RESEARCH-ARTICLE

SSH-T3 : A Hierarchical Pre-training Framework for Multi-Scenario Financial Risk Assessment

ZEHAO GU, Fudan University, Shanghai, China

YATENG TANG, Tencent, Shenzhen, Guangdong, China

JIARONG XU, School of Management Fudan University, Shanghai, China

ZHANG SIWEI, Fudan University, Shanghai, China

XUEHAO ZHENG, Tencent, Shenzhen, Guangdong, China

XI CHEN, Fudan University, Shanghai, China

View all

Open Access Support provided by:

Fudan University

Tencent

School of Management Fudan University

Published: 10 November 2025

Citation in BibTeX format

CIKM '25: The 34th ACM International Conference on Information and Knowledge Management
November 10 - 14, 2025
Seoul, Republic of Korea

Conference Sponsors:
SIGWEB
SIGIR

SSH-T^3: A Hierarchical Pre-training Framework for Multi-Scenario Financial Risk Assessment

Zehao Gu*
Shanghai Key Laboratory of Data Science, College of Computer Science and Artificial Intelligence, Fudan University
Shanghai, China
guzh22@m.fudan.edu.cn

Yateng Tang*
Tencent Weixin Group
Shenzhen, China
fredyttang@tencent.com

Jiarong Xu
School of Management, Fudan University
Shanghai, China
jiarongxu@fudan.edu.cn

Siwei Zhang
Shanghai Key Laboratory of Data Science, College of Computer Science and Artificial Intelligence, Fudan University
Shanghai, China
swzhang24@m.fudan.edu.cn

Xuehao Zheng†
Tencent Weixin Group
Shenzhen, China
xuehaozheng@tencent.com

Xi Chen†
Yun Xiong
Shanghai Key Laboratory of Data Science, College of Computer Science and Artificial Intelligence, Fudan University
Shanghai, China
x_chen21@m.fudan.edu.cn
yunx@fudan.edu.cn

Abstract

Efficiently modeling user behavior on online payment platforms is crucial for accurately identifying potential financial risks. With the rapid growth of online payment platforms, the volume of user transaction data has significantly increased. Moreover, users' payment behaviors often encompass diverse activities and interactions across multiple scenarios. Based on observations from online payment platforms, we identify three key challenges: scarce labels and poor representation robustness, long user payment behavior sequences, and complex and heterogeneous amount-aware scenarios.

To address these challenges, we propose a novel Self-Supervised Hierarchical Two-Tower Transformer (SSH-T^3), specifically designed for multi-scenario financial risk assessments. We introduce a masked modeling pre-training task to reconstruct multi-scenario day-level transaction amount distributions, effectively mitigating behavior-level noise and enhancing representation robustness. Additionally, we propose a hierarchical Multi-Scenario Payment Behavior Sequence (MS-PBS) modeling approach tailored to business needs, which significantly reduces complexity while capturing user behavior patterns more effectively through day-level representations. Furthermore, we highlight the critical importance of correlating multi-scenario data in MS-PBS modeling to better identify defaulter patterns. To this end, we design a Two-Tower Transformer equipped with a specialized attention mechanism that captures intricate user patterns across scenarios. Extensive experiments conducted on both offline and online real-world business datasets demonstrate the effectiveness and applicability of SSH-T^3.

CCS Concepts

• **Computing methodologies** → **Temporal reasoning**.

Keywords

Financial Risk Assessment, Multi-scenario Long Sequence Modeling, Pre-training

ACM Reference Format:
Zehao Gu, Yateng Tang, Jiarong Xu, Siwei Zhang, Xuehao Zheng, Xi Chen, and Yun Xiong. 2025. SSH-T^3: A Hierarchical Pre-training Framework for Multi-Scenario Financial Risk Assessment. In *Proceedings of the 34th ACM International Conference on Information and Knowledge Management (CIKM '25), November 10–14, 2025, Seoul, Republic of Korea.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3746252.3761504

1 Introduction

With the rapid advancement of Internet payment technology, major online payment platforms have introduced a variety of inclusive financial services [13, 33, 38] that reach broader user bases than traditional institutions. However, this also meets challenges. The surge in users and low default costs heighten default risks in online inclusive finance [29, 37]. Widespread defaults could destabilize financial systems, necessitating urgent research on financial risk assessment methods to safeguard the modern financial ecosystem.

Financial risk assessment evaluates default likelihood via users' financial histories [3]. While statistical and machine learning-based methods have advanced in financial risk assessment [1, 15, 23], they rely heavily on explicit features like user loan repayments

*Equal Contribution.
†Corresponding author.

CIKM '25, Seoul, Republic of Korea.

ACM ISBN 979-8-4007-2040-6/2025/11
https://doi.org/10.1145/3746252.3761504

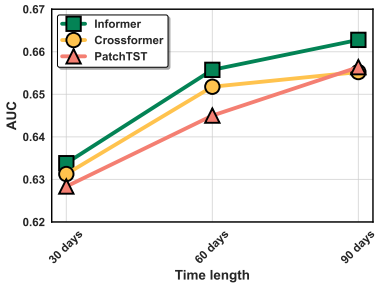

Figure 1: Model Performance of 3 selected Transformer-based models improves with increased input sequence length.

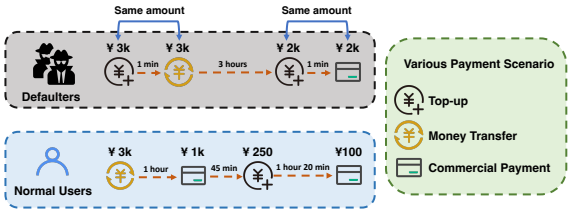

Figure 2: This illustration elucidates the divergence in online payment patterns between defaulters and normal users

and credit card usage. These data are often inaccessible to online payment platforms. Instead, these platforms leverage numerous payment behavior sequences across various scenarios collected by themselves. In our financial risk assessment business, we have access to comprehensive payment behavior sequences (PBS), which capture transaction details (e.g. time, channels, and card numbers). Furthermore, in our business, the PBS encompasses multiple scenarios (e.g. commercial payments, money transfers, top-ups, etc.), which reflect users' varied transactional behaviors. In other words, our PBS are Multi-Scenario Behavior Sequences (MS-PBS).

The MS-PBS implicitly reveals users' consumption cycles, spending habits, and creditworthiness, making them ideal for PBS modeling to enhance risk management and reduce default risks. Transformer-based PBS modeling methods have advanced domains like recommendation systems [36], advertising platforms [32], and risk management [18] by capturing dynamics and internal correlations of PBS. However, applying these methods to real-world MS-PBS in our business faces several challenges, which we outline below.

Challenge 1: Scarce Labels and poor representations robustness. In practical online payment scenarios, it's noteworthy that the proportion of labeled user payment sequences is quite limited, representing merely 0.02% [31]. Self-supervised learning [12, 30, 44] with pre-training task has been deployed to alleviate this scarcity, yet it is confined to the behavior-level, involving tasks such as masked behavior modeling or next behavior prediction. However, This approach overlooks the significant noise present in behavior sequences, yielding representations with less robustness.

Challenge 2: Long user payment behavior sequences. Methods based on Transformers [39] excel at processing sequence data. As shown in Fig. 1, these methods perform better with longer input sequences. However, the proliferation of PBS on online payment platforms is expanding rapidly, imposing quadratic time and storage complexity, overlooked in prior work [8, 24]. Some studies have tried to segment longer PBS, but each has had issues. Time series work [34, 50] uniformly split sequences into patches, unsuitable for MS-PBS's irregular inter-behavior intervals. Other works [20, 24] employ a defined webpage architecture to divide PBS, yet these methods are heavily reliant on expert knowledge.

Challenge 3: Heterogeneous amount-aware scenarios. MS-PBS owns heterogeneous payment scenarios. For instance, when remitting funds to supermarkets, opt for 'commercial payments,' and utilize 'money transfer' for trading within group chats. Previous methods simply integrate scenario-specific features without explicit modeling, which is insufficient: MS-PBS varies across different payment scenarios, and uniform modeling causes conflicting parameter gradient adjustments. Furthermore, as depicted in Fig. 2, scenarios are also directly linked to transaction amounts. Defaulters often make quick cross-scenario transfers due to detection concerns.

To this end, we engineer a Self-Supervised Hierarchical Two-Tower Transformer (SSH-T^3) specifically tailored for multi-scenario financial risk assessments. Our approach is a two-stage method including pre-training and fine-tuning. We employ masked modeling to reconstruct the day-level representation for pre-training, using the distribution of transaction amounts across each scenario as the reconstruction target. This target is more robust, and the transaction amounts align with our business requirements **(Addressed Challenge 1)**. For the downstream task, we fine-tune SSH-T^3 to estimate the users' default likelihood. Our model does not focus on the self-supervised task of behavior-level MS-PBS, as individual behaviors are inherently noisy and hard to align with our downstream targets. Instead, given the vast scale of MS-PBS, we adopt a hierarchical pre-training framework to effectively reduce the high complexity while using Transformers for sequence data **(Addressed Challenge 2)**. For behavior-level MS-PBS aggregation, we use our designed Two-Tower Multi-Scenario Transformer, which comprises a scenario tower and a behavior tower. The scenario tower extracts defaulter transaction patterns from amounts and scenario sequences. Its output signal highlights corresponding behavior representations. This interpretable design strengthens extraction of scenario heterogeneity and behavioral defaulter patterns **(Addressed Challenge 3)**. We highlight our key contributions:

- **Robust Representation Learning.** We design a masked modeling pre-training task to reconstruct multi-scenario day-level transaction amount distributions, transcending behavior-level noise for robust representation learning.
- **Low-complexity interpretable MS-PBS Modeling.** We propose a hierarchical MS-PBS modeling approach tailored to our business, which significantly decreases the complexity. Besides, day-level representation reflects user patterns.
- **Efficient Heterogeneous Scenario Capture.** We emphasize the critical role of correlating multi-scenario data in MS-PBS modeling and identifying defaulter patterns. We introduce a Two-Tower Transformer that leverages a specialized Attention mechanism to capture user patterns.
- **Experiments Findings.** Experiments on our 4 large-scale real-world business datasets demonstrate the effectiveness and applicability of our proposed method (SSH-T^3).

2 Related Work

2.1 Financial Risk Assessment

Financial risk assessment generally pertains to evaluating the risk exposure and potential default likelihood of individuals or corporations through an analysis of their financial traits, such as transactional behavior. Initially, financial risk assessments were primarily based on the financial health and fundamental characteristics of the enterprise [1].

It is evident that, with the surge in data volume, the incorporation of machine learning technology has become indispensable. This technology enhances the efficiency and precision of statistical methods without the need for restrictive assumptions [3]. Techniques such as decision trees [2] and support vector machines [23, 31] capitalize on comprehensive user feature data for their models. Recently, more approaches have employed deep learning to assess financial risk. These include identifying various forms of financial malfeasance, such as fraud [14, 21, 25–27, 42, 43], cash-out [17, 19], money laundering [18], failure to repay [40]. We mainly discuss defaulters who fail to repay.

2.2 Behavior Sequential Modeling

The modeling of behavior data has seen significant advancements and growth in applications such as risk management, recommendation systems, and fraud detection [4, 6, 7, 35, 46, 48, 49]. Within the field of time sequences analysis [5, 34, 41, 45, 50], CNN-based studies [41, 45] perform sequence modeling by capturing periodic patterns, while Transformer-based architectures [34, 50] mitigate complexity and handle long sequences through patching. However, these approaches are not suitable for our research. The payment behavior time intervals in MS-PBS vary significantly. Simple patching leads to significant semantic information discrepancies. In the recommendation systems domain, there is a growing industry focus on modeling long sequences of users to enhance goods and video recommendations. For instance, studies [10, 36, 52] have concentrated on leveraging self-supervised methods such as contrastive learning and Masked Autoencoders for long sequence recommendations. Meanwhile, [11, 30, 47] have delved into the nuances of different behaviors for sequence recommendation tasks. Unlike our work, these recommendation systems primarily aim to predict the next items. Their downstream tasks are different from ours, necessitating our exploration of alternative self-supervised tasks and methodologies for modeling across diverse scenarios.

3 Methodology

3.1 Problem Definition

Definition 1 (Multi-Scenario Payment Behavior Sequences (MS-PBS)). *Assume that a certain user has C payment behaviors on the online payment platform, each of which contains payment features (such as amount, card number, etc.) and corresponding payment scenarios (such as money transfers, commercial payments, etc.). We denote behavior sequences as $\boldsymbol{B}_u \in \mathbb{R}^{C\times f}$ and scenario sequences as $\boldsymbol{s}_u \in \mathbb{R}^{C}$. C and f denote the number of behaviors and features, respectively.*

Definition 2 (Financial Risk Assessment). *We denote the users set as $\mathcal{U} = \{u_1, u_2, \ldots, u_{|\mathcal{U}|}\}$, where $|\mathcal{U}|$ denotes the users count. Suppose there are $|\mathcal{S}|$ types of scenarios. The scenarios set denotes as $\mathcal{S} = \{S_1, S_2, \ldots, S_{|\mathcal{S}|}\}$. We hope to predict whether a user is a defaulter through his or her long MS-PBS. Precisely, our objective is to forecast ground-truth $y \in \{0, 1\}$ for a given specific user PBS $\boldsymbol{B}_u$ and the corresponding scenario sequence $\boldsymbol{s}_u$, where $y = 1$ indicates a defaulter and $y = 0$ indicates a normal user.*

3.2 Two-Tower Multi-Scenario Transformer

In the actual online payment transaction business, defaulters expedite the transfer of funds through alternative scenarios to evade scrutiny from regulatory platforms. However, current methodologies do not adequately account for the tactics commonly employed by defaulters, leading to a deficiency in modeling these behaviors. Moreover, the distribution of transaction amounts significantly differs across various scenarios. These practical challenges have driven us to propose a detailed module that captures the heterogeneity of transaction scenarios.

Hence, we introduce a novel Two-Tower Multi-Scenario Transformer to address this challenge. This framework comprises a scenario tower and a behavior tower, which model sequential dependencies with a focus on behavior level. The former extracts user payment patterns across various scenarios using Multi Amount-Aware Scenario (MAAS) Transformer blocks, while the latter focuses on behavioral dependency. We use sigmoid fusion to integrate the output of the scenario tower into the output of the behavior tower. The final output serves as a day-level representation, which is then used for our day-level pre-training tasks and financial risk assessment. The pipeline of our model is shown in Fig. 3.

3.2.1 Multi-feature Encoding. MS-PBS encompasses diverse features. Prior to feeding these into our model, it is essential that we first encode the features across various fields. Regarding the transactions, we have taken into account the amount, specific scenarios, types, and the cards utilized. Furthermore, we have incorporated hour-of-day temporal information and textual descriptions of the transactions, formulated as Eqn. (1), where, $\boldsymbol{H}_{trans}$, $\boldsymbol{H}_{text}$, and $\boldsymbol{H}_{h2d}$ denotes transaction embeddings, payment embeddings and hour-of-day temporal embeddings, respectively.

$$\boldsymbol{H}^{(0)} = Concat\,(\boldsymbol{H}_{trans},\ \boldsymbol{H}_{text},\ \boldsymbol{H}_{h2d}). \tag{1}$$

In our scenario, periodicity is a critical factor to consider, as users exhibit a tendency to make payments within the same hours across different days. However, the relative position encoding typically employed in sequence models fails to capture this periodic nature fully. Moreover, directly utilizing numerical signals as inputs does not resolve this issue. For instance, the times 23:00 p.m. and 0:00 a.m. are conceptually close, yet the numerical representation does not reflect this proximity. Consequently, we have adopted a specialized transaction time encoding method to address this challenge. The specific formula for this method is detailed in Eqn. (2). Here, L represents the length of a time period. For $\boldsymbol{H}_{h2d}$, we set L as 24.

$$\phi\ (t, L) = (cos(2\pi t/L), sin(2\pi t/L)), \quad \boldsymbol{H}_{h2d}[i] = \phi\ (t[i], L). \tag{2}$$

We use the vanilla Transformer [39] to generate $\boldsymbol{H}_{text}$. For $\boldsymbol{H}_{trans}$, the amount (denoted as $\boldsymbol{A}$) is a numerical feature. Transaction features such as type, scenario, and card, which are categorical, are converted into embeddings using three distinct lookup matrices

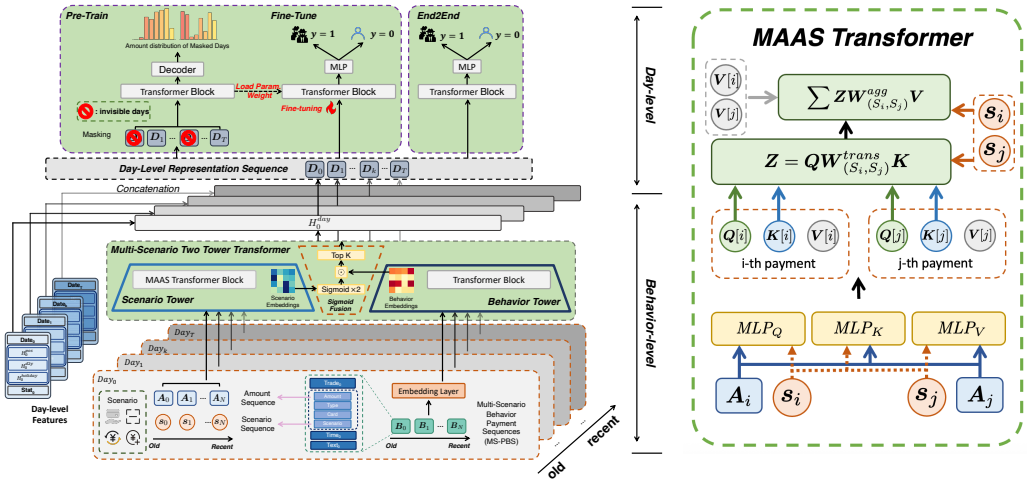

Figure 3: (Left) The SSH-T^3 pipeline processes behavior and scenario sequences using a Multi-Scenario Two-Tower Transformer for heterogeneous scenario modeling. It is enhanced by hierarchical pre-training for robust, noiseless day-level representations. The goal of the model is to detect defaulters. (Right) MAAS Transformer Block.

$\boldsymbol{E}_{ty} \in \mathbb{R}^{|Ty|\times d_{ty}}$, $\boldsymbol{E}_{sc} \in \mathbb{R}^{|Sc|\times d_{sc}}$, and $\boldsymbol{E}_{ca} \in \mathbb{R}^{|Ca|\times d_{ca}}$, respectively. $\boldsymbol{H}_{trans}$ is the concatenation of above four features.

3.2.2 Scenario Transformer Tower. Defaulters frequently engage in rapid fund transfers of identical amounts across two different scenarios. However, when employing a vanilla Transformer block, these special amount-aware patterns tend to be overlooked. Consequently, we propose the *MAAS Transformer Block* in the scenario tower. We take the numerical transaction amount sequences $\boldsymbol{A} \in \mathbb{R}^N$ and its corresponding scenario sequences $\boldsymbol{s} \in \mathbb{R}^N$ within a certain day as the input of this module. $\boldsymbol{A}$ has undergone a logarithmic transformation. Here, N refers to daily transaction counts.

For **Queries**, **Keys**, and **Values** utilized in the self-attention calculation, we use different Multi-Layer Perceptrons (MLP) to map amount sequences $\boldsymbol{A}$ (denoted as $\boldsymbol{A}^{(0)}$) into the Embedding representation of different scenarios respectively, which reflects our consideration of heterogeneous scenarios. Specific Details are shown in Eqn. (3). Here, MLPs are determined by $s_u[i]$, which represents a corresponding scenario, $*$ can be $\boldsymbol{Q}$, $\boldsymbol{K}$ or $\boldsymbol{V}$. In other words, for $|\mathcal{S}|$ distinct scenarios, we set $|\mathcal{S}|$ different MLPs respectively. $\boldsymbol{A}_u^{(l-1)}[i]$ represents the output corresponding to the i-th payment behavior at the $(l-1)$-th layer.

$$*_u^{(l)}[i] = MLP_{*,s_u[i]}(\boldsymbol{A}_u^{(l-1)}[i], s_u[i]), \tag{3}$$

Instead of computing the dot product between **Queries** and **Keys**, we set a pair-wise transfer matrix $\boldsymbol{W}^{trans} \in \mathbb{R}^{(|\mathcal{S}|\times|\mathcal{S}|)\times d\times d}$. This matrix enables us to capture the inter-correlations within two scenarios, explicitly revealing the patterns of users employing diverse scenarios for swift fund transfers. Similarly, when focusing on modeling relationships within more than two scenarios, we can achieve this by stacking multiple transfer matrices. The detailed formula is shown in Eqn. (4):

$$\boldsymbol{Z}^{(l)}_{(i,\,j)} = (\boldsymbol{Q}_u^{(l)}[i]\,\boldsymbol{W}^{trans}_{(S_i,\,S_j)}\,(\boldsymbol{K}_u^{(l)}[j])^T)/\sqrt{d}. \tag{4}$$

$\boldsymbol{Z}^{(l)} \in \mathbb{R}^{(|\mathcal{S}|\times|\mathcal{S}|)\times N\times N}$ stands for the attention map of all scenario-pairs. Similar to the transform matrix, we set a pair-wise aggregation matrix $\boldsymbol{W}^{agg} \in \mathbb{R}^{(|\mathcal{S}|\times|\mathcal{S}|)\times d\times d}$. This matrix is used to aggregate all transactions to obtain the representation of the entire sequence finally. The aggregation operation is shown in Eqn. (5).

$$\boldsymbol{A}^{(l)}_{(i,\,j)} = \boldsymbol{Z}^{(l)}_{(i,\,j)}\,\boldsymbol{W}^{agg}_{(S_i,\,S_j)}\,\boldsymbol{V}_u^{(l)}[j], \quad \boldsymbol{A}_u^{(l)}[i] = \sum_{j=1}^{N} \boldsymbol{A}^{(l)}_{(i,\,j)}. \tag{5}$$

3.2.3 Behavior Transformer Tower. While emphasizing the amount-aware heterogeneity within MS-PBS, it is imperative to enhance the feature richness of individual behaviors for sequence modeling. To accomplish this, we employ Behavior Transformer Tower, which not only mitigates potential noise from the scenario tower but also bolsters the robustness of our pipeline. For the input embedding $\boldsymbol{H}^{(0)}$, we adopt the multi-head self-attention mechanism to model the sequence dependency. The operation is described in Eqn. (6). Here, $\boldsymbol{W}_t^Q, \boldsymbol{W}_t^K, \boldsymbol{W}_t^V$ are learnable parameter matrix to generate **Queries**, **Keys**, and **Values** respectively. $\boldsymbol{H}_t$ is the behavior representations of a specific head. The block also including residual connection, FFN and layer-norm.

$$\begin{aligned}
\boldsymbol{H}_t^{(l)} &= Softmax(\boldsymbol{Q}_t(\boldsymbol{K}_t)^T/\sqrt{d_{h_1}})\boldsymbol{V}_t.\\
\boldsymbol{Q}_t, \boldsymbol{K}_t, \boldsymbol{V}_t &= (\boldsymbol{H}_t^{(l-1)})^T\boldsymbol{W}_t^Q, (\boldsymbol{H}_t^{(l-1)})^T\boldsymbol{W}_t^K, (\boldsymbol{H}_t^{(l-1)})^T\boldsymbol{W}_t^V.\\
\boldsymbol{H}^{(l)} &= \textbf{Concat}\,(\boldsymbol{H}_1^{(l)}, \boldsymbol{H}_2^{(l)}, \ldots, \boldsymbol{H}_{h_1}^{(l)}).
\end{aligned} \tag{6}$$

3.2.4 Two-Tower Transformer Fusion. To integrate the amount-aware scenario signals derived from the Scenario Tower with the behavior representations from the Behavior Tower, we employ a Sigmoid fusion. This involves mapping the scenario signals onto the interval (0, 2) [22] and subsequently engaging in element-wise multiplication with the behavior representations. Our focus is on identifying the behavior patterns of defaulters, particularly the

rapid transfer of identical funds across various scenarios. By highlighting the pertinent transaction pairs through this method, our model is directed to allocate greater attention to these critical behavior patterns, thereby enhancing its ability to discern defaulters from such sequences. The specific formula can be found in Eqn. (7). $\boldsymbol{A}^{(l)}$ and $\boldsymbol{H}^{(l)}$ represent the outputs from the Scenario Tower and Behavior Tower, respectively.

$$\boldsymbol{A}^{(l)} = 2 * Sigmoid\,(\boldsymbol{A}^{(l)}), \quad \boldsymbol{H}^{(l)} = \boldsymbol{H}^{(l)} \odot \boldsymbol{A}^{(l)}. \tag{7}$$

Ultimately, we consolidate the integrated representations to form a day-level representation, which is shown in Eqn. (8). For this aggregation, we opt for the Top-k method. Alternatives such as average pooling and max pooling could also be utilized to achieve this consolidation. $\boldsymbol{H}^{day} \in \mathbb{R}^d$ denotes the day-level representation of a certain day. $\boldsymbol{D} \in \mathbb{R}^{T\times d}$ denotes a day-level sequence.

$$\boldsymbol{H}^{day} = Topk\,(\boldsymbol{H}^{(l)}), \quad \boldsymbol{D} = \{\boldsymbol{H}_1^{day}, \boldsymbol{H}_2^{day}, \ldots, \boldsymbol{H}_T^{day}\}. \tag{8}$$

3.3 Hierarchical MS-PBS Pre-training

As mentioned in the introduction, pre-training tasks based on behavior-level sequences often ignore the high noise and low semantic information of individual behaviors. In addition, predicting the next behavior or reconstructing a masked individual behavior is difficult to align with our downstream tasks. Therefore, we designed a hierarchical MS-PBS pre-training framework. The Two-Tower Transformer is used to process the representation of the behavior level, and we use the representation of the day level for pre-training tasks, that is, to reconstruct the amount distribution of each scenario every day.

We obtain day-level sequences $\boldsymbol{D} \in \mathbb{R}^{T\times d}$ as the output of the Two-Tower Transformer. In order to better describe the day-level representation, we incorporate some other features that are unique to the day, such as the date feature representing the day and some statistical features of transactions within the day. The detailed process is shown in Eqn. (9). $\boldsymbol{D}_{stat}$ and $\boldsymbol{D}_{date}$ are statistics features and date features respectively. For $\boldsymbol{D}_{date}$, we consider the relative position in the entire day-level MS-PBS $\boldsymbol{H}_{pos}$, $\boldsymbol{H}_{d2y}$, and holiday encoding features $\boldsymbol{H}_{holiday}$. Similar to $\boldsymbol{H}_{h2d}$, L is set to 365 here.

$$\boldsymbol{D}^{(0)} = Concat\,(\boldsymbol{D}, \boldsymbol{D}_{stat}, \boldsymbol{D}_{date}). \tag{9}$$

3.3.1 Masked modeling. Masked autoencoders have demonstrated their effectiveness as a robust backbone for self-supervised learning in both natural language processing (NLP) [9] and computer vision (CV) [16]. Inspired by this, we employ a method that involves masking sequences of day-level tokens and reconstructing them for self-supervised pre-training. Notably, the representations of day-level tokens learned through our approach align more closely with real-world business and downstream tasks. These day-level representations reflect the periodic payment patterns of each user. In our downstream tasks, we aim to assess whether a user is at risk of default based on long-term MS-PBS. If we bypass our hierarchical design and directly apply masked modeling to behavior-level tokens, the model is likely to learn significant noise. Furthermore, predicting the next behavior or reconstructing individual transactions may not be suitable for our downstream objectives. Our hierarchical MS-PBS pre-training exhibits enhanced robustness.

Specifically, we randomly set $T_\alpha = \lfloor T \times \alpha \rfloor$ days to be masked. Here, α is a hyper-parameter which stands for the masking ratio. Suppose $\mathcal{M}$ as the masked index matrix and the input sequences will be $\boldsymbol{D}^{(0)} = \boldsymbol{D}^{(0)} \odot \mathcal{M}$. Then, we use the Transformer blocks to extract day-level dependencies. The detailed formula is shown in Eqn. (10). Here, $d_{h_2} = d/h_2$. $\boldsymbol{W}_g^Q, \boldsymbol{W}_g^K, \boldsymbol{W}_g^V$ are learnable parameter matrix to generate **Queries**, **Keys**, and **Values** respectively.

$$\begin{aligned}
&\boldsymbol{D}_g^{(l)} = Softmax(\boldsymbol{Q}_g(\boldsymbol{K}_g)^T/\sqrt{d_{h_2}})\boldsymbol{V}_g.\\
&\boldsymbol{Q}_g, \boldsymbol{K}_g, \boldsymbol{V}_g = (\boldsymbol{D}_g^{(l-1)})^T\boldsymbol{W}_g^Q, (\boldsymbol{D}_g^{(l-1)})^T\boldsymbol{W}_g^K, (\boldsymbol{D}_g^{(l-1)})^T\boldsymbol{W}_g^V.\\
&\boldsymbol{D}^{(l)} = Concat\,(\boldsymbol{D}_1^{(l)}, \boldsymbol{D}_2^{(l)}, \ldots, \boldsymbol{D}_{h_2}^{(l)}).
\end{aligned} \tag{10}$$

We utilize an MLP as our decoder, mapping the output of encoder $\boldsymbol{D}^{(l)}$ into $|\mathcal{S}|$ dimensions to represent the distribution of transaction amounts of a user on invisible days. The specific formula is shown in Eqn. (11). $\boldsymbol{W}_1, \boldsymbol{W}_2, \boldsymbol{b}_1, \boldsymbol{b}_2$ are learnable parameters of the MLP. It is important to note that we only reconstruct and calculate the loss for the masked days. Regarding soft labels, we employ the normalized transaction distributions of $|\mathcal{S}|$ scenarios during the masked days, scaling them to the range $[0, 1]$. $\hat{\boldsymbol{D}}$ stands for the amount distribution of masked days in different scenarios. We use the Kullback-Leibler divergence loss function $\mathcal{L}_{kl}$ to represent the reconstruction loss, which is shown in Eqn. (12).

$$p_\theta(\hat{\boldsymbol{D}}[i,j]|\boldsymbol{B}, \boldsymbol{A}, \boldsymbol{s}) = Softmax\,(\boldsymbol{W}_1\sigma(\boldsymbol{W}_2\boldsymbol{D}^{(l)} + \boldsymbol{b}_1) + \boldsymbol{b}_2). \tag{11}$$

$$\min_\theta \mathcal{L}_{kl} = \min_\theta \frac{1}{T_\alpha}\frac{1}{|\mathcal{S}|}\sum_{i=1}^{T_\alpha}\sum_{j=1}^{|\mathcal{S}|}\left(\hat{\boldsymbol{D}}[i,j]\log\left(\frac{\hat{\boldsymbol{D}}[i,j]}{p_\theta(\hat{\boldsymbol{D}}[i,j]|\boldsymbol{B}, \boldsymbol{A}, \boldsymbol{s})}\right)\right). \tag{12}$$

3.3.2 Training and Inference. While pre-training, our model is trained by minimizing the loss function shown in Eqn. (12). For inference, when processing day-level sequences, we no longer set any [MASK] tokens and remove the Decoder. Instead, we directly predict the likelihood of a user defaulting based on the input long-term MS-PBS $\boldsymbol{B}$ as well as its corresponding scenario sequences $\boldsymbol{s}$ and amount sequences $\boldsymbol{A}$. Specifically, our loss function is as shown in Eqn (13). y_{pred} and $\hat{y}$ stands for the prediction of the model and the ground truth, respectively. N_t denotes the represents the number of labeled training MS-PBS.

$$\mathcal{L} = -\frac{1}{N_t}\sum_{i=1}^{N_t}(\hat{y}_i\ log(y_{pred,i}) + (1-\hat{y}_i)\ log(1-y_{pred,i})). \tag{13}$$

4 Experiments

4.1 Experiments Settings

4.1.1 Dataset Description. We collect three real-world datasets for offline experiments (Small, Medium and Large) and another dataset for online simulation experiments[1], sourced from Tencent Mobile Payment, while strictly adhering to security and privacy policies.

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

Table 1: Model Performance Comparison. The performance is evaluated using Area Under the Curve (AUC) , Kolmogorov Smirnov (KS), and Recall at 10% (Recall@10). These metrics collectively reflect both the discriminative ability and the recall rate of the highest-risk individuals. The best results are with a Red background.

Type	Methods	Small (60 days)			Medium (90 days)			Large (90 days)		
		AUC↑	KS↑	Recall@10↑	AUC↑	KS↑	Recall@10↑	AUC↑	KS↑	Recall@10↑
End2End	Transformer	0.6582	0.2287	0.2597	0.6477	0.2104	0.2456	0.6611	0.2364	0.2574
	iTransformer	0.6560	0.2251	0.2542	0.6521	0.2157	0.2520	0.6505	0.2151	0.2502
	Informer	0.6557	0.2222	0.2582	0.6628	0.2401	0.2674	0.6543	0.2221	0.2571
	Crossformer	0.6518	0.2170	0.2458	0.6552	0.2258	0.2598	0.6515	0.2187	0.2484
	PatchTST	0.6450	0.2090	0.2493	0.6564	0.2254	0.2542	0.6404	0.2046	0.2365
Pre-training	BERT4Rec	0.6415	0.2021	0.2381	0.5990	0.1458	0.1903	0.6387	0.2015	0.2392
	S^3-Rec	0.6489	0.2179	0.2432	0.6168	0.1723	0.2138	0.6427	0.2035	0.2373
	CBiT	0.6001	0.1466	0.1992	0.6084	0.1600	0.2021	0.6394	0.2037	0.2391
SSH-T^3	End2End	0.6650	0.2366	0.2671	0.6762	0.2567	0.2813	0.6742	0.2551	0.2773
	Pre-train	0.6842	0.2683	0.2857	0.6977	0.2869	0.3094	0.6904	0.2742	0.3009
	Improve	+4.0%	+17.3%	+10.0%	+5.3%	+26.1%	+15.7%	+4.4%	+16.0%	+16.9%

All datasets include 8 scenarios. In accordance with the company's policy, we only provide examples of these scenarios (e.g., Money Transfer, Top-up). Small Dataset has a 60-day time window while Medium and Large has a 90-day time window.

4.1.2 Baselines. To better assess the performance of SSH-T^3, we compare it with 8 Transformer-based baselines: i) End-to-end sequential modeling methods: Transformer [39], Informer [51], Crossformer [50], PatchTST [34], iTransformer [28]. ii) Pre-trained Sequential Modeling Methods: Bert4Rec [36], S^3-rec [52], CBiT [10].

4.1.3 Implementation Details. Experiments run on a 16×V100 GPU server (up to 4 GPUs per experiment), with learning rates: 2e-4 for pre-training, 1e-4 for fine-tuning and end-to-end training. AdamW optimizer is used. The model dimension and the batch size for the offline Large dataset and the online dataset is 128. For the offline Small and Medium datasets, the batch size is 64. For the online experiments, we use pre-trained SSH-T^3 on the Large offline dataset and inference on our online deployment platforms.

4.2 Performance Validation

Tab. 1 presents the overall performance results of SSH-T^3 and other baselines. SSH-T^3 outperforms baselines in both End2End and the self-supervised framework, with an average 4.56% AUC improvement. We attribute this enhancement to several factors: **i)** Our hierarchical modeling framework models the behavior-level MS-PBS and day-level MS-PBS. The latter reflects the user's payment pattern aligned with our real business and the downstream task, whereas baselines solely rely on noisy behavior-level MS-PBS. **ii)** Our Two-Tower Multi-Scenario Transformer captures defaulters' special payment signals via the MAAS Transformer, which models the distribution of amounts under various scenarios by the amount-aware self-attention. During the fusion stage, the signals generated by the scenario tower are amplified and integrated into the behavior tower. **iii)** Our self-supervised representation learning also bolsters model prediction performance. This is because positive samples are sparse in financial risk assessment. The pre-training task we designed alleviates this difficulty and enhances robustness.

Table 2: Results of ablation study.

Model	AUC↑	KS↑	Recall@10↑
w/o MAAS	0.6592	0.2280	0.2666
w/o TE	0.6551	0.2229	0.2550
w/o SF	0.6604	0.2326	0.2645
SSH-T^3	0.6650	0.2366	0.2671

Unlike prior Transformer-based models (e.g., Transformer, Informer, Crossformer, PatchTST) assuming uniform time intervals, our MS-PBS data has various time intervals. Interval modeling is critical because defaulters rapidly transfer funds to avoid account freezes. Variants of Transformers capture time series seasonality but suffer from MS-PBS noise. PatchTST's uniform sequence patching will incorrectly merge transactions in two days into one patch, reducing robustness. iTransformer focuses more on the feature correlations over sequence inner dependencies further degrading performance on our task. Pre-trained models like Bert4rec [36] and S^3-rec [52] use behavior-level self-supervision tasks (e.g., masked behavior modeling) that overlook inherent behavior-level noise. Predicting future behaviors misaligns with the defaulter prediction target. CBiT [10] requires multi-pair contrastive learning and suffers from "out-of-memory" issues when processing larger datasets.

4.3 Ablation Study

In this section, we conduct ablation studies to validate the effectiveness of the key components of SSH-T^3. We set three variants:

- w/o MAAS: Without adopting MAAS Transformer.
- w/o TE: Without temporal encodings. We remove $\boldsymbol{H}_{h2d}$, $\boldsymbol{H}_{pos}$, $\boldsymbol{H}_{d2y}$, and $\boldsymbol{H}_{holiday}$.
- w/o SF: Without sigmoid fusion. Here we concatenate the output of two towers instead.

Fig. 2 shows the results of ablation study: our model performs best across all experiments. The MAAS Transformer explicitly captures defaulters' trading patterns, helping the model learn their unique day-level representations and accelerate convergence. Temporal

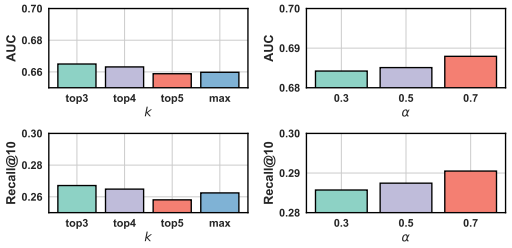

Figure 4: Hyperparameter study on our dataset.

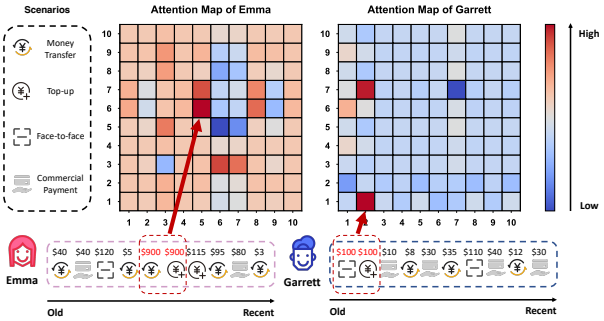

Figure 5: Case Study. Emma and Garrett are two real defaulters in the dataset. Their behavior of transferring the same amount of funds across multiple scenarios is captured by the scenario tower, as highlighted in the attention map.

Encoding, integrated into both behavior-level and day-level features, enables the model to better comprehend special temporal signals like periodicity and holidays. Sigmoid Fusion outperforms single concatenation by highlighting signals of the scenario tower on the behavior tower, avoiding performance degradation.

4.4 Parameter Sensitivity

We analyze hyperparameter impacts on our dataset, focusing on the masking ratio α and k. k denotes the number of top behavior-level representations aggregated into day-level features. Fig. 4 shows SSH-T^3 performs best at $k = 3$: defaulter signals rarely rely on single behaviors (max pooling may discard relevant information), while $k > 3$ introduces more noise. For α, higher values improve performance but increase computational resources. Hence, $\alpha = 0.3$ is used for our experiment.

4.5 Case Study

We conduct a case analysis using our real dataset. Fig. 5 displays the transaction sequences of two actual defaulting users on a certain day, along with the attention map captured by the MAAS Transformer. During the defaulters' online payment, both of them exhibit a peculiar payment pattern where the money of the same amount transfers across different scenarios. This unusual payment pattern is effectively detected and assigned a high weight by the MAAS Transformer Block. These payment behaviors are highlighted, and SSH-T^3 identifies these two users as potential defaulters. This demonstrates the effectiveness of our Two-Tower Multi-Scenario Transformer in capturing amount dependencies across multiple scenarios, and it also underscores the robustness of the SSH-T^3.

4.6 Complexity Analysis

Fig. 1 shows longer MS-PBS improves model performance, but Transformer-based methods face high time complexity. SSH-T^3 uses hierarchical sequence modeling to significantly reduce complexity. The complexity details are in Tab. 3.

Specifically, $\mathcal{L}$ denotes the total length of MS-PBS, $\mathcal{D}$ the feature dimension, $\mathcal{S}$ the stride for patching (PatchTST [34]), $\mathcal{L}_{seg}$ the segment length (Crossformer [50]). For SSH-T^3, $\mathcal{N}$ and $\mathcal{T}$ represent daily payment counts and total days, respectively, with $\mathcal{N} \times \mathcal{T} \approx \mathcal{L}$. Thus, $Max(\mathcal{N}, \mathcal{T})^2 \ll \mathcal{L}^2$. In real business, financial risk assessment features usually satisfy $\mathcal{L} \ll \mathcal{D}$. Notably, the time complexity of SSH-T^3 is close to PatchTST [34] and Crossformer [50], which divide MS-PBS into behaviors patches with equal length but fail to account for non-uniform time intervals, leading to a noisy input.

Table 3: Time complexity comparison.

Model	Time Complexity
Transformer [39]	$O(\mathcal{L}^2 \times \mathcal{D})$
Informer [51]	$O(\mathcal{L} \times log\mathcal{L} \times \mathcal{D})$
PatchTST [34]	$\approx O((\mathcal{L}/\mathcal{S})^2 \times \mathcal{D})$
Crossformer [50]	$O((\mathcal{L}/\mathcal{L}_{seg})^2 \times \mathcal{D})$
iTransformer [28]	$O(\mathcal{L} \times \mathcal{D}^2)$
SSH-T^3 (Ours)	$O(Max(\mathcal{N}, \mathcal{T})^2 \times \mathcal{D})$

By contrast, our day-level representation captures comprehensive user's consumption habits. Shown in Sec. 3.2, SSH-T^3 outperforms these methods in the financial risk assessment tasks.

4.7 Online Result

Utilizing the online dataset detailed in Sec. 4.1.1, we deploy and test proposed SSH-T^3 on our real business platform. Owing to the company's security policy, we are restricted to presenting relative enhancements instead of actual numbers. Relative to the online baseline model, we improve AUC by **3.746%**, KS by **15.760%**, and Recall@10 by **11.004%**. These advancements signify substantial progress in the real-world financial risk assessment industry and contribute to the robustness of the modern financial ecosystem.

5 Conclusion

We propose SSH-T^3, a novel self-supervised Two-Tower Transformer for multi-scenario financial risk assessment. The Two-Tower Transformer includes a Scenario Tower (capturing defaulters' cross-scenario identical-amount transfer patterns via MAAS Transformer) and a Behavior Tower (extracting sequence dependencies from rich features). Our hierarchical MS-PBS pre-training framework mitigates behavior-level noise and Transformer computational complexity. SSH-T^3 outperforms SOTA sequence models and excels in real-world risk assessment scenarios.

Acknowledgements

This work is sponsored by the Tencent Rhino-Bird Focused Research Program.

GenAI Usage Disclosure

We confirm that generative AI tools were used only for improving the grammar and clarity of the manuscript. No part of the research design, data analysis, interpretation of results, or core writing was performed by AI tools. All scientific content is the original work of the authors.

References

[1] E.I. Altman. 1968. Financial ratios, discriminant analysis and the prediction of corporate bankruptcy. *The journal of finance* 23, 4 (1968), 589–609.

[2] B. Baesens, T. Van Gestel, S. Viaene, M. Stepanova, J. Suykens, and J. Vanthienen. 2003. Benchmarking state-of-the-art classification algorithms for credit scoring. *Journal of the operational research society* 54, 6 (2003), 627–635.

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

[4] Xi Chen, Yongxiang Liao, Yun Xiong, Yao Zhang, Siwei Zhang, Jiawei Zhang, and Yiheng Sun. 2023. Speed: Streaming partition and parallel acceleration for temporal interaction graph embedding. *arXiv preprint arXiv:2308.14129* (2023).

[5] Xi Chen, Yateng Tang, Jiarong Xu, Jiawei Zhang, Siwei Zhang, Sijia Peng, Xuehao Zheng, and Yun Xiong. 2025. Rethinking Time Encoding via Learnable Transformation Functions. In *Forty-second International Conference on Machine Learning*.

[6] Xi Chen, Yun Xiong, Siwei Zhang, Jiawei Zhang, Yao Zhang, Shiyang Zhou, Xixi Wu, Mingyang Zhang, Tengfei Liu, and Weiqiang Wang. 2024. Dtformer: A transformer-based method for discrete-time dynamic graph representation learning. In *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management*. 301–311.

[7] Xi Chen, Siwei Zhang, Yun Xiong, Xixi Wu, Jiawei Zhang, Xiangguo Sun, Yao Zhang, Feng Zhao, and Yulin Kang. 2024. Prompt learning on temporal interaction graphs. *arXiv preprint arXiv:2402.06326* (2024).

[8] D. Cheng, S. Xiang, C. Shang, Y. Zhang, F. Yang, and L. Zhang. 2020. Spatio-temporal attention-based neural network for credit card fraud detection. In *Proceedings of the AAAI conference on artificial intelligence (AAAI)*, Vol. 34. 362–369.

[9] J. Devlin. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805* (2018).

[10] H. Du, H. Shi, P. Zhao, D. Wang, V.S. Shen, Y. Liu, G. Liu, and L. Zhao. 2022. Contrastive learning with bidirectional transformers for sequential recommendation. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management (CIKM)*. 396–405.

[11] S. Elsayed, A. Rashed, and L. Schmidt-Thieme. 2024. Multi-Behavioral Sequential Recommendation. In *Proceedings of the 18th ACM Conference on Recommender Systems (RecSys)*. 902–906.

[12] C. Fu, W. Wu, X. Zhang, J. Hu, J. Wang, and J. Zhou. 2023. Robust user behavioral sequence representation via multi-scale stochastic distribution prediction. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management (CIKM)*. 4567–4573.

[13] James Guild. 2017. Fintech and the Future of Finance. *Asian Journal of Public Affairs* (2017), 17–20.

[14] J. Guo, G. Liu, Y. Zuo, and J. Wu. 2018. Learning sequential behavior representations for fraud detection. In *2018 IEEE international conference on data mining (ICDM)*. IEEE, 127–136.

[15] D.J. Hand and W.E. Henley. 1997. Statistical classification methods in consumer credit scoring: a review. *Journal of the Royal Statistical Society: Series A (Statistics in Society)* 160, 3 (1997), 523–541.

[16] K. He, X. Chen, S. Xie, Y. Li, P. Dollár, and R. Girshick. 2022. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (CVPR)*. 16000–16009.

[17] B. Hu, Z. Zhang, C. Shi, J. Zhou, X. Li, and Y. Qi. 2019. Cash-out user detection based on attributed heterogeneous information network with a hierarchical attention mechanism. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, Vol. 33. 946–953.

[18] S. Huang, Y. Xiong, Y. Xie, T. Qiu, and G. Wang. 2024. Robust Sequence-Based Self-Supervised Representation Learning for Anti-Money Laundering. In *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management (CIKM)*. 4571–4578.

[19] Y. Ji, Z. Zhang, X. Tang, J. Shen, X. Zhang, and G. Yang. 2022. Detecting cash-out users via dense subgraphs. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)*. 687–697.

[20] H. Li, X. Fu, R. Wu, J. Xu, K. Xiao, X. Chang, W. Wang, S. Chen, L. Shi, T. Xiong, et al. 2022. Design Domain Specific Neural Network via Symbolic Testing. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)*. 3219–3229.

[21] K. Li, T. Yang, M. Zhou, J. Meng, S. Wang, Y. Wu, B. Tan, H. Song, L. Pan, F. Yu, et al. 2024. SEFraud: Graph-based Self-Explainable Fraud Detection via Interpretative Mask Learning. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)*. 5329–5338.

[22] P. Swietojanskiand J. Li and S. Renals. 2016. Learning hidden unit contributions for unsupervised acoustic model adaptation. *IEEE/ACM Transactions on Audio, Speech, and Language Processing (TASLP)* 24, 8 (2016), 1450–1463.

[23] S-T. Li, W. Shiue, and M-H. Huang. 2006. The evaluation of consumer loans using support vector machines. *Expert Systems with Applications* 30, 4 (2006), 772–782.

[24] W. Lin, L. Sun, Q. Zhong, C. Liu, J. Feng, X. Ao, and H. Yang. 2021. Online credit payment fraud detection via structure-aware hierarchical recurrent neural network.. In *IJCAI*. 3670–3676.

[25] X. Ling, D. Yan, B. Alsallakh, A. Pandey, M. Bakshi, and P. Bhattacharya. 2023. Learned Temporal Aggregations for Fraud Classification on E-Commerce Platforms. In *Companion Proceedings of the ACM Web Conference 2023 (WWW)*. 1365–1372.

[26] C. Liu, L. Sun, X. Ao, J. Feng, Q. He, and H. Yang. 2021. Intention-aware heterogeneous graph attention networks for fraud transactions detection. In *Proceedings of the 27th ACM SIGKDD conference on knowledge discovery & data mining (KDD)*. 3280–3288.

[27] C. Liu, Q. Zhong, X. Ao, L. Sun, W. Lin, J. Feng, Q. He, and J. Tang. 2020. Fraud transactions detection via behavior tree with local intention calibration. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD)*. 3035–3043.

[28] Y. Liu, T. Hu, H. Zhang, H. Wu, S. Wang, L. Ma, and M. Long. [n. d.]. iTransformer: Inverted Transformers Are Effective for Time Series Forecasting. In *The Twelfth International Conference on Learning Representations (ICLR)*.

[29] Nicholas Loubere. 2017. China's internet finance boom and tyrannies of inclusion. *China Perspectives* 2017, 2017/4 (2017), 9–18.

[30] J. Luo, M. He, X. Lin, W. Pan, and Z. Ming. 2022. Dual-task learning for multi-behavior sequential recommendation. In *Proceedings of the 31st ACM international conference on information & knowledge management (CIKM)*. 1379–1388.

[31] J. Luo, X. Yan, and Y. Tian. 2020. Unsupervised quadratic surface support vector machine with application to credit risk assessment. *European Journal of Operational Research* 280, 3 (2020), 1008–1017.

[32] X. Ma, L. Zhao, G. Huang, Z. Wang, Z. Hu, X. Zhu, and K. Gai. 2018. Entire space multi-task model: An effective approach for estimating post-click conversion rate. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval (SIGIR)*. 1137–1140.

[33] C. Musto, G. Semeraro, P. Lops, M. De Gemmis, and G. Lekkas. 2015. Personalized finance advisory through case-based recommender systems and diversification strategies. *Decision Support Systems* 77 (2015), 100–111.

[34] Y. Nie, N.H. Nguyen, P. Sinthong, and J. Kalagnanam. [n. d.]. A Time Series is Worth 64 Words: Long-term Forecasting with Transformers. In *The Eleventh International Conference on Learning Representations (ICLR)*.

[35] Emanuele Rossi, Ben Chamberlain, Fabrizio Frasca, Davide Eynard, Federico Monti, and Michael Bronstein. 2020. Temporal graph networks for deep learning on dynamic graphs. *arXiv preprint arXiv:2006.10637* (2020).

[36] F. Sun, J. Liu, J. Wu, C. Pei, X. Lin, W. Ou, and P. Jiang. 2019. BERT4Rec: Sequential recommendation with bidirectional encoder representations from transformer. In *Proceedings of the 28th ACM international conference on information and knowledge management (CIKM)*. 1441–1450.

[37] G. Sun, T. Li, Y. Ai, and Q. Li. 2023. Digital finance and corporate financial fraud. *International Review of Financial Analysis* 87 (2023), 102566.

[38] Hamed Taherdoost. 2023. Fintech: Emerging trends and the future of finance. *Financial technologies and DeFi: a revisit to the digital finance revolution* (2023), 29–39.

[39] A. Vaswani. 2017. Attention is all you need. *Advances in Neural Information Processing Systems (NeurIPS)* (2017).

[40] D. Wang, Z. Zhang, Y. Zhao, K. Huang, Y. Kang, and J. Zhou. 2023. Financial Default Prediction via Motif-preserving Graph Neural Network with Curriculum Learning. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)*. 2233–2242.

[41] H. Wang, J. Peng, F. Huang, J. Wang, J. Chen, and Y. Xiao. 2023. Micn: Multi-scale local and global context modeling for long-term series forecasting. In *The eleventh international conference on learning representations (ICLR)*.

[42] L. Wang, H. Zhao, C. Feng, W. Liu, C. Huang, M. Santoni, Manuel Cristofaro, P. Jafrancesco, and J. Bian. 2023. Removing Camouflage and Revealing Collusion: Leveraging Gang-crime Pattern in Fraudster Detection. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)*. 5104–5115.

[43] Z. Wang, Q. Wu, B. Zheng, J. Wang, K. Huang, and Y. Shi. 2023. Sequence as genes: An user Behavior modeling framework for fraud transaction detection in E-commerce. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)*. 5194–5203.

[44] C. Wu, F. Wu, T. Qi, and Y. Huang. 2022. Userbert: Pre-training user model with contrastive self-supervision. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*. 2087–2092.

[45] H. Wu, T. Hu, Y. Liu, H. Zhou, J. Wang, and M. Long. [n. d.]. TimesNet: Temporal 2D-Variation Modeling for General Time Series Analysis. In *The Eleventh International Conference on Learning Representations (ICLR)*.

[46] Biao Yang, Yun Xiong, Xi Chen, Xuejing Feng, Meng Wang, and Jun Ma. 2024. ST-ECP: A Novel Spatial-Temporal Framework for Energy Consumption Prediction of Vehicle Trajectory. In *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management*. 2807–2816.

[47] E. Yuan, W. Guo, Z. He, H. Guo, C. Liu, and R. Tang. 2022. Multi-behavior sequential transformer recommender. In *Proceedings of the 45th international ACM SIGIR conference on research and development in information retrieval (SIGIR)*. 1642–1652.

[48] Siwei Zhang, Xi Chen, Yun Xiong, Xixi Wu, Yao Zhang, Yongrui Fu, Yinglong Zhao, and Jiawei Zhang. 2024. Towards adaptive neighborhood for advancing temporal interaction graph modeling. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 4290–4301.

[49] Siwei Zhang, Yun Xiong, Yao Zhang, Yiheng Sun, Xi Chen, Yizhu Jiao, and Yangyong Zhu. 2023. Rdgsl: Dynamic graph representation learning with structure learning. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*. 3174–3183.

[50] Y. Zhang and J. Yan. 2023. Crossformer: Transformer utilizing cross-dimension dependency for multivariate time series forecasting. In *The eleventh international conference on learning representations (ICLR)*.

[51] H. Zhou, S. Zhang, J. Peng, S. Zhang, J. Li, H. Xiong, and W. Zhang. 2021. Informer: Beyond efficient transformer for long sequence time-series forecasting. In *Proceedings of the AAAI conference on artificial intelligence (AAAI)*, Vol. 35. 11106–11115.

[52] K. Zhou, H. Wang, W.X. Zhao, Y. Zhu, S. Wang, F. Zhang, Z. Wang, and J-R. Wen. 2020. S3-rec: Self-supervised learning for sequential recommendation with mutual information maximization. In *Proceedings of the 29th ACM international conference on information & knowledge management (CIKM)*. 1893–1902.