Dean&Francis

ISSN 2959-6130

Research on Stock and Futures Price Forecasting Based on Machine Learning

Wendi Wang

Big Data and Artificial Intelligence School, Anhui Xinhua University, Hefei 230000, China

Abstract:

Impact on their decision-making process. Applying machine learning techniques to futures price forecasting can assist investors in making more rational investment decisions. Based on this, this paper selects the trading data of the "CSI 300 Futures" from 9:31:00 on January 3, 2017, to 15:00:00 on December 31, 2021, as a sample, and uses the Transformer model, Prophet model, and XGBoost model to predict and analyze its stock price trends. The final evaluation results show that the XGBoost model performs best in terms of prediction accuracy, while the Transformer model shows greater potential.

Keywords: Futures Price Forecasting, Transformer, Prophet, XGBoost

1.Introduction

With the acceleration of globalization and the rapid advancement of information technology, financial markets have become increasingly complex and interconnected. As a key component of the financial system, stock market price fluctuations are influenced by a variety of factors, including the macroeconomic environment, changes in policies and regulations, the financial health of companies, and investor sentiment[6]. Traditional financial theory, based on the efficient market hypothesis, posits that stock prices reflect all available information, making future stock price trends difficult to predict. However, behavioral finance research indicates that due to irrational investor behavior and information asymmetry, stock prices may overreact or underreact to new information, thus offering potential for prediction[5].

Despite these advancements, existing research still faces numerous challenges. On one hand, traditional statistical models like ARIMA can effectively capture the characteristics of linear time series, but they are less effective in handling high-dimensional and nonlinear market data. On the other hand, while early machine learning methods have improved prediction accuracy, they often overlook the time-dependent and long-term memory characteristics of time series data[7]. Furthermore, although deep learning models such as LSTM excel at processing sequence data, optimizing these models to adapt to the dynamic changes in specific markets remains a significant challenge[6].

Given the aforementioned background and challenges, this study focuses on the CSI 300 Futures as the research subject. Using approximately 29,000 historical trading data points, it explores the performance of three advanced machine learning models—Trans-

ISSN 2959-6130

former, Prophet, and XGBoost—in stock price prediction. Specifically, we evaluate the performance of each model by calculating the coefficient of determination (R^2), mean absolute error (MAE), and mean squared error (MSE), and compare their applicability in various scenarios. This study aims to validate the practical utility of modern machine learning technologies in the financial sector, provide investors with a more scientific basis for investment decisions, and advance the academic research on modeling methods for complex financial markets [8].

The data set used in this study is large in scale and covers market activities over a long period of time, which helps to improve the quality and generalization ability of model training. By comparing the results of the three models, we can more clearly understand their advantages and limitations, so as to choose the most appropriate solution for practical application.

2. literature review

In recent years, as big data technology and machine learning methods have matured, financial market predictions have shifted from traditional statistical models to more efficient intelligent algorithms. To address the complex characteristics of market nonlinearity and dynamic changes in stock and futures prices, researchers have developed various models. The Transformer, Prophet, and XGBoost models used in this paper are among the most prominent methods in this field that have gained significant attention in recent years.

The Transformer model, initially proposed by Vaswani et al., is based on the self-attention mechanism, which uniquely excels in capturing long-range dependencies [1]. Despite its significant achievements in natural language processing and other fields, the model's reliance on large-scale data and high computational complexity pose challenges in financial time series prediction. Additionally, when dealing with high-frequency market fluctuations, the Transformer faces challenges in parameter tuning and model stability.

The Prophet model, developed by Taylor and Letham, is designed to provide a user-friendly and automated time series prediction tool for business applications [2]. It constructs a prediction framework by decomposing trends, seasonality, and special events (such as holiday effects), offering robustness and low tuning costs. However, the decomposition-based design limits Prophet's ability to capture sudden nonlinear fluctuations in financial markets, making it challenging to fully capture complex market dynamics.

The XGBoost model, an efficient gradient boosting tree method introduced by Chen and Guestrin, excels in mining nonlinear features and handling large-scale data [3]. In financial prediction tasks, XGBoost can effectively capture the complex relationships among multidimensional features through ensemble learning. However, its description of the inherent time dependence in time series data is relatively weak, which has led some researchers to integrate it with other time series models to compensate for the limitations of a single model [4].

Although existing literature has explored the application of Transformer, Prophet, and XGBoost in financial forecasting, comparative studies of multiple models on specific datasets, such as the CSI 300 futures data, remain scarce. This paper aims to systematically compare these three models using actual CSI 300 futures trading data, explore their respective strengths and limitations, and propose potential model integration strategies to provide more precise technical support for short-term price prediction in the futures market.

3. Model introduction

3.1 Transformer model

The Transformer model was originally proposed by Vaswani et al., whose core lies in the self-attention mechanism, which can effectively capture the global dependence relationship between different positions in the sequence, so as to make up for the deficiency of traditional cyclic models in long sequence processing [1].

For the input sequence X, the query (Q), key (K) and value (V) matrices are generated through linear transformation. The basic attention calculation formula is as follows:

$$\mathrm{Attention}(Q,K,V)=\mathrm{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

Among them, d_k represents the dimension of the key vector. In the multi-head attention mechanism, each attention head calculates the above expression separately, and then concatenates the outputs of each head and obtains the final result through linear mapping:

MultiHead(Q, K, V) = Concat($head_1$, $head_2$, …, head_h) W^o

Among them,

$head_i$ = Attention (QW_i^Q, KW_i^K, VW_i^V)

W_i^Q, W_i^K and W_i^V are their respective weight matrices, and W^o is the output transformation matrix.

In this study, we use the Transformer model for futures price prediction. By constructing historical trading data as sequence input and using multi-layer encoder structure to capture long-term trends and short-term fluctuations in the data, we can achieve high-precision prediction of future prices.

3.2 Prophet model

The Prophet model, proposed by Taylor and Letham, is a time series prediction method based on additive decomposition [2]. This method assumes that the time series can be decomposed into four parts: trend, seasonality, holiday effect and random error, whose mathematical expression is as follows:

$$\boldsymbol{y}(t) = \boldsymbol{g}(t) + \boldsymbol{s}(t) + \boldsymbol{h}(t) + \varepsilon(t)$$

Among them, g(t) describes the trend component, which is usually adopted as a piecewise linear or logarithmic growth function; s(t) represents the seasonal variation, which is a periodic function, and its modeling method is often based on Fourier coefficients:

$$s(t) = \sum_{n=1}^{N} [a_n \cos(2\pi nt / P) + b_n \sin(2\pi nt / P)]$$

H (t) captures the impact of holidays or special events on the data, and ε (t) is the noise term.

The Prophet model is simple to design and relatively easy to optimize, making it highly practical for business forecasting. This paper uses this model to model cyclical changes and sudden events in the CSI 300 futures data. However, its additive assumption may limit the depiction of complex nonlinear relationships to some extent.

3.3 XGBoost model

The XGBoost (extreme gradient boosting) model is an ensemble learning algorithm based on gradient boosting decision trees, proposed by Chen and Guestrin [3]. The model fits the data residuals by iteratively building a series of regression trees, gradually improving the overall prediction performance. Its basic prediction formula is:

$$\hat{y}_i = \sum_{k=1}^{K} f_k(x_i),\ f_k \in \mathcal{F}$$

Among them, represents the function space composed of all possible regression trees. The objective function of XGBoost consists of loss terms and regularization terms:

$$\text{Obj} = \sum_{i=1}^{n} l(y_i, \hat{y}_i) + \sum_{k=1}^{K} \Omega(f_k)$$

The regularization term Ω(f) is used to control the complexity of the model. Its expression is:

$$\Omega(f) = \gamma T + \frac{1}{2}\lambda \|w\|^2$$

Among them, T is the number of leaves of the tree, w is the weight of leaf nodes, and γ and λ are regularization parameters. XGBoost uses second-order Taylor expansion to approximate the solution of the objective function, which effectively improves the optimization efficiency and generalization ability of the model.

In this study, XGBoost model was applied to the futures price prediction task. By capturing the complex nonlinear relationship between input features, it realized high-precision fitting of price fluctuations. Its excellent feature extraction and anti-overfitting ability made it perform well in financial time series prediction.

4. Empirical analysis

4.1 data sources

In financial time series prediction, data quality and sample size are crucial for model performance. To ensure the reliability and generalization of our predictions, we selected historical trading data from the CSI 300 Futures as our research subject. The data spans from January 3, 2017, at 9:31:00 to December 31,2021, at 15:00:00, comprising approximately 29,000 transaction records that include key indicators such as opening price, closing price, highest price, lowest price, and trading volume. To reduce noise and enable different models to more effectively learn market trends, we normalized the data and used a sliding

ISSN 2959-6130

window method to construct time series samples.

In terms of data partitioning, the dataset is divided into an 80% training set and a 20% test set to ensure that the model can learn from a sufficient amount of data while avoiding overfitting. All data undergoes rigorous cleaning and preprocessing to ensure the accuracy and stability of the analysis [3]. This study uses CSI 300 futures data and employs three machine learning models—Transformer, Prophet, and XGBoost—to conduct comparative experiments, exploring the performance of different models in financial time series prediction.

4.2 Transformer model training and results

(1) data preprocessing

During the training of the Transformer model, a sliding window method is used to construct time series samples. For each input sequence, we select 64 time steps (sequence_length=64), meaning the data from the first 64 time points are used to predict future prices. All data is normalized using the MinMaxScaler to enhance the model's stability and convergence speed.

(2) Model architecture

The Transformer model consists of an input layer, a Transformer encoder layer, a multi-head self-attention mechanism and a fully connected output layer. Its structure is as follows:

Input layer: The data is transformed linearly to map it to the feature space of dimension d_model=128.

Transformer encoder: It contains three layers of Transformer encoder with each layer consisting of a multi-head attention layer and a feedforward neural network.

Output layer: The full connection layer is used for price prediction.

(3) Model training and optimization

The model training uses Adam optimizer (learning_rate=0.001) and MSE as the loss function. Meanwhile, we adopt the Early Stopping mechanism to stop the training when the verification loss does not decrease for 5 consecutive times to prevent overfitting.

(4) Prediction results and model evaluation

After 10 epochs of training, the final test set prediction results of the Transformer model are as follows:

MSE = 0.00048

MAE = 0.01947

R^2 = 0.9799

4.3 Prophet model training and results

(1) data preprocessing

The Prophet model requires that the time series data have a timestamp (datetime) as the primary key, so we convert the data set to the format suitable for Prophet:

DS (time column)

Y (target variable, i.e. closing price)

At the same time, we use StandardScaler to avoid the influence of numerical scale on model fitting.

(2) Prediction results and model evaluation

After training, the Prophet model's prediction performance on the test set is as follows:

MSE = 0.00231

MAE = 0.03285

R^2 = 0.8912

4.4 XGBoost model training and results

(1) data preprocessing

XGBoost needs to convert the data into a feature matrix (X) and target variable (y), so we construct the following features:

Moving average (MA) of closing price, Volatility (Volatility), KDJ index, RSI index, volume, etc.

All features are normalized by MinMaxScaler.

(2) Prediction results and model evaluation

XGBoost performance on the test set:

MSE = 0.00032

MAE = 0.01521

R^2 = 0.9845

Table 1 Comparison of model prediction results

model	MSE	MAE	R^2
Transformer	0.00048	0.01947	0.9799
Prophet	0.00231	0.03285	0.8912
XGBoost	0.00032	0.01521	0.9845

Based on the results in Table 1

This paper uses data from the CSI 300 futures to predict futures price trends using three machine learning models: Transformer, Prophet, and XGBoost. By comparing the prediction performance of these models, we found that the XGBoost model outperforms the others, excelling in all evaluation metrics (Mean Squared Error (MSE), Mean Absolute Error (MAE), and R^2).

References

[1] Taylor, S. J., & Letham, B. (2018) Forecasting at scale. The American Statistician, 72(1): 37–45.

[2] [2] Chen, T., & Guestrin, C. (2016) XGBoost: A Scalable Tree Boosting System. Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining: 785–794.

[3] [3] Vaswani, A., Shazeer, N., Parmar, N., et al. (2017) Attention is All You Need. Advances in Neural Information Processing Systems, 30: 5998–6008.

[4] [4] Zhang, Y., Qin, X., & Huang, T. (2019) A Hybrid Forecasting Framework for Financial Time Series. Expert Systems with Applications, 129: 132–142.

[5] Chen Haiqiang, Chen Liqiong, Li Yingxing, Luo Xiangfu (2025) Can high-frequency data improve stock price prediction? —— An empirical study based on functional data. In: China Financial Engineering Annual Conference. Beijing. Page number not provided.

[6] Hou Yanan (2025) Stock Price Prediction Based on Bi-LSTM Deep Learning. In: Global AI and Finance Summit, Shanghai. Page number not provided.

[7] Gumparthi, S., & Prasad, V. V. (2025). Using statistical and machine learning techniques for stock price prediction. Presented at the International Conference on Quantitative Trading, Singapore. No page number provided.

[8] WYL, S. (2025). Original stock prediction based on deep learning. In: Asian Fintech Innovation Forum, Hong Kong