
A Generative Approach to Credit Prediction with Learnable Prompts for Multi-scale Temporal Representation Learning

Yu Lei^{1,2} Zixuan Wang¹ Yiqing Feng¹ Junru Zhang³ Yahui Li² Chu Liu¹ Tongyao Wang¹

Abstract

Recent industrial credit scoring models remain heavily reliant on manually tuned statistical learning methods. While deep learning offers promising solutions, its effectiveness is often limited by the complexity of financial data, particularly in long-horizon scenarios. In this work, we propose FinLangNet¹, which addresses credit scoring by reframing it as the task of generating multi-scale distributions of a user’s future behavior. Within this framework, tabular data is transformed into sequential representations, enabling the generation of user embeddings across multiple temporal scales. Inspired by the recent success of prompt-based training in Large Language Models (LLMs), FinLangNet also introduces two types of prompts to model and capture user behavior at both the feature-granularity and user-granularity levels. Experimental results demonstrate that FinLangNet outperforms the online XGBoost benchmark, achieving a 7.2% improvement in KS metric performance and a 9.9% reduction in the relative bad debt rate. Furthermore, FinLangNet exhibits superior performance on public UEA archives, underscoring its scalability and adaptability in time series classification tasks.

1. Introduction

Credit risk prediction is a cornerstone for financial institutions to devise effective lending policies and informed decisions evaluating the solvency of borrowers (Genovesi et al., 2023). This analytical endeavor is critical in managing and minimizing loan default risks, which is essential for preserving low bad debt levels and mitigating financial losses in the multi-billion dollar credit industry (Cheng et al., 2020).

¹Didi International Business Group ²Beijing University of Posts and Telecommunications ³Zhejiang University. Correspondence to: Yu Lei <leiyu0210@gmail.com>, Zixuan Wang <maxwangzixuan@didiglobal.com>.

¹<https://github.com/leiyu0210/FinLangNet>

Credit risk models perform binary classifications to discern good from bad customers, improving overdue risk prediction, and effectively managing bad debt while maintaining profitability (Zhao et al., 2023).

In the field of credit risk for industrial scenarios, user data generally comes from multiple sources such as credit reports and product usage behavior data (Lu and Zhang, 2023). For reasons of stability and interpretability, risk control industry basically relies on XGBoost to handle the irregular multi-source financial data. The prevalent practice in the industry involves developing derivative features through feature engineering, followed by feature selection and processed with a XGBoost model (Li et al., 2022). Given the inherent noise in multi-source data and the high-dimensional, sparse, and discrete nature of financial behavior data, which often have a relatively high proportion of missing values (Elia et al., 2023), the XGBoost model generally needs to capitalize on its strengths when combined with human prior knowledge (Song et al., 2023). However, this process is time-consuming and costly, and the model struggles to extract meaningful representational information about users. Moreover, XGBoost faces difficulties in tracking user credit profiles over the long term (Niazkar et al., 2024), which poses a substantial challenge for practical downstream tasks.

To overcome these challenges, recent advancements in time-series (Yao et al., 2022), sequential models (Yousefi and Tosarkani, 2022), and graph models (Xue et al., 2024) have fueled exciting endeavors in the field, such as Derisk (Liang et al., 2023). However, these methods typically focus on prediction or classification at a single time point. While the widely adopted XGBoost in the industry demonstrates strong performance and interpretability at individual time points, its effectiveness declines over different time periods. In practical scenarios, it is crucial to model user behavior across various time windows. Therefore, we reformulate this task as a user behavior representation problem across future time periods.

In this work, we proposed a novel FinLangNet framework to handle multi-source and multi-modal data effectively, where non-sequential data representations, which capture the advantages of modeling non-sequential features such as feature interactions and high-dimensional sparse data, are

learned through a DeepFM-based (Guo et al., 2017) architecture, and sequential data are processed by the innovative sequential module, which independently captures user representations from each data source and jointly trains them. Moreover, the sequential module incorporates dual prompts at the feature-granularity and user-granularity levels. The outputs are then jointly trained to capture the overall user representation.

The main contributions are summarized as follows:

1. Unlike traditional credit scoring tasks, we reformulate the problem as a generative task. Specifically, we aim to generate future multi-scale distributional representations based on users’ historical behaviors, enabling a more fine-grained and dynamic understanding of credit risks.
2. We propose a novel framework, FinLangNet, to effectively handle multi-source and multi-modal data. In the sequential module, we introduce two classification prompts: A feature-granularity prompt to learn fine-grained feature-level patterns. A user-granularity prompt to capture user-level representations. The head vector of the sequential module serves as the user representation for each data source.
3. It has been deployed in our Behavior Card service, which is used in the operational model during the loan process. It achieves a 9.9% reduction in the relative bad debt rate, proving its practical utility in financial risk assessment. Notably, our sequential module achieves superior performance on the UEA benchmark.

2. Related Work

Credit risk prediction is a crucial task in the financial sector, typically focused on estimating the likelihood of borrower default over a specific time period. Credit scores, such as FICO (Maiden and Maiden, 2024), are widely used evaluation tools (Jensen, 1992; Bucker et al., 2022), generated by algorithms that analyze various user-related data to assess a borrower’s creditworthiness. Extensive research has explored the application of machine learning techniques for credit risk prediction, including decision-tree-based methods like XGBoost (He et al., 2018), graph models such as GraphSAGE (Balmaseda et al., 2023), ChebConv (Liu, 2022), and their combinations (Fein-Ashley et al., 2024). While deep learning methods are often considered to offer enhanced modeling capabilities, existing studies have found that XGBoost generally outperforms deep learning approaches in this domain (Xu et al., 2021).

The processing of irregular, multi-source financial data is a critical challenge in credit risk prediction. Such data are typically presented in tabular form and often involve high-dimensional features, necessitating effective feature

selection techniques. Several methods have been proposed to improve performance, including filter methods (Janane et al., 2023), wrapper methods (Ahadzadeh et al., 2023), and embedded methods (Raghu et al., 2023), which enhance both model accuracy (Xu et al., 2024; Ha et al., 2019; Li et al., 2020) and interpretability (Ma et al., 2018; Xu et al., 2021). Deep learning approaches have also been applied to handle these diverse data types (Gorishniy et al., 2021; Borisov et al., 2022), but they do not consistently outperform XGBoost models (Gorishniy et al., 2021) when working with tabular data.

From data perspective, another approach is to treat financial information as sequential data for processing. Structured data based on temporal sequences, such as transaction records or historical behaviors, can be represented in the standard time-series format, which encompass a wide range of methodologies. Models like EDI, DTWI, and DTWD (Bagnall et al., 2018) rely on calculating distances reflective of temporal warping or deviations in time sequences. MLSTM-FCNs (Karim et al., 2019) combine LSTM and CNN layers for feature generation, while WEASEL-MUSE (Schäfer and Leser, 2017) transforms series into symbolic representations. In the domain of credit risk, time-series models include approaches from both statistical and machine learning (El-Qadi et al., 2022), as well as deep learning methods (Forough and Momtazi, 2022; Ala’raj et al., 2021; Wang and Xiao, 2022; Liang et al., 2023) such as LSTM (Yu et al., 2019) and Transformer (Vaswani et al., 2017). However, existing methods are predominantly built on preprocessed, well-structured financial time-series data, which limits their ability to handle irregular, multi-source records. Additionally, most approaches rely on relatively plain model architectures. To overcome these limitations, we propose FinlangNet, a novel approach for credit risk prediction that transforms irregular multi-source financial data into sequential formats. Through a carefully designed prompt mechanism and network modules tailored to different data types, FinlangNet significantly improves credit risk prediction performance in real-world scenarios.

Meanwhile, we also investigated several LLM-based approaches. In real-world applications, general-purpose large language models often fall short in meeting specific business needs. For example, although CALM (Feng et al., 2023) is a financial-domain LLM, it is not tailored for credit scoring tasks. CALM outputs binary classifications (e.g., Good/Bad), while our model, FinLangNet, produces continuous delinquency probabilities, which can be directly mapped to credit scores. These scores are crucial for making fine-grained decisions such as customer acceptance or rejection based on cutoff thresholds. Moreover, model size and output stability are also critical considerations in production environments, where large models like CALM may not perform reliably or efficiently.

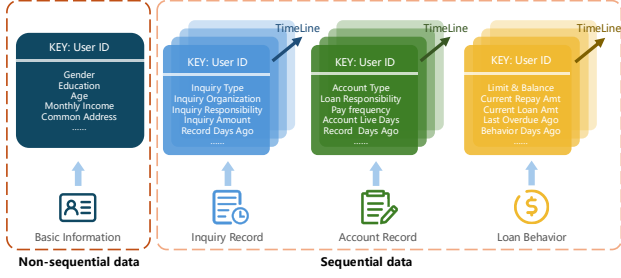


Figure 1: The dataset consists of two parts: Non-Sequential Data and Sequential Data. Non-Sequential Data includes static user information. Sequential Data represents time-dependent information. Together, they provide a comprehensive view of user characteristics and behaviors over time.

3. Preliminary

Our task is to learn an effective representation of the potential risk in a user’s future credit behavior by leveraging irregular, cross-modal user data. The data we used is derived from real user data of the company’s financial business and is presented in tabular form, comprising three main parts: i) *Basic Information*, ii) *Credit Report* and iii) *Loan Behavior*. We systematically disassembled and reorganized these multi-source data, with each source containing various dimensions, into several distinct parts. For contents with temporal information, we unified them in chronological order, enabling processing in a sequential format.

Overall, we categorized the multi-source data into two types: i) non-sequential data, *Basic Information* m , where $m \in \mathbb{R}^M$; ii) sequential data, including the *Credit Report* (which contains *Inquiry Record* and *Account Record*) and *Loan Behavior* information, each comprising various attributes. An overview of the data is presented in Figure 1. More details about datasets are provided in Appendix Table 5. We denote all these sequential data as $z = \{z_1, z_2, \dots, z_S\}$, where $z_s \in \mathbb{R}^{C \times T}$. The capital S stands for the number of different data sources (In our case, S sources include *Inquiry Record* and *Account Record* from *Credit Report*, and *Loan Behavior* information), while s represents each source. C is the number of features, reflecting market data, transaction details, financial indicators and other quantitative measurements. T indicates the sequence length of each feature. For data from different sources, C and T take different values.

The input is defined as $X = (m, z)$. The objective is to predict a probability distribution of overdue behavior across different time scales, expressed as: $f_\theta : X \rightarrow [0, 1]^L$, where L represents the number of time scales and each entry in $[0, 1]^L$ indicates the predicted probability of overdue behavior at a specific time scale. The model is trained under supervision using ground-truth labels, indicating whether overdue behavior occurred at each corresponding time scale. These labels guide the model in learning an accurate proba-

bility distribution over future time horizons.

4. Methodology

As Figure 2 illustrates, our framework consists of two parts: a sequential module and a non-sequential module. For the sequential module, we proposed SRG (Sequence Representation Generator) module to process multi-source sequential data. The output from the sequential and non-sequential modules are fused to form a unified shared embedding. Finally, this output is passed through different task to generate representations across different time scales.

4.1. Sequential Module

As shown in Figure 2 within the sequential module, we designed a module called SRG (Sequence Representation Generator) to process sequence data. Each sequential data source $z_s \in \mathbb{R}^{C \times T}$ (where $s \in \{1, 2, \dots, S\}$) refers to the index of a data source) undergoes a transformation using a specialized module f_{SRG} , which is responsible for learning temporal dependencies and effectively representing sequential dynamics. The transformation can be expressed as: $O_{\text{ns}_s} = f_{\text{SRG}}(z_s)$, where O_{ns_s} represents the processed output of the s -th data source after temporal feature learning.

Considering the issues of data sparsity and model complexity in financial scenarios, we address these problems by applying discretization to sequential data during the data processing stage. Details can be found in Appendix B.1. The tokenizer is a function that converts features in \mathbb{R} into discrete integer tokens in \mathbb{N} . For each channel c , the tokenizer is defined as: $\mathcal{T}_c : \mathbb{R}^T \rightarrow \mathbb{N}^T$

- **Input:** A sequential feature vector $w_c = (w_{c,1}, w_{c,2}, \dots, w_{c,T}) \in \mathbb{R}^T$ for channel c .
- **Output:** A sequence of discrete integer tokens $t_c = (t_{c,1}, t_{c,2}, \dots, t_{c,T}) \in \mathbb{N}^T$.

To incorporate a global representation inspired by BERT’s [class] token, we prepend a prompt of feature granularity $\tilde{\phi}_c$ to the sequence representation for each channel. For each tokenized sequence $t_c = (t_{c,1}, t_{c,2}, \dots, t_{c,T})$, the modified sequence is defined as: $t'_c = (\tilde{\phi}_c, t_{c,1}, t_{c,2}, \dots, t_{c,T})$, where $\tilde{\phi}_c \in \mathbb{N}$ is a unique special prompt specifically assigned to represent the entire channel c . Thus, the final representation for all channels becomes: $t = \{t'_1, t'_2, \dots, t'_C\}$.

After prepending the feature prompt $\tilde{\phi}_c$, we process each modified sequence t'_c with an embedding layer to transform the discrete tokens into dense representations. Specifically, for each channel c , we use an embedding layer $E_c : \mathbb{N} \rightarrow \mathbb{R}^d$, where d is the embedding dimension. This yields the sequence representation: $h_c = (h_{c,0}, h_{c,1}, h_{c,2}, \dots, h_{c,T}) \in \mathbb{R}^{(T+1) \times d}$, where: $h_{c,0} = E_c(\tilde{\phi}_c)$ is the embedding of the

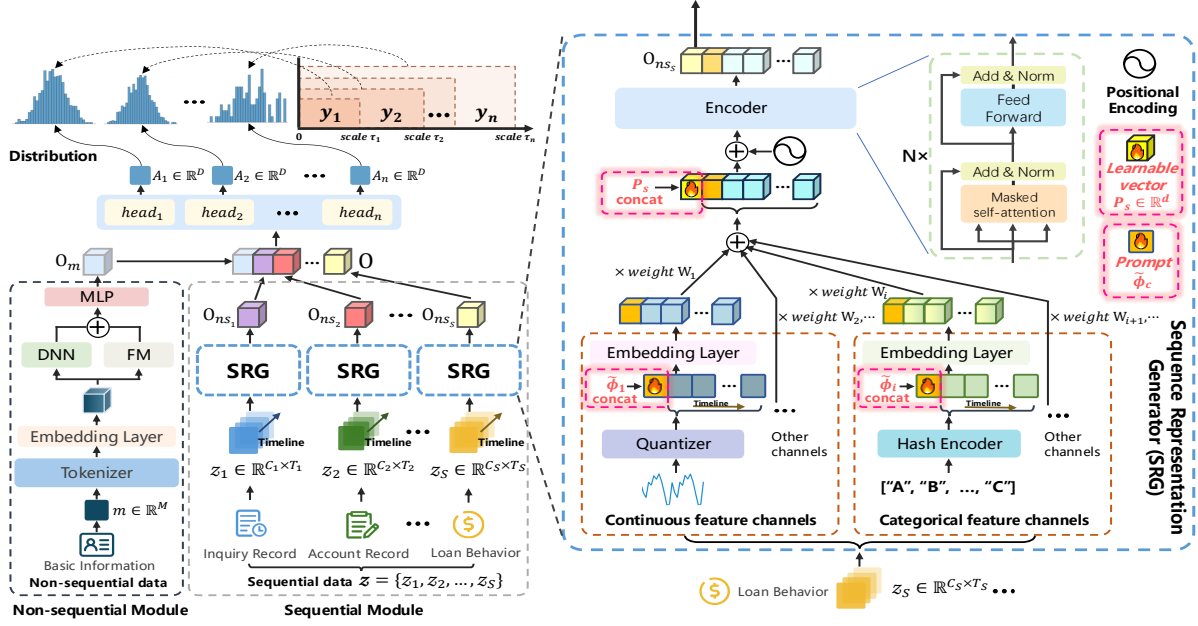


Figure 2: FinLangNet Framework Overview. The architecture incorporates two pivotal sub-modules to harness both sequential and non-sequential data effectively. The Sequential Module (SRG) for sequential features. These components are independently trained during the intermediary phase.

prompt $\tilde{\phi}_c$, $h_{c,j} = E_c(t_{c,j})$, $\forall j \in \{1, 2, \dots, T\}$, represents the embeddings for the individual tokens of the sequence.

Thus, for all features $c \in \{1, 2, \dots, C\}$, we obtain the representations: $\{h_1, h_2, \dots, h_C\}$, where each h_c is the densely embedded representation of the modified token sequence for feature channel c .

Inspired by the recent success of prompt-based training in Large Language Models (LLMs), we aim to use a learnable vector to represent different data sources. Specifically, to construct the final representation H' , a summary prompt $P_s \in \mathbb{R}^d$ is concatenated with the aggregated feature h' .

To compute the global representation h' from the features $h_c \in \mathbb{R}^{(T+1) \times d}$ along the C components, we perform a weighted sum pooling operation across the C components as follows:

$$h' = \sum_{c=1}^C w_c h_c, \quad h' \in \mathbb{R}^{(T+1) \times d}, \quad (1)$$

where the weights w_c are generated by a weight function $a(h_c)$, ensuring $\sum_{c=1}^C w_c = 1$.

Next, we concatenate the learnable vector P_s (representing the summary prompt) with the global feature h' to get the final representation H' :

$$H' = [P_s; h'], \quad H' \in \mathbb{R}^{(T+2) \times d}. \quad (2)$$

Here, $[\cdot; \cdot]$ denotes the concatenation operation. The resulting representation H' combines both the learnable vector P_s

and the aggregated global features h' , providing enriched semantic information for downstream tasks.

The process begins with the input representation H' and concludes with the final output O_{ns_S} . The input $H' \in \mathbb{R}^{(T+2) \times d}$ is processed through a Transformer encoder, which consists of multiple layers of self-attention and feed-forward modules. Let $H^{(l)} \in \mathbb{R}^{(T+2) \times D}$ represent the input to the l -th layer, where $H^{(0)} = H'$, and D denotes the embedding dimension of the Transformer.

Each Transformer layer begins by computing the multi-head attention outputs. The outputs of k attention heads are concatenated and projected back to the dimension D using a learnable projection matrix W_O :

$$H_{\text{attn}}^{(l)} = [\text{head}_1; \text{head}_2; \dots; \text{head}_k] W_O. \quad (3)$$

A residual connection is then combined with the input of the layer, followed by layer normalization to stabilize training:

$$H_{\text{out}}^{(l)} = \text{LayerNorm}(H^{(l)} + H_{\text{attn}}^{(l)}). \quad (4)$$

The output is then passed through a two-layer feed-forward network (FFN) for further transformation. The FFN is defined as:

$$\text{FFN}(x) = \text{ReLU}(x W_1 + b_1) W_2 + b_2, \quad (5)$$

where $W_1, W_2 \in \mathbb{R}^{D \times D}$ and $b_1, b_2 \in \mathbb{R}^D$ are learnable parameters of the network.

To retain consistency with previous layers, a second residual connection is applied to the FFN output, followed by layer normalization:

$$H^{(l+1)} = \text{LayerNorm}(H_{\text{out}}^{(l)} + \text{FFN}(H_{\text{out}}^{(l)})). \quad (6)$$

This process is repeated for L layers. $H^{(L)}$ represents the final output of the Transformer encoder, where $H^{(L)} \in \mathbb{R}^{(T+2) \times D}$. We extract the first token vector, representing the output of corresponding position P_s as the feature expression learned by the current source. The final output O_{ns_S} is computed: $O_{\text{ns}_S} = H^{(L)}[0] \in \mathbb{R}^D$.

After processing each individual data source through f_{SRG} , the outputs $O_{\text{ns}_1}, O_{\text{ns}_2}, \dots, O_{\text{ns}_S}$ are finally concatenated along the source dimension to generate a unified representation of the entire multi-source dataset. Formally, the concatenation process can be expressed as:

$$O_{\text{ns}} = [O_{\text{ns}_1}; O_{\text{ns}_2}; \dots; O_{\text{ns}_S}]. \quad (7)$$

The aggregated output O_{ns} serves as the sequential feature representation of the entire dataset, capturing temporal patterns as well as multi-source dependencies.

Through this approach, the method effectively learns both individual temporal dependencies from each sequential source and cross-source relationships, providing a comprehensive representation for the sequential feature set z .

4.2. Non-Sequential Module

As described in Fig 2. within the non-sequential module. Given a set of non-sequential features denoted as $m \in \mathbb{R}^M$, we process m using the DeepFM (Guo et al., 2017) model to effectively capture both low-order and high-order feature interactions. We construct a custom dictionary that maps potential feature values or their ranges to unique tokens. Due to the potentially large feature space, we employ a hashing technique to quickly map continuous feature values to integer indices, approximating dense feature representations without storing a full mapping. These integer indices are then passed to the embedding layer of the DeepFM model, where the sparse indices are transformed into dense low-dimensional latent vectors.

The embedding vectors $\{V_m\}_{m=1}^M$ generated for each feature are passed to the FM component to model pairwise (second-order) interactions between features. This is achieved by computing the inner product $\langle V_i, V_j \rangle$, which measures the interaction strength between feature i and feature j . Additionally, the FM component provides a linear modeling of first-order feature importance. The FM output is defined as:

$$y_{\text{FM}} = \langle w, m \rangle + \sum_{j_1=1}^M \sum_{j_2=j_1+1}^M \langle V_{j_1}, V_{j_2} \rangle \cdot m_{j_1} \cdot m_{j_2}, \quad (8)$$

where $w \in \mathbb{R}^M$ is the weight for first-order features, and $V_j \in \mathbb{R}^k$ (where k is user-defined) is the latent vector for feature j . Both w and V_j are jointly learned during training.

The same embedding vectors are also fed into the deep component, which is a feed-forward neural network designed to learn high-order non-linear feature interactions.

$$y_{\text{DNN}} = \text{MLP}\left(\frac{1}{2} \left[\left(\sum_{i=1}^M m_{i_1} \mathbf{v}_{i_1} \right)^2 - \sum_{i=1}^M (m_{i_1}^2 \cdot \mathbf{v}_{i_1}^2) \right] \right). \quad (9)$$

The final prediction of the DeepFM model combines the outputs of the FM and DNN components:

$$O_m = \text{MLP}(y_{\text{FM}} + y_{\text{DNN}}). \quad (10)$$

This comprehensive processing effectively leverages both linear and non-linear interactions between non-sequential features, enhancing the model's ability to capture complex patterns within the static feature set m .

4.3. Design of Multi-scale Distribution Representation and Loss Function

4.3.1. MULTI-SCALE DISTRIBUTION REPRESENTATION

In our application scenario, we hope to model the representation of users at different time scales in the future so that we can more comprehensively evaluate the credit risk of a user. In our model, the output vector is defined as:

$$O = [O_m; O_{\text{ns}}]. \quad (11)$$

Suppose we aim to learn representations at n different scales. The shared representation O is first processed through a multi-layer perceptron and subsequently passed through n separate heads, generating a set of vectors: $A = \{A_1, A_2, \dots, A_n\}$. Each A_i corresponds to the user's representation at a specific future time scale. To intuitively evaluate the model's performance, we map each A_i to a downstream task y_i , where y_i represents the overdue status of the user at the corresponding time scale, resulting in predictions $\mathbf{y} = \{y_1, y_2, \dots, y_n\}$.

4.3.2. LOSS FUNCTION

This design dynamically adjusts the model's focus on different samples. This loss combines the Mean Squared Error (MSE) and Weighted Logarithmic Loss (WLL), while dynamically adjusting sample-level weights to emphasize harder examples. The total loss function is defined as:

$$\mathcal{L}_{\text{total}} = \frac{1}{n} \sum_{i=1}^n \omega_i [\beta (y'_i - y_i)^2 - (1 - \beta) (\mathcal{L}_{\text{WLL}, i})], \quad (12)$$

Table 1: Performance comparison across multiple models and labels. The upper part lists models using tabular features, while the lower part includes models combining sequential and non-sequential features. All models share the same training and evaluation settings.

Label	$y_1(\tau = 1)$		$y_2(\tau = 2)$		$y_3(\tau = 3)$		$y_4(\tau = 4)$		$y_5(\tau = 5)$		$y_6(\tau = 6)$		Avg KS
Metrics	AUC	KS	AUC	KS	AUC	KS	AUC	KS	AUC	KS	AUC	KS	(%)
XGBoost(baseline)	72.78	32.85	75.76	37.42	70.89	30.00	73.04	33.18	68.69	26.96	69.70	28.31	31.45
MLP (Goodfellow et al., 2016)	71.97	31.81	74.76	36.14	69.95	28.70	72.00	31.76	67.59	25.38	68.45	26.55	30.06
GraphSAGE (Hamilton et al., 2017)	72.19	32.04	75.09	36.49	70.26	29.09	72.37	32.16	68.03	26.00	68.95	27.25	30.51
GRU (Cho et al., 2014)	72.59	32.40	75.68	37.16	70.93	30.05	73.37	33.57	69.06	27.44	70.62	29.75	31.73
Transformer (Vaswani et al., 2017)	72.54	32.62	75.95	37.98	70.97	30.12	73.76	34.54	69.30	27.82	71.19	30.67	32.29
Mamba (Gu and Dao, 2023)	72.28	32.06	75.66	37.28	70.65	29.67	73.16	33.47	68.79	26.88	70.23	29.02	31.40
TimesNet (Wu et al., 2023)	72.49	32.54	75.90	37.98	70.83	29.99	73.48	34.15	69.26	27.73	71.05	30.53	32.15
DIN (Zhou et al., 2018)	73.16	33.63	76.51	38.98	71.56	31.16	74.13	35.19	70.03	28.89	71.84	31.67	33.25
DIEN (Zhou et al., 2019)	73.45	33.83	76.91	39.33	71.72	31.30	74.42	35.33	70.03	28.79	71.86	31.58	33.36
BST (Chen et al., 2019)	73.28	33.76	76.80	39.35	71.62	31.23	74.37	35.52	69.95	28.78	71.89	31.75	33.40
FinLangNet(ours)	73.55	34.08	76.96	39.46	71.92	31.60	74.51	35.67	70.33	29.27	72.12	32.16	33.71

y_i represents the task label corresponding to the representation of the observation period at different time scales
 τ represents different time scales (such as 60 days, 90 days, 180 days, etc.)

where the dynamic sample weight $\omega_i = \frac{\left(\frac{1}{g_i + \epsilon}\right)^\alpha}{\sum_{j=1}^N \left(\frac{1}{g_j + \epsilon}\right)^\alpha}$, and the gradient magnitude $g_i = \left|\frac{\partial \mathcal{L}_i}{\partial y_i}\right|$. Here, $\beta \in [0, 1]$ is a hyperparameter controlling the trade-off between MSE and WLL. The Weighted Logarithmic Loss (WLL) is defined as:

$$\mathcal{L}_{\text{WLL},i} = - \left(w_i^+ y_i \log(y_i' + \epsilon) + w_i^- (1 - y_i) \log(1 - y_i' + \epsilon) \right), \quad (13)$$

where w_i^+ and w_i^- represent the weights for positive and negative samples, and ϵ is a small constant to prevent numerical instability. Details can be found in Appendix B.2.

5. Experiment

5.1. Experiment Setup

Initially, our experiments are carried out on real business data. We sampled data from over 7 million users. The dataset contains 6 evaluation tasks, representing overdue status at different time scales. The evaluation indicators of each task are KS (Massey Jr, 1951) and AUC. We compared the online benchmark model and some deep model methods under the same training parameter conditions. Additional details of our dataset are provided in Appendix Table 5. Detailed training parameters can be found in Appendix Table 7.

Furthermore, we have chosen to evaluate the scalability of our framework’s sequential module by analyzing 5 multivariate time series classification (MTSC) tasks sourced from the publicly available UEA archive (Bagnall et al., 2018). These datasets encompass a diverse array of practical applications, including gesture and action recognition, audio analysis, and medical diagnostics through heartbeat

monitoring. Our assessment entails a comparison of our FinLangNet with 11 existing MTSC SOTA methods, such as TStamp (Zerveas et al., 2021) and SVP-T (Zuo et al., 2023). See the Appendix for more details.

5.2. Performance on Credit Prediction

As illustrated in Table 1, we evaluate model performance across six temporal scales, denoted as τ ranging from 1 to 6, with prediction results labeled as y_1 through y_6 . Each task is assessed using AUC and KS as evaluation metrics.

In the upper part of the table, we compare traditional tabular-based models. XGBoost, the current online baseline, exhibits strong performance across most tasks and remains highly competitive in risk control scenarios. However, our proposed FinLangNet consistently outperforms XGBoost and other deep learning baselines on all six tasks, achieving an average improvement of 7.2% in key metrics. This demonstrates both the robustness and scalability of our framework in handling complex business data.

The lower section of Table 1 focuses on models tailored for sequential inputs, including GRU (Cho et al., 2014), Transformer (Vaswani et al., 2017), Mamba (Gu and Dao, 2023), TimesNet (Wu et al., 2023), DIN (Zhou et al., 2018), DIEN (Zhou et al., 2019) and BST (Chen et al., 2019). All models are trained under identical hyperparameter settings. Even in this setting, FinLangNet achieves superior performance with negligible additional computational cost. We attribute the superior performance of our model to its enhanced ability to generalize across heterogeneous channels and data sources, a limitation commonly observed in standard sequential models that often struggle in real-world scenarios despite performing well on academic datasets. To address this, we introduce two tailored prompts that facilitate cross-source information integration. Furthermore, we reformulate credit risk modeling as a multi-scale representation learning task,

Table 2: Performance Metrics for Model across Multiple Labels Tested with Various Lengths of Time-Series Data

Label	$y_1(\tau = 1)$		$y_2(\tau = 2)$		$y_3(\tau = 3)$		$y_4(\tau = 4)$		$y_5(\tau = 5)$		$y_6(\tau = 6)$		Avg KS
Metrics	AUC	KS	AUC	KS	AUC	KS	AUC	KS	AUC	KS	AUC	KS	(%)
1.00 * Length	73.55	34.08	76.96	39.46	71.92	31.60	74.51	35.67	70.33	29.27	72.12	32.16	33.71
0.50 * Length	73.45	34.00	76.80	39.34	71.80	31.55	74.45	35.61	70.22	29.14	72.07	32.11	33.62
0.25 * Length	73.36	33.82	76.77	39.18	71.72	31.30	74.37	35.54	70.14	29.07	71.99	32.00	33.49

enabling comprehensive user risk profiling across the full credit lifecycle. This multi-scale design promotes mutual learning across time horizons, contributing to consistently optimal performance across all tasks.

5.3. Ablation Study

5.3.1. IMPACT OF SEQUENTIAL DATA LENGTH

To assess the effect of sequence length on model performance, we conducted a set of ablation experiments in which only the length of the sequential inputs was varied, while non-sequential features were held fixed. As reported in Table 2, shortening the sequence led to only marginal performance degradation, suggesting that the model exhibits robustness to sequence length variation. This also underscores the presence of critical information in early parts of the sequence and highlights the model’s ability to generalize under conditions of incomplete or truncated data.

Table 3: Influence of Different Modules on task $y_1(\tau = 1)$

Configuration	KS(%)	AUC(%)
Baseline (No Prompts)	32.62	72.54
+ Prompt ($\tilde{\phi}_c$)	32.79	72.65
+ Prompt (P_s)	32.99	72.78
+ Both Prompts ($\tilde{\phi}_c + P_s$)	34.08	73.55
w/o Sequential model (O_{ns})	12.53	58.62
w/o Non-sequential model (O_m)	32.75	72.66

These findings suggest that although there was a reduction in the length of the sequential data, there was only a slight decrease in model performance, which may indicate a certain robustness of the model to changes in sequence length. It also highlights the importance of key information contained within the sequential data for maintaining model performance. Additionally, the results demonstrate the model’s capacity to adapt to variations in data streams that are typical in real-life situations.

5.3.2. ROLE OF DIFFERENT MODULE

We conducted ablation studies to evaluate the contribution of different components in our model, as summarized in Table 3. First, we assess the impact of the two prompts in the SRG module: the *feature-granularity prompt* $\tilde{\phi}_c$ and the

data-source prompt P_s . Both prompts individually improve performance over the no-prompt baseline, with P_s yielding a larger gain, indicating that modeling cross-source interactions is more critical than capturing fine-grained feature context.

Additionally, we hypothesize that the larger gain from P_s is due to its ability to disentangle and organize information from disparate data sources more effectively, improving the model’s ability to generalize and capture global patterns. On the other hand, $\tilde{\phi}_c$ enhances the model’s fine-grained representation at the feature level, complementing P_s . This synergy between the two prompts explains their combined impact in boosting overall performance (as shown in Table 1), where the integration of both $\tilde{\phi}_c$ and P_s results in the best overall performance.

Furthermore, we examine the role of the sequential and non-sequential modules. Removing the sequential module results in a substantial performance drop, confirming its critical role in capturing temporal dynamics. In contrast, removing the non-sequential module leads to only a slight decline, implying that sequential information carries the dominant signal for our task.

These results validate the effectiveness of our dual-prompt strategy and modular design, highlighting the importance of incorporating both hierarchical prompts and multi-modal input streams for robust credit risk modeling.

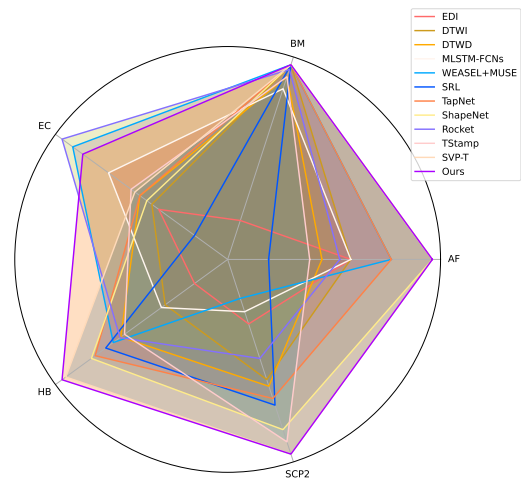


Figure 3: Model Performance on the UEA archive.

Table 4: Swap Set Analysis on the Performance of Online Baseline Model (XGBoost) and FinLangNet in Actual Business

Baseline\Ours	(-inf, 611.0]	(611.0, 697.0]	(697.0, 766.0]	(766.0, 844.0]	(844.0, inf]	Bad_Rate	Pass_Rate	Pass_Bad_Rate
(-inf, 478.0]	41.45%	22.36%	14.78%	8.68%	8.00%	35.5%	100.0%	16.7%
(478.0, 560.0]	31.44%	20.42%	14.54%	9.77%	6.40%	20.5%	79.9%	11.9%
(560.0, 634.0]	28.58%	18.89%	13.05%	8.97%	5.70%	14.0%	59.9%	9.1%
(634.0, 730.0]	27.09%	16.60%	11.56%	7.72%	4.69%	8.9%	40.0%	6.6%
(730.0, inf]	27.78%	14.50%	10.35%	6.61%	3.08%	4.2%	20.0%	4.2%
Bad_Rate	38.43%	20.13%	13.04%	7.98%	3.54%			
Pass_Rate	100.00%	79.93%	59.81%	39.74%	19.89%			
Pass_Bad_Rate	16.67%	11.20%	8.20%	5.76%	3.54%			
RBDR	9.90%							

Yellow_cell - No change (Users who are not replaced under the current decision)

Green_cell - Swapping in user (Users whose historical decision is Reject and whose new decision is Approve)

Orange_cell - Swapping out user (Users whose historical decision is Approve and whose new decision is Reject)

Bad_Rate - The overdue rate of users in the corresponding binning

Pass_Rate - The proportion of users who pass the corresponding binning decision

Pass_Bad_Rate - The overdue rate of users under the corresponding binning decision

RBDR - Relatively bad decline rate. $RBDR = (Pass_Bad_Rate(Baseline) - Pass_Bad_Rate(Ours)) / (Pass_Bad_Rate(Baseline))$

5.4. Scalability

We also evaluate the scalability of our method to demonstrate its benefits in time series classification tasks. The accuracy comparison results, illustrated in Figure 3, showcase the superior performance of our methods across the majority of public datasets. These results underscore the versatility of our model, highlighting its effectiveness across various real-world domains. Please refer to the detailed results presented in Appendix Table 8.

6. Online A/B Test

In practical business applications, the predicted probability output by the model is transformed into a credit score using a predefined fixed-parameter function. A higher predicted probability of delinquency corresponds to a lower credit score, and vice versa. During the decision-making phase, user scores are partitioned into distinct binning intervals through equal-frequency binning. This process facilitates the determination of a cutoff threshold, where applications with scores above the cutoff are approved, while those below it are rejected. Typically, there exists a positive correlation between the approval rate and the bad debt rate, meaning that an increase in the approval rate often results in a corresponding rise in the bad debt rate.

Although technical metrics such as AUC and KS provide valuable insights into model performance, the ultimate goal of the model is its integration into downstream risk control strategies. In the deployment of credit risk models, Swap Set Analysis (Kritzing and van Vuuren, 2019) is commonly employed to assess performance. This analysis involves “swapping out” a subset of customer accounts—typically those classified as high-risk (bad accounts)—and “swapping in” a new set of accounts, generally consisting of lower-risk (good) customers. This approach provides a more intuitive understanding of the model’s impact on credit decision-

making and overall portfolio risk.

Table 4 presents the Swap Set Analysis comparing our model with the baseline in a real business scenario. The table’s rows and columns represent the distribution of credit scores assigned by the two models to the same batch of customers, with bin boundaries determined using equal-frequency binning into five groups. By mapping the scores of both models into a $5 \times 5 = 25$ grid, we obtain a comparative view of their performance. The values in the table indicate the actual overdue rate (bad rate) within each bin.

Beyond the table, we calculate the bad rate for each bin, the cumulative pass rate, and the bad rate of passed samples (i.e., the actual overdue rate of customers approved under the given cutoff). In practical business applications, the bad rate of passed samples is the key performance metric. Swap Set Analysis allows us to compare the new and baseline models under the same approval rate to observe differences in the bad rate of passed samples.

For illustration, we analyze a cutoff decision boundary at a 60% pass rate, meaning that the top 60% of customers are approved, while the bottom 40% are rejected. Under this setting, the baseline model exhibits a bad rate of 9.1% for the approved population, whereas our new model achieves a lower bad rate of 8.2% under the same pass rate. This indicates an absolute risk reduction of 0.9%, corresponding to a relative improvement of 9.9% over the baseline. As a result, our model effectively mitigates nearly 10% of potential asset losses, delivering substantial economic benefits to the credit business.

For deployment, the model is converted to ONNX format and served on an L20 GPU cluster. Given the low-frequency nature of credit applications, the system operates under moderate QPS (~ 100) and achieves inference latency within 100 ms, meeting real-time business requirements.

To ensure transparency and trustworthiness, we incorporate

a lightweight interpretability framework based on the LIME (Local Interpretable Model-agnostic Explanations) (Ribeiro et al., 2016) method. LIME perturbs the input—by masking or shuffling elements in the sequence—and learns a local surrogate model to identify the most influential features driving predictions. This allows us to interpret the behavior of our sequential model in a locally faithful manner.

We apply this mechanism across different data sources to rank feature importance, which serves multiple practical purposes: (i) guiding feature selection during model development, (ii) supporting issue diagnosis after deployment, and (iii) monitoring feature drift in production.

7. Conclusion

In this work, we propose an innovative framework FinLangNet to solve practical credit risk business problems. We treat credit scoring as a generative task, predicting multi-scale distributions of users’ future behavior. The design of FinLangNet incorporates a DeepFM-based architecture for non-sequential data and the innovative SRG module for sequential data. In SRG module, we introduce dual prompts at the feature and user-granularity levels, allowing for fine-grained feature learning and holistic user representation. Experimental results demonstrate FinLangNet’s effectiveness, achieving a 7.2% improvement in KS metric and a 9.9% reduction in the relative bad debt rate compared with online baseline. Furthermore, FinLangNet achieves superior performance on public UEA archives, showcasing its scalability and adaptability in time-series classification tasks.

References

- Behrouz Ahadzadeh, Moloud Abdar, Fatemeh Safara, Abbas Khosravi, Mohammad Bagher Menhaj, and Ponnuthurai Nagarathnam Suganthan. Sfe: A simple, fast and efficient feature selection algorithm for high-dimensional data. *IEEE Transactions on Evolutionary Computation*, 2023.
- Maher Ala’raj, Maysam F Abbod, and Munir Majdalawieh. Modelling customers credit card behaviour using bidirectional lstm neural networks. *Journal of Big Data*, 8(1): 69, 2021.
- Anthony Bagnall, Hoang Anh Dau, Jason Lines, Michael Flynn, James Large, Aaron Bostrom, Paul Southam, and Eamonn Keogh. The uea multivariate time series classification archive, 2018. *arXiv preprint arXiv:1811.00075*, 2018.
- Vicente Balmaseda, María Coronado, and Gonzalo de Cadenas-Santiago. Predicting systemic risk in financial systems using deep graph learning. *Intelligent Systems with Applications*, 19:200240, 2023.
- Vadim Borisov, Tobias Leemann, Kathrin Seßler, Johannes Haug, Martin Pawelczyk, and Gjergji Kasneci. Deep neural networks and tabular data: A survey. *IEEE Transactions on Neural Networks and Learning Systems*, 2022.
- Michael Bücker, Gero Szepannek, Alicja Gosiewska, and Przemyslaw Biecek. Transparency, auditability, and explainability of machine learning models in credit scoring. *Journal of the Operational Research Society*, 73(1):70–90, 2022.
- Weichen Chen, Canru Wang, Yaming Wang, Wei Liu, Ying Lyu, Taifeng Chen, and Yong Yu. Behavior sequence transformer for e-commerce recommendation in alibaba. In *Proceedings of the 1st Workshop on Deep Learning Practice for High-Dimensional Sparse Data*, pages 1–4. ACM, 2019.
- Dawei Cheng, Zhibin Niu, and Yiyi Zhang. Contagious chain risk rating for networked-guarantee loans. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 2715–2723, 2020.
- Kyunghyun Cho, Bart van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder–decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*, 2014.
- Angus Dempster, François Petitjean, and Geoffrey I Webb. Rocket: exceptionally fast and accurate time series classification using random convolutional kernels. *Data Mining and Knowledge Discovery*, 34(5):1454–1495, 2020.
- Ayoub El-Qadi, Maria Trocan, Thomas Frossard, and Natalia Díaz-Rodríguez. Credit risk scoring forecasting using a time series approach. In *Physical Sciences Forum*, volume 5, page 16. MDPI, 2022.
- Gianluca Elia, Valeria Stefanelli, and Greta Benedetta Ferilli. Investigating the role of fintech in the banking industry: what do we know? *European Journal of Innovation Management*, 26(5):1365–1393, 2023.
- Jacob Fein-Ashley, Tian Ye, Sachini Wickramasinghe, Bingyi Zhang, Rajgopal Kannan, and Viktor Prasanna. A single graph convolution is all you need: Efficient grayscale image classification. *arXiv preprint arXiv:2402.00564*, 2024.
- Duanyu Feng, Yongfu Dai, Jimin Huang, Yifang Zhang, Qianqian Xie, Weiguang Han, Zhengyu Chen, Alejandro Lopez-Lira, and Hao Wang. Empowering many, biasing a few: Generalist credit scoring through large language models. *arXiv preprint arXiv:2310.00566*, 2023.

- Javad Forough and Saeedeh Momtazi. Sequential credit card fraud detection: A joint deep neural network and probabilistic graphical model approach. *Expert Systems*, 39(1):e12795, 2022.
- Jean-Yves Franceschi, Aymeric Dieuleveut, and Martin Jaggi. Unsupervised scalable representation learning for multivariate time series. *Advances in neural information processing systems*, 32, 2019.
- Sergio Genovesi, Julia Maria Mönig, Anna Schmitz, Maximilian Poretschkin, Maram Akila, Manoj Kahdan, Romina Kleiner, Lena Krieger, and Alexander Zimmermann. Standardizing fairness-evaluation procedures: interdisciplinary insights on machine learning algorithms in creditworthiness assessments for small personal loans. *AI and Ethics*, pages 1–17, 2023.
- Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016.
- Yury Gorishniy, Ivan Rubachev, Valentin Khrulkov, and Artem Babenko. Revisiting deep learning models for tabular data. *Advances in Neural Information Processing Systems*, 34:18932–18943, 2021.
- Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces. *arXiv preprint arXiv:2312.00752*, 2023.
- Huifeng Guo, Ruiming Tang, Yunming Ye, Zhenguo Li, and Xiuqiang He. Deepfm: a factorization-machine based neural network for ctr prediction. *arXiv preprint arXiv:1703.04247*, 2017.
- Van-Sang Ha, Dang-Nhac Lu, Gyoo Seok Choi, Ha-Nam Nguyen, and Byeongnam Yoon. Improving credit risk prediction in online peer-to-peer (p2p) lending using feature selection with deep learning. In *2019 21st International Conference on Advanced Communication Technology (ICACT)*, pages 511–515. IEEE, 2019.
- William L. Hamilton, Rex Ying, and Jure Leskovec. Inductive representation learning on large graphs. In *Advances in Neural Information Processing Systems*, volume 30, 2017.
- Hongliang He, Wenyu Zhang, and Shuai Zhang. A novel ensemble method for credit scoring: Adaption of different imbalance ratios. *Expert Systems with Applications*, 98: 105–117, 2018.
- Fatima Zahra Janane, Tayeb Ouaderhman, and Hasna Chamlal. A filter feature selection for high-dimensional data. *Journal of Algorithms & Computational Technology*, 17: 17483026231184171, 2023.
- Herbert L Jensen. Using neural networks for credit scoring. *Managerial finance*, 18(6):15–26, 1992.
- Fazle Karim, Somshubra Majumdar, Houshang Darabi, and Samuel Harford. Multivariate lstm-fcns for time series classification. *Neural networks*, 116:237–245, 2019.
- Nico Kritzing and Gary W van Vuuren. A statistical technique to enhance application scorecard monitoring. *Journal of Credit Risk*, 15(2), 2019.
- Xuan-May Le, Ling Luo, Uwe Aickelin, and Minh-Tuan Tran. Shapeformer: Shapelet transformer for multivariate time series classification. *arXiv preprint arXiv:2405.14608*, 2024.
- Guozhong Li, Byron Choi, Jianliang Xu, Sourav S Bhowmick, Kwok-Pan Chun, and Grace Lai-Hung Wong. Shapenet: A shapelet-neural network approach for multivariate time series classification. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pages 8375–8383, 2021.
- Wei Li, Shuai Ding, Hao Wang, Yi Chen, and Shanlin Yang. Heterogeneous ensemble learning with feature engineering for default prediction in peer-to-peer lending in china. *World Wide Web*, 23:23–45, 2020.
- Yixuan Li, Charalampos Stasinakis, and Wee Meng Yeo. A hybrid xgboost-mlp model for credit risk assessment on digital supply chain finance. *Forecasting*, 4(1):184–207, 2022.
- Yancheng Liang, Jiajie Zhang, Hui Li, Xiaochen Liu, Yi Hu, Yong Wu, Jinyao Zhang, Yongyan Liu, and Yi Wu. Derisk: An effective deep learning framework for credit risk prediction over real-world financial data. *arXiv preprint arXiv:2308.03704*, 2023.
- Xinyan Liu. Fast recommender system combining global and local information: Construction of large-scale commodity information recommendation system. In *2022 2nd International Conference on Big Data, Artificial Intelligence and Risk Management (ICBAR)*, pages 166–169. IEEE, 2022.
- Tian Lu and Yingjie Zhang. Profit vs. equality? the case of financial risk assessment and a new perspective on alternative data. *MIS Quarterly*, 47(4), 2023.
- Xiaojun Ma, Jinglan Sha, Dehua Wang, Yuanbo Yu, Qian Yang, and Xueqi Niu. Study on a prediction of p2p network loan default based on the machine learning lightgbm and xgboost algorithms according to different high dimensional data cleaning. *Electronic Commerce Research and Applications*, 31:24–39, 2018.

- Stephen E Maiden and Stephen E Maiden. Fico score. *Darden Business Publishing Cases*, pages 1–11, 2024.
- Frank J Massey Jr. The kolmogorov-smirnov test for goodness of fit. *Journal of the American statistical Association*, 46(253):68–78, 1951.
- Majid Niazkar, Andrea Menapace, Bruno Brentan, Reza Piraei, David Jimenez, Pranav Dhawan, and Maurizio Righetti. Applications of xgboost in water resources engineering: A systematic literature review (dec 2018–may 2023). *Environmental Modelling & Software*, page 105971, 2024.
- Aniruddh Raghu, Payal Chandak, Ridwan Alam, John Guttag, and Collin Stultz. Sequential multi-dimensional self-supervised learning for clinical time series. In *International Conference on Machine Learning*, pages 28531–28548. PMLR, 2023.
- Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. "why should i trust you?": Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1135–1144. ACM, 2016. doi: 10.1145/2939672.2939778.
- Patrick Schäfer and Ulf Leser. Multivariate time series classification with weasel+ muse. *arXiv preprint arXiv:1711.11343*, 2017.
- Yu Song, Yuyan Wang, Xin Ye, Russell Zaretzki, and Chuanren Liu. Loan default prediction using a credit rating-specific and multi-objective ensemble learning scheme. *Information Sciences*, 629:599–617, 2023.
- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.
- Chongren Wang and Zhuoyi Xiao. A deep learning approach for credit scoring using feature embedded transformer. *Applied Sciences*, 12(21):10995, 2022.
- Haixu Wu, Tengge Hu, Yong Liu, Hang Zhou, Jianmin Wang, and Mingsheng Long. Timesnet: Temporal 2d-variation modeling for general time series analysis. *arXiv preprint arXiv:2210.02186*, 2022.
- Haixu Wu, Zonghan Xu, Yuxuan Liu, Yihan Wu, Jing Lin, Li Zeng, and Xing Xie. Timesnet: Temporal 2d-variation modeling for general time series analysis. *International Conference on Learning Representations (ICLR)*, 2023.
- Jinxin Xu, Han Wang, Yuqiang Zhong, Lichen Qin, and Qishuo Cheng. Predict and optimize financial services risk using ai-driven technology. *Academic Journal of Science and Technology*, 10(1):299–304, 2024.
- Junhui Xu, Zekai Lu, and Ying Xie. Loan default prediction of chinese p2p market: a machine learning methodology. *Scientific Reports*, 11(1):18759, 2021.
- Gang Xue, Shifeng Liu, Long Ren, and Daqing Gong. Risk assessment of utility tunnels through risk interaction-based deep learning. *Reliability Engineering & System Safety*, 241:109626, 2024.
- Yuantao Yao, Minghan Yang, Jianye Wang, and Min Xie. Multivariate time-series prediction in industrial processes via a deep hybrid network under data uncertainty. *IEEE Transactions on Industrial Informatics*, 19(2):1977–1987, 2022.
- Samuel Yousefi and Babak Mohamadpour Tosarkani. The adoption of new technologies for sustainable risk management in logistics planning: A sequential dynamic approach. *Computers & Industrial Engineering*, 173:108627, 2022.
- Yong Yu, Xiaosheng Si, Changhua Hu, and Jianxun Zhang. A review of recurrent neural networks: Lstm cells and network architectures. *Neural computation*, 31(7):1235–1270, 2019.
- George Zerveas, Srideepika Jayaraman, Dhaval Patel, Anuradha Bhamidipaty, and Carsten Eickhoff. A transformer-based framework for multivariate time series representation learning. In *Proceedings of the 27th ACM SIGKDD conference on knowledge discovery & data mining*, pages 2114–2124, 2021.
- Xuchao Zhang, Yifeng Gao, Jessica Lin, and Chang-Tien Lu. Tapnet: Multivariate time series classification with attentional prototypical network. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 6845–6852, 2020.
- Yang Zhao, John W Goodell, Yong Wang, and Mohammad Zoynul Abedin. Fintech, macroprudential policies and bank risk: Evidence from china. *International Review of Financial Analysis*, 87:102648, 2023.
- Guorui Zhou, Xiaoqiang Zhu, Chenru Song, Yingqiang Fan, Xiao Ma, Yanghui Yan, Junqi Jin, Han Li, and Kun Gai. Deep interest network for click-through rate prediction. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2018.
- Guorui Zhou, Xiao Ma, Xiaoqiang Zhu, Yingqiang Fan, Yuan Wang, Yanghui Yan, Junqi Jin, Han Li, and Kun Gai. Deep interest evolution network for click-through rate prediction. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 5941–5948, 2019.

Rundong Zuo, Guozhong Li, Byron Choi, Sourav S Bhowmick, Daphne Ngar-yin Mah, and Grace LH Wong. Svp-t: a shape-level variable-position transformer for multivariate time series classification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 11497–11505, 2023.

A. Data

A.1. The Credit Dataset Details

We sampled real data from over 700,000 active users to participate in the modeling. The sample size level is about 2 million. The assembled datasets consist of three primary sources illustrating different dimensions of consumer financial behavior. The visualization presented in Figure 4, demonstrates a data visualization approach for a subset of our dataset. We randomly selected 256 individuals and focused on four distinct features per individual. From these features, only 10% of their lengths were extracted to perform a focused analysis. The more details is provided in Appendix Table 5.

Delving into the datasets themselves, the Basic Information dataset aggregates a compilation of over 700,000 entries, with each entry endowed with 11 attributes. These attributes relay a combination of demographic insights and rudimentary financial information on the subjects. Concurrently, the Credit Report dataset is divided into two intricate sub-datasets: Firstly, the Inquiry Feature sub-dataset encapsulates the trajectory of credit inquiries with more than 700,000 entries, each chronicled through a sequence of 120 timestamped records, and each record is intricately detailed with 5 distinctive features. Secondly, the Account Feature sub-dataset expands this vista by supplying over 700,000 entries each backed by 200 time-based observations, and an elaborate array of 29 different features for every observation, presenting a fulsome view of individual account behaviors over time. Complementing these, the Loan Behaviour dataset meticulously records transactional data concerning loans for the same sprawling number of subjects, parsed into 26 discrete time points, each defined by 241 unique attributes.

Cumulatively, these datasets form an extensive resource for analyzing consumer credit patterns, with a breadth of information ranging from personal identifiable information to intricate details of credit inquiries, account management, and loan repayment behaviors. In the intricate realm of industrial scenarios, the employment of meticulously processed datasets is quintessential. As portrayed in the table 5, the data employed in this study is the fruition of a complex transformation process, hailing from disparate data sources. Each segment has undergone a tailored procedure to distill the information into a usable format for analysis and model training. The foundation of our dataset is the Basic Information segment, where we extract key personal and financial details from relevant text sources. This extraction focuses on gathering the most pertinent attributes, ensuring a solid base for our dataset.

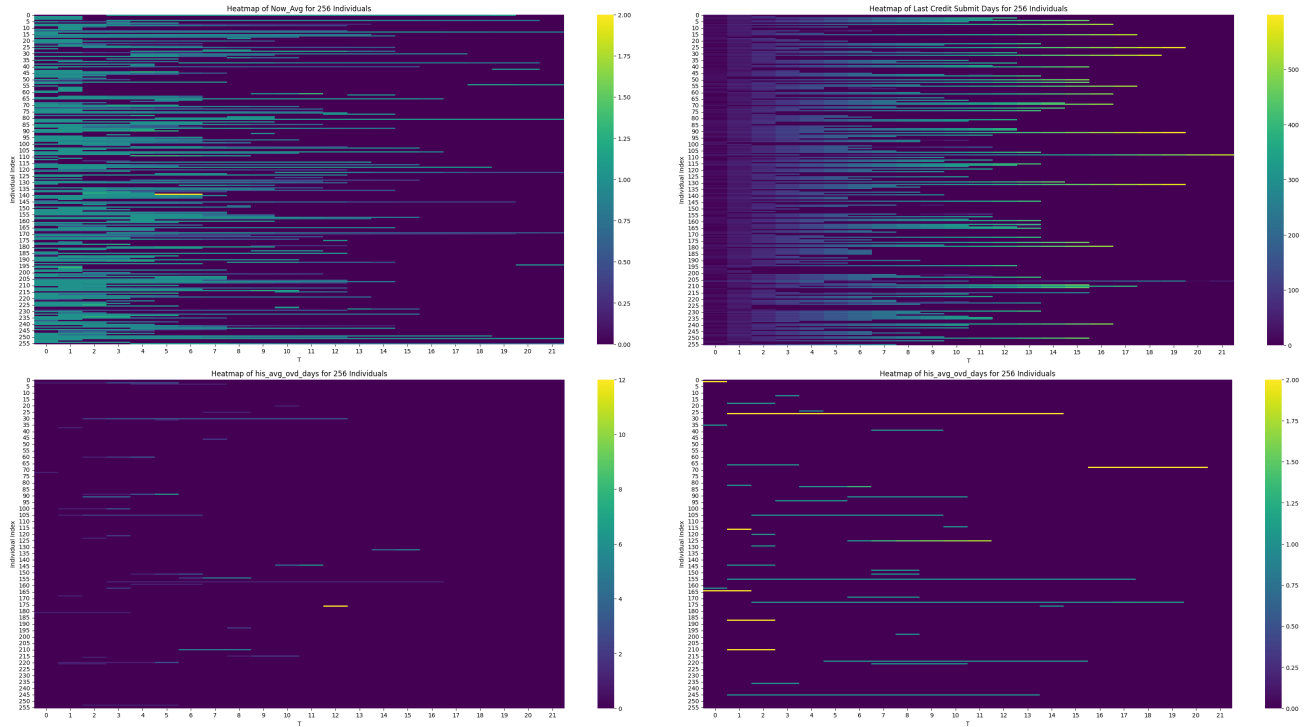


Figure 4: Feature Sub-Section Heatmap of Selected users.

Within the Credit Report database, we divide the data into Inquiry and Account segments, each offering unique insights into the credit reporting process. Through sequence alignment, we achieve a unified and chronological view of the data, which facilitates a more nuanced understanding of credit behavior over time. The Loan Behaviour data is managed through time series analysis, which allows us to navigate its temporal complexities. This approach precisely tracks the evolution of loan transactions, capturing the authentic rhythm of financial movements. Together, these datasets present a comprehensive and nuanced view of the industrial dataset landscape, primed for detailed exploration and advanced modeling. Each segment’s specialized handling enriches the overall data pool, providing a robust foundation for rigorous analysis.

Table 5: Dataset descriptions.

Data Source	Channel	Sequential Length	Dataset Size	Description
Basic Information	11	None	(~ 3000000 , 0, ~ 600000)	Personal Information
Credit Report: Inquiry	5	120	(~ 3000000 , 0, ~ 600000)	Credit Records
Credit Report: Account	29	200	(~ 3000000 , 0, ~ 600000)	Credit Records
Loan Behaviour	241	26	(~ 3000000 , 0, ~ 600000)	Borrowing or Repayment Behavior

A.2. The UEA Dataset Details

For the purpose of testing and validation, we utilized the core module on the UEA archive. Nonetheless, we undertook a random selection encompassing all 5 types of multivariate time series datasets available within the UEA archive. Details of these datasets are shown in Table 6.

Table 6: UEA Classification Dataset Details

Dataset	Train Cases	Test Cases	Dimensions	Length	Classes
AtrialFibrillation(AF)	15	15	2	640	3
BasicMotions(BM)	40	40	4	100	4
EthanolConcentration(EC)	261	263	3	1751	4
Heartbeat(HB)	204	105	61	495	2
SelfRegulationSCP2(SCP2)	200	180	7	1152	2

B. Method

B.1. Data Processing

For the input data, the features of each channel $c \in \{1, 2, \dots, C\}$ are classified into two types: continuous feature channel set $\mathcal{C}_{\text{cont}}$ and categorical feature channel set \mathcal{C}_{cat} , where

$$|\mathcal{C}_{\text{cont}}| = C_{\text{cont}}, \quad |\mathcal{C}_{\text{cat}}| = C_{\text{cat}},$$

and the total number satisfies $C_{\text{cont}} + C_{\text{cat}} = C$.

For continuous feature channels $c \in \mathcal{C}_{\text{cont}}$, the input is a temporal feature vector:

$$W_c = (w_{c,1}, w_{c,2}, \dots, w_{c,T}) \in \mathbb{R}^T.$$

These continuous features are discretized through a quantization function $Q_c : \mathbb{R} \rightarrow \mathbb{N}$, resulting in a discrete integer token sequence:

$$t_c = (t_{c,1}, t_{c,2}, \dots, t_{c,T}) \in \mathbb{N}^T, \quad t_{cj} = Q_c(w_{cj}), \quad \forall j \in \{1, \dots, T\}.$$

For categorical feature channels $c \in \mathcal{C}_{\text{cat}}$, the input is a temporal categorical feature vector:

$$W_c = (w_{c,1}, w_{c,2}, \dots, w_{c,T}), \quad w_{cj} \in \mathcal{X}_c,$$

where \mathcal{X}_c represents the set of categorical features for channel c . These categorical features are discretized via a hash encoding function $\mathcal{H}_c : \mathcal{X}_c \rightarrow \{1, 2, \dots, H_c\}$, producing the token sequence:

$$t_c = (t_{c,1}, t_{c,2}, \dots, t_{c,T}) \in \mathbb{N}^T, \quad t_{cj} = \mathcal{H}_c(w_{cj}), \quad \forall j \in \{1, \dots, T\}.$$

Finally, the discretized outputs for all channels are unified into a single representation as discrete integer token sequences:

$$t_c = (t_{c1}, t_{c2}, \dots, t_{cT}) \in \mathbb{N}^T, \quad \forall c \in \{1, 2, \dots, C\}.$$

For $c \in \mathcal{C}_{\text{cont}}$:

$$t_{cj} = Q_c(w_{cj});$$

For $c \in \mathcal{C}_{\text{cat}}$:

$$t_{cj} = \mathcal{H}_c(w_{cj}).$$

This unified discretization approach ensures that all features, regardless of type, are represented in a consistent format, enabling further processing and modeling.

B.2. Loss Function

To address the issue of class imbalance in our binary classification tasks, we propose a modified loss function that incorporates both a reweighting scheme and dynamic adjustment mechanisms. Specifically, at each iteration, the loss weights are recalculated based on the gradient norms, allowing the model to focus more on harder examples. We denote by ω_r the weight for the r -th loss component, adjusted using a tuning parameter α .

Our total loss function $\mathcal{L}_{\text{total}}$ combines the weighted Dice-BCE loss and the Focal Tversky loss as follows:

$$\mathcal{L}_{\text{total}} = \omega_1 \mathcal{L}_{\text{Dice-BCE}} + \omega_2 \mathcal{L}_{\text{Focal-Tversky}}, \quad (14)$$

where ω_1 and ω_2 are the dynamic weights for each loss component.

The weighted Dice-BCE loss $\mathcal{L}_{\text{Dice-BCE}}$ is defined as:

$$\mathcal{L}_{\text{Dice-BCE}} = \lambda \mathcal{L}_{\text{Dice}} + (1 - \lambda) \mathcal{L}_{\text{BCE}}, \quad (15)$$

where $\lambda \in [0, 1]$ balances the contribution between the Dice loss and the Binary Cross-Entropy (BCE) loss.

The Dice loss $\mathcal{L}_{\text{Dice}}$ is calculated by:

$$\mathcal{L}_{\text{Dice}} = 1 - \frac{2 \sum_{i=1}^N p_i y_i + \epsilon}{\sum_{i=1}^N p_i + \sum_{i=1}^N y_i + \epsilon}, \quad (16)$$

and the BCE loss \mathcal{L}_{BCE} is given by:

$$\mathcal{L}_{\text{BCE}} = -\frac{1}{N} \sum_{i=1}^N [y_i \log(p_i) + (1 - y_i) \log(1 - p_i)], \quad (17)$$

where $p_i \in [0, 1]$ is the predicted probability for the i -th sample, $y_i \in \{0, 1\}$ is the corresponding binary label, N is the total number of samples, and ϵ is a small constant to prevent division by zero.

The Focal Tversky loss $\mathcal{L}_{\text{Focal-Tversky}}$ is defined as:

$$\mathcal{L}_{\text{Focal-Tversky}} = (1 - \text{Tversky Index})^\gamma, \quad (18)$$

where $\gamma \geq 1$ is the focal parameter that emphasizes harder-to-classify examples.

The Tversky Index is calculated by:

$$\text{Tversky Index} = \frac{\sum_{i=1}^N p_i y_i + \epsilon}{\sum_{i=1}^N p_i y_i + \alpha \sum_{i=1}^N p_i (1 - y_i) + \beta \sum_{i=1}^N (1 - p_i) y_i + \epsilon}, \quad (19)$$

where α and β are constants that control the penalties for false positives and false negatives, respectively.

The dynamic weights ω_r are recalculated at each iteration based on the gradient norms G_r of each loss component:

$$\omega_r = \frac{\left(\frac{1}{G_r + \epsilon}\right)^\alpha}{\sum_k \left(\frac{1}{G_k + \epsilon}\right)^\alpha}, \quad (20)$$

where

$$G_r = \|\nabla_\theta \mathcal{L}_r\|. \quad (21)$$

$\nabla_\theta \mathcal{L}_r$ is the gradient of the r -th loss component with respect to the model parameters θ , and α is a tuning parameter that adjusts the sensitivity of the weight updates.

By incorporating this dynamic reweighting scheme, the model adaptively emphasizes loss components that contribute more to the overall gradient, effectively handling class imbalance. The combined loss function $\mathcal{L}_{\text{total}}$ facilitates improved convergence and performance by balancing the influence of each loss component according to the current training dynamics.

This design allows the predicted probabilities p_i (ranging from 0 to 1) to be effectively compared with the binary labels y_i in the loss computations, optimizing the model's performance on the binary classification tasks across different downstream applications.

The proposed loss function builds on the principles of Dice-BCE loss and Focal Tversky loss, combining their emphasis on focusing on harder examples with dynamic adjustment of sample weights. Below, we detail the formulation of the loss function step by step.

The Weighted Logarithmic Loss is defined as:

$$\mathcal{L}_{\text{WLL},i} = - \left(w_i^+ y_i \log(y_i' + \epsilon) + w_i^- (1 - y_i) \log(1 - y_i' + \epsilon) \right), \quad (22)$$

where: - y_i is the ground truth label for sample i , and y_i' is the predicted value. - w_i^+ and w_i^- are weights for positive and negative examples, respectively, used to handle class imbalance. - $\epsilon > 0$ is a small constant for numerical stability.

This logarithmic loss penalizes predictions more harshly when they deviate significantly from the true labels.

To prioritize harder examples, dynamic weights are calculated for each sample based on the gradient magnitude of the loss with respect to the prediction. The sample weight ω_i is defined as:

$$\omega_i = \frac{\left(\frac{1}{g_i + \epsilon}\right)^\alpha}{\sum_{j=1}^n \left(\frac{1}{g_j + \epsilon}\right)^\alpha}. \quad (23)$$

$g_i = \left| \frac{\partial \mathcal{L}_i}{\partial y_i'} \right|$ is the gradient magnitude for the i -th sample. α is a tunable exponent that controls the emphasis on harder examples (higher α increases their importance). $\epsilon > 0$ ensures numerical stability.

This weight assigns higher importance to samples with larger gradient magnitudes, focusing the training process on harder examples.

The final total loss function combines the Mean Squared Error (MSE) term and the Weighted Logarithmic Loss (WLL) term, controlled by a trade-off parameter $\beta \in [0, 1]$. For each sample i , the loss is expressed as:

$$\mathcal{L}_i = \omega_i \left[\beta (y_i' - y_i)^2 - (1 - \beta) \mathcal{L}_{\text{WLL},i} \right]. \quad (24)$$

The MSE term, $(y_i' - y_i)^2$, penalizes the squared error between the prediction and ground truth. The WLL term, $\mathcal{L}_{\text{WLL},i}$, introduces a logarithmic penalty for classification errors. - β balances the trade-off between MSE and WLL.

The total loss over all n samples is:

$$\mathcal{L}_{\text{total}} = \frac{1}{n} \sum_{i=1}^n \omega_i [\beta(y'_i - y_i)^2 - (1 - \beta)\mathcal{L}_{\text{WLL},i}]. \quad (25)$$

Substituting the definition of WLL into the equation, the total loss becomes:

$$\mathcal{L}_{\text{total}} = \frac{1}{n} \sum_{i=1}^n \omega_i \left[\beta(y'_i - y_i)^2 - (1 - \beta) \left(w_i^+ y_i \log(y'_i + \epsilon) + w_i^- (1 - y_i) \log(1 - y'_i + \epsilon) \right) \right]. \quad (26)$$

Dynamic Weighting (ω_i): Focuses on harder examples by assigning larger weights to samples with higher gradient magnitudes. Trade-off Hyperparameter (β): Controls the balance between the MSE term (for regression-like behavior) and the WLL term (for robust classification). By integrating MSE and WLL dynamically, the approach benefits from both regression accuracy and log-scale penalties, while adapting to the varying difficulty of examples. This loss formulation retains the focus on harder examples, similar to Focal Tversky and Dice-BCE, while unifying classification and regression objectives in a single framework.

C. Experiments

C.1. Training Details and Evaluation Metric

The integration of these metrics, along with the Kolmogorov-Smirnov statistic, furnishes a multi-faceted evaluation framework for the risk assessment model. The Kolmogorov-Smirnov statistic, by measuring the maximum divergence in cumulative distributions between positive and negative classes, directly addresses the model’s discriminative power. Similarly, the AUC provides a measure of the model’s ability to differentiate between classes over the entire range of classification thresholds. Table 7 shows the key hyperparameters used in the training of the risk assessment model and their set values.

Parameter	Value	Description
Optimizer	AdamW	Optimizer with weight decay for better generalization.
Learning Rate	0.0005	Step size for parameter updates.
Betas	(0.9, 0.999)	Decay rates for first and second moment estimates in Adam.
Epsilon	1e-08	Smoothing term to avoid division by zero.
Weight Decay	0.01	Regularization to prevent overfitting.
Balancing Coefficients α	0.5	Balances different loss components.
Balancing Coefficients β	0.5	Balances regression (MSE) and classification (WLL).
Balancing Coefficients γ	1.0	Focuses on hard-to-learn samples.
Epochs	12	Number of complete passes through the training dataset.

Table 7: Training Parameters and Descriptions.

C.2. Case Study

In a case study, we analyzed two models: XGBoost (XGB) and FinLangNet. The XGB utilized manually engineered features informed by domain expertise and relied on a single feature for making predictions. FinLangNet, however, constructed sequential features from a subset of the original features using advanced algorithms to extract insights without preconceived notions, and it broadened its scope by applying six different labels to enrich the feature set.

A fair comparison required focusing on the common label $y_1(\tau = 1)$. The evaluation of model performance was conducted over a one-month future testing window. In the context of risk control, the threshold marking a "high risk" case is essential. As seen in Figure 5, the XGB achieves high Recall at low thresholds, capturing most positive cases but with reduced Precision, leading to a high false positive rate. Conversely, FinLangNet displays more balanced performance, offering high Precision and slightly lower Recall at low thresholds, resulting in fewer misclassifications.

Our examination of the $y_1(\tau = 1)$ label as illustrated in Figure 6 revealed that FinLangNet outperforms the XGBoost model in identifying users at high risk of default. Even though it sometimes incorrectly classifies low-risk users, FinLangNet was

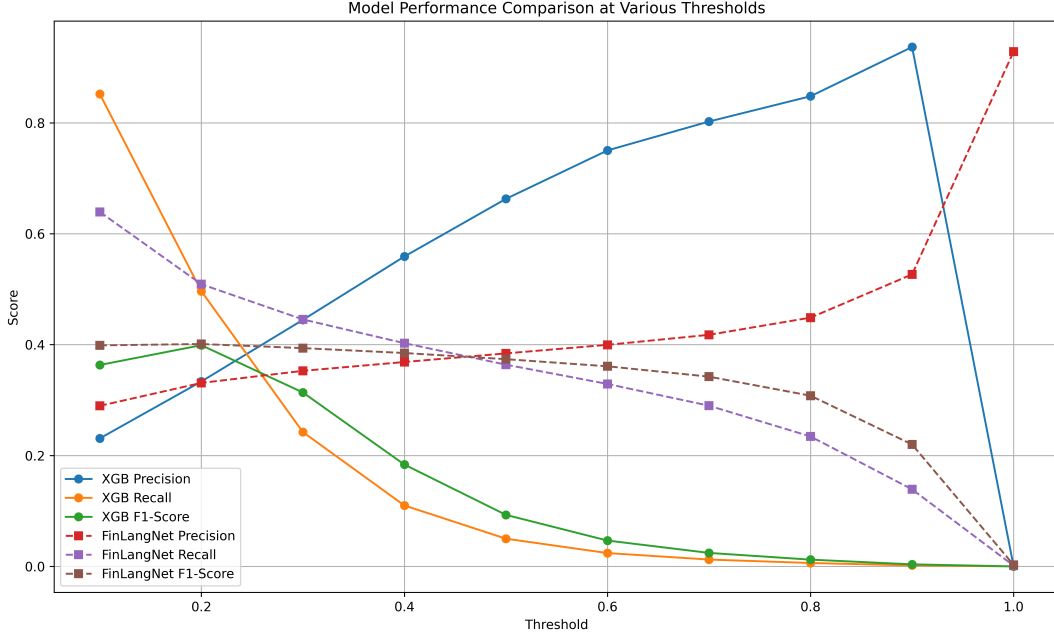


Figure 5: Model Performance Comparison at Various Thresholds on the $y_1(\tau = 1)$.

found to be more adaptable and robust against imbalanced labels, ensuring it retains its superior performance in skewed datasets.

C.3. Multivariate Time Series Classification Task

C.3.1. BASELINES DETAILS

For classification, we refer to the SOTA methods: EDI, DTWI, and DTWD (Bagnall et al., 2018) rely on calculating distances reflective of temporal warping or deviations in time sequences. MLSTM-FCNs (Karim et al., 2019) combine LSTM and CNN layers for feature generation, while WEASEL-MUSE (Schäfer and Leser, 2017) transforms series into symbolic representations. Scalable Representation Learning (SRL) (Franceschi et al., 2019) builds representations via negative sampling and encoders, TapNet (Zhang et al., 2020) is a recent model with an attentional prototype learning in its deep learning-based network, ShapeNet (Li et al., 2021) discovers shapelets through clustering in a unified space, and Rocket (Dempster et al., 2020) employ random convolutional kernels for rapid feature extraction from univariate series. These methods like TStamp (Zerveas et al., 2021) and SVP-T (Zuo et al., 2023) collectively elevate the precision and understanding of temporal data analysis.

C.3.2. RESULTS

Impressively, even bereft of the full suite of FinLangNet capabilities, the core module’s performance was observed to either mirror or exceed the benchmarks established by the currently trending models in the domain. The specifics of our performance metrics and the accuracy scores for these datasets are meticulously detailed in Table 8.

It is important to note that two most recently SOTA models in Multivariate Time Series Classification, such as TimesNet (Wu et al., 2022) and ShapeFormer (Le et al., 2024). We tested the performance of TimesNet on our dataset, as shown in Table 1. Regarding the shapeFormer, although it performs exceptionally well on the UEA archive, it could not be validated in our setting. This limitation is due to the model’s dependency on the Shapelet module, which requires the extraction of discriminative sub-sequences from the entire dataset in advance. Our scenario, characterized by an ultra-large-scale data stream, makes the implementation of this approach impractical. Therefore, we were unable to verify the method.

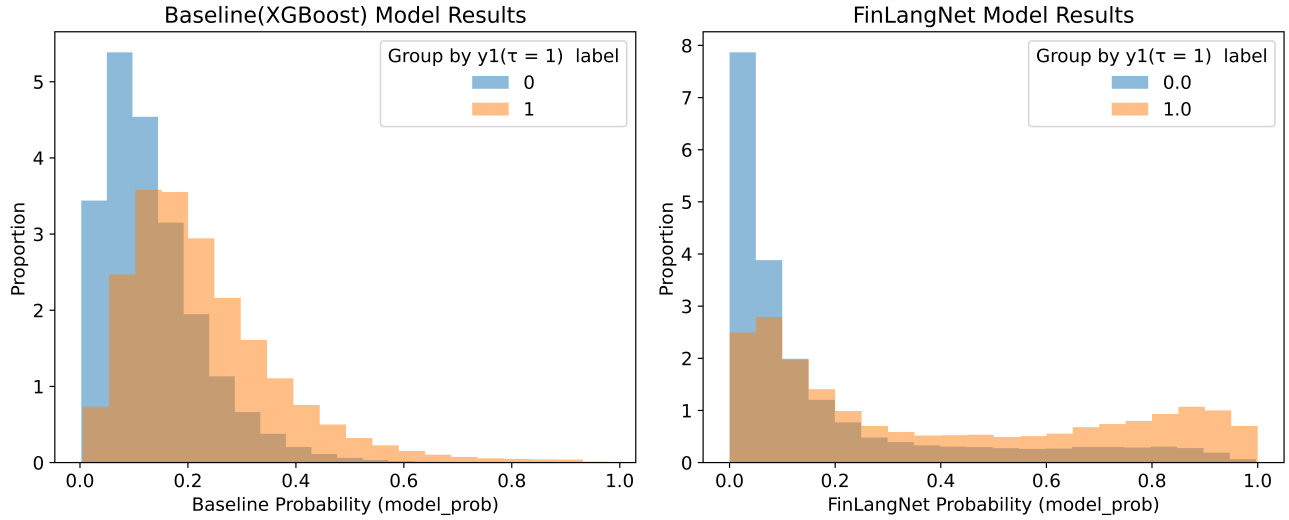


Figure 6: Comparison of Predictive and the $y_1(\tau = 1)$ Distributions for XGBoost and FinLangNet.

Nonetheless, we have compared the performances of TimesNet, ShapeFormer, and FinLangNet across different datasets in Table 8 and 9.

Table 8: Accuracies on Five Datasets of the UEA Archive

	EDI	DTWI	DTWD	MLSTM-FCNs	WEASEL+MUSE	SRL	TapNet	ShapeNet	Rocket	TStamp	SVP-T	FinLangNet
AF	0.267	0.267	0.220	0.267	0.333	0.133	0.333	0.400	0.249	0.200	0.400	0.400
BM	0.676	1.000	0.975	0.950	1.000	1.000	1.000	1.000	0.990	0.975	1.000	1.000
EC	0.293	0.304	0.323	0.373	0.430	0.236	0.323	0.312	0.447	0.337	0.331	0.414
HB	0.619	0.658	0.717	0.663	0.727	0.737	0.751	0.756	0.718	0.712	0.790	0.795
SCP2	0.483	0.533	0.539	0.472	0.460	0.556	0.550	0.578	0.514	0.589	0.600	0.600

Table 9: Comparison with Two Most Recently Advanced TSC Methods on Public UEA and Our Dataset.

	TimesNet	ShapeFormer	FinLangNet
AF \diamond	N/A	0.600	0.400
BM \diamond	N/A	1.000	1.000
EC \diamond	0.357	0.378	0.414
HB \diamond	0.780	0.800	0.795
SCP2 \diamond	0.572	0.633	0.600
Our Dataset *	0.322	N/A	0.337

* Using KS as metric; \diamond Using Acc as metric.