Temal: A Time Encoding Module Augmented LLM for Financial Forecasting

Mingjun Ma[0009-0001-6264-515X] , Chenyu Wang, Ruyao Xu, Shuxiao Chen, Zhongchen Miao(*)

Innovation Lab, Shanghai Financial Futures Information Technology Co., Ltd,
Shanghai, China
miaozc@cffex.com.cn

Abstract. In the domain of time series analysis, financial forecasting presents itself as a pinnacle of intricacy. Despite the multitude of models, even those powered by cutting-edge transformer architectures, their practical efficacy on financial datasets has remained unexplored. This challenge stems from the unique nature of financial derivatives: various time scales, multifaceted attributes, and volatile patterns. Therefore, this study introduces an innovative multi-modal fine-tuning framework, which harnesses the semantic comprehension capabilities of Large Language Models (LLMs) and encodes both time-series data and its domain-specific knowledge. To mitigate the shortcomings of LLMs in capturing temporal dynamics, we propose two pivotal innovations: a Time-series Encoding Module (TEM) and a Multi-Patch Method. The TEM seamlessly embeds sophisticated temporal representation algorithms within the LLM architecture. Concurrently, the Multi-Patch Method transforms 1D time series into multiple sets of 2D tensors, each representing distinct temporal segments, thereby enriching the model's temporal analysis capabilities. Our empirical evaluations reveal that the Multi-Patch Method adeptly handles the complex temporal fluctuations across varied intervals. The proposed model outperforms other competing methods, marking a 20.2％ enhancement in forecasting accuracy for Turnover Ratio and an 9.1％ improvement in zero-shot forecasting performance. Crucially, The TEM and Multi-Patch offer modular improvements for LLM-based time-series forecasting, with potential applications across various domains.

Keywords: Time Series Forecasting, Large Language Models, Financial Derivatives.

1 Introduction

The derivatives market plays a crucial role in the financial area, essential for risk management and market efficiency. It offers hedgers tools [2] to mitigate losses from price volatility, enables speculators to transfer risks, and provides arbitrageurs with opportunities to exploit market price discrepancies. Therefore, the accurate prediction of key financial derivatives indicators, such as the Turnover Ratio [25], is paramount for comprehending and evaluating market dynamics. Numerous researchers have explored market trend predictions by employing machine learning and deep learning techniques [27]. However, the emergence of Large Language Models (LLMs) has been a game-changer [1], demonstrating extraordinary capabilities across various fields, including time-series forecasting [26]. These models show the great potential in financial forecasting [28], which aids in effective risk management on the financial markets.

Furthermore, when compared with the current mainstream deep learning-based time series models, the zero-shot learning capability of LLMs on time-series data is also noteworthy [3,4]. As studied in [19], LLMs can leverage their pre-trained knowledge, enabling them to make predictions or analyze trends with minimal task-specific training. In light of this, we conduct comparative experiments on Chinese financial derivatives datasets, specifically focusing on the models' zero-shot learning abilities, and the results reveal that our model exhibits outstanding performance in this regard.

Despite the strong generality exhibited by LLMs, it's noted that time-series data differs fundamentally from textual data, showing greater complexity and randomness in its variations, and more notably, the techniques for time-series segmenting and interpreting [30] lag behind those for text segmentation [29] in Natural Language Processing (NLP). Therefore, in our study, we propose an innovative **T**ime **E**ncoding **M**odule **A**ugmented **L**LM **(Temal)**, we adapt the structure of the LLM to handle the complexities inherent in predicting financial derivatives indicators. We conduct comprehensive assessments in both long-term forecasting tasks and zero-shot learning scenarios, evaluating our approach across daily-level and minute-level financial data, as well as across a spectrum of financial derivative products. The findings confirm that our method consistently outperforms other leading algorithms, showcasing its sustained improvements in performance under these challenging conditions.

To summarize, this paper makes three key contributions:

- **Plug-and-Play Time Encoding Upgrade in LLM Forecasting:** To deal with the challenge of stochastic volatility in financial derivatives, we introduce a replaceable Time-series Encoding Module (TEM) for Large Language Models (LLMs), which integrates a suite of cutting-edge time-series algorithms [6-8] to boost the temporal perception of LLMs.
- **Adaptive Multi-Patch Temporal Feature Extraction:** Similar to word tokenization in the NLP domain, patching [7] is an effective temporal partitioning way. To enhance the LLM's sensitivity to diverse time scales in financial time-series data, we refine our Time Encoding Module further by introducing the Multi-Patch approach. This strategy uses time patching in different time intervals to improve the model's effectiveness in representing time-series data across different time granularities.
- **Real-World Industry Application in Financial Derivatives:** This paper leverages the authentic Chinese Financial Derivative Data and showcases the practical application of our refined model in the financial derivatives market.

2 Related Work

Traditional Models: Traditional time series models applied to financial forecasting date back to the ARIMA models [9]. Subsequently, with the advent of deep neural networks, Recurrent Neural Networks (RNNs), specifically Long Short-Term Memory (LSTM) and Gated Recurrent Units (GRU) tailored for time-series forecasting tasks [10-12], gained popularity. Due to the suboptimal training efficiency of RNNs caused by their iterative structure, there has been a trend towards adopting modern time series frameworks that leverage the Transformer [13] architecture. These architectures [7] have achieved advancements in tackling the issue of long-range dependencies by using its self-attention mechanisms, significantly enhancing the performance of time series prediction tasks. Recent developments in time series forecasting within the Transformer family primarily concentrate on two aspects: the specific refinement of Transformer-based models tailored for time series data, and the application of Pretrained LLMs to these forecasting tasks.

Transformer-based: Autoformer [14] contributes to time series forecasting by innovating upon the Transformer architecture with an adaptive attention mechanism, reducing computational complexity and enhancing scalability. Informer [15] builds on this progress, employing causal sparse attention and self-attention enhancements to efficiently handle long-term dependencies in large-scale time series data. The subsequent FEDFormer [16] advances further by integrating multiple feature extraction strategies, optimizing its performance for multivariate time series prediction. PatchTST [7] introduces a novel concept from computer vision, using patches to better understand and represent complex, dynamic time series, thus refining the evolution of time series modeling techniques. When applied to time series forecasting, Pretrained LLMs can leverage their extensive learned knowledge to understand and predict patterns in time series data, even though their primary design was for language-related tasks.

Large Language Models and Prompt-based: PromptCast [17] pioneers the exploration of leveraging the inherent linguistic comprehension abilities of LLMs for temporal sequence forecasting, which capitalizes on Prompt Engineering techniques to transform numerical data into textual form, thereby engaging LLMs in predictive endeavors. It provides a fresh approach for addressing challenges in the field of time series forecasting. LLMTime [3] has a similar idea for zero-shot time series forecasting with LLMs by encoding numbers as text and sampling possible extrapolations as text completions. Another highly notable study is Google's TEMPO [18], which concentrates on time series forecasting and incorporates some processing enhancements, such as time series decomposition and soft prompts.

Large Language Models and Fintune-based: GPT4TS [19] proposes a unified framework based on partially frozen LLMs, only fine-tuning the embedding and normalization layers while keeping self-attention and feed-forward layers frozen, to achieve a state-of-the-art or comparable performance in all major types of time series analysis tasks, including time series classification, short/long-term forecasting, imputation, anomaly detection, and few-shot. Different from the above methods, a recent work Time-LLM [20] is proposed to reprogram time series with the source data modality along with natural language-based prompting to unleash the potential of LLMs as effective time series machines, which achieves state-of-the-art performance in various forecasting scenarios, as well as excels in both few-shot and zero-shot settings. Time-LLM is also lightweight and efficient, since it neither edits the input time series directly nor finetunes the backbone LLM.

3 Problem Formulation

Given the potential interactions and causal connections among various financial indicators, we choose a multivariate-to-univariate forecasting approach. So we utilize a range of indicators, from basic price and volume metrics such as closing price, opening price, highest price, lowest price, trading volume, and turnover, to more advanced technical indicators like Moving Average (MA), Relative Strength Index (RSI), and Moving Average Convergence Divergence (MACD). To enhance prediction accuracy, we utilize multivariate time series data as the input for our model, thereby taking potential interactions and influences into account in our analysis.

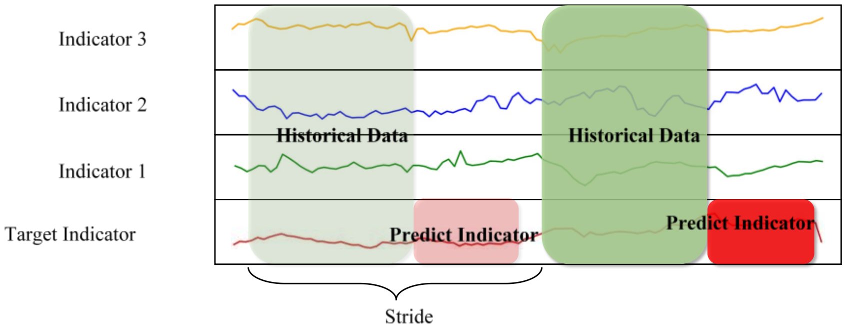

Fig. 1. Problem formulation for prediction of Turnover Ratio: Given a multivariate derivative time series, a sliding window is used to capture sequential segments. Each segment is divided into two parts: Historical Data $X_L = (x_1, \ldots, x_L)$ and Predict Indicator $X_T = (x_{L+1}, \ldots, x_{L+T})$.

As shown in Fig. 1, history data contains past observations from multiple indicators with a look-back window length of L, which is visualized in green part. The Predict Indicator, shown in red, is the target indicator to forecast.

Also, it's noted that $S_{shape}(X_L) = (B, L, M)$ and $S_{shape}(X_T) = (B, T)$, where multivariate predicts univariate. Moreover, the stride of the sliding window should be set to a value greater than or equal to 1.

- L denotes the sequence length of Historical Data.
- T represents the prediction length.
- M stands for the number of indicators.
- B indicates the batch size.

4 Methodology

In this paper, we apply a method that synergizes advanced time series representation algorithms with LLMs for financial derivatives forecasting. The overall structure of the model is depicted as follows.

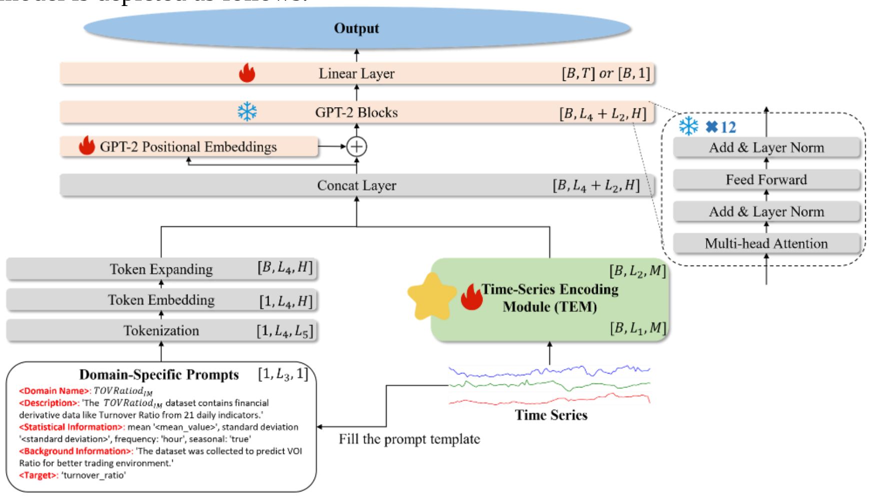
Output
Linear Layer $[B,T]$ or $[B,1]$
GPT-2 Blocks $[B,L_4+L_2,H]$
GPT-2 Positional Embeddings
Concat Layer $[B,L_4+L_2,H]$
×12
Add & Layer Norm
Feed Forward
Add & Layer Norm
Multi-head Attention
Token Expanding $[B,L_4,H]$
Token Embedding $[1,L_4,H]$
Tokenization $[1,L_4,L_5]$
$[B,L_2,M]$
Time-Series Encoding Module (TEM)
$[B,L_1,M]$
Time Series
Fill the prompt template
Domain-Specific Prompts $[1,L_3,1]$
<Domain Name>: $TOVRatiod_{IM}$
<Description>: 'The $TOVRatiod_{IM}$ dataset contains financial derivative data like Turnover Ratio from 21 daily indicators.'
<Statistical Information>: mean '<mean_value>', standard deviation '<standard deviation>', frequency: 'hour', seasonal: 'true'
<Background Information>: 'The dataset was collected to predict VOI Ratio for better trading environment.'
<Target>: 'turnover_ratio'

Fig. 2. A multi-modal framework based on the LLM. Marked with the red flame, TEM and GPT-2 positional embeddings and output linear layer are open for training and other parts are frozen, where B as batch size, M as feature size, H as hidden size of LLMs, L as sequence length, and T as prediction length.

Instead of direct fine-tuning of the LLM's original parameters, we use an approach that combines a fixed template prompt and a custom time-series encoding module to adapt to the prediction task. This approach both utilizes the original capabilities of the LLM and extends its understanding of times-series data through a specific design.

As shown in the left corner of Fig. 2, we design Domain-Specific Prompts based on a template for prompt tuning, aimed at enhancing the model's comprehension of product fundamental information. Each prompt is constructed by extracting relevant features from every batch of time-series data. Prompts are tokenized using Byte Pair Encoding (BPE) [22], and each token is transformed into a fixed-size vector $(1, L_4, \mathrm{H})$ through the token embedding layer. The token embedding layer utilizes the pre-trained embeddings from GPT-2, which is retained for language understanding.

Simultaneously, multivariate financial time series undergoes transformation via a Time-Series Encoding Module (TEM), which can be a time series algorithm, like TimesNet [6] and PatchTST [7], to enhance temporal representation. This pivotal step entails a transformation of the input data, reshaping it from an initial format of $(\mathrm{B}, L_1, \mathrm{M})$ to an intermediary configuration that aligns seamlessly with the LLMs' requirements, shown as $(\mathrm{B}, L_2, \mathrm{H})$. Notably, during the temporal encoding stage, the second dimension expands from L_1 to L_2, reflecting the altered sequence length. Concurrently, the dimension M undergoes a transformation into H to match the hidden layer size of the LLM, thereby ensuring compatibility and optimal interaction between the encoded data and the model's architecture.

This rich information combined from time series and textual prompts is sent to the attention layers of LLMs. To maintain the original capabilities of the LLM, we freeze all 12 layers of the GPT-2. We use the last linear layer to connect the GPT-2 Blocks and output the prediction results, this layer can be a linear or dense layer. Our approach is highly scalable and each module in the framework is replaceable.

4.1 Time-Series Encoding Module (TEM)

In the construction of Time-Series Encoding Module, we utilize those well-validated methods such as TimesNet [6] and the NonStationary Transformer [23] to enhance the expressiveness of time series data. Specifically:

TimesNet excels in identifying and extracting dynamic characteristics in financial futures data, good at capturing short-term volatility and irregular price movement. This methodology employs the TimesBlock to capture information across different time granularities. It transforms a 2D tensor into multiple variations with inherent periodicity and inter-periodicity changes. This is achieved through a parameter-efficient inception block designed to discover multiple periods. Such an approach effectively overcomes the representational limitations typically associated with 1D time series. NonStationary transformer focuses on analyzing and interpreting the non-stationarity in time series data, including changes in long-term trends, seasonal fluctuations, and structural breaks.

These algorithms ensures that the time series, before being passed to the large model, contains richer and more regularized features. For different time series representation algorithms, some layer fitting work is necessary to ensure that their outputs can be effectively integrated into the large model. This can be achieved by concatenating the data or adding feed-forward layers, ensuring seamless integration of diverse time series representations, thus providing the LLM with comprehensive input data.

Furthermore, to address the issue of inadequate time series representation capabilities in existing approaches, we propose a new TEM algorithm: **Multi-Patch**, as shown in Fig. 3. Inspired by [7], this method improves PatchTST with more patches to capture multi-granularity temporal features. Multi-Patch has two advantages: First, By dividing a continuous financial time series into a series of temporal windows or patches, the model gains the ability to delve deeper into the temporal characteristics within each segment, including short-term fluctuations, seasonality, cyclical changes, and long-term trends. Second, LLMs typically excel at processing and interpreting contextual relationships and dependency structures in natural language texts. When applied to properly partitioned time series data, LLMs can draw upon their abilities in semantics and reasoning to parse the underlying logic and structural information within time series data.

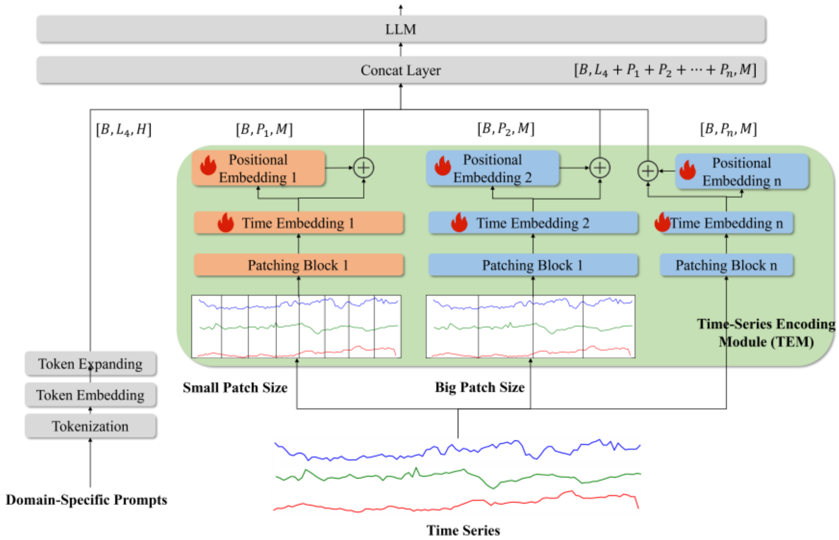

Fig. 3. When TEM is Multi-Patch, the time series is split into multiple groups with different patch sizes, maintaining the independence of multiple channels. Each patching module has independent time embedding and positional embedding layers, both of which are open for training. Finally, the multi-channel data is gathered with token information, then output to the LLM.

4.2 Domain-Specific Prompts

The LLMs inherently excel in language understanding. Related studies [20] have found that integrating temporal and textual information can enhance the performance of time-series tasks. Therefore, this paper concatenates the word embeddings of prompts with temporal representations, maximizing the utilization of pretrained LLMs.

For efficiency in practical usage, we employ a simple domain-specific prompt template for ease of construction. This prompt integrates information from 5 aspects, as illustrated in Fig. 1: *Domain Name, Description, Statistical Information, Background Information of the task, and the Prediction Target*. The statistical information of the dataset is aggregated based on each batch serving as a statistical window. It should be noted that the length of the prompts cannot be unlimited. To ensure that the time series data plays a dominant role, we limit the number of words in the prompt: $N_W \leq 64$ and token length is smaller than time length $L_4 \leq L_2$.

5 Experiment

Our objective is to validate the feasibility and effectiveness of our method in forecasting financial deritives indicators, and subsequently identify its strengths and superiority.

5.1 Experiment Settings

Datasets: We collect a financial time series dataset comprising Chinese financial derivatives, which includes data on stock index futures and their corresponding options. This dataset is broadly classified into seven product types, including four futures products: IC, IF, IH, IM; and three options products: IO, MO, HO. The timeframe for all products extend from their respective inception dates to January 12, 2024, except IF which covers only from September 7, 2015 to January 12, 2024. These datasets consist of two distinct subsets: the Minute-level Financial Futures Dataset (FFDm) and the Daily Financial Futures Dataset (FFDd).

Minute-Level Financial Futures Dataset (FFDm) contains approximately 3 million records, focusing on micro-level analyses and intraday metrics, where metrics include latest PRICE, BASIS and VIX. The minute-level price updates that closely track and analyze the price movements of futures contracts in time. And the basis between spot and futures prices, a key component of this dataset, serves as an essential gauge for monitoring market expectations, supply-demand dynamics, and potential risks. The VIX is a measure of market risk and investor sentiment, highly related with options and it often increases when investors anticipate that stock prices will become more erratic. A higher VIX value typically indicates a higher level of fear or uncertainty in the market. The VIX is used to understand the mood of the market, hedge against market downturns, or speculate on changes in market volatility.

Daily Financial Futures Dataset (FFDd) holds 9,580 daily observations, capturing medium to long-term trends. Among its features are critical indicators such as the Turnover Ratio [25], which provides insights into market sentiment and participation levels. In addition, it offers daily oscillation measures including the Relative Strength Index (RSI) and Moving Average Convergence Divergence (MACD), empowering users to identify pivotal turning points and shifts in momentum within prices.

SOTA Benchmarks: Beyond validating the feasibility of our method, to further analyze the performance of our model, we replicate other SOTA methods and compare our approach against these up-to-date methods. Our baselines include deep network methods such as TimesNet [6], and Transformer-based methods like PatchTST, GPT4TS, and TimeLLM. Notably, among these Transformer based methods, we also incorporate a commercialized model, TimeGPT-1 [24].

Model Details: In our method, we utilize GPT-2 as the backbone. Besides, the Time-Series Encoding Module (TEM) adopts two distinct variants: $Temal_{tn}$, which utilizes TimesNet as the TEM part, and $Temal_{mp}$, employing a Multi-Patch approach. Specifically, in $Temal_{tn}$, the encoding layers of TimesNet $E_L = 2$ are configured to 2. Meanwhile, $Temal_{mp}$ integrates three different patch sizes, 16, 8, and 3. The base model is GPT-2, with number of layers $T_L = 12$ and hidden size $\mathrm{d} = 512$. In this setup, all parameters of GPT-2, except for the position embeddings, are frozen. And the batch size $\mathrm{B} = 64$, the stride of sliding window $\mathrm{s} = \frac{1}{2} * \mathrm{L}$, L is the predict length.

Devices: Our fine-tuning and testing processes were all conducted using 4 A100 GPUs, each with 40GB of memory. Given that the base model underlying Temal is GPT-2, most models in this paper could be fine-tuned on a single GPU. However, for TimeLLM specifically, multiple GPUs are used due to its parameter sizes. And other configurations: we follow the pipeline of the project **Time-series-library** [21], which includes the implementation of other baselines, utilizing PyTorch as our training tool, and Python 3.9.12, Torch 1.13.0, Pandas 1.4.2, Scikit-Learn 1.0.2.

5.2 Long-Term Forecasting

Long-term Forecasting on dataset FFDd. Due to the limited size of product IM, we select only IC, IF, and IH for our product. Experiments were conducted with configurations of prediction length {30-15, 30-30, 30-60}. We choose three indicators: Daily Turnover Ratio (TOVRatiod), Daily Relative Strength Index (RSId), and Daily Moving Average Convergence Divergence (MACDd). In the course of our experiments, we unify the parameters across different methods. The experiments are repeated three times for epochs of 3, 5, and 10, and their average results were reported.

Due to the small magnitudes of features such as Turnover Ratio, we opt for Root Mean Square Error (RMSE) and Mean Absolute Error (MAE) as our evaluation metrics to ensure clear discrimination, and we use Mean Square Error (MSE) as the loss function.

MAE measures the average magnitude of the errors in a set of predictions, without considering their direction. MSE calculates the average of the squares of the errors, which means it gives more weight to larger errors. RMSE is the square root of the MSE. It's a measure that provides the standard deviation of the prediction errors, showing how spread out the errors are. Where, n is the number of observations, y_i is the actual value for the i-th observation, and $\hat{y_i}$ is the predicted value for the i-th observation. The full results of long-term forecasting on FFDd are summarized in Tab. 1. Evaluation of $Improvement$ is a synthesis of RMSE and MAE, where IMP$_{RMSE}$ is the improvement of our method compared to the average RMSE of PatchTST, TimesNet, TimeLLM and Tempo, and IMP_{MAE} is the improvement of our method compared to other algorithms using average MAE.

$$\text{RMSE} = \sqrt{\frac{1}{n} * \sum_{i=1}^{n} (y_i - \hat{y_i})^2} \tag{1}$$

$$\text{MAE} = \frac{1}{\text{n}} * \sum_{i=1}^{n} |y_i - \hat{y_i}| \tag{2}$$

$$\text{MSE} = \frac{1}{n} * \sum_{i=1}^{n} (y_i - \hat{y_i})^2 \tag{3}$$

$$\text{Improvement} = \frac{IMP_{RMSE} + IMP_{MAE}}{2} \tag{4}$$

As illustrated in Tab. 1, our method demonstrates better performance on the dataset *TOVRatiod*. In an overarching assessment across all time series forecasts, the improvements on Datasets *TOVRatiod*, *RSId*, and *MACDd* reach **10.8**%, **2.2**% and **3.5**% respectively. Particularly, for the IH product, our method exhibits a significant enhancement compared to other SOTA models, with Turnover Ratio predictions showing an impressive improvement of **20.2**%. Moreover, the table data clearly indicates that our $Temal_{mp}$ method performs well in the {30-60} prediction tasks, achieving an overall improvement of **6.1**%.

Long-term Forecasting on FFDm. FFDm has a more abundant data volume, we select the futures product IM and options products IO, HO, MO for experimentation, with configurations of prediction length {60-60, 60-120, 60-240}. For the product IM, we choose PRICE and BASIS as the forecasting targets to construct the datasets PRICEm$_{IM}$ and BASISm$_{IM}$. In the case of options, we choose VIX as the forecasting target to build the datasets VIXm$_{IO}$, VIXm$_{MO}$, and VIXm$_{HO}$. The experiments are also repeated three times for epochs of 3, 5, and 10, and their average results were reported.

Table 1. Daily Financial Futures Dataset Prediction. RMSE, MAE for {30-15, 30-30, 30-60} prediction on Datasets TOVRatiod, RSId, MACDd of FFDd, comparing Temal$_{mp}$, $Temal_{tn}$with PatchTST, TimesNet, TimeLLM, Tempo. The numbers with bold denote the best.

Datasets	**Prod**	**P**	**Temal$_{mp}$**		**Temal$_{tn}$**		**PatchTST**		**TimesNet**		**TimeLLM**		**Tempo**	
			RMSE	**MAE**	**RMSE**	**MAE**	**RMSE**	**MAE**	**RMSE**	**MAE**	**RMSE**	**MAE**	**RMSE**	**MAE**
TOV-Ratiod	IC	15	0.056	**0.045**	**0.055**	0.046	0.055	0.046	0.057	0.058	0.061	0.05	0.055	0.046
		30	**0.058**	0.061	0.061	**0.049**	0.066	0.054	0.067	0.054	0.068	0.055	0.068	0.056
		60	0.051	0.043	**0.051**	**0.041**	0.065	0.05	0.061	0.047	0.065	0.05	0.064	0.049
	IF	15	0.683	0.573	0.787	0.679	0.649	**0.538**	0.71	0.582	0.688	0.572	**0.6**	0.58
		30	0.759	0.649	0.834	0.7	0.743	0.625	0.795	0.673	0.759	0.631	**0.729**	**0.617**
		60	**0.633**	**0.525**	0.733	0.608	0.676	0.551	0.778	0.625	0.675	0.551	0.664	0.545
	IH	15	**0.074**	**0.06**	0.077	0.064	0.078	0.061	0.085	0.068	0.088	0.065	0.086	0.067
		30	**0.077**	**0.065**	0.078	0.065	0.087	0.069	0.095	0.075	0.089	0.07	0.097	0.076
		60	**0.072**	**0.059**	0.073	0.059	0.094	0.075	0.108	0.087	0.089	0.071	0.1	0.078
RSId	IC	15	1.113	0.918	1.168	0.96	1.17	0.964	**1.058**	**0.862**	1.111	0.907	1.111	0.908
		30	1.159	0.954	1.173	0.978	1.153	0.947	**1.08**	**0.884**	1.224	0.992	1.155	0.949
		60	1.114	0.918	1.186	0.982	1.17	0.965	**1.108**	**0.912**	1.179	0.968	1.156	0.951
	IF	15	**1.096**	0.9	1.157	0.963	1.096	**0.899**	1.68	0.883	1.1	0.906	1.115	0.919
		30	**1.12**	1.1006	1.207	1.013	1.147	0.955	1.128	**0.932**	1.133	0.942	1.148	0.957
		60	**1.102**	**0.892**	1.201	1.017	1.208	0.994	1.16	0.959	1.2	0.988	1.211	0.955
	IH	15	1.07	0.873	1.102	0.898	1.074	0.979	**1.042**	**0.857**	1.072	0.873	1.068	0.872
		30	1.134	0.94	1.15	0.952	1.144	0.942	1.099	**0.907**	1.134	0.937	**1.128**	0.93
		60	**1.12**	**0.911**	1.2	1	1.174	0.997	1.121	0.918	1.178	0.97	1.178	0.973
MACDd	IC	15	0.292	0.214	0.353	0.264	0.296	0.215	0.292	0.214	**0.283**	**0.209**	0.306	0.223
		30	0.388	0.286	0.454	0.342	0.39	0.285	**0.363**	**0.273**	0.387	0.289	0.432	0.32
		60	**0.415**	**0.307**	0.468	0.35	0.423	0.313	0.453	0.34	0.435	0.318	0.433	0.324
	IF	15	0.57	0.445	0.582	0.441	**0.56**	**0.42**	0.563	0.441	0.568	0.426	0.571	0.435
		30	**0.687**	**0.545**	0.792	0.599	0.775	0.583	0.722	0.561	0.751	0.582	0.783	0.613
		60	**0.908**	**0.709**	0.947	0.734	0.947	0.73	0.979	0.77	0.913	0.719	0.94	0.737
	IH	15	0.512	**0.377**	0.521	0.401	**0.509**	0.379	0.51	0.394	0.545	0.406	0.512	0.377
		30	0.717	0.536	0.745	0.54	0.735	0.545	**0.68**	**0.53**	0.742	0.564	0.716	0.535
		60	**0.795**	**0.625**	0.945	0.712	0.898	0.672	0.824	0.632	0.889	0.67	0.931	0.73

As shown in Tab. 2, we compare the Temal series with TimesNet and TEMPO. The best experimental results are highlighted in bold. The results demonstrate that the Temal series also performs well in forecasting on minute-level datasets, particularly the Temal$_{mp}$, which reuse the TimesNet structure and achieve a **5.7**% improvement in RMSE and **7.4**% in MAE over TimesNet itself. And we also find that compared to making particularly long sequence predictions of {60-120} and {60-240}, our method excels more in mid-range predictions like {60-60}. When the prediction length exceeds a certain range, the predictive capability diminishes.

Table 2. Minute-Level Financial Futures Datset Prediction, RMSE, MAE for {60-60, 60-120, 60-240} prediction on Datasets PRICEm$_{IM}$, BASISm$_{IM}$, VIXm$_{IO}$, VIXm$_{MO}$, VIXm$_{HO}$ and comparing Temal$_{mp}$, Temal$_{tn}$ with PatchTST, TimesNet and Tempo. The numbers with bold denote the best.

Datasets	**P**	**Temal$_{\text{mp}}$**		**Temal$_{tn}$**		**TimesNet**		**TEMPO**	
		RMSE	**MAE**	**RMSE**	**MAE**	**RMSE**	**MAE**	**RMSE**	**MAE**
PRICEm$_{IM}$	60	0.14	0.094	0.135	**0.83**	0.151	0.1	**0.13**	0.091
	120	0.163	0.112	**0.157**	**0.108**	0.165	0.113	0.16	0.109
	240	0.193	0.138	**0.184**	**0.133**	0.189	0.135	0.185	0.133
BASISm$_{IM}$	60	0.48	0.359	**0.458**	**0.32**	0.503	0.379	0.48	0.359
	120	**0.556**	**0.416**	0.556	0.42	0.575	0.437	0.556	0.417
	240	0.626	0.488	**0.598**	**0.466**	0.625	0.486	0.654	0.502
VIXm$_{IO}$	60	**0.161**	**0.123**	0.162	0.124	0.179	0.136	0.162	0.123
	120	**0.182**	**0.14**	0.184	0.143	0.214	0.154	0.187	0.143
	240	0.211	0.164	0.214	0.165	0.24	0.186	**0.207**	**0.161**
VIXm$_{MO}$	60	0.246	0.169	**0.243**	**0.165**	0.245	0.172	0.242	0.166
	120	0.281	0.194	0.279	0.194	0.28	0.194	**0.278**	**0.19**
	240	0.317	0.212	0.318	0.221	0.328	0.24	**0.311**	**0.221**
VIXm$_{HO}$	60	0.268	0.147	**0.258**	**0.139**	0.264	0.149	0.26	0.141
	120	**0.273**	**0.159**	0.282	0.166	0.297	0.182	0.283	0.166
	240	**0.302**	**0.191**	0.304	0.192	0.341	0.221	0.31	0.195
AVG		0.293	0.207	**0.288**	**0.202**	0.306	0.219	0.294	0.208

5.3 Zero-Shot Forecasting

To assess the zero-shot prediction capabilities of different methods, we initially train our model using generic datasets: ETT datasets (ETTh1, ETTh2, ETTm1, ETTm2), and Weather datasets [6]. The results of the zero-shot prediction tasks are presented in Tab. 3, our Temal$_{mp}$ method achieves an impressive improvement **9.1**%, reflected as a **4.3**% improvement compared to the second best method TimeGPT-1, a **17.9**% improvement over TimesNet, and **5.2**% improvement over GPT4TS.

As shown in Fig. 4, we compare the performance of our $Temal_{mp}$ method with TimeGPT-1 in the {30-15} forecast. It can be observed that TimeGPT-1 makes more conservative predictions with smaller fluctuations, while our method exhibits higher variability. In terms of curve fitting and trend prediction, our method aligns more closely with the ground-truth.

Table 3. RMSE on daily financial Dataset of TOVRatiod compared with GPT4TS, TimeGPT-1. The numbers with bold denote the best.

	Methods	**Temal$_{\mathbf{mp}}$**	**GPT4TS**	**TimeGPT-1**	**TimesNet**
DataSets	**Period**	**RMSE**	**RMSE**	**RMSE**	**RMSE**
	30-15	**0.112**	0.122	0.118	0.138
$TOVRatiod_{IC}$	30-30	**0.141**	0.147	0.144	0.178
	30-60	**0.125**	0.136	0.133	0.183
	30-15	0.178	0.184	**0.176**	0.201
$TOVRatiod_{IF}$	30-30	**0.217**	0.231	0.224	0.262
	30-60	**0.199**	0.211	0.217	0.245
	30-15	**0.178**	0.186	0.178	0.211
$TOVRatiod_{IH}$	30-30	**0.208**	0.219	0.218	0.5
	30-60	**0.189**	0.195	0.214	0.214

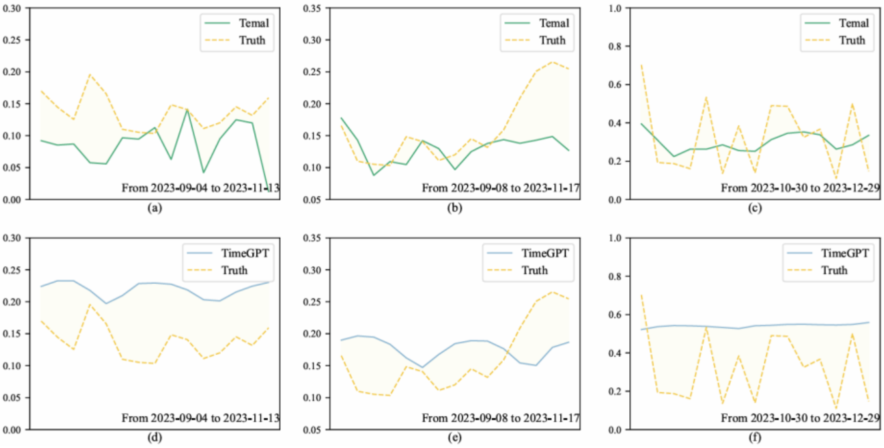

Fig. 4. The six images represent comparisons between Temal, TimeGPT-1, and ground truth for the 30-15 forecast task on different dates. The upper images (a), (b), and (c) display the prediction curves of Temal, while the lower images labeled (d), (e), and (f) illustrate the prediction curves of TimeGPT-1.

6 Discussion and Future Direction

This study explores the efficacy of Large Language Models (LLMs) in financial forecasting through significant modifications, including the integration of a Time-Series

Encoding Module (TEM), the development of a Multi-Patch technique, and the adoption of a hybrid fine-tuning approach that combines prompts with time-series data. Despite these enhancements, the model encounters two primary challenges. Firstly, the acquisition of time-series data is more difficult than that of textual data, which hinders the full utilization of LLMs' extensive parameters during fine-tuning with limited datasets, potentially leading to suboptimal model performance. Secondly, in line with the feature collapse phenomenon detailed in [31], a similar issue has been observed in LLMs when processing time-series data, further complicated by the significant difference between time and word embeddings, potentially deteriorating the model's effectiveness.

In our future work, our research will focus on two main objectives. Initially, we plan to expand our dataset and conduct fine-tuning using a more varied collection of data, including public opinion metrics and sentiment analysis, to improve the model’s generalization capabilities. Furthermore, we aim to thoroughly investigate the feature collapse issue within LLMs and devise specific strategies to address it, especially in terms of their applicability to time-series analysis. Our ultimate goal is to enhance the adaptability and efficiency of LLMs for handling time-series data, making them more suitable for such applications.

References

1. Yu, X., Chen, Z., Ling, Y., Dong, S., Liu, Z., Lu, Y.: Temporal Data Meets LLM-Explainable Financial Time Series Forecasting (2023)
2. Julio J. Lucia, Angel Pardo: On measuring speculative and hedging activities in futures markets from volume and open interest (2010)
3. Nate Gruver and Marc Anton Finzi and Shikai Qiu and Andrew Gordon Wilson.: Large Language Models Are Zero-Shot Time Series Forecasters
4. Liu, H., Zhao, Z., Wang, J., Kamarthi, H., Prakash, B.A.: LSTPrompt: Large Language Models as Zero-Shot Time Series Forecasters by Long-Short-Term Prompting (2024)
5. Radford, A., Wu, J., Child, R., Luan, D., Amodei, D.,Sutskever, I.: Language Models are Unsupervised Multitask Learners (2019)
6. Wu, H., Hu, T., Liu, Y., Zhou, H., Wang, J., Long, M.: Timesnet: Temporal 2d-variation modeling for general time series analysis (2022)
7. Nie, Y., Nguyen, N. H., Sinthong, P., Kalagnanam, J.: A time series is worth 64 words: Long-term forecasting with transformers (2022)
8. Liu, Y., Wu, H., Wang, J., Long, M.: Non-stationary transformers: Exploring the stationarity in time series forecasting (2022) Advances in Neural Information Processing Systems, 35, 9881-9893.
9. Nelson, B. K.: Time series analysis using autoregressive integrated moving average (ARIMA) models (1998) Academic emergency medicine, 5(7), 739-744.
10. Medsker, L. R., Jain, L. C.: Recurrent neural networks (2001) Design and Applications, 5(64-67), 2.
11. Kim, S., Kang, M.: Financial series prediction using Attention LSTM (2019)
12. Shen, G., Tan, Q., Zhang, H., Zeng, P., Xu, J.: Deep learning with gated recurrent unit networks for financial sequence predictions (2018) Procedia computer science, 131, 895-903.
13. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N.,...Polosukhin, I.: Attention is all you need (2017). Advances in neural information processing systems, 30.

14. Wu, H., Xu, J., Wang, J., Long, M.: Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting (2021) Advances in Neural Information Processing Systems, 34, 22419-22430.
15. Zhou, H., Zhang, S., Peng, J., Zhang, S., Li, J., Xiong, H., Zhang, W.: Informer: Beyond efficient transformer for long sequence time-series forecasting (2021, May) In Proceedings of the AAAI conference on artificial intelligence (Vol. 35, No. 12, pp. 11106-11115).
16. Zhou, T., Ma, Z., Wen, Q., Wang, X., Sun, L., Jin, R.: Fedformer: Frequency enhanced decomposed transformer for long-term series forecasting (2022, June) In International Conference on Machine Learning (pp. 27268-27286). PMLR.
17. Xue, H., Salim, F. D. Promptcast: A new prompt-based learning paradigm for time series forecasting. (2023). IEEE Transactions on Knowledge and Data Engineering.
18. Cao, D., Jia, F., Arik, S. O., Pfister, T., Zheng, Y., Ye, W., Liu, Y.: Tempo: Prompt-based generative pre-trained transformer for time series forecasting. (2023).
19. Zhou, T., Niu, P., Sun, L., Jin, R.:One fits all: Power general time series analysis by pre-trained lm. (2024). Advances in neural information processing systems, 36.
20. Jin, M., Wang, S., Ma, L., Chu, Z., Zhang, J. Y., Shi, X., ...Wen, Q.: Time-llm: Time series forecasting by reprogramming large language models. (2023).
21. https://github.com/thuml/Time-Series-Library
22. Rico S., Barry H., and Alexandra B.: Neural Machine Translation of Rare Words with Subword Units. (2016) In Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 1715–1725, Berlin, Germany. Association for Computational Linguistics.
23. Yong L. and Haixu W. and Jianmin W. and Mingsheng L.: Non-stationary Transformers: Exploring the Stationarity in Time Series Forecasting (2022) Advances in Neural Information Processing Systems
24. TimeGPT-1 https://docs.nixtla.io/
25. C. Wang et al.: Modeling Price and Risk in Chinese Financial Derivative Market with Deep Neural Network Architectures (2020) 5th International Conference on Computational Intelligence and Applications (ICCIA), Beijing, China, pp. 13-18
26. Zhang, X., Chowdhury, R.R., Gupta, R.K., Shang, J. (2024). Large Language Models for Time Series: A Survey. ArXiv, abs/2402.01801.
27. Ankit T., Kinjal C.: A comprehensive survey on deep neural networks for stock market: The need, challenges, and future directions. (2021) Expert Systems with Applications, Volume 177, 2021, 114800, ISSN 0957-4174
28. Zhao, H., Liu, Z., Wu, Z., Li, Y., Yang, T., Shu, P., Xu, S., Dai, H., Zhao, L., Mai, G., Liu, N., Liu, T. (2024). Revolutionizing Finance with LLMs: An Overview of Applications and Insights. ArXiv, abs/2401.11641.
29. Mielke, S.J., Alyafeai, Z., Salesky, E., Raffel, C., Dey, M., Gallé, M., Raja, A., Si, C., Lee, W.Y., Sagot, B., Tan, S. (2021). Between words and characters: A Brief History of Open-Vocabulary Modeling and Tokenization in NLP. ArXiv, abs/2112.10508.
30. Keogh, Eamonn, Chu, Selina, Hart, David, Pazzani, Michael. (2003). Segmenting Time Series: A Survey and Novel Approach. Data Mining in Time Series Databases. 57. 10.1142/9789812565402_0001.
31. Tang, Yehui and Han, Kai and Xu, Chang and Xiao, An and Deng, Yiping and Xu, Chao and Wang, Yunhe.: Augmented Shortcuts for Vision Transformers. (2021) Advances in Neural Information Processing Systems