OPTIMIZATIONS IN APPLIED MACHINE LEARNING
Research Article | Volume 4 | Issue 4 | Dec 2024
Received: 4 Nov 2024 | Revised: 30 Nov 2024
Accepted: 3 Dec 2024 | Published Online: 9 December 2024

An End-to-End Stock Recommendation Algorithm Study Based on Time-Frequency Consistency

Shuguang Xiong, Microsoft Inc., Email: shuxiong@microsoft.com;

Huitao Zhang*, Northen Arizona University, Email: hz345@nau.edu;

Meng Wang, Newmark Group, Email: wang070210@gmail.com;

Ning Zhou, Zhejiang Future Technology LLC, zhouning723@gmail.com

*Corresponding Author

Abstract: The volatility and complexity of stock prices in the financial market make precise trend prediction a formidable challenge. Traditional stock prediction approaches often rely solely on either time-domain or frequency-domain information, which limits their ability to fully capture the multi-scale dynamics of stock prices, resulting in suboptimal prediction accuracy. To overcome these limitations, this paper presents an end-to-end stock recommendation algorithm grounded in time-frequency consistency. First, we introduce a time-frequency consistency analysis method that extracts both time-domain and frequency-domain features of stock prices concurrently, offering a more holistic view of trend fluctuations. Next, by applying prompt learning strategies, the model leverages pre-set prompts to identify optimal low-risk buying points within targeted time intervals, enhancing the decision-making process for stock recommendations. Finally, end-to-end model training facilitates seamless integration and automation from data input to stock recommendation output, enabling a fully streamlined prediction workflow. Experimental results indicate that this method surpasses traditional approaches in prediction accuracy and risk control, providing more dependable support for investor decisions.

Keywords: T*ime-Frequency Consistency; Stock Recommendation; Multi-Scale Dynamic Characteristics; Prompt Learning; Risk Control*

1. Introduction

In today's highly volatile and complex financial markets, accurately predicting stock price trends is crucial for investors. Stock price fluctuations are influenced by various factors, including macroeconomic indicators, company performance, market sentiment, and international political situations. Due to the complexity of the interactions among these factors, predicting stock prices has become one of the most challenging tasks in the financial field [1]. Traditional stock prediction methods, such as those based on time series analysis, moving averages, or regression models, provide tools for trend analysis to some extent but are often limited to single-domain time analysis. This one-dimensional approach fails to fully capture the multi-scale dynamic characteristics of stock prices and shows limitations, especially when dealing with nonlinear and complex market signals [2].

With advances in computing power and data processing technologies, machine learning algorithms have gradually become emerging tools for stock prediction. These methods include Support Vector Machines (SVM), Random Forests (RF), and Neural Networks (NN), which can make predictions by learning complex patterns from large amounts of historical data [3]. In particular, deep learning models, such as Long Short-Term Memory (LSTM) networks and Convolutional Neural Networks (CNNs), have shown great potential in handling nonlinear time series prediction tasks due to their strong feature extraction and pattern recognition capabilities [4]. However, most of these methods still rely on single-domain time analysis, making it difficult to comprehensively capture the multi-scale dynamic characteristics of stock prices. Frequency domain analysis methods have emerged as a response, revealing cyclical patterns hidden in price fluctuations by decomposing time series data into frequency components. For example, Fourier Transform can convert time series data into a sum of sine waves of different frequencies, helping to identify market behavior at different time scales [5]. Wavelet Transform further extends the application scope of frequency domain analysis by decomposing signals into sub-waves of different scales, providing joint time and frequency analysis [6]. Although these methods offer new tools for capturing hidden periodicity in stock prices, solely relying on frequency domain analysis still has limitations, especially when dealing with complex and volatile market environments, making it difficult to fully capture dynamic characteristics. Therefore, time-frequency analysis methods have gained popularity in recent years. These methods can analyze signals in both the time and frequency domains simultaneously, providing a more comprehensive tool for complex financial markets.

Despite the potential value of time-frequency analysis methods, their application in stock prediction still faces challenges. Firstly, effectively integrating time-domain and frequency-domain information to comprehensively characterize the dynamic changes in stock prices is a key focus of current research. Secondly, existing prediction algorithms often lack effective risk control in decision optimization. Merely improving prediction accuracy is not enough to cope with market uncertainty. Introducing intelligent decision support mechanisms into the prediction process to help investors make more robust investment decisions in complex and volatile markets is equally important. Additionally, traditional stock prediction processes often involve multiple independent steps (e.g., feature extraction, model training, decision generation). The separation of these steps not only reduces efficiency but also introduces errors at various stages [7]. To address these issues, this paper proposes an end-to-end stock recommendation algorithm based on time-frequency consistency, aiming to improve stock prediction accuracy and practicality by integrating time-domain and frequency-domain information and incorporating prompt learning strategies. Firstly, the theoretical foundation of time-frequency consistency analysis is explored in depth, and specific methods for stock price analysis are developed. Secondly, prompt learning strategies are designed to enable effective risk assessment and decision optimization by utilizing market features to guide the model. Subsequently, an end-to-end stock recommendation model is constructed and trained, integrating time-frequency feature extraction and prompt learning processes, ensuring seamless integration throughout the prediction and decision-making process. Finally, experimental design and empirical analysis are conducted to comprehensively evaluate the performance of the model, verifying its effectiveness and stability in different market environments.

The structure of this paper is arranged as follows: The first part introduces the background, problems, objectives, significance, research methods, and content of the study. The second part reviews existing research in the field of stock prediction, particularly the progress in the application of time-frequency analysis and prompt learning strategies, and clarifies the research direction of this paper. The third part details the proposed end-to-end stock recommendation algorithm based on time-frequency consistency, including time-frequency consistency analysis, the design and application of prompt learning strategies, and the construction and implementation of the end-to-end model. The fourth chapter focuses on experimental design and result analysis, validating the effectiveness of the proposed algorithm through experiments, covering prediction accuracy tests on different datasets, model comparison experiments, and result analysis. The fifth part summarizes the main research findings and theoretical contributions of this paper, discusses the limitations of the study, and proposes future research directions. The main contributions of this paper are as follows:

1. This paper applies the time-frequency consistency analysis method to stock price prediction, successfully integrating time-domain and frequency-domain information to comprehensively capture the multi-scale dynamic characteristics of stock prices. Compared with traditional single-domain time or frequency analysis methods, this algorithm demonstrates higher prediction accuracy and robustness in handling complex market signals.

2. To achieve more robust investment decisions in stock recommendations, this paper designs and applies a prompt learning strategy, guiding the model to identify low-risk buying and selling points through pre-designed market feature prompts. This strategy not only enhances the decision-making ability of the model but also shows significant advantages in risk control.

3. This paper develops an end-to-end stock recommendation model that integrates time-frequency consistency analysis with prompt learning strategies, simplifying multiple independent steps in traditional prediction processes. This model achieves full-process automation from data input to stock recommendation, not only improving prediction efficiency but also enhancing the model's integration and practicality in real-world applications.

2. Related Work

As the global economy continues to develop and financial markets become increasingly complex, accurately predicting stock prices has become a growing challenge. This challenge mainly stems from the non-stationarity, high volatility, frequent fluctuations, and inherent randomness of stock market data. These characteristics often make traditional statistical models and fundamental analysis methods inadequate when dealing with complex time series data. An increasing number of scholars and practitioners are dedicated to developing more precise and efficient predictive models. Traditional stock prediction methods primarily include time series analysis, technical analysis, and fundamental analysis. Building on these methods, researchers have proposed improved time series analysis approaches. For example, Shakir Khan et al. [9] proposed an ARIMA model-based method to accurately predict stock time series. By analyzing five years of historical data for Netflix stock, they compared an automated ARIMA model with a custom ARIMA(p, D, q) model and found that ARIMA(1,1,33) performed best in terms of accuracy, demonstrating the effectiveness of the ARIMA model in stock prediction. Lu Wang et al. [10] proposed a GARCH-MIDAS model that combines asymmetry and extreme volatility effects to model and predict stock price volatility more accurately. Their research indicates that the asymmetric effect has a significantly greater impact on volatility in both the long and short term compared to extreme volatility effects. Through a series of robustness tests, their study also confirmed the model's superior performance in predicting short-term volatility. These methods assume that market prices follow certain historical patterns that can be captured by statistical models. However, as the market environment continues to evolve, especially in the face of nonlinear and highly volatile market conditions, the predictive effectiveness of such methods is often limited.

In recent years, with the rise of deep learning technologies, models like Long Short-Term Memory (LSTM) networks have shown great potential in handling nonlinear time series prediction tasks. However, they mainly rely on time-domain data and struggle to capture more complex market dynamics. Hum Nath Bhandari et al. [11] proposed an LSTM-based method for predicting the next day's closing price of the S&P 500 index. They developed single-layer and multi-layer LSTM models using nine predictive factors, including market data, macroeconomic data, and technical indicators. The study results showed that the single-layer LSTM model outperformed the multi-layer LSTM model in prediction accuracy and fit. Burak Gülmez et al. [12] proposed a deep LSTM network combined with the Artificial Rabbit Optimization (ARO) algorithm (LSTM-ARO) for stock price prediction. They applied this model to Dow Jones Industrial Average (DJIA) stock data and compared it with traditional Artificial Neural Networks (ANN), three other LSTM models, and an LSTM model optimized using Genetic Algorithms (GA). The LSTM-ARO model exhibited higher predictive accuracy across various evaluation metrics. In technical analysis, researchers and investors widely use indicators like Moving Averages (MA) and the Relative Strength Index (RSI). These technical indicators analyze historical price and volume data to predict future market trends. While these methods are intuitive and easy to use, they often overlook fundamental market information, especially in long-term predictions. Additionally, technical analysis methods typically struggle to handle unexpected events or abnormal market fluctuations, limiting their application in complex market environments. Frequency domain analysis methods offer a different perspective from traditional time-domain analysis by transforming time series data into frequency components to reveal cyclical patterns hidden in price fluctuations. Donghwan Song et al. [14] proposed a Padding-Fourier Transform Denoising (P-FTD) method to improve the prediction accuracy of financial time series data. This method addresses the issue of data divergence at both ends when restoring the original time series by eliminating noise waveforms in the frequency domain. Applying the denoised data to several time series-based deep learning models demonstrated that deep learning models combined with P-FTD technology outperformed basic models in prediction performance and effectively mitigated time lag issues. Satya Verma et al. [15] proposed a feature engineering method based on Discrete Wavelet Transform (DWT) and Chicken Swarm Optimization (CSO) (DWT-CSO) for stock market prediction. Their model decomposed data using DWT and used CSO to select the optimal feature subset to address data noise and the problem of too many features. However, relying solely on frequency domain analysis has limitations, particularly in integrating time-domain and frequency-domain information to fully capture the dynamic characteristics of the market.

Time-frequency analysis methods, such as Short-Time Fourier Transform (STFT) and Wavelet Packet Decomposition (WPD), attempt to analyze signals in both the time and frequency domains simultaneously, providing more comprehensive market information. Yaqing Luo et al. [16] proposed a wavelet neural network model combined with time-frequency analysis, using Gaussian wavelets as the activation function and refining stock price data through wavelet decomposition to enhance the model's sensitivity to data. This model significantly reduced the mean squared error in London stock market data. These methods can better capture market dynamics by jointly analyzing time and frequency components. However, in complex and volatile market environments, the key challenge in current research lies in how to make models more accurately identify favorable investment opportunities and potential risks. Prompt learning strategies, an emerging machine learning optimization method, have also begun to show potential in the financial field. Defu Cao et al. [17] proposed a new framework called TEMPO, based on generative pre-trained Transformers for time series prediction. This framework leverages two key inductive biases in time series tasks: the complex interaction decomposition of trend, seasonality, and residual components, and promotes the adaptation of different types of time series distributions by designing prompts. Tian Guo et al. [18] proposed a method for predicting stock returns based on large language models (LLMs) combined with financial news streams. By fine-tuning LLMs, they integrated text representation with the prediction module and compared the impact of differences between encoder-only and decoder-only LLMs on prediction performance. Their aggregated representation improved the performance of long and long-short portfolios in stock return prediction. Prompt learning optimizes the decision-making process by guiding the model to learn specific market features through pre-designed prompts. In the field of financial prediction, end-to-end models integrate feature extraction, model training, and decision generation into a unified framework, avoiding errors that may be introduced by multiple independent steps in traditional methods. Existing stock prediction methods still have many shortcomings in capturing market multi-scale features, achieving effective risk control, and decision support. Building on this foundation, this paper proposes an innovative end-to-end stock recommendation algorithm based on time-frequency consistency, aiming to enhance stock prediction accuracy and practicality and provide more reliable decision support for investors.

3. Method

Figure 1 illustrates the overall architecture of the proposed stock recommendation algorithm, which is trained in an end-to-end manner to ensure time-frequency consistency. Initially, the input stock data is processed through time and frequency encoders to extract temporal and frequency features, respectively. These features are then patched and combined with prompt information before being embedded into the model. The model further processes and integrates these features, and finally, a fully connected neural network is used to make predictions, outputting stock recommendations. The entire process leverages time-frequency consistency analysis to comprehensively capture the dynamic characteristics of stock prices, achieving a complete end-to-end workflow from data input to stock recommendation.

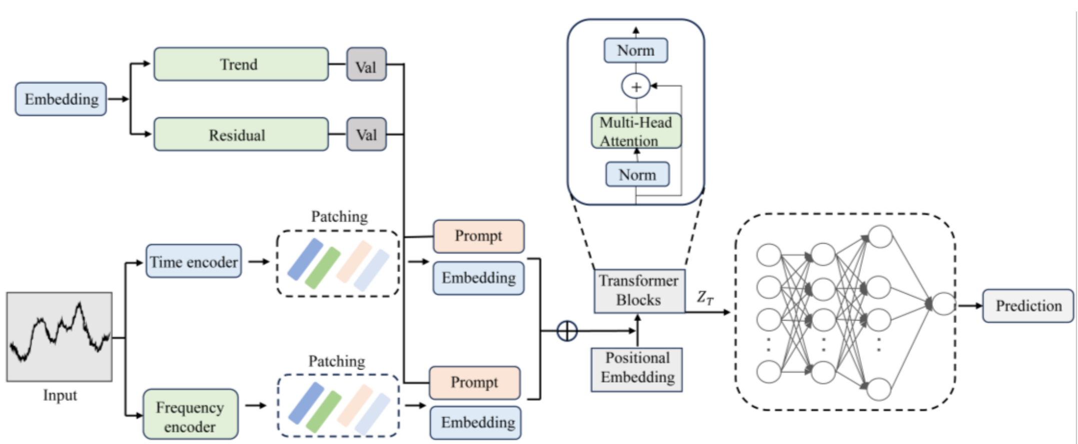

Figure 1. Overall algorithm architecture.

3.1 Time-Frequency Consistency Model

The time-frequency consistency model is an innovative method proposed in recent years for time series analysis. It aims to represent time series data in both the time domain and frequency domain simultaneously, ensuring consistency in a unified time-frequency space. This approach is particularly suitable for handling time series data with complex dynamic characteristics. Time series data are prevalent across various fields, such as financial markets, medical diagnostics, and traffic analysis. In financial markets, stock price fluctuations not only reflect trends over time (time domain features) but also contain various periodic and non-periodic components (frequency domain features). Traditional time series analysis methods typically focus on either time domain or frequency domain analysis, making it challenging to comprehensively capture these complex features. The time-frequency consistency model was proposed to overcome this limitation. The core idea is to simultaneously learn feature representations in both the time and frequency domains and enforce consistency between these representations in a latent time-frequency space. This enables the model to better understand and predict the dynamic changes in time series data. The algorithm architecture diagram is shown in Figure 2.

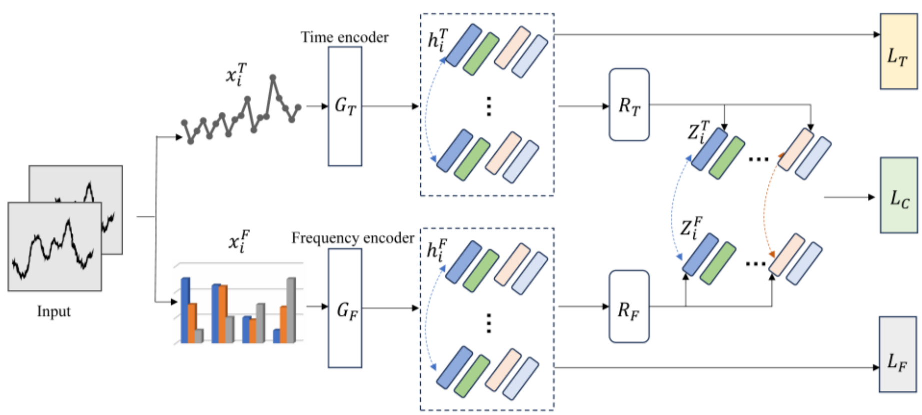

Figure 2. Time-frequency consistency model architecture diagram.

The model comprises a time encoder G_T and a frequency encoder G_F. The time encoder receives the time series input $x(t)$ and maps it to a latent representation in the time domain z_T:

$$z_T = G_T(x(t)) \tag{1}$$

The frequency encoder G_F receives the frequency representation of the time series $X(f)$ and maps it to a latent representation in the frequency domain z_F:

$$z_F = G_F(X(f)) \tag{2}$$

Here, $X(f)$ is the frequency domain representation of the signal obtained through Fourier transform or other spectral analysis methods. To compare the representations in the time and frequency domains within the same space, the model introduces two projectors: the time domain projector R_T and the frequency domain projector R_F. These projectors map the time and frequency representations into a unified time-frequency consistency space:

$$z_T^{(p)} = R_T(z_T), \quad z_F^{(p)} = R_F(z_F) \tag{3}$$

where $z_T^{(p)}$ and $z_F^{(p)}$ represent the projected time domain and frequency domain representations, respectively. The model is designed with a loss function that ensures the representations of the same time series in the time and frequency domains are as close as possible in the projected time-frequency space. To achieve this, a time-frequency consistency loss function L_C is introduced:

$$L_C = \sum_{\text{Spair}} \left(d(z_T^{(p)}, z_F^{(p)}) - d(z_T^{(p)}, \tilde{z}_F^{(p)}) + \delta \right) \tag{4}$$

where $d(\cdot,\cdot)$ denotes a distance measure in the projection space, $\tilde{z}_F^{(p)}$ is the representation after frequency domain perturbation, and δ is a constant used to maintain negative sample separation.

This loss function encourages the model to pull the time domain and frequency domain representations of the same time series closer together in the time-frequency space while pushing apart the representations of different time series or perturbed representations. This maintains consistency in the latent space. The model is trained using a contrastive learning framework by constructing positive and negative sample pairs. Positive sample pairs consist of time and frequency domain representations of the same time series, while negative sample pairs are composed of representations from different time series or the original and perturbed representations. The total loss function of the model consists of three parts. The time domain contrastive loss L_T is used to optimize the time encoder G_T to generate representations invariant to time perturbations:

$$L_T = \sum_i d(z_T, \tilde{z}_T) \tag{5}$$

The frequency domain contrastive loss L_F is used to optimize the frequency encoder G_F to generate representations invariant to spectral perturbations:

$$L_F = \sum_i d(z_F, \tilde{z}_F) \tag{6}$$

The time-frequency consistency loss L_C ensures consistency between the time and frequency representations in the time-frequency space:

$$L_C = \sum_{\text{Spair}} \left(d(z_T^{(p)}, z_F^{(p)}) - d(z_T^{(p)}, \tilde{z}_F^{(p)}) + \delta \right) \tag{7}$$

The total loss function is:

$$L = \lambda(L_T + L_F) + (1-\lambda)L_C \tag{8}$$

where λ is a hyperparameter that balances the contrastive loss and the consistency loss. By minimizing this total loss function, the model can learn both time domain and frequency domain feature representations while maintaining consistency between them in the time-frequency space, thereby enhancing the model's ability to capture the complex dynamic characteristics of time series data.

3.2 Prompt Learning Model

Prompt learning aims to guide the model in identifying the lowest-risk entry points within a specific time frame by using pre-designed prompts. The design of these prompts relies on the analysis of historical stock price data, combined with prior market knowledge and specific investment strategies. These prompts can be specific patterns in the time series, threshold values of indicators, or time windows for certain key events. Subsequently, based on these time-frequency consistency features, prompt learning is used to optimize investment decisions. The architecture diagram is shown in Figure 3.

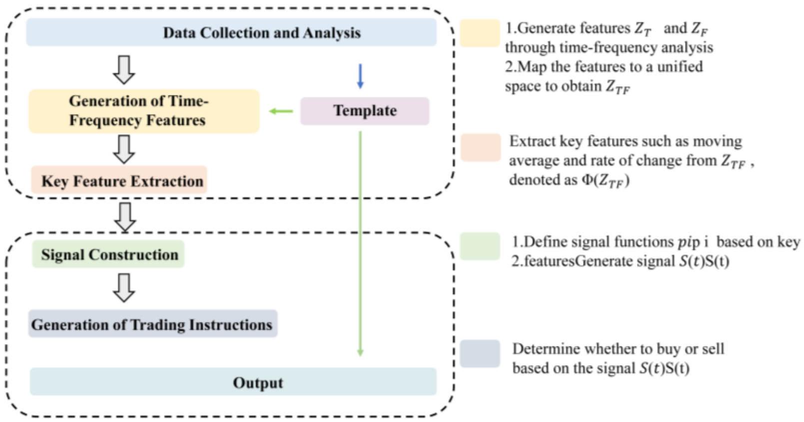

Figure 3. Prompt learning model architecture diagram.

Assume that the time series data $X = \{x_1, x_2, \ldots, x_T\}$ and the time-frequency consistency features generated through time-frequency analysis are represented as follows:

$$\begin{aligned} Z_T &= \{z_T(t_1), z_T(t_2), \ldots, z_T(t_T)\} \\ Z_F &= \{z_F(f_1), z_F(f_2), \ldots, z_F(f_F)\} \end{aligned} \tag{9}$$

where $z_T(t_i)$ represents the time-domain feature, and $z_F(f_j)$ represents the frequency-domain feature. Through the time-frequency consistency model, these features are mapped into a unified latent space to obtain the time-frequency consistency feature Z_{TF}:

$$Z_{TF} = \{z_{TF}(t_1), z_{TF}(t_2), \ldots, z_{TF}(t_T)\} \tag{10}$$

These features represent the consistency information of the time series data in both time and frequency dimensions, capturing the multi-scale dynamic characteristics of the data. After obtaining the time-frequency consistency feature Z_{TF}, a set of prompt signals is constructed using these time-frequency features to guide the model in making buy or sell decisions. To more effectively utilize these features, further processing and feature extraction are performed. By calculating the moving averages, rate of change, frequency domain energy, etc., of these features, different market signals are captured:

$$\Phi(Z_{TF}) = \{\phi_1(Z_{TF}), \phi_2(Z_{TF}), \ldots, \phi_k(Z_{TF})\} \tag{11}$$

where $\phi_i(Z_{TF})$ is the i-th feature extracted from the time-frequency consistency features, and k is the number of features.

Based on the extracted features $\Phi(Z_{TF})$, prompt functions p_i are defined, which are used to generate buy or sell prompt signals. A prompt function based on time-frequency energy aggregation can be defined as follows:

$$p_i(Z_{TF}, t) = \sum_{f \in F_{\text{selected}}} z_{TF}(t, f) \tag{12}$$

where F_{selected} represents the selected frequency range, and $z_{TF}(t, f)$ represents the time-frequency feature value at time t and frequency f. This prompt function indicates the degree of energy aggregation within a specific frequency range, corresponding to a certain market signal such as a trend reversal or price breakout. To provide more comprehensive decision-making guidance, the outputs of multiple prompt functions are integrated to generate a final prompt signal $S(t)$:

$$S(t) = \sum_{i=1}^{N} w_i \cdot p_i(Z_{TF}, t) \tag{13}$$

where w_i is the weight of the prompt function p_i, indicating the importance of each prompt in the combined signal. These weights are automatically adjusted through the model training process to maximize prediction performance. After obtaining the combined prompt signal $S(t)$, decision rules can be generated based on this signal value to issue buy or sell instructions. When $S(t)$ exceeds a certain threshold θ, it indicates that the current market state is suitable for buying: if $S(t) > \theta$, then buy at time t, similarly, if $S(t)$ falls below another threshold θ', it may suggest selling.

3.3 End-to-End Learning

In the stock recommendation algorithm proposed in this paper, the end-to-end learning method is the core of the entire model, enabling the full automation of the process from raw data input to final decision output. Within the end-to-end learning framework, all steps, from time-frequency consistency feature extraction and prompt signal generation to decision optimization, are integrated into a unified model, where joint training directly optimizes the final investment decisions.

The end-to-end learning model consists of several sub-modules that work collaboratively to achieve the overall goal of stock recommendation. The input representation module receives the raw time-series data $X = \{x_1, x_2, \ldots, x_T\}$ and performs initial feature extraction. The time-frequency consistency module extracts time-frequency features Z_{TF} from the input representation, capturing the dynamic changes in stock prices across both the time and frequency domains. The prompt learning module constructs prompt signals $S(t)$ based on the time-frequency features, which guide the buy or sell decisions. The decision module generates the final investment decisions based on the prompt signals. The architecture of the entire model can be expressed as the following function composition:

$$D(X) = f_{\text{decision}}\left(f_{\text{prompt}}\left(f_{\text{TF}}(X)\right)\right) \tag{14}$$

where f_{TF} represents the time-frequency consistency module, f_{prompt} represents the prompt learning module, and $f_{decision}$ represents the final decision module. The output $D(X)$ is the final decision result. The joint loss function in end-to-end learning simultaneously optimizes the parameters of all sub-modules, thereby directly enhancing the quality of the final decisions. The total loss function L_{total} is composed of the following three parts:

Time-frequency consistency loss L_{TF} is used to optimize the representation of time-frequency features, ensuring consistency across both time and frequency domains.

$$L_{\text{TF}} = \sum_{t=1}^{T} \left\| f_{\text{time}}(X_t) - f_{\text{freq}}(X_t) \right\|^2 \tag{15}$$

where $f_{time}(X_t)$ and $f_{freq}(X_t)$ represent the feature representations in the time and frequency domains, respectively. Prompt learning loss L_{prompt} optimizes the generation of prompt signals to accurately reflect potential market trends and risks.

$$L_{\text{prompt}} = \sum_{t=1}^{T} \left\| S(t) - \hat{S}(t) \right\|^2 \tag{16}$$

where $S(t)$ is the prompt signal generated by the model, and $\hat{S}(t)$ is the target signal based on historical data. Decision loss $L_{decision}$ directly optimizes the final investment decisions, minimizing risk while maximizing returns.

$$L_{\text{decision}} = -\sum_{t \in B} r(t) + \lambda \sum_{t \in B} \sigma(t) \tag{17}$$

where $r(t)$ represents the expected return at time t, $\sigma(t)$ represents the corresponding risk measure, and λ is a balancing coefficient. The joint loss function is defined as:

$$L_{\text{total}} = \alpha L_{\text{TF}} + \beta L_{\text{prompt}} + \gamma L_{\text{decision}} \tag{18}$$

where α, β, and γ are hyperparameters that adjust the weights of each loss component and are automatically tuned during model training. By minimizing the joint loss function L_{total}, the parameters of all modules are optimized simultaneously. The model is trained using gradient descent, gradually updating the parameters to reduce the overall loss:

$$\theta^* = \arg\min_{\theta} L_{\text{total}}(\theta) \tag{19}$$

where θ represents all the model parameters, including those of the time-frequency consistency module, prompt learning module, and decision module. During training, the model continuously learns from historical data, gradually improving its ability to predict and make decisions for future markets. Through end-to-end joint training, the model can automatically capture complex patterns in the input data and directly use these patterns to guide decision-making.

4. **Experiment**

4.1 Experimental Environment

The experimental environment in this study includes both hardware and software configurations, as well as data sources. On the hardware side, the experiments were conducted on a computer equipped with an Intel Core i7 processor, 64GB of RAM, and an NVIDIA GTX 3090 graphics card. On the software side, Python was primarily used as the programming language, along with the TensorFlow deep learning framework for model training and inference. Additionally, data processing libraries such as Pandas and NumPy were utilized to efficiently manage and process time-series data.

4.2 Experimental Data

- S&P 500 Index Constituents Dataset

The S&P 500 Index Constituents dataset [19] contains the historical trading data of the 500 most representative companies in the U.S. stock market. This dataset includes key information for each constituent stock, such as daily opening price, closing price, highest price, lowest price, and trading volume. Since the S&P 500 index covers leading companies across multiple industries, this dataset reflects the overall performance of the U.S. market. By using this dataset, one can test stock recommendation models under broad market conditions, making it particularly suitable for large-cap market analysis, cross-sector comparisons, and diversified investment strategy experiments.

- Shanghai A-Share Dataset

The Shanghai Stock Exchange A-Share dataset [20] contains the historical trading data of companies listed in mainland China, including key metrics such as opening price, closing price, highest price, lowest price, and trading volume. As a representative of emerging markets, China's A-share market is characterized by significant volatility and unique market mechanisms. This makes the dataset highly suitable for evaluating stock recommendation models in the context of emerging markets. Conducting experiments on this dataset allows for an in-depth study of the generalization ability and stability of time-frequency consistency models under different market conditions.

- NASDAQ 100 Index Constituents Dataset

The NASDAQ 100 Index Constituents dataset [21] gathers the daily trading data of the top 100 non-financial companies in the U.S. NASDAQ market. These companies primarily operate in sectors such as technology, communications, and biotechnology, and their stock prices tend to exhibit high volatility. This dataset provides detailed information, including opening price, closing price, highest price, lowest price, and trading volume, making it suitable for testing stock recommendation models in highly volatile markets. Using this dataset, one can assess the model's prediction accuracy and risk control capabilities in high-risk stock environments.

- FTSE 100 Index Constituents Dataset

The FTSE 100 Index Constituents dataset [22] contains the historical trading data of the 100 largest companies by market capitalization on the London Stock Exchange. This index represents the overall performance of the UK market, and the dataset covers key information such as daily opening price, closing price, highest price, lowest price, and trading volume. FTSE 100 companies span multiple industries, and the market is relatively mature and stable. By conducting experiments on this dataset, one can test stock recommendation models in mature markets, making it especially suitable for studying the model's adaptability to different economic environments and the effectiveness of long-term investment strategies.

4.3 Evaluation Metrics

- Mean Absolute Error (MAE)

MAE is used to measure the average absolute error between the model's predicted results and the actual stock prices. The lower the MAE value, the more accurate the model's prediction of stock prices. Since MAE only considers the absolute value of the errors, it avoids the issue of positive and negative errors canceling each other out. Therefore, MAE directly reflects the magnitude of the model's prediction bias. The formula for MAE is as follows:

$$\mathrm{MAE} = \frac{1}{n}\sum_{i=1}^{n}\left|y_i - \hat{y}_i\right| \tag{20}$$

where y_i is the actual stock price, $\hat{y}_1$ is the model's predicted stock price, and n is the number of samples.

- Mean Squared Error (MSE)

MSE is used to measure the overall error of the model by taking the average of the squared prediction errors. It not only measures the size of the prediction error but also amplifies the effect of large errors due to the squaring operation, which makes it particularly sensitive to extreme market fluctuations. The lower the MSE value, the better the overall performance of the model's predictions. The formula for MSE is as follows:

$$\mathrm{MSE} = \frac{1}{n}\sum_{i=1}^{n}\left(y_i - \hat{y}_i\right)^2 \tag{21}$$

- Coefficient of Determination (R^2)

R^2 measures the model's ability to explain the variability in stock prices, with a value between 0 and 1. The closer it is to 1, the better the model's fit. In stock recommendation algorithms, R^2 can help us understand the correlation between the predicted stock prices and the actual prices. A high R^2 value indicates that the model captures stock price trends well, whereas a low R^2 value may suggest that the model has not fully utilized the data to make accurate predictions. The formula is as follows:

$$R^2 = 1 - \frac{\sum_{i=1}^{n}\left(y_i - \hat{y}_i\right)^2}{\sum_{i=1}^{n}\left(y_i - \bar{y}\right)^2} \tag{22}$$

where $\bar{y}$ is the mean of all actual stock prices.

- Normalized Discounted Cumulative Gain (NDCG)

NDCG is commonly used as an evaluation metric in recommendation systems, suitable for assessing the performance of ranking tasks. NDCG effectively measures whether the positions of highly relevant items in the recommendation list are reasonable and whether the model can correctly identify and prioritize important stocks or other financial products. A high NDCG value indicates that the model can accurately rank the most relevant items at the top, thus improving user satisfaction and the success rate of investment decisions.

$$\mathrm{NDCG} = \frac{\mathrm{DCG}}{\mathrm{IDCG}} \tag{23}$$

where DCG is the Discounted Cumulative Gain, which considers the importance of the position by calculating the cumulative gain with a discount factor for the ranking position.

$$\mathrm{DCG} = \sum_{i=1}^{n} \frac{rel_i}{\log_2(i+1)} \tag{24}$$

where i is the position of the recommended item in the list, and rel_i is the relevance score of the i-th recommended item. Generally, the top few items in the list are more important, and by discounting for position, the quality of the recommendation system can be more accurately reflected. IDCG represents the Ideal Discounted Cumulative Gain, which refers to the maximum DCG value that can be achieved in the ideal case (where the most relevant items are ranked at the top).

$$\mathrm{IDCG} = \sum_{i=1}^{|REL|} \frac{rel_i}{\log_2(i+1)} \tag{25}$$

where $|REL|$ represents the total number of recommended items in the ideal ranking.

4.4 Experimental Comparison and Analysis

Table1. Comparison of relevant indicators of this method with other methods on S&P 500 Index Constituents Dataset and Shanghai A-Share Dataset.

Model	S&P 500 Index Constituents Dataset				Shanghai A-Share Dataset			
	MAE	MSE	R^2	NDCG	MAE	MSE	R^2	NDCG
Yang et al. [23]	0.264	0.136	0.862	0.374	0.253	0.124	0.883	0.382
Liu et al. [24]	0.194	0.112	0.894	0.384	0.186	0.101	0.912	0.391
Lu et al. [25]	0.284	0.153	0.849	0.362	0.261	0.135	0.857	0.376
Chaudhari et al. [26]	0.337	0.197	0.837	0.354	0.303	0.142	0.869	0.379
Wijerathne et al. [27]	0.163	0.073	0.912	0.391	0.158	0.064	0.924	0.412
Mahmoodi et al. [28]	0.192	0.092	0.896	0.388	0.176	0.087	0.922	0.396
Ours	0.078	0.036	0.944	0.436	0.064	0.027	0.952	0.447

From the data in Table 1, it is evident that our proposed algorithm outperforms other methods on the S&P 500 Index Constituents Dataset and the Shanghai A-Share Dataset. Specifically, in terms of MAE and MSE, our method achieved the lowest values of 0.078 and 0.036 on the S&P 500 dataset, and 0.064 and 0.027 on the Shanghai A-Share dataset, indicating that our method significantly outperforms others in error control. Additionally, for the R^2 and NDCG metrics, our method also stands out, reaching 0.944 and 0.436 on the S&P 500 dataset, and 0.952 and 0.447 on the Shanghai A-Share dataset, far exceeding other comparative methods. In contrast, while the method by Wijerathne et al. also has relatively high R^2 and NDCG values, it still falls short of our method in terms of error metrics. Figure 4 provides a visual comparison of these results.

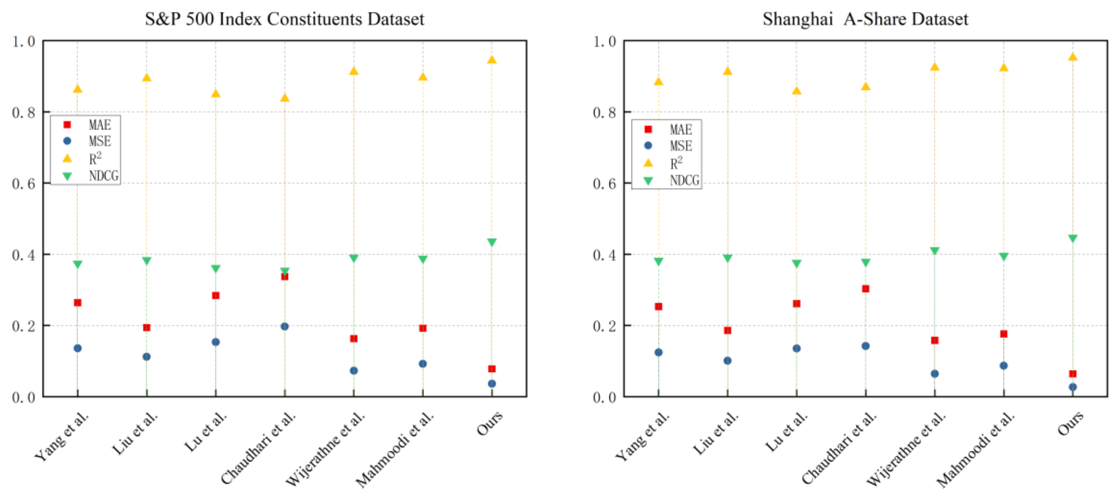

Figure 4. Visual comparison of relevant indicators on S&P 500 Index Constituents Dataset and Shanghai A-Share Dataset.

Table2. Comparison of relevant indicators of this method with other methods on NASDAQ 100 Index Constituents Dataset and FTSE 100 Index Constituents Dataset.

Model	NASDAQ 100 Index Constituents Dataset				FTSE 100 Index Constituents Dataset			
	MAE	MSE	R^2	NDCG	MAE	MSE	R^2	NDCG
Yang et al.	0.324	0.195	0.842	0.362	0.335	0.194	0.842	0.358
Liu et al.	0.242	0.117	0.882	0.372	0.257	0.129	0.871	0.374
Lu et al.	0.276	0.142	0.877	0.368	0.286	0.162	0.841	0.359
Chaudhari et al.	0.312	0.163	0.853	0.351	0.322	0.185	0.841	0.339
Wijerathne et al.	0.154	0.071	0.905	0.396	0.169	0.079	0.907	0.388
Mahmoodi et al.	0.176	0.086	0.893	0.384	0.172	0.081	0.899	0.386
Ours	0.094	0.063	0.924	0.415	0.069	0.035	0.942	0.436

From the data in Table 2, our method significantly outperforms other comparative methods on the NASDAQ 100 and FTSE 100 Index Constituents Datasets. Specifically, in terms of MAE and MSE, our model achieved the lowest error values on both datasets, demonstrating its significant advantage in prediction accuracy. Additionally, the R^2 metric shows that our method reached 0.924 and 0.942 on these datasets, reflecting its strong ability to fit the real data. For the NDCG metric, our method achieved 0.415 and 0.436 on the NASDAQ 100 and FTSE 100 datasets, surpassing all other comparative methods. Overall, our algorithm outperforms existing methods across multiple metrics, showcasing superior overall performance. Figure 5 provides a visual comparison of these trends.

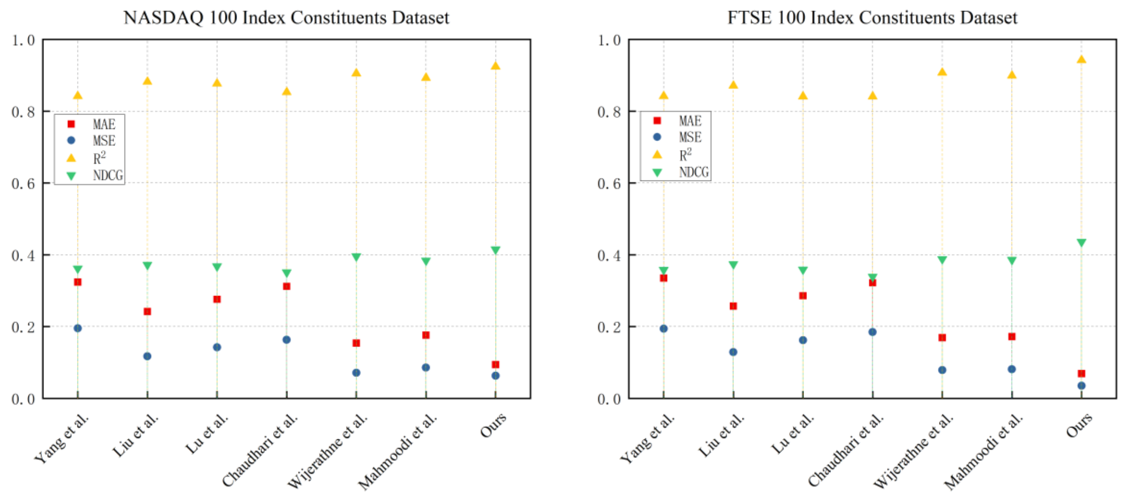

Figure 5. Visual comparison of relevant indicators on NASDAQ 100 Index Constituents Dataset and FTSE 100 Index Constituents Dataset.

Table3. Comparison of training indicators on four datasets.

Model	S&P 500 Index Constituents Dataset					
	Parameters(M)	Inference Time(ms)	Trainning Time(s)	Parameters(M)	Inference Time(ms)	Trainning Time(s)
Yang et al.	391.04	396.70	184.80	351.70	356.89	285.14
Liu et al.	377.57	363.19	183.82	374.00	365.12	265.35
Lu et al.	358.40	392.19	202.62	372.88	331.20	242.99
Chaudhari et al.	384.77	383.27	207.68	368.49	355.77	263.88
Wijerathne et al.	390.79	373.21	258.84	359.01	367.47	267.78
Mahmoodi et al.	366.53	342.79	214.83	353.25	389.96	208.06
Ours	336.54	306.46	171.24	338.69	316.73	176.52
Model	NASDAQ 100 Index Constituents Dataset			FTSE 100 Index Constituents Dataset		
	Parameters(M)	Inference Time(ms)	Trainning Time(s)	Parameters(M)	Inference Time(ms)	Trainning Time(s)
Yang et al.	373.22	393.86	253.88	381.06	314.66	235.83
Liu et al.	382.11	372.26	239.43	372.00	301.52	287.33
Lu et al.	355.96	382.87	235.64	359.78	295.74	235.82
Chaudhari et al.	374.27	379.35	284.10	368.58	376.55	291.35
Wijerathne et al.	361.01	386.80	247.54	357.61	340.72	257.43
Mahmoodi et al.	356.60	359.99	284.54	377.29	325.23	229.29
Ours	342.64	322.59	177.84	342.57	275.74	220.34

From the data in Table 3, our method excels in terms of parameter size, inference time, and training time. Firstly, for the number of parameters, our method has the smallest parameter scale across all four datasets. For example, on the S&P 500 and NASDAQ 100 datasets, our model parameters are 336.54M and 342.64M, significantly reduced compared to other models, indicating that our model is more lightweight and efficient. Secondly, for inference time, our model showed the fastest inference speed across all datasets. On the S&P 500 and FTSE 100 datasets, the inference times are 306.46ms and 275.74ms, lower than other methods, indicating that our model responds faster in real-time inference. Finally, for training time, our method also demonstrated the shortest training time across all datasets. On the NASDAQ 100 dataset, our training time was 177.84 seconds, and on the FTSE 100 dataset, it was 220.34 seconds, more efficient than other models. This proves that while maintaining prediction accuracy, our model can significantly reduce computation costs and training time, improving overall efficiency. Figure 6 provides a visual comparison of these results.

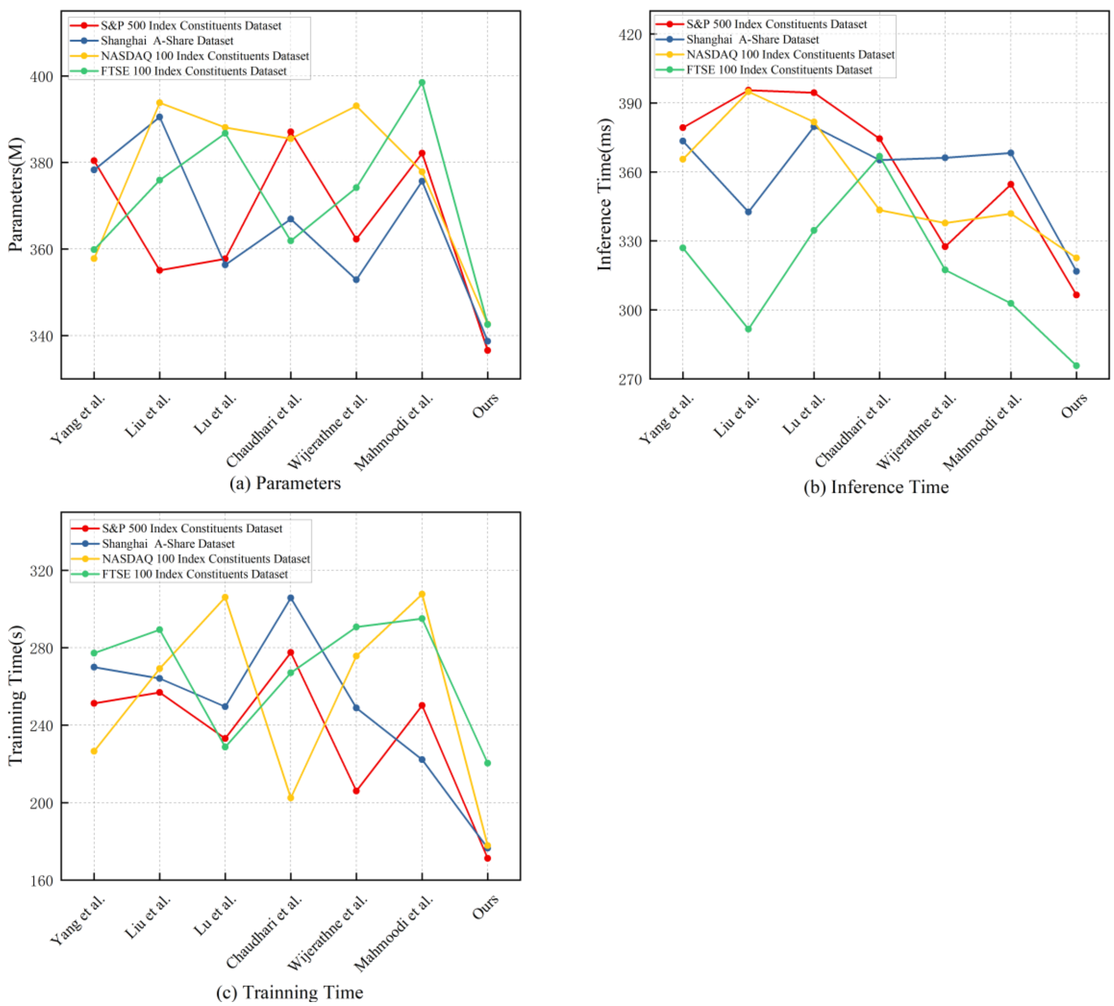

Figure 6. Visual comparison of training indicators.

Table4. Ablation experiments on S&P 500 Index Constituents Dataset and Shanghai A-Share Dataset.

Model	S&P 500 Index Constituents Dataset				Shanghai A-Share Dataset			
	MAE	MSE	R^2	NDCG	MAE	MSE	R^2	NDCG
baseline	0.286	0.163	0.784	0.364	0.273	0.151	0.789	0.381
+TFC	0.185	0.069	0.864	0.396	0.177	0.052	0.873	0.411
+Prompt	0.124	0.043	0.916	0.419	0.112	0.035	0.927	0.426
+TFC-Prompt	0.078	0.036	0.944	0.436	0.064	0.027	0.952	0.447

Table5. Ablation experiments on NASDAQ 100 Index Constituents Dataset and FTSE 100 Index Constituents Dataset.

Model	NASDAQ 100 Index Constituents Dataset				FTSE 100 Index Constituents Dataset			
	MAE	MSE	R^2	NDCG	MAE	MSE	R^2	NDCG
baseline	0.316	0.196	0.754	0.352	0.279	0.159	0.776	0.373
+TFC	0.224	0.069	0.826	0.381	0.182	0.062	0.869	0.399
+Prompt	0.151	0.064	0.896	0.401	0.121	0.041	0.921	0.420
+TFC-Prompt	0.094	0.063	0.924	0.415	0.069	0.035	0.942	0.436

The data in Tables 4 and 5 shows that the stock recommendation algorithm, which combines Time-Frequency Consistency (TFC) and Prompt learning, significantly outperforms the baseline model across multiple datasets. On the S&P 500 Index Constituents Dataset, the baseline model's MAE is 0.286, MSE is 0.163, R^2 is 0.784, and NDCG is 0.364. On the Shanghai A-Share Dataset, the baseline model's MAE is 0.273, MSE is 0.151, R^2 is 0.789, and NDCG is 0.381. After introducing Time-Frequency Consistency (TFC), the MAE on the S&P 500 dataset dropped to 0.185, MSE dropped to 0.069, R^2 increased to 0.864, and NDCG rose to 0.396. Similarly, on the Shanghai A-Share dataset, the MAE and MSE dropped to 0.177 and 0.052, R^2 increased to 0.873, and NDCG increased to 0.411. Further introducing Prompt learning improved performance even more, with the S&P 500 dataset's MAE and MSE reaching 0.124 and 0.043, R^2 rising to 0.916, and NDCG to 0.419; and on the Shanghai A-Share dataset, the MAE dropped to 0.112, MSE to 0.035, and R^2 and NDCG increased to 0.927 and 0.426, respectively. Ultimately, when using the TFC-Prompt strategy, the S&P 500 dataset achieved the lowest MAE of 0.078, MSE of 0.036, R^2

of 0.944, and NDCG of 0.436. On the Shanghai A-Share dataset, the MAE decreased to 0.064, MSE to 0.027, R^2 reached 0.952, and NDCG increased to 0.447. A similar trend can be seen in the NASDAQ 100 and FTSE 100 Index Constituents Datasets in Table 5. When the TFC-Prompt strategy was introduced, the MAE on the NASDAQ 100 dataset dropped to 0.094, MSE to 0.063, R^2 reached 0.924, and NDCG rose to 0.415. On the FTSE 100 dataset, the MAE decreased to 0.069, MSE to 0.035, R^2 reached 0.942, and NDCG increased to 0.436. This indicates that the synergy between Time-Frequency Consistency and Prompt learning can provide more accurate and reliable investment strategies for stock recommendations. Figures 7 and 8 visually depict these trends.

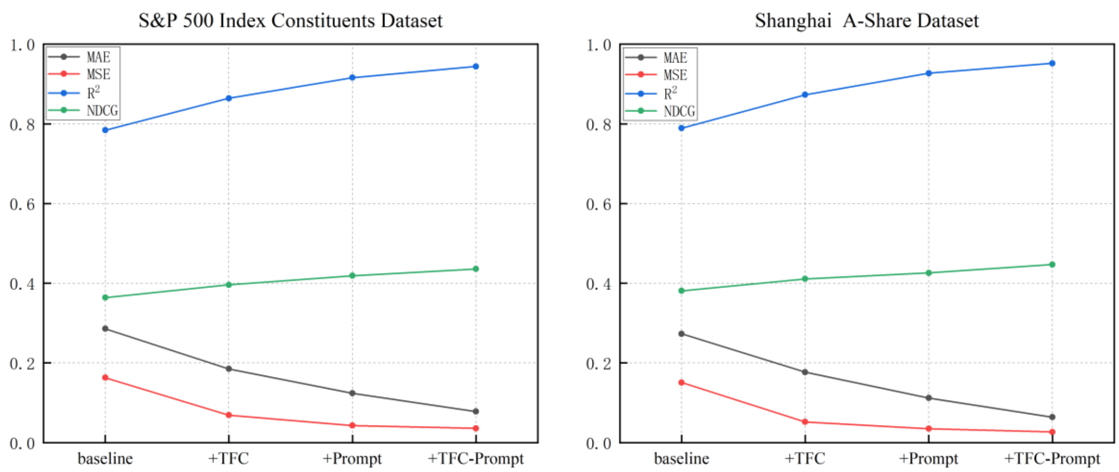

Figure 7. Visual comparison of ablation experiments on S&P 500 Index Constituents Dataset and Shanghai A-Share Dataset.

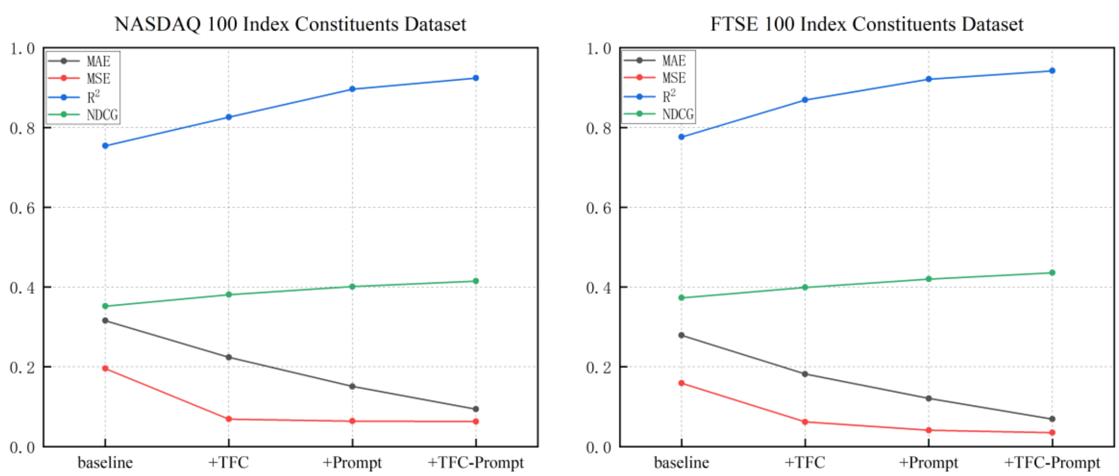

Figure 8. Visual comparison of ablation experiments on NASDAQ 100 Index Constituents Dataset and FTSE 100 Index Constituents Dataset.

5. Conclusion

This paper proposes an end-to-end stock recommendation algorithm based on time-frequency consistency analysis and prompt learning strategies. It addresses the limitations of traditional stock prediction methods that focus only on time-domain or frequency-domain information, enabling the model to fully capture the multi-scale dynamic characteristics of stock prices. Experimental results show that the algorithm significantly outperforms existing models in terms of prediction accuracy, risk control, and computational efficiency across multiple datasets. On the S&P 500 Index Constituents Dataset, the proposed algorithm achieved the lowest MAE of 0.078 and MSE of 0.036, with an R^2 value of 0.944 and an NDCG score of 0.436. Additionally, the model continued to perform well across multiple datasets, demonstrating its advantage in prediction accuracy. By introducing time-frequency consistency analysis, the model simultaneously considers both time-domain and frequency-domain features, resulting in more accurate stock price predictions. The prompt learning strategy further optimizes the decision-making process by identifying low-risk entry points, enhancing risk control in stock recommendations. Furthermore, the end-to-end model simplifies the entire prediction process, from data input to final recommendation, significantly improving prediction efficiency and reducing model complexity. However, this study still has some limitations. The model's performance is partially dependent on the quality of input data and the design of prompts, which may require further optimization for different market conditions and financial instruments. Moreover, the application of the model in real-time trading needs further exploration to ensure it can adapt to rapidly changing market environments. Future research will focus on improving the model's adaptability by introducing advanced machine learning techniques such as reinforcement learning and adversarial training. Expanding the application of the model to other financial markets and instruments will also help to further understand its robustness and generalizability.

Funding
Not applicable

Author Contributions

Conceptualization, S. X. and H. Z.; writing—original draft preparation, M. W. and N. Z.; writing—review and editing, S.X. and M.W.; All of the authors read and agreed to the published the final manuscript.

Institutional Reviewer Board Statement
Not applicable

Informed Consent Statement
Not applicable

Data Availability Statement
Not applicable

Conflict of Interest

The authors declare no conflict of interest.

Reference

[1] C. Zhao, M. Wu, J. Liu, Z. Duan, L. Shen, X. Shangguan, D. Liu, Y. Wang et al., “Progress and prospects of data-driven stock price forecasting research,” International Journal of Cognitive Computing in Engineering, vol. 4, pp. 100–108, 2023.

[2] D. Sheth and M. Shah, “Predicting stock market using machine learning: best and accurate way to know future stock prices,” International Journal of System Assurance Engineering and Management, vol. 14, no. 1, pp. 1–18, 2023.

[3] S. S. Pashankar, J. D. Shendage, and J. Pawar, “Machine learning techniques for stock price prediction-a comparative analysis of linear regression, random forest, and support vector regression.” Journal of Advanced Zoology, vol. 45, 2024.

[4] Z. Nourbakhsh and N. Habibi, “Combining lstm and cnn methods and fundamental analysis for stock price trend prediction,” Multimedia Tools and Applications, vol. 82, no. 12, pp. 17 769–17 799, 2023.

[5] J. Olbrys and M. Mursztyn, “Measuring stock market resiliency with discrete fourier transform for high frequency data,” Physica A: Statistical Mechanics and Its Applications, vol. 513, pp. 248–256, 2019.

[6] D. K. Sharma, H. Hota, and A. R. Rababaah, “Forecasting us stock price using hybrid of wavelet transforms and adaptive neuro fuzzy inference system,” International Journal of System Assurance Engineering and Management, vol. 15, no. 2, pp. 591–608, 2024.

[7] C. Ya˜nez, W. Kristjanpoller, and M. C. Minutolo, “Stock market index prediction using transformer neural network models and frequency decomposition,” Neural Computing and Applications, pp. 1–21, 2024.

[8] Q. Zhang, Y. Zhang, F. Bao, Y. Liu, C. Zhang, and P. Liu, “Incorporating stock prices and text for stock movement prediction based on information fusion,” Engineering Applications of Artificial Intelligence, vol. 127, p. 107377, 2024.

[9] S. Khan and H. Alghulaiakh, “Arima model for accurate time series stocks forecasting,” International Journal of Advanced Computer Science and Applications, vol. 11, no. 7, 2020.

[10] L. Wang, F. Ma, J. Liu, and L. Yang, “Forecasting stock price volatility: New evidence from the garch-midas model,” International Journal of Forecasting, vol. 36, no. 2, pp. 684–694, 2020.

[11] H. N. Bhandari, B. Rimal, N. R. Pokhrel, R. Rimal, K. R. Dahal, and R. K. Khatri, “Predicting stock market index using lstm,” Machine Learning with Applications, vol. 9, p. 100320, 2022.

[12] B. G¨ulmez, “Stock price prediction with optimized deep lstm network with artificial rabbits optimization algorithm,” Expert Systems with Applications, vol. 227, p. 120346, 2023.

[13] Y. A. GIWA, “Over the counter stocks data acquisition & analysis with time series prediction,” International Journal of Social Sciences and Scientific Studies, vol. 4, no. 1, pp. 3643–3670, 2024.

[14] D. Song, A. M. C. Baek, and N. Kim, “Forecasting stock market indices using padding-based fourier transform denoising and time series deep learning models,” IEEE Access, vol. 9, pp. 83 786–83 796, 2021.

[15] S. Verma, S. P. Sahu, and T. P. Sahu, “Discrete wavelet transform-based feature engineering for stock market prediction,” International Journal of Information Technology, vol. 15, no. 2, pp. 1179–1188, 2023.

[16] Y. Luo, “Wavelet neural network model with time-frequency analysis for accurate share prices prediction,” in Intelligent Computing: Proceedings of the 2021 Computing Conference, Volume 3. Springer, 2021, pp. 286–297.

[17] D. Cao, F. Jia, S. O. Arik, T. Pfister, Y. Zheng, W. Ye, and Y. Liu, “Tempo: Prompt-based generative pre-trained transformer for time series forecasting,” arXiv preprint arXiv:2310.04948, 2023.

[18] T. Guo and E. Hauptmann, “Fine-tuning large language models for stock return prediction using newsflow,” arXiv preprint arXiv:2407.18103, 2024.

[19] Y. Jiao and J. Jakubowicz, “Predicting stock movement direction with machine learning: An extensive study on s&p 500 stocks,” in 2017 IEEE International Conference on Big Data (Big Data). IEEE, 2017, pp. 4705–4713.

[20] S. Feng, C. Xu, Y. Zuo, G. Chen, F. Lin, and J. XiaHou, “Relation-aware dynamic attributed graph attention network for stocks recommendation,” Pattern Recognition, vol. 121, p. 108119, 2022.

[21] Q. Bui and R. ´Slepaczuk, “Applying hurst exponent in pair trading strategies on nasdaq 100 index,” Physica A: Statistical Mechanics and its Applications, vol. 592, p. 126784, 2022.

[22] V. Cioli, T. Giacomo, A. Giannozzi, and O. Roggi, “Stock liquidity and corporate investment policy after ftse 100 index additions,” International Journal of Business and Management, vol. 17, no. 10, 2022.

[23] Z. Yang, T. Zhao, S. Wang, and X. Li, “Mdf-dmc: A stock prediction model combining multi-view stock data features with dynamic market correlation information,” Expert Systems with Applications, vol. 238, p. 122134, 2024.

[24] Y. Liu, S. Huang, X. Tian, F. Zhang, F. Zhao, and C. Zhang, “A stock series prediction model based on variational mode decomposition and dual-channel attention network,” Expert Systems with Applications, vol. 238, p. 121708, 2024.

[25] M. Lu and X. Xu, “Trnn: An efficient time-series recurrent neural network for stock price prediction,” Information Sciences, vol. 657, p. 119951, 2024.

[26] K. Chaudhari and A. Thakkar, “Data fusion with factored quantization for stock trend prediction using neural networks,” Information Processing & Management, vol. 60, no. 3, p. 103293, 2023.

[27] N. Wijerathne, J. Samarathunge, K. Rathnayake, S. Jayasinghe, S. Ahangama, I. Perera, V. Dhananjaya, and L. Sivaneasharajah, “Deep learning based personalized stock recommender system,” in International Conference on Neural Information Processing. Springer, 2023, pp. 362–374.

[28] A. Mahmoodi, L. Hashemi, M. Jasemi, S. Mehraban, J. Lalibert´e, and R. C. Millar, “A developed stock price forecasting model using support vector machine combined with metaheuristic algorithms,” Opsearch, vol. 60, no. 1, pp. 59–86, 2023.