Stock Price Prediction: A Comparative Study of Random Forest and LSTM Models

Xun Wang*

School of Mathematics, Statistics and Mechanics, Beijing University of Technology, Beijing, 100097, China

*Corresponding author: jerry@emails.bjut.edu.cn

Abstract. With the escalating globalization and intricacies of financial markets, accurate stock price forecasting has become paramount for investors, analysts, and researchers. This study compares the effectiveness of Random Forest and Long Short-Term Memory (LSTM) in predicting Tesla stock prices. Utilizing historical data refined for sequential analysis, both models were trained and tested. Evaluation metrics such as Mean Square Error (MSE) and accuracy were used to assess predictive prowess. Results indicate that LSTM exhibits superior accuracy in forecasting Tesla stock prices, owing to its proficiency in managing long-term dependencies and nonlinear relationships inherent in stock price time series data. Conversely, Random Forest's performance was relatively limited. On the other hand, the random forest model has the advantage of running time much lower than LSTM. This research underscores the significance of model selection tailored to the unique characteristics of financial data and opens avenues for future explorations in optimizing predictive models and translating insights into practical stock trading strategies.

Keywords: Stock price; time series analysis; random forest; long short-term memory network.

1. Introduction

In the financial market domain, stock price prediction has always been a focal point and a challenging task for researchers. With the rapid advancement of information technology and the advent of the big data era, researchers have begun to employ various cutting-edge data analysis techniques to unravel the inherent patterns of the stock market. This study aims to systematically compare the performance disparities of two models: random forest, and Long Short-Term Memory (LSTM) in stock price prediction through time series analysis, while assessing their predictive accuracy.

In the realm of stock price prediction, time series forecasting emerges as a pivotal approach, as evidenced by the exhaustive examination by Hellstrom, who presents a comprehensive analysis and introduction to time series prediction, encompassing diverse techniques of technical and fundamental analysis, setting the stage for future predictive efforts [1]. Nonetheless, the pursuit of developing a precise predictive model continues to be a challenging task. Linear Regression, as a straightforward statistical method, has seen widespread application in stock price predictions. Studies by Karim et al. have utilized linear regression and decision tree regression to study into the stock market, confirming their effectiveness in tracking trends in stock prices [2]. Nonetheless, when confronted with potentially complex nonlinear relationships inherent in stock market fluctuations, linear regression's limitations are evident.

The researchers embarked on a quest to overcome the limitations of linear regression by delving into and experimenting with more sophisticated machine learning methods. The use of Random Forest in ensemble learning demonstrates notable efficacy in predicting stock prices. Khaidem et al. successfully leveraged random forest models to predict stock market price directions, thereby confirming its superiority in handling high-dimensional data and nonlinear associations [3]. Moreover, Nti et al. employed random forest for feature selection based on macroeconomic variables, further improving the precision of stock market forecasts [4]. The performance of random forests, however, remains constrained by the selection of features and the configuration of model parameters.

Lately, the utilization of deep learning techniques for forecasting stock prices has become increasingly popular. Algorithms based on deep learning are capable of autonomously deriving complex features from data and tackle intricate nonlinearities. Within this context, Mukherjee et al. proposed the utilization of artificial neural network (ANN) and convolutional neural network (CNN) models for stock price prediction, yielding remarkably high accuracy [5]. Further, Rikukawa et al. employed a Recurrent Neural Network (RNN) model, utilizing dynamic time warping as a similarity measure for stock prices in time series data, attesting to its effectiveness [6]. Expanding on this, Li et al. integrated recurrent neural networks with sentiment analysis for stock volatility prediction [7]. This multisource data fusion approach introduces a novel perspective to stock price prediction. Traditional RNNs, however, may encounter issues like vanishing or exploding gradients when dealing with long sequences, impeding their ability to capture long-term dependencies. To address this, LSTM was introduced. First proposed by Hochreiter et al., the Long Short-Term Memory network (LSTM) boasts a unique memory mechanism and gating structures, making it particularly adept at processing sequential data [8]. Subsequent research by Walaa Makhlouf and Yash Upadhyay et al. verified the efficiency of LSTM in predicting stock prices [9, 10].

Although significant progress has been made in the field of stock price forecasting through numerous research endeavors, there remain certain issues that merit further exploration and investigation. The performance disparity across different prediction models varies significantly across datasets and environments, emphasizing the criticality of model selection in diverse contexts and the need to further investigate their applicability and generalization capabilities.

This study aims to extend previous research by delving into the application of random forest, and LSTM in stock price time series prediction. Through a comparative analysis of these models' predictive accuracy, stability, and computational efficiency, this study strives to provide constructive recommendations for the selection of mathematical theories and methods in time series prediction. It is hoped that this research will offer valuable insights and inspiration to researchers in related mathematical fields, enhancing the deeper use of mathematics in analyzing time series data.

2. Methods

2.1. Data Source

Within this paper's methodology segment, this paper will elaborate on the employed stock price prediction method and offer detailed numerical setups to guarantee the reproducibility of the study. Initially, At the outset, the paper analyzed Kaggle's public dataset and selected Tesla's recent stock price as the target for historical trading data. The data set contains the opening price, day high price, day low price, closing price and total number of trades of Tesla stock from 2016 to 2021.

2.2. Index Selection and Description

In the data set, the date and closing price are selected as indicators. The closing price is the last traded price of a stock at the end of the trading day, and it reflects the combined result of all the buying and selling activity of the stock in the market that day. The advantage of choosing the daily closing price as a predictor is its comprehensiveness and accessibility. Comprehensiveness is reflected in the fact that it can reflect the overall performance of the stock in a day, and easy access means that this data can be easily obtained from major financial data platforms or exchanges to provide a basis for subsequent predictive analysis. Visualize the dataset using a chart, and the result is shown in Figure 1:

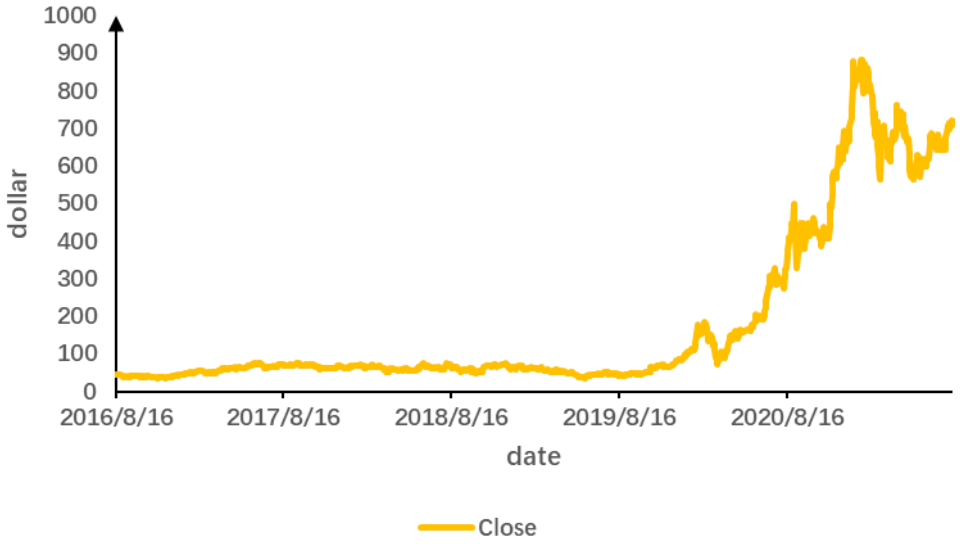

Fig. 1 2016-2020 Tesla stock price data

2.3. Method Introduction

To guarantee data quality and integrity, I performed preprocessing tasks such as handling missing values and detecting outliers. For the machine learning model selected for analysis, I will evaluate the performance by mean square error (MSE) and R-square scores, and the calculation formula is as follows:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \quad (1)$$

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

The closer the R-square score is to 1, the better the model fits the data, and the higher the prediction accuracy of the model. The mean square error is a measure of the difference between the predicted value and the true value, and the smaller the value, the smaller the prediction error of the model.
In the choice of programming language, the paper use Python to build the model. The first method adopted is random forest. Random forest is a supervised learning algorithm based on ensemble learning that improves the accuracy of classification or regression tasks by constructing multiple decision trees and aggregating their predictive results.

Next, the second model employed in this study is the long short-term memory model (LSTM), which will be compared to the random forest regression model. The Long Short-Term Memory (LSTM) model is a special type of recurrent neural network (RNN), which is designed to avoid long-term dependency problems, remember long-term information, and access it when needed. In terms of the use of the dataset, the first 80% of the data is used to train the model and the remaining 20% is used to test the model. Finally, this paper uses the Matplotlib library to plot the real prices and the predicted prices on the test set for the results of these two models, respectively.

3. Results and Discussion

3.1. Random Forest Results

Based on the aforementioned methodology, the dataset is directly inputted into the random forest model for stock price prediction. The data is divided into a training set and a test set, with the first 80% used for training and the remaining 20% reserved for testing. The running result is shown in the figure 2.

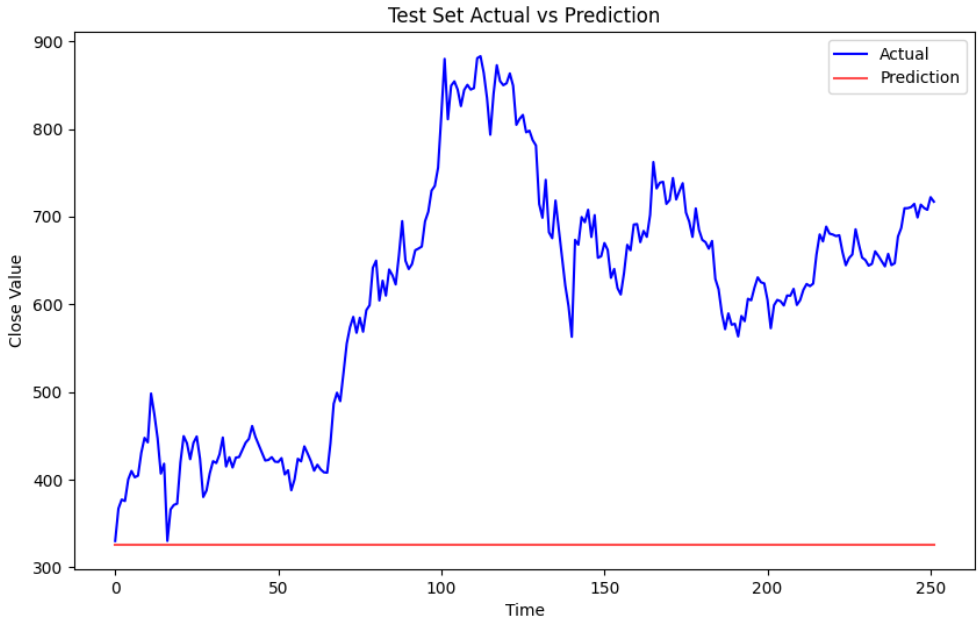

Fig. 2 Comparison of the predicted value and the actual value from the random forest model

As can be seen from the figure 2, if the random forest regression model is directly trained using the dataset, the final prediction result is represented as a straight line, which deviates significantly from the actual values. This indicates that due to factors such as limited input features for time series prediction and high nonlinearity in the data, the model fails to capture the fundamental relationships within the data, resulting in underfitting. Therefore, it provides no valuable reference for stock price prediction.

To accomplish this, the paper optimized the model by creating a lag feature. By using past observations as features to predict future target values, lag features provide the historical information the model needs to make predictions, helping the model capture time dependencies and trends in price movements. Set the lag number to 1000, that is, consider the stock price data of the past 1000 trading days to predict the future price. Through the lag feature function, a data frame containing 1000 lag features is generated, each feature corresponds to the historical price data with different lag. Then, the data set is divided into a training set and a test set, with a ratio of 80-20. Specifically, the first 80% of the data is used to train the model and the remaining 20% is used to test the model. After that, the number of trees in the random forest is set to 100 and the random seeds are set to 42 to ensure consistent results every time the code is run.

According to the aforementioned method, the results obtained from executing improved random forest model are as follows:

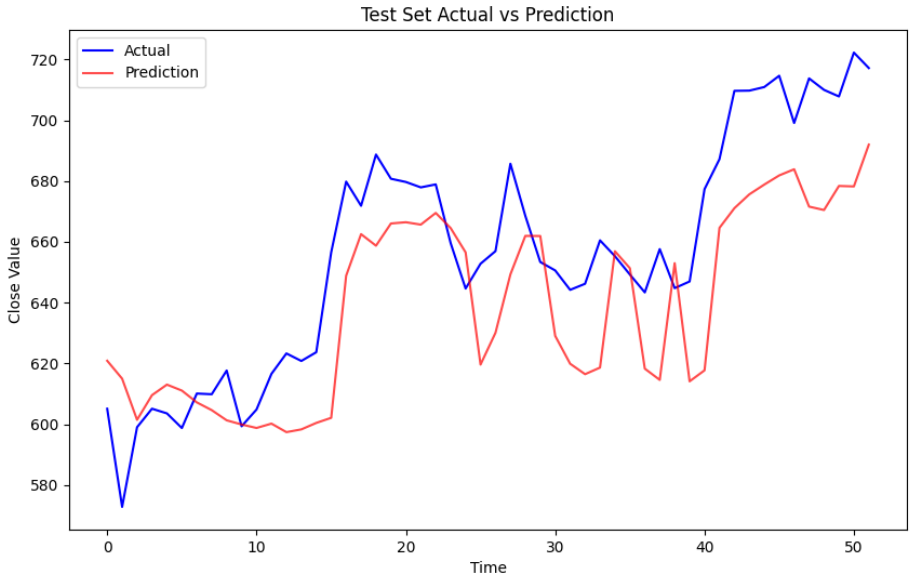

Fig. 3 Comparison of the predicted and the actual value from the improved random forest model

It can be observed according to figure 3 that although there are still some errors in the predictions made by the optimized model, it has nonetheless demonstrated a certain degree of effectiveness in stock time series prediction. Specifically, the Test Mean Squared Error (MSE) of 706.409indicates the presence of discrepancies between the model's predicted values and the actual stock prices in the test dataset. On the other hand, the Test R-squared (R^2) score of 0.518 suggests that the model accounts for approximately 51.87% of the variability in the stock prices. While this score is not exceptional, it does imply that the model has captured some significant patterns and trends in the data, enabling it to make predictions that are moderately correlated with the actual outcomes. These results suggest that the optimized model, despite its limitations, has shown some promise in stock time series prediction.

3.2. LSTM Model Results

Regarding the execution of the LSTM model, firstly, MinMaxScaler is used to normalize the data and scale its range between 0 and 1 to eliminate the influence of different dimensions on the model. Next, in the construction of the model, a neural network structure consisting of two LSTM layers is adopted. The first layer LSTM sets up 50 neurons and then passes the output sequence to the second layer LSTM. The second layer of LSTM also contains 50 neurons, which ultimately produce a single output. Finally, a Dense layer is used to map the LSTM output to the final predicted value. During model training, Mean Squared Error is used as the loss function, and the ADAM optimizer is selected for parameter optimization. Subsequently, the dataset is split into a training set and a test set, with the test set accounting for 20% of the total data. The ModelCheckpoint callback function is employed to save the model weights with the minimum loss on the validation set. Lastly, the model weights that perform best on the validation set are loaded, and predictions are made on the test set. And the results obtained from executing the LSTM model are as follows:

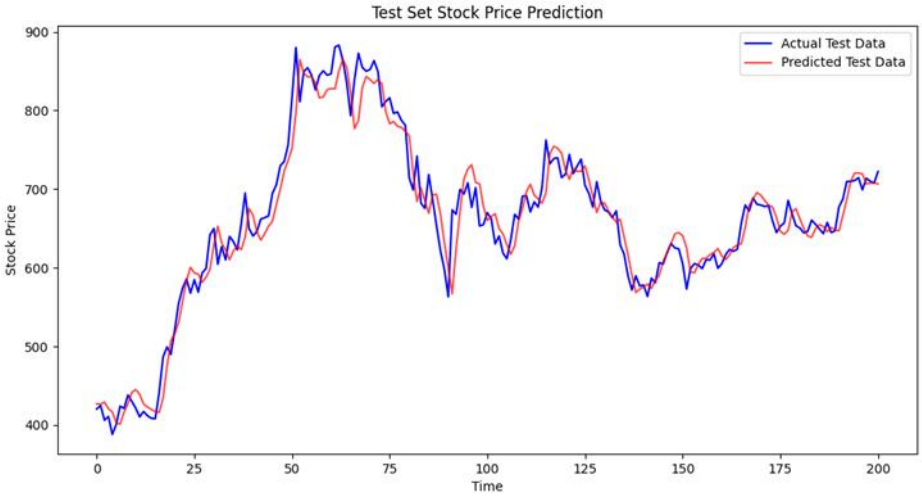

Fig. 4 Comparison of the predicted value and the actual value from the LSTM model

It can be seen from Figure 4 that the test results for the LSTM model are highly encouraging. Specifically, the model achieved an impressive R^2 score of 0.9430944069184133 and a Mean Squared Error (MSE) of 684.407. These metrics suggest that the model has achieved excellent prediction accuracy on the test set, indicating a strong correlation between the predicted and actual stock prices. The high R^2 score signifies that the LSTM model explains a significant portion of the variance in the stock price data, capturing key patterns and trends effectively. This score is very close to 1, which is the ideal value, indicating that the model's predictions align closely with the actual outcomes. At the same time, the relatively low MSE value demonstrates that the model's predictions deviate minimally from the true stock prices, on average. A lower value is desirable as it suggests less variance in the errors.

These results suggest that the LSTM model exhibits strong learning and generalization abilities when dealing with stock price prediction. And the high accuracy demonstrated in time series prediction suggests that this model is more suitable for stock price forecasting.

Comparing the results shown in Figure 3 and Figure 4, it is clear that even though the accuracy of the Random Forest model in time series prediction has been improved by incorporating lagged features, the LSTM model still demonstrates better performance in this aspect, making it more suitable for stock price forecasting in a time series context. Obviously, the LSTM's ability to capture temporal dependencies and patterns effectively allows it to outperform the Random Forest in predicting stock prices, highlighting its strength in handling sequential data.

4. Conclusion

Using Tesla stock as a case study, the research thoroughly examines the use of random forest regression model (RFR) and long short-term memory network (LSTM) in predicting stock prices over time, and thoroughly assesses the pros and cons of both models.

Initially, regarding the precision of predictions, the LSTM model demonstrates notable benefits. Experimental findings indicate that Long-Term Memory (LSTM) is markedly more accurate in forecasting Tesla stock prices, largely due to its distinctive memory feature, enabling it to accurately record long-term patterns in time series data. Conversely, the stochastic forest regression model exhibits significant predictive inaccuracies in handling intricate time series data, making it challenging to precisely represent the market's dynamic shifts.

Additionally, each model presents unique benefits and drawbacks regarding their design and functionality. Random forest regression model is simple, easy to implement and fast to train, which is suitable for processing time series data with fewer features and short time span. However, when faced with time series data with a large number of features and long-term dependencies, its predictive

performance may be limited. In contrast, LSTM models have more complex structures and higher computational requirements, requiring more time and resources to train. However, once trained, the LSTM model performed well on time series data with long-term dependencies, providing more accurate predictions.

Furthermore, the LSTM model demonstrates significant benefits regarding stability. In contrast to the random forest regression model, the LSTM model demonstrates superior resilience against market variations and uncertainties. Consequently, in real-world scenarios, the LSTM model demonstrates greater flexibility in response to the evolving dynamics of the stock market.

In summary the research indicates the superiority of the LSTM model over the random forest regression model regarding the precision and steadiness of stock price time series forecasts, yet the random forest regression model is easy to construct and has a short running time. Therefore, the selection of the right model should be based on the specific data characteristics, forecasting needs, and computing resources should be considered.

References

[1] Karim R, Alam M K, Hossain M R. Stock Market Analysis Using Linear Regression and Decision Tree Regression. International Conference on Emerging Smart Technologies and Applications, IEEE, 2021.

[2] Nti I K, Adekoya A F, Weyori B A. Random Forest Based Feature Selection of Macroeconomic Variables for Stock Market Prediction. American Journal of Applied Sciences, 2019.

[3] Mukherjee S, et al. Stock market prediction using deep learning algorithms. Journal of intelligent technology, 2023, 8(1): 82-94.

[4] Rikukawa S, Mori H, Harada T. RECURRENT NEURAL NETWORK BASED STOCK PRICE PREDICTION USING MULTIPLE STOCK BRANDS. International journal of innovative computing, information and control, 2020, 16.

[5] Li P, Wan T, Liu Y, et al. Stock Volatility Prediction Using Recurrent Neural Networks with Sentiment Analysis. International Conference Industrial, Engineering & Other Applications Applied Intelligent Systems, 2017.

[6] Hochreiter S, Schmidhuber J. Long short-term memory. Neural computation, 1997, 9(8), 1735-1780.

[7] Fang Yiqiu, Lu Zhuang, Ge Junwei. Stock price prediction using the combined RMSE loss LSTM-CNN model. Computer Engineering and Applications, 2022.

[8] Zhang Ruixue, Hao Yongtao. Research on Stock Price Prediction Based on Deep Learning. Computer Knowledge and Technology, 2023, 33: 8-10.

[9] Deng Dejun, Xu Hongzhen, Wei Shiyue. Stock Price Prediction of E-V-ALSTM Model. Computer Engineering and Applications, 2023, 59(6): 12.

[10] Jiang Minghua, Chen Yun. Application of Improved Multilayer Graph Attention Network in Stock Price Prediction. Computer Engineering and Applications, 2022, 58(3): 7.