contract demonstrations **……**

Results:
- [**LLM generation: a summary of input contract**]
- [**LLM generation: a ranked prediction list**]
- [**LLM generation: explanations for the predictions**].

Fig. 4. Prompt flow instance.

Downloaded from https://spj.science.org on March 09, 2026

2. Domain knowledge: Key concepts, terminology, and background are supplied to ensure that the LLM is well versed in the financial domain.
3. Demonstrations: Several risk rating examples are provided to help the LLM generate responses that are consistent with expert practices.
4. Chain-of-thought: Chain-of-thought [24], a step-by-step reasoning process, is used to produce the answer. Few-shot prompting has been proven to perform poorly on intricate tasks. Thus, we exploit chain-of-thought to instruct the LLM to summarize first, which implicitly forces the LLM to recall relevant financial knowledge before generation, leading to more reliable outputs.

Fine-tuning LM interpreter

The purpose of this stage is to distill the structured knowledge generated by the LLM into a trainable encoder. Specifically, we use the risk labels annotated by experts as the ground truth to conduct supervised fine tuning of FinBERT. The model input includes customized outputs from the LLM. This training stage enables FinBERT to learn contextual representations that contain both original semantics and internalized expert inference patterns.

Let $\mathbf{x} \in \mathcal{X}$ denote a raw guarantee contract, and let $\mathbf{z} = \mathrm{LLM}(\mathbf{x})$ be the LLM-structured augmentation. The contextual embedding is then obtained via the FinBERT encoder:

$$\mathbf{e} = \mathrm{FinBERT}(\mathbf{z}), \tag{3}$$

where $\mathbf{e}$ serves as the expert-aligned representation. A multilayer perceptron (MLP) classifier head predicts the risk label as

$$\hat{y}_c = \mathrm{Softmax}(\mathrm{MLP}(\mathbf{e})), \tag{4}$$

and the model is trained using the cross-entropy loss:

$$L = -\frac{1}{|C|}\sum_{c=1}^{|C|} y_c \log\left(\hat{y}_c\right), \tag{5}$$

where y is the ground-truth label, C is the number of labels, and θ are the learnable parameters.

Temporal graph network encoder

Our encoder utilizes a temporal graph network (TGN) [25] as the backbone. Given a graph G representing a sequence of time-stamped guarantee relation interactions, for each time t, our encoder learns the temporal embeddings for all nodes denoted as $Z(t) = \left(z_1(t), \ldots, z_i(t), \ldots\right)$. In addition, a memory M stores the memory state for each node at time t, denoted as $S(t) = \left(s_1(t), \ldots, s_i(t), \ldots\right)$, capturing the node's temporal evolution up to time t. Our encoder consists of 5 important modules as follows.

Time encoding

To distinguish different timestamps, we utilize the time encoding function $cos\left(t_n\omega\right)$ proposed by GraphMixer [26] to map each timestamp to a d_T dimensional vector, where $\omega_i = \alpha^{-(i-1)/\beta}$, $i = 1, \ldots, d_T$. The cosine function projects the scaled time values into the range $[-1, +1]$, while the exponentially decaying frequencies ω_i ensure that higher-dimensional components capture fine-grained temporal differences and lower-frequency components model long-term trends. Following GraphMixer [26], we set $\alpha = \beta = \sqrt{d_T} = 10$ and keep ω fixed during training. This configuration has become a standard choice in continuous-time graph representation learning. The resulting encoding is temporally smooth, with similar timestamps yielding similar encodings. As timestamps increase, the components of the encoding vector progressively saturate toward +1, starting from the higher-frequency dimensions.

Message function

Given an event interaction $\delta(t) = (u, v, t)$, messages update the memory state of node u at time t as follows:

$$m_u(t) = Msg\left(s_u(t^-), s_v(t^-), \phi(\Delta t), X_{u,v}(t)\right). \tag{6}$$

We implement $Msg(\cdot)$ using concatenation. $s_u(t^-)$ and $s_v(t^-)$ denote the memory states of nodes u and v before t. Δt denotes the time interval since the last update, $\phi(\cdot)$ represents the time encoding, and $X_{u,v}(t)$ are the edge features.

Message aggregator

For batch processing, we need to aggregate multiple messages within the same batch. Each node u may have multiple interactions in the time interval $\left[t_1, t\right]$, and each event generates a message. We use mean pooling for aggregating the most recent k messages $m_u\left(t_{b-k}\right), \ldots, m_u\left(t_b\right)$ for $t_1 \le t_{b-k}, \ldots, t_b \le t$.

If there are fewer than k messages, we apply zero padding to maintain a fixed length. The rationale is that recent events are more representative of the current state in the temporal network [27].

$$\overline{m}_u(t) = \mathrm{Agg}\left(m_u\left(t_{b-k}\right), \ldots, m_u\left(t_b\right)\right). \tag{7}$$

Memory updater

Historical information enables the model to recognize evolving patterns and long-term dependency and thus is crucial in updating current states. Consequently, state $s_u(t)$ updates based on $\overline{m}_u(t)$ and its previous state $s_u(t^-)$. This update can be formulated as follows:

$$s_u(t) = \mathrm{Mem}\left(s_u(t^-), \overline{m}_u(t)\right). \tag{8}$$

We implement the $\mathrm{Mem}(\cdot)$ function, which captures the temporal patterns of node states, as follows:

$$\begin{aligned}
i_t &= \sigma\left(W_i\left[s_u(t^-), \overline{m}_u(t)\right]\right) \\
f_t &= \sigma\left(W_f\left[s_u(t^-), \overline{m}_u(t)\right]\right) \\
\widetilde{C}_t &= \tanh\left(W_c\left[s_u(t^-), \overline{m}_u(t)\right]\right) \\
o_t &= \sigma\left(W_o\left[s_u(t^-), \overline{m}_u(t)\right]\right) \\
C_t &= f_t \odot C_{t^-} + i_t \odot \widetilde{C}_t \\
s_u(t) &= o_t \odot \tanh\left(C_t\right),
\end{aligned} \tag{9}$$

where $\odot$ denotes element-wise multiplication. The tth LSTM cell takes C_{t^-} and s_{t^-} as inputs and produces C_t and h_t as outputs.

Downloaded from https://spj.science.org on March 09, 2026

Self-attention mechanism

To avoid the so-called memory staleness problem [28] and obtain the final embedding $z_u(t)$ of any node u at any t, inspired by TGAT [29], we utilize the self-attention mechanism (SAM) [30].

In a series of L graph attention layers, node u's embedding is computed by aggregating information from its L-hop temporal neighbors. At each layer l, the inputs include node u's representation $h_u^{(l-1)}(t)$, timestamp t, and the representations of its neighbors $h_1^{(l-1)}(t), \ldots, h_N^{(l-1)}(t)$ with their respective timestamps $t_1, \ldots, t_N$. The initial representation $h_u^{(0)}(t)$ for node u is set to its memory $s_u(t)$.

$$\begin{aligned} C_u^{(l)}(t) &= \left[h_1^{(l-1)}(t) \,\|\, \phi(t-t_1), \ldots, h_N^{(l-1)}(t) \,\|\, \phi(t-t_N)\right] \\ q_u^{(l)}(t) &= \left(h_u^{(l-1)}(t) \,\|\, \phi(0)\right) W_Q \\ K_u^{(l)}(t) &= C_u^{(l)}(t) W_K \\ V_u^{(l)}(t) &= C_u^{(l)}(t) W_V \\ \tilde{h}_u^{(l)}(t) &= \text{MultiHeadAttention}^{(l)}\left(q_u^{(l)}(t), K_u^{(l)}(t), V_u^{(l)}(t)\right) \\ h_u^{(l)}(t) &= MLP^{(l)}\left(h_u^{(l-1)}(t) \,\|\, \tilde{h}_u^{(l)}(t)\right) \end{aligned} \tag{10}$$

We replace origin positional encoding with time encoding $\phi(\cdot)$. Here, $\|$ denotes concatenation, and W_Q, W_K, and W_V are learnable weight matrices. Each layer performs multihead attention, using $q^{(l)}(t)$, $K^{(l)}(t)$, and $V^{(l)}(t)$ as the query, key, and value, respectively. An MLP layer combines the node's representation with the aggregated information from its neighbors, yielding the final embedding $z_u(t) = h_u^{(l)}(t)$.

Loan default epidemic prediction

For each node u, the final embedding $z_u(t)$ is used to classify the company's epidemic states at t. Given a labeled set $D_t = \{(v,y)\}$ at t, we apply a cross-entropy loss as follows:

$$L = -\frac{1}{|D_t|} \sum_{u=1}^{|D_t|} \sum_{k=1}^{4} y_{uk} \log\left(\hat{y}_{uk}\right). \tag{11}$$

The predicted label $\hat{y}_u$ is computed as follows:

$$\hat{y}_u(t) = \text{pred}\left(z_u(t); \theta\right). \tag{12}$$

Here, $\text{pred}\left(z_u(t); \theta\right)$ is the prediction function that sets $z_u(t)$ to a real-valued score. We implement this using 2 layers of MLP with LeakyReLU activation in the hidden layer and apply the Softmax activation function in the output layer.

Related work

LLM-enhanced graph learning

The capability to capture textual information and topology renders GNNs potent instruments [31]. However, many early methods relied on shallow, noncontext-aware text embeddings (such as word2vec), which made it difficult to capture fine-grained semantic information in node attributes. The semantic limitations of such embeddings considerably affect the performance of downstream tasks, especially in scenarios that require deep language understanding [32].

The introduction of LLMs into the enhancement of GNNs provides new perspectives on graph representation learning. Recent research has begun to explore the integration of knowledge generated by LLMs into the GNN framework. For example, He et al. [21] combined the predicted results generated by an LLM with explanatory text to enhance the initial node features, thereby improving classification performance. Xu et al. [33] utilized an LLM to enhance attribute representation in molecular graphs, in order to improve property prediction tasks in drug discovery. However, existing methods often treat LLMs as black box annotators and do not optimize the semantic representation specifically for graph-based risk modeling. In contrast, the proposed method utilizes the collaborative design of an LLM and a domain-specific LM to explicitly construct high-quality, expert-aligned text embeddings, fundamentally enhancing the input representation capability of the GNN.

Dynamic GNNs

Based on time granularity, dynamic GNNs can be categorized into discrete time dynamic graphs (DTDGs) and CTDGs.

A DTDG can be represented as a chronological sequence of graph snapshots taken at fixed time intervals: $G = \{G_0, G_1, \ldots, G_n\}$. The core is combining spatial GNNs with temporal recurrent neural networks such as long short-term memory relational graph convolutional network [34] and EvolveGCN [35]. However, DTDGs have 2 main drawbacks: The fixed time intervals of DTDGs obscure key details and event sequences, and DTDGs cannot model multiedge graphs.

A CTDG can be represented as a chronological sequence of interactions between specific node pairs: $G = \{(u_0, v_0, t_0), G = (u_1, v_1, t_1) \ldots (u_n, v_n, t_n)\}$. CTDGs excel in dynamic graph benchmarks by using various advanced methods. Approaches like Know-Evolve [36] utilize temporal point processes to capture the time dependence of event occurrences. Methods based on recurrent neural networks, such as the dynamic GNN [37] and the TGN [25], focus on the influence intensity of interactions. In addition, random walk models like the causal anonymous walk network [38] enable inductive learning of temporal networks. Given their superior performance and characteristics, our study primarily focuses on CTDGs.

Results and Discussion

Experimental settings

Datasets

We evaluated our method using 2 real-world datasets. Dataset 1 is from a major international bank in Asia and covers the time period from 2004 January 1 to 2016 December 31. It encompasses 0.48 million companies, over US$200 billion in registered capital, 0.79 million guarantee relationships, and a total loan volume of over US$3 trillion. Dataset 2 is the Guarantee Status Table of Shanghai A-Share Listed Companies from the Wind Economic Database (EDB) (https://www.wind.com.cn/). Each contains detailed loan-level and firm-level information for each participant.

Baseline methods and settings

To substantiate the effectiveness of FinDoctor on the networked-loans, we considered 4 categories of methods as baselines on SEIR state classification: (a) standard financial methods: LightGBM [39] and XGBoost [40]; (b) static GNNs: GCN [41] and GAT [42]; (c) DTDGs: EvolveGCN and DySAT [43]; (d) CTDGs: JODIE [44], TGAT, TGN, and GraphMixer. In addition, we used 2

Downloaded from https://spj.science.org on March 09, 2026

versions of TGN: TGN-attn and TGN-mean. The key difference is in the final embedding stage: The former uses SAM, while the latter uses mean pooling. Notably, for all baseline models, we used the same initial node embeddings generated by our enhancer module.

FinDoctor and its variants

We conducted an ablation study to assess the impact of the 2 main components of FinDoctor, the enhancer and encoder modules.

The FinDoctor-noEnhancer variant relies on the bag-of-words (BoW) approach to producing shallow embeddings for the downstream GNN, the same as baseline methods. The FinDoctor-TGN variant utilizes the unmodified TGN-atten as the encoder module. FinDoctor-ALL denotes the complete model. We set the number of attention heads to 2, the hidden size of *pred* to 128, the initial learning rate to 0.0001, and the batch size to 200.

Evaluation metrics

Given the multiple states in SEIR, we evaluated the accuracy of predictions using area under the curve (AUC) and the following multilabel classification metrics: micro F1, macro F1, and weighted F1. We set ground-truth labels for each company based on the spread scope of real default contagion. In our experiments, the 2 datasets were split chronologically into training, validation, and test sets. The respective ratios for training, validation, and test sets in Dataset 1 were 8:1:1, while those for Dataset 2 were 10:1:1.

Results

Table 1 summarizes the experimental results, showing the average AUC, micro F1, macro F1, and weighted-F1 values over 10 trials. Methods based on static GNNs outperform standard financial methods, highlighting the importance of network topology information. The key difference in static GNNs is GAT's spatial attention mechanism, which better adapts to directed graphs like networked-loans. EvolveGCN and DySAT exceed the static graph models, emphasizing the importance of temporal information in networked-loans.

CTDG-based methods improve AUC by 3% to 7% and multiclass metrics by 1% to 5% over DTDG-based methods, highlighting the benefit of continuous-time interaction modeling in temporal networks. Among them, SAM-based approaches (TGN-atten and TGAT) achieve the best overall performance. TGN-atten surpasses TGN-mean with a 2% boost in AUC and a 2% to 3% gain in multilabel metrics, directly proving the effectiveness of the SAM. The comparison between TGN-atten and TGAT shows the advantage of the memory module in real-time updates and long-term dependencies.

TGN-atten, with its superior performance and extensibility, serves as the foundation for integrating other methods' strengths. FinDoctor-noEnhancer, the extended version of TGN, surpasses the unmodified model, illustrating the success of our modifications. When comparing FinDoctor-TGN to the original TGN, the introduction of the enhancer gives all metrics a substantial boost varying from 2% to 3%, proving more impactful. Consequently, FinDoctor-ALL delivers the highest performance across all metrics.

Ablation study on hyperparameters

In our implementation, the key architectural hyperparameters were selected as follows. For components inherited from the TGN backbone [25], including the memory dimension and message dimension, we adopted the default settings from the original TGN paper to ensure compatibility. For other components, we set the number of attention heads to 2, the hidden size of the MLP in the prediction module to 128, and the message aggregation window size to 3. To assess the robustness of these choices, we conducted an ablation study on 3 critical hyperparameters: (a) the number of attention heads, (b) the hidden size of the prediction MLP, and (c) the message aggregation window size k. Results on Dataset 1 are summarized in Table 2.

Table 1. Comparison of the prediction results. The best values are shown in bold.

		Dataset 1				Dataset 2			
Type of method	Model	Micro F1	Macro F1	Weighted F1	AUC	Micro F1	Macro F1	Weighted F1	AUC
Standard methods	Lightgbm	0.695	0.289	0.682	0.717	0.711	0.219	0.642	0.740
	XGBoost	0.706	0.334	0.704	0.720	0.721	0.232	0.702	0.738
Static GNNs	GCN	0.715	0.342	0.707	0.733	0.720	0.210	0.732	0.747
	GAT	0.731	0.351	0.710	0.743	0.749	0.348	0.702	0.758
DTDGs	EvolveGCN	0.728	0.358	0.729	0.756	0.779	0.279	0.752	0.777
	DySAT	0.737	0.360	0.734	0.762	0.761	0.318	0.756	0.770
CTDGs	JODIE	0.746	0.372	0.726	0.796	0.764	0.303	0.742	0.765
	TGN-mean	0.755	0.373	0.739	0.810	0.771	0.325	0.757	0.784
	GraphMixer	0.760	0.402	0.748	0.818	0.780	0.319	0.756	0.789
	TGAT	0.770	0.387	0.757	0.822	0.790	0.328	0.774	0.840
	TGN-atten	0.771	0.398	0.752	0.831	0.785	0.347	0.784	0.858
Proposed method	FinDoctor-BoW	0.635	0.348	0.682	0.705	0.703	0.328	0.687	0.761
	FinDoctor-TGN	0.784	0.422	0.769	0.858	0.801	**0.349**	0.789	0.878
	FinDoctor-ALL	**0.790**	**0.436**	**0.778**	**0.882**	**0.820**	0.340	**0.808**	**0.893**

Downloaded from https://spj.science.org on March 09, 2026

The results show that FinDoctor is stable across a range of reasonable values. Our settings achieve near-optimal performance while maintaining computational efficiency. The larger MLP sizes may lead to overfitting. Notably, the choice of aggregation window size aligns with the temporal locality principle proposed in CPDG [27], confirming the importance of recent interactions in risk contagion modeling.

Enhancer module analysis

In this section, we demonstrate the effectiveness of our enhancer module across 3 dimensions on Dataset 1.

Text embedding quality

To assess the effectiveness of our enhancer in generating high-quality initial node representations, we compared it against multiple embedding strategies: (a) traditional BoW, (b) a general-purpose semantic encoder (Sentence-BERT [45]), (c) expert handcrafted features engineered by financial professionals based on domain rules, and (d) FinBERT trained only on raw contracts without LLM augmentation. All embeddings were fed into the same FinDoctor encoder for the SEIR state classification task.

As shown in Table 3, our deep embeddings achieve the best performance in the downstream task. Notably, our method even outperforms expert handcrafted features.

To further validate the semantic discriminability of our embeddings, we constructed 100 positive and 100 negative contract pairs based on expert risk labels. As shown in Table 4, our method achieves the highest positive-pair similarity (Sim^+) and the lowest negative-pair similarity (Sim^-), resulting in a considerably larger margin ($\Delta Sim = 0.37$) compared to all baselines. This confirms that our LLM-enhanced embeddings effectively capture risk-consistent semantics, enabling meaningful distinction between contracts of different risk profiles.

Visualization

To provide a more intuitive and interpretable evaluation of the effectiveness of our deep embeddings, we visualized 100 expert-labeled contracts using t-distributed stochastic neighbor embedding (t-SNE) and compared our deep embeddings

Table 2. Ablation study on key hyperparameters on Dataset 1. The best values are shown in bold.

Hyperparameter	Value	AUC
Attention heads	1	0.778
	2	0.882
	4	**0.883**
MLP hidden size	64	0.736
	128	**0.882**
	256	0.846
Aggregation window *k*	1	0.806
	3	**0.882**
	5	0.879

Table 3. Downstream SEIR state classification performance on Dataset 1 using different initial node embeddings. The best values are shown in bold.

Model	Micro F1	Macro F1	Weighted F1	AUC
BoW	0.635	0.348	0.682	0.705
Sentence-BERT	0.742	0.389	0.730	0.762
FinBERT (without LLM)	0.768	0.406	0.747	0.814
Expert-handcrafted features	0.782	0.428	0.768	0.875
Ours	**0.790**	**0.434**	**0.778**	**0.882**

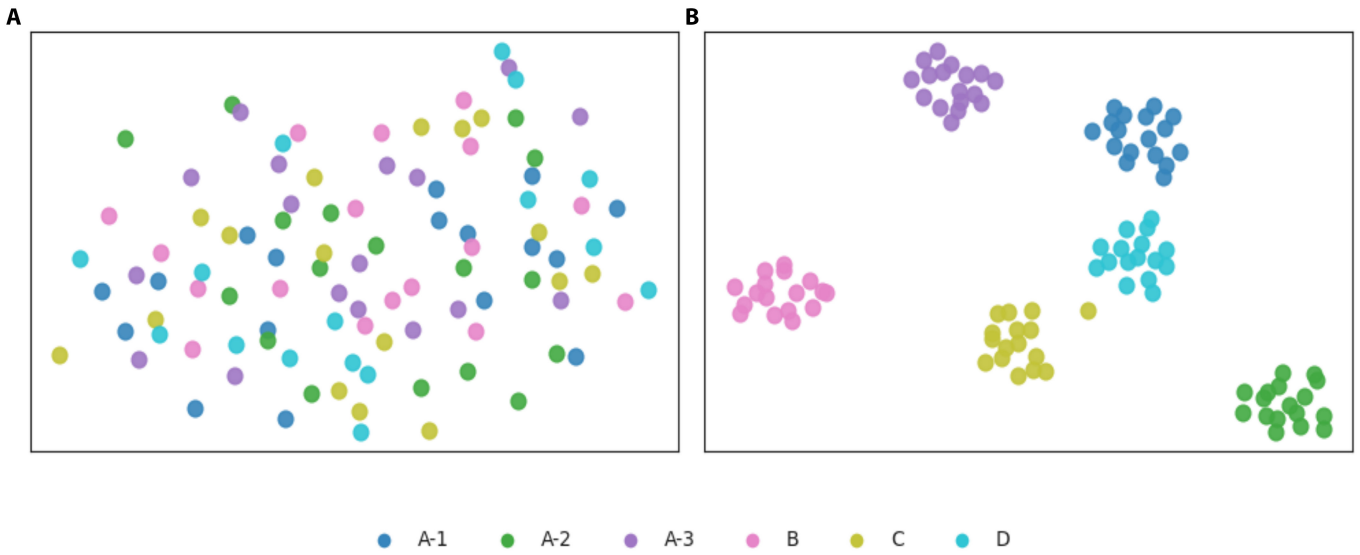

Fig. 5. Visualizations of t-SNEs. (A) Shallow embeddings from BoW. (B) Deep embeddings from enhancer of FinDoctor.

Downloaded from https://spj.science.org on March 09, 2026

with shallow representations based on BoW. As shown in Fig. 5, BoW embedding exhibits severe interclass overlap, with contracts with the same risk label scattered in the embedding space, indicating the inability to capture distinct financial semantics. In contrast, our LLM-enhanced embedding form compact clusters that are precisely aligned with the risk labels. This clear separation indicates that our enhancer successfully encodes domain-specific risk signals into meaningful representations, providing an interpretable basis for downstream risk contagion modeling.

Table 4. Semantic similarity evaluation on positive and negative contract pairs. Higher Sim^+ and lower Sim^- yield better separation. The best values are shown in bold.

Embedding method	$Sim^+(\uparrow)$	$Sim^-(\downarrow)$	$\Delta Sim = Sim^+ - Sim^-(\uparrow)$
BoW	0.38	0.31	0.07
Sentence-BERT	0.47	0.28	0.19
FinBERT (without LLM)	0.52	0.30	0.22
Expert-handcrafted features	0.51	0.34	0.17
Ours	**0.58**	**0.21**	**0.37**

Generality

To affirm our enhancer module's generality and broad applicability, we incorporated it into baseline methods. The performance of all the models was boosted as shown in Fig. 6. In addition, the enhancer provides the largest improvement to the TGN-atten model.

Risk mitigation case study

To identify highly infectious companies early and curb systemic risk, we developed an interactive system for the dynamic presentation of risk contagion in networked-loans. Based on

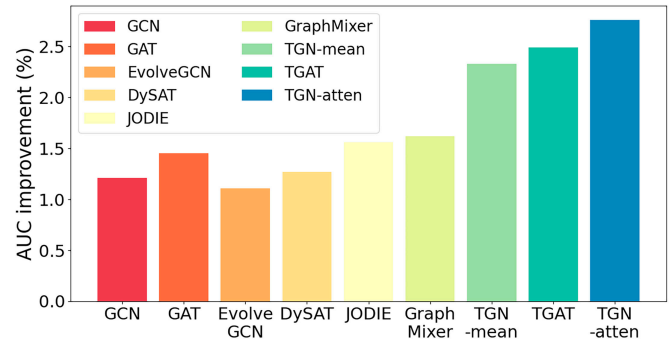

Fig. 6. AUC improvement with deep embeddings.

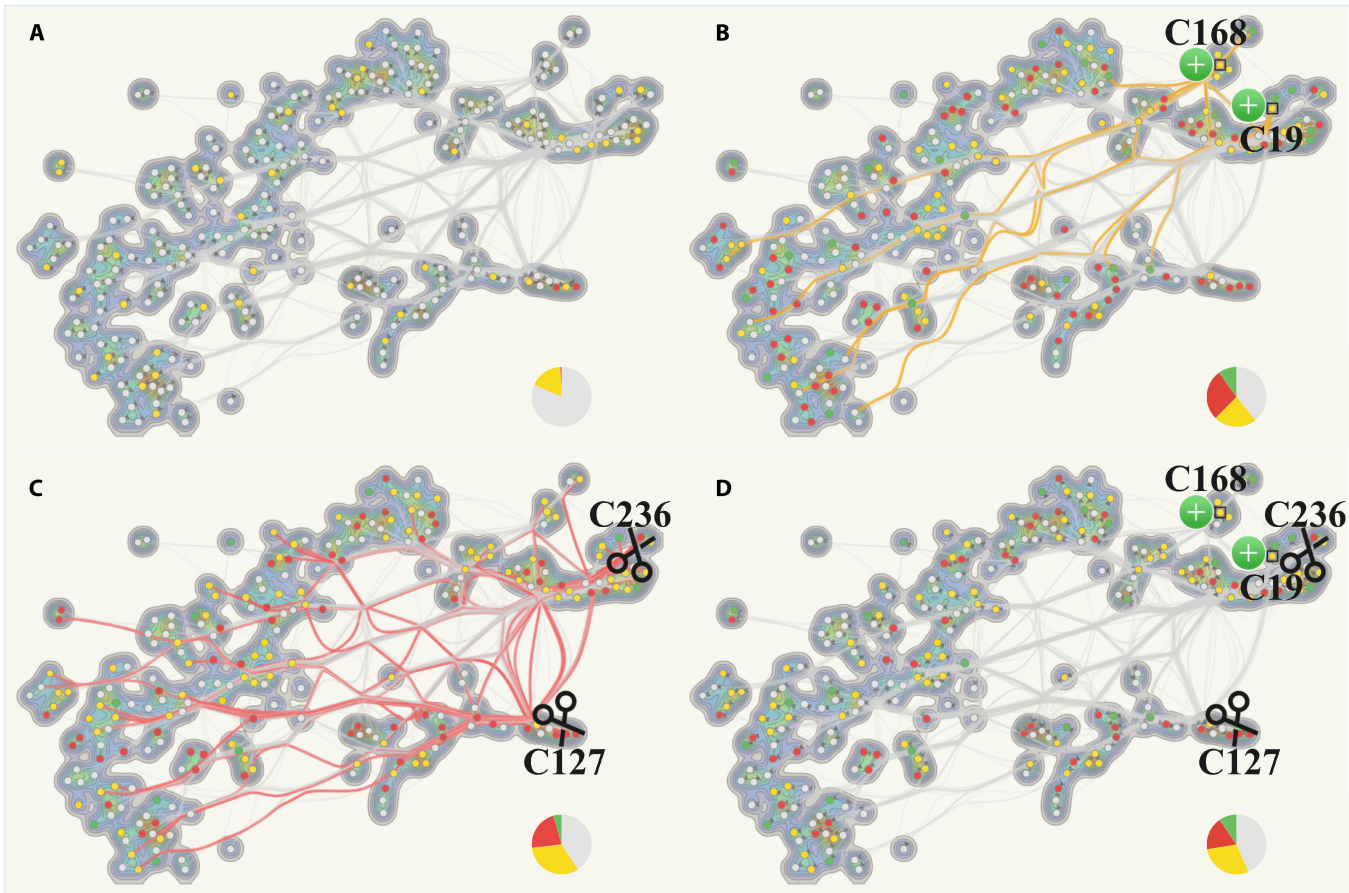

Fig. 7. Case study on real-world networked-loans. (A) Initial SEIR states. (B to D) Final SEIR states after using different risk mitigation methods (restoration, isolation, and both). Gray, susceptible; yellow, exposed; red, infectious; green, recovered.

Downloaded from https://spj.science.org on March 09, 2026

customized t-SNE, the system clusters nodes by risk label without disrupting the network topology, enhancing the efficiency of risk monitoring and mitigation. In our system, nodes are colored by default status, as shown in Fig. 2, and a pie chart displays the proportions of the 4 SEIR states. Using this system, we conducted a case study that focuses on a real-world networked-loan consisting of 250 companies, 500 guarantee contracts, and $1.7 billion in loans.

Figure 7A shows the status of the network on 2015 May 3, with only 2 companies infectious and 44 exposed, suggesting a healthy network. We used FinDoctor to predict the status of the network based on the events at the next timestamp (2015 May 31). Our prediction demonstrates that 83 companies are infectious and 68 exposed, covering 73.5% of the loan amount. The variation between these 2 points yields an R_0 of 1.2, indicating that risk contagion could escalate into a full-blown systemic crisis without timely mitigation.

The first mitigation method, called restoration, provides financial subsidies or postpones expiration dates for exposed individuals to avoid infection. Latent infections are not contagious until a default occurs, and the risk quickly spreads to neighboring nodes. As shown in Fig. 7B, nodes C19 and C168, which were predicted to transition into the exposed state, have the largest number of direct connections to other nodes (guarantors), as shown by yellow lines. The simulation results of remediating these 2 companies show a considerable reduction in R_0, saving 14 companies from early infection, reducing the overall default rate by 8.7%, and increasing total loan repayments to US$15.1 million.

The second mitigation method is isolation, which means severing all relationships involving infectious individuals. Given the network's ability to amplify infections, we focused on the contagion paths of infectious nodes. After traversing the risk contagion paths of all predicted infectious nodes by breadth-first search, we identified nodes C127 and C236 as those with the longest paths and the highest total loan amounts, as shown by red lines in Fig. 7C. In our simulation, quarantining these 2 firms saved 27 firms from default, reduced the overall default rate by 10.8%, and brought the total loan repayment to US$18.7 million.

Figure 7D depicts encouraging simulated results where both methods were applied, saving 38 companies from default, reducing the overall default rate by 15.2%, and enabling the repayment of US$22.8 million in loans, achieving a more optimal result than using either intervention alone. This case study validates the effectiveness of our approach in predicting and stemming risk contagion.

Conclusion

This paper introduces FinDoctor, a method to identify financial risk contagion. Our model seamlessly integrates advanced artificial intelligence with domain expertise. The enhancer uses knowledge distillation to guide LLMs and pretrained LMs in generating semantically enriched embeddings, capturing subtle financial risk indicators and terms. The encoder utilizes these embeddings to model risk contagion dynamics across networked-loans. Based on the SEIR epidemic model and predictions, FinDoctor computes R_0, a critical metric for contagion risk assessment, enabling proactive interventions and reducing cascading defaults. Through rigorous validation, FinDoctor has proven its accuracy and effectiveness in predicting and mitigating risk contagion. Our collaborative effort with financial experts and regulators underpins the model's development, supporting the growth of SMEs globally, ensuring financial market stability, and advancing the goals of United Nations Sustainable Development Goal 8.

Acknowledgments

Funding: The authors acknowledge that they did not receive funding for this work.
Author contributions: M.L. conducted and designed the experiments and contributed to the writing of the manuscript. L.Z. conducted the ablation experiments, prepared the datasets, and contributed to the writing of the manuscript. J.C. and S.Z. researched the literature and reviewed the manuscript. Z.N. contributed to the conception of the study and guided it throughout. X.L. guided the study throughout and provided experimental equipment.
Competing interests: The authors declare that they have no competing interests.

Data Availability

Dataset 2 is available in the Wind Economic Database (https://www.wind.com.cn/). Due to legal constraints, we are unable to provide the complete raw Dataset 1 in this submission. A nondisclosure agreement prevents us from providing details about the companies involved.

References

1. Jian M, Xu M. Determinants of the guarantee circles: The case of Chinese listed firms. *Pac Basin Financ J*. 2012;20(1):78–100.
2. He Q, Liu J, Gan J, Qian Z. Systemic financial risk and macroeconomic activity in China. *J Econ Bus*. 2019;102:57–63.
3. Gai P, Kapadia S. Contagion in financial networks. *Proc R Soc A Math Phys Eng Sci*. 2010;466(2120):2401–2423.
4. Ivashina V, Scharfstein D. Bank lending during the financial crisis of 2008. *J Financ Econ*. 2010;97(3):319–338.
5. Gai P, Haldane A, Kapadia S. Complexity, concentration and contagion. *J Monet Econ*. 2011;58(5):453–470.
6. Niu Z, Li R, Wu J, Cheng D, Zhang J. iConViz: Interactive visual exploration of the default contagion risk of networked-guarantee loans. In: *2020 IEEE conference on visual analytics science and technology (VAST)*. Salt Lake City (UT): IEEE; 2020. p. 84–94.
7. Niu Z, Cheng D, Zhang L, Zhang J. Visual analytics for networked-guarantee loans risk management. In: *2018 IEEE pacific visualization symposium (PacificVis)*. Los Alamitos (CA): IEEE Computer Society; 2018. p. 160–169.
8. Nagurney A. Networks in finance. In: Seese D, Weinhardt C, Schlottmann F, editors. *Handbook on information technology in finance*. Berlin, Heidelberg (Germany): Springer Berlin Heidelberg; 2008. p. 383–419.
9. Cheng D, Zou Y, Xiang S, Jiang C. Graph neural networks for financial fraud detection: A review. *Front Comp Sci*. 2025;19(9):Article 199609.
10. Xu B, Shen H, Sun B, An R, Cao Q, Cheng X. Towards consumer loan fraud detection: Graph neural networks with role-constrained conditional random field. In: *Proceedings of the AAAI conference on artificial intelligence*. Vol. 35. 5. Virtual Event: AAAI Press; 2021. p. 4537–4545.

Downloaded from https://spj.science.org on March 09, 2026

11. Cheng D, Niu Z, Zhang L. Delinquent events prediction in temporal networked-guarantee loans. *IEEE Trans Neural Networks Learn Syst*. 2023;34(4):1692–1704.
12. Lu Z, Li T, Zhang J, Liu M, Li X, Cui L, Chen J, Niu Z. RisQNet: Rescuing SMEs from financial shocks with a novel networked-loan risk assessment. In: *Proceedings of the thirty third international joint conference on artificial intelligence. IJCAI '24*. Jeju (Korea): IJCAI; 2024. p. 7385–7393.
13. Cheng D, Niu Z, Zhang Y. Contagious chain risk rating for networked-guarantee loans. In: *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining. KDD '20*. Virtual Event (CA): Association for Computing Machinery; 2020. p. 2715–2723.
14. Kostylenko O, Rodrigues HS, Torres DFM. Banking risk as an epidemiological model: An optimal control approach. In: Vaz AIF, Almeida JP, Oliveira JF, Pinto AA, editors. *Operational research*. Cham: Springer International Publishing; 2018. p. 165–176.
15. Bucci A, LaTorre D, Liuzzi D, Marsiglio S. Financial contagion and economic development: An epidemiological approach. *J Econ Behav Organ*. 2019;162:211–228.
16. Sui X, Wen H, Gao J, Lu S. Research on contagion and the influencing factors of personal credit risk based on a complex network. *Discret Dyn Nat Soc*. 2022;2022(1):4730479.
17. Van den Driessche P. Reproduction numbers of infectious disease models. *Infectious Dis Model*. 2017;2(3):288–303.
18. Xu P, Yu X. Research on the application of risk contagion model of mutual guarantee financing for SMEs clusters. *Account Res*. 2018;1:82–88.
19. Giudici P, Sarlin P, Spelta A. The interconnected nature of financial systems: Direct and common exposures. *J Bank Financ*. 2020;112:Article 105149.
20. Zhao C, Li M, Wang J, Ma S. The mechanism of credit risk contagion among internet P2P lending platforms based on a SEIR model with time-lag. *Res Int Bus Financ*. 2021;57:Article 101407.
21. He X, Bresson X, Laurent T, Perold A, LeCun Y, Hooi B. Harnessing explanations: LLM-to LM interpreter for enhanced text-attributed graph representation learning. In: *The twelfth international conference on learning representations*. Vienna (Austria): http://OpenReview.net; 2023.
22. Liang T, Zeng G, Zhong Q, Chi J, Feng J, Ao X, Tang J. Credit risk and limits forecasting in e-commerce consumer lending service via multi-view-aware mixture-of-experts nets. In: *Proceedings of the 14th ACM international conference on web search and data mining*. WSDM '21. Virtual Event (Israel): Association for Computing Machinery; 2021. p. 229–237.
23. Liu Z, Huang D, Huang K, Li Z, Zhao J. FinBERT: A pre-trained financial language representation model for financial text mining. In: *Proceedings of the twenty-ninth international joint conference on artificial intelligence, IJCAI-20*. Bessiere C, editor. Special Track on AI in FinTech. Yokohama (Japan): International Joint Conferences on Artificial Intelligence Organization; 2020. p. 4513–4519.
24. Wei J, Wang X, Schuurmans D, Bosma M, Xia F, Chi E, Le QV, Zhou D. Chain-of-thought prompting elicits reasoning in large language models. In: Koyejo S, Mohamed S, Agarwal A, Belgrave D, Cho K, Oh A, editors. *Advances in neural information processing systems*. Vol. 35. Red Hook (NY): Curran Associates, Inc; 2022. p. 24824–24837.
25. Rossi E, Chamberlain B, Frasca F, Eynard D, Monti F, Bronstein M. Temporal graph networks for deep learning on dynamic graphs. arXiv. 2020. https://doi.org/10.48550/arXiv.2006.10637
26. Cong W, Zhang S, Kang J, Yuan B, Wu H, Zhou X, Tong H, Mahdavi M. Do we really need complicated model architectures for temporal networks? In: *The eleventh international conference on learning representations*. Kigali (Rwanda): http://OpenReview.net; 2023.
27. Bei Y, Xu H, Zhou S, Chi H, Wang H, Zhang M, Li Z, Bu J. CPDG: A contrastive pre-training method for dynamic graph neural networks. In: *2024 IEEE 40th international conference on data engineering (ICDE)*. Utrecht (Netherlands): IEEE; 2024. p. 1199–1212.
28. Kazemi SM, Goel R, Jain K, Kobyzev I, Sethi A, Forsyth P, Poupart P. Representation learning for dynamic graphs: A survey. *J Mach Learn Res*. 2020;21(70):1–73.
29. Xu D, Ruan C, Korpeoglu E, Kumar S, Achan K. Inductive representation learning on temporal graphs. In: *International conference on learning representations*. Addis Ababa (Ethiopia): http://OpenReview.net; 2020.
30. Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, Kaiser L, Polosukhin I. Attention is all you need. In: *Proceedings of the 31st international conference on neural information processing systems*. NIPS'17. Long Beach (CA): Curran Associates Inc; 2017. p. 6000–6010.
31. Nguyen VH, Sugiyama K, Nakov P, Kan MY. FANG: Leveraging social context for fake news detection using graph representation. *Commun ACM*. 2022;65:124–132.
32. Chen Z, Mao H, Li H, Jin W, Wen H, Wei X, Wang S, Yin D, Fan W, Liu H, et al. Exploring the potential of large language models (LLMs) in learning on graphs. *ACM SIGKDD Explor Newsl*. 2024;25(2):42–61.
33. Xu J, Wu Z, Lin M, Zhang X, Wang S. LLM and GNN are complementary: Distilling LLM for multimodal graph learning. arXiv. 2024. https://doi.org/10.48550/arXiv.2406.01032
34. Li J, Han Z, Cheng H, Su J, Wang P, Zhang J, Pan L. Predicting path failure in time-evolving graphs. In: *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining. KDD '19*. Anchorage (AK): Association for Computing Machinery; 2019. p. 1279–1289.
35. Pareja A, Domeniconi G, Chen J, Ma T, Suzumura T, Kanezashi H, Kaler T, Schardl T, Leiserson C. EvolveGCN: Evolving graph convolutional networks for dynamic graphs. In: *Proceedings of the AAAI conference on artificial intelligence*. Vol. 34. 04. New York (NY): AAAI Press; 2020. p. 5363–5370.
36. Trivedi R, Dai H, Wang Y, Song L. Know-Evolve: Deep temporal reasoning for dynamic knowledge graphs. In: *Proceedings of the 34th international conference on machine learning-volume 70. ICML'17*. Sydney (Australia): JMLR.org; 2017. p. 3462–3471.
37. Ma Y, Guo Z, Ren Z, Tang J, Yin D. Streaming graph neural networks. In: *Proceedings of the 43rd International ACM SIGIR conference on research and development in information retrieval. SIGIR'20*. Virtual Event (China): Association for Computing Machinery; 2020. p. 719–728.
38. Wang Y, Chang YY, Liu Y, Leskovec J, Li P. Inductive representation learning in temporal networks via causal anonymous walks. In: *International conference on learning representations*. Vienna (Austria); http://OpenReview.net; 2021.
39. Ke G, Meng Q, Finley T, Wang T, Chen W, Ma W, Ye Q, Liu TY. LightGBM: A highly efficient gradient boosting decision tree. In: *Proceedings of the 31st international conference on neural*

Downloaded from https://spj.science.org on March 09, 2026

information processing systems. NIPS'17. Long Beach (CA): Curran Associates Inc; 2017. p. 3149–3157.
40. Chen T, Guestrin C. XGBoost: A Scalable tree boosting system. In: *Proceedings of the 22nd ACMSIGKDD international conference on knowledge discovery and data mining. KDD'16*. San Francisco (CA): Association for Computing Machinery; 2016. p. 785–794.
41. Kipf TN, Welling M. Semi-supervised classification with graph convolutional networks. In: *International conference on learning representations*. Toulon (France): http://OpenReview.net; 2017.
42. Veličković P, Cucurull G, Casanova A, Romero A, Liò P, Bengio Y. Graph attention networks. In: *International conference on learning representations*. Vancouver Convention Center: Vancouver (Canada): http://OpenReview.net; 2018.
43. Sankar A, Wu Y, Gou L, Zhang W, Yang H. 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. WSDM'20*. Houston (TX): Association for Computing Machinery; 2020. p. 519–527.
44. Kumar S, Zhang X, Leskovec J. Predicting dynamic embedding trajectory in temporal interaction networks. In: *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining. KDD'19*. Anchorage (AK): Association for Computing Machinery; 2019. p. 1269–1278.
45. Reimers N, Gurevych I. Sentence-BERT: Sentence embeddings using Siamese BERT networks. In: Inui K, Jiang J, Ng V, Wan X, editors. *Proceedings of the 2019 conference on empirical methods in natural lan guage processing and the 9th international joint conference on natural language processing (EMNLP-IJCNLP)*. Hong Kong (China): Association for Computational Linguistics; 2019. p. 3982–3992.

Downloaded from https://spj.science.org on March 09, 2026

Intelligent Computing
A SCIENCE PARTNER JOURNAL

Identifying Financial Risk Contagion with Large Language Models and Temporal Graph Learning

Moyang Liu, Letian Zhao, Junqi Chen, Xiang Li, Shuo Zhang, and Zhibin Niu

Citation: Liu M, Zhao L, Chen J, Li X, Zhang S, Niu Z. Identifying Financial Risk Contagion with Large Language Models and Temporal Graph Learning. *Intell Comput*. 2026;**5**:0292. DOI: 10.34133/icomputing.0292

Fostering economic growth, particularly the development of small and medium-sized enterprises, is central to United Nations Sustainable Development Goal 8. Small and medium-sized enterprises can secure bank loans with the help of guarantors, forming interconnected networked-loans. While essential for access to credit, networked-loans can also amplify risk contagion and thus are vulnerable to epidemic financial crises. Previous research has failed to capture nuanced financial semantics and analyze the mechanisms of risk contagion. We designed FinDoctor to tackle these challenges. FinDoctor leverages large language models to extract context-aware features from unstructured financial text. It then combines temporal graph neural networks with the susceptible, exposed, infectious, recovered epidemiological model to identify financial risk contagion. This enables both accurate prediction and targeted risk mitigation. Evaluated on 2 real-world datasets, FinDoctor achieves state-of-the-art performance, with area under the curve scores of 88% and 89%. We further develop an interactive application that visualizes risk propagation and supports targeted intervention. Our work provides a full-cycle solution to identify financial risk contagion in networked-loans.
Image

View the article online
https://spj.science.org/doi/10.34133/icomputing.0292

Downloaded from https://spj.science.org on March 09, 2026

Use of this article is subject to the Terms of service

Intelligent Computing (ISSN 2771-5892) is published by the American Association for the Advancement of Science. 1200 New York Avenue NW, Washington, DC 20005.

Copyright © 2026 Moyang Liu et al.

Exclusive licensee Zhejiang Lab. No claim to original U.S. Government Works. Distributed under a Creative Commons Attribution License (CC BY 4.0).