Proceedings of the 4th International Conference on Signal Processing and Machine Learning
DOI: 10.54254/2755-2721/55/20241572

Stock price forecasting by ARIMA, linear regression, LSTM and decomposition linear models

Chenyu Wang

Department of Systems Engineering and Engineering Management, The Chinese University of Hong Kong, Hong Kong, China

1155124593@link.cuhk.edu.hk

Abstract. Stock price forecasting has piqued the interest of academics studying finance and economics since it is a challenging but critical task for investors in financial markets. A variety of methods has been deployed, attempting to extract useful information to tackle the prediction problem. This paper presents rigorous research on the task of predicting the next-day price of Tesla stock by its past prices. A variety of methods are tested, including the long short-term memory (LSTM), the neural network model (NN), the autoregressive integrated moving average model (ARIMA), and the decomposition linear model (DL). The result is measured with mean absolute error (MAE). As shown in the experimentation result, the MAE of the ARIMA model is 7.2400, the MAE of the neural network is 5.8770, the MAE of the LSTM model is 6.8390 and the MAE of the decomposition model is 5.5890. The result suggests that the DL model performs the best in terms of MAE.

Keywords: ARIMA, Linear Regression, LSTM, Decomposition Linear.

1. Introduction

Forecasting stock prices remains a continual focal point of interest in the field of financial studies. Researchers are constantly seeking improved methods to predict stock price movements and make informed decisions. The dynamic and often unpredictable nature of financial markets makes this task particularly challenging.

Currently, numerous researchers made attempts in price forecasting. Adebiyi, etc., treat stock prices as time series and use the statistical model, ARIMA, to analyze the pattern of price movements [1]. The order that generates the lowest Bayesian information criterion (BIC) value on the training set is selected to predict on the test set. Besides, machine learning methods have also been experimented on the stock price forecasting tasks. Support vector machine (SVM) [2], artificial neural networks (ANN) [3], deep neural networks (DNN) [4], recurrent neural network (RNN) [5], LSTM [6] etc., have been commonly used in the early research and have generated satisfactory results in stock forecasting. With the advancement of machine learning techniques, transformer models with the attention mechanism are applied to the stock price forecasting [7, 8]. Moreover, Zhang, e.g., deploys the generative adversarial network (GAN) to predict the stock price and attained the cutting-edge results against other machine learning models [9]. However, Zeng, etc., expressed doubts on the effectiveness of stunningly complicated transformer models [10]. They showcased a simple single linear layer model could beat transformer models and to extend their model, a decomposition model was proposed to be applied to price forecasting.

Tesla, Inc. is one of the corporations that have the biggest capitalization and an industry leader in technology. Its stock price is representative of the US stock market. Therefore, this study is focused on the forecasting of Tesla stock prices.

This paper experiments four models, i.e., ARIMA, neural network, LSTM and decomposition linear, on the Tesla stock price forecasting task. The data are segmented into three groups, labeled as training, validation and test. The first 80% of the data make up the training set as it is irrational to learn on future values. The remaining 10% makes up the validation portion. And the remaining 10% comprises the test set. The experimentation process presents that the DL model achieves the best performance (the lowest MAE).

2. Data

This study uses the adjusted closed stock prices of Tesla from the python package yfinance 0.2.31, incorporating the yahoo finance API, ranging from 2016/01/04 to 2023/10/05, which contains 1953 samples. The minimum is 9.578 and the maximum is 409.97. The prices have a standard deviation of 102.56 and a mean of 110.15 (Basic information can be seen in Table 1). The price of stocks is predicted for the following day using data from the previous 20 days.

Table 1. Dataset statistics

Statistic	Value
count	1953
mean	110.15
Standard deviation	102.56
min	9.578
max	409.97

Unprocessed data can cause weaker training efficiency and forecasting performance. Here this study adopts standard scaling to preprocess the data, which is to subtract the mean from all values and have the results divided by the standard deviation. The scaling process can make the data regularized, which ensures that the samples' mean is zero and their standard deviation is one.

$$\mathrm{x_i} = \frac{\mathrm{x_i} - \mu}{\sigma} \tag{1}$$

Where x_i is the actual price, μ represents the prices' mean and σ represents the standard deviation. Even though the trained model may perform admirably on the training dataset, the test set performance may fall short of expectations, which is called model overfitting. Therefore, in order to address the issue of model overfitting, the validation set is used. As a result, training, validation, and test sets of data are created. The first 80% of the data are included in the training set. The next 10% make up the validation set. And the test set is the remaining 10%.

The data is divided into two sets for the ARIMA section: a training set and a test set. The ARIMA training set consists of the first 90% of the data. The ARIMA test set is comprised of the remaining amount. The purpose of the training set is to determine the optimal order for the ARIMA model.

3. Methodology

3.1. ARIMA

ARIMA is a well-established statistical model, integrating autoregressive and moving average components with differencing to model and comprehend the temporal dependencies and trends within a time series. ARIMA models are adept at capturing short-term fluctuations and stationary behavior, rendering them particularly suitable for short-range stock price forecasting. An ARIMA model is a generalization of autoregressive moving average models with differencing, which is composed of a differencing, an autoregressive and a moving average part.

An autoregressive (AR) process suggests the value is related to the past values. Consider an AR process of order p, it can be expressed as follows.

$$y_t = \alpha_0 + \sum_{i=1}^{p} \alpha_i y_{t-i} + \varepsilon_t \quad (2)$$

Where y_i is the data point of the time series at the time i, α_j is the coefficient and ε_t is the residual. The time series may be non-stationary, bringing about the imprecision of the model. Therefore, a non-stationary time series can be converted to a stationary one using differencing. The first order differencing is expressed as follows.

$$z_t = y_t - y_{t-1} \quad (3)$$

$\{z_1, z_2, \dots\}$ is the differenced time series. Differencing can be conducted multiple times until the time series becomes stationary. Denote $y_t^* = \Delta^d y_t$ as the order-d differencing process.

In a moving average (MA) process, the value of a time series is related to the errors between the past values and its mean. Consider a MA(q) process of order q, it can be expressed as follows.

$$y_t = \mu + \varepsilon_t + \sum_{i=1}^{q} \beta_i \varepsilon_{t-i} \quad (4)$$

Where μ is the mean of the time series, ε_i is the error term at the time i, β_i is the coefficient at the time i. Thus, an ARIMA (p, d, q) process is express as follows.

$$y_t^* = \Delta^d y_t \quad (5)$$

$$y_t^* = \mu + \sum_{i=1}^{p} \alpha_i y_{t-p}^* + \sum_{i=1}^{q} \beta_i \varepsilon_{t-i} + \varepsilon_t \quad (6)$$

3.2. Neural Network

Drawing inspiration from the intricacies of the human brain's architecture, neural networks comprise a wide-ranging family of machine learning models., which are constructed from layers of interconnected artificial neurons, enabling them to capture complex, non-linear patterns in data. Neural networks offer a versatile framework for stock price forecasting, as they can effectively adapt to diverse data patterns and relationships. However, they may require substantial training data and meticulous hyperparameter tuning to achieve optimal performance. Denote the input as $\boldsymbol{x}_i = \{x_{i1}, \dots, x_{in}\}^T$, the weight matrix as W and the feature as y_i.

$$\widehat{y_i} = W \boldsymbol{x}_i + \boldsymbol{b}_i \quad (7)$$

where $\widehat{y_i}$ is the predicted value, $\mathbf{b_i}$ is the bias.

3.3. LSTM

LSTM, a particular variant of recurrent neural networks, is specifically engineered to effectively recognize and represent extended temporal relationships within sequential data. LSTMs excel in modeling time series data due to their remarkable capacity to remember and encode information over extended time intervals. This characteristic renders them highly effective at capturing nuanced and long-range trends in stock prices, making them particularly suited for financial time series analysis. Compared with RNN, LSTM incorporates three gates, forget, input and output. Based on the present and previous

DOI: 10.54254/2755-2721/55/20241572

cell states, they can determine what data from the former cell state should be kept or deleted, which enables the LSTM to capture long-term dependencies. The process of LSTM is explained as follows.

$$\text{Forget gate: } f_t = \sigma(W_f x_t + b_f) \tag{8}$$

$$\text{Candidate: } \widetilde{C_t} = tanh(W_C x_t + b_C) \tag{9}$$

$$\text{Input: } i_t = \sigma(W_i x_t + b_i) \tag{10}$$

$$\text{Output: } o_t = \sigma(W_o x_t + b_o \tag{11}$$

$$\text{Cell state: } C_t = f_t * C_{t-1} + i_t * \widetilde{C_t} \tag{12}$$

$$\text{Hidden state: } h_t = o_t * tanh(C_t) \tag{13}$$

$$\text{Final: } y_t = W_y h_t + b_y \tag{14}$$

Where $tanh$ represents the hyperbolic tangent function and σ represents the sigmoid function. The symbols W_f, W_C, W_i, W_o represent the forget, candidate, input, and output gate weight matrices.

3.4. Decomposition Linear

Decomposition linear (DL) encompasses an array of methods aimed at dissecting a time series into its constituent components, including the moving average and the residual. Compared with linear models, this model offers a valuable framework for isolating specific patterns and structures within the data, capturing the seasonal trend of the time series.

In this model, first the moving average is calculated for each sample and subtract it from the sample to form a seasonal part and a residual part, each connected to a 2-layer network. The moving average is calculated by a moving average kernel. In order to preserve the length of each input, paddings are added to both the head and the tail.

For each input $x_t = \{x_{t1}, \ldots, x_{t1}\}$, it is decomposed into a seasonal part $s_t = \{s_{t1}, \ldots, s_{t1}\}$ and a residual part $r_t = \{r_{t1}, \ldots, r_{t1}\}$. Each part is connected to a single-layer neural network. The predicted value is given by the following.

$$\widehat{y_t} = W_s s_t + b_s + W_r r_t + b_r \tag{15}$$

Where W_s, b_s is the seasonal part's weight matrix and bias, W_r, b_r is the residual part's weight matrix and bias.

4. Results

4.1. ARIMA

For the ARIMA model to work, the time series must be stationary; otherwise, the trained model would be imprecise. The ARIMA training set is subjected to the ADF (Augmented Dickey Fuller) test. The results are shown in Table 2.

Table 2. ADF test results

Time series	Adf statistic	p-value
price	-1.3522	0.6049
1-order differenced	-7.5220	3.7688e-11

The ADF test results for the price suggest the series of tesla stock prices is non-stationary while the first order differencing is stationary. This study experiments from (0, 1, 0) to (8, 1, 8) to find the optimal model order. The result shows (4, 1, 3) achieves the lowest Akaike information criterion (AIC) score on the ARIMA training set. Therefore, the order (4, 1, 3) is decided to be the order of the ARIMA model applied on the test set. After forecasting on the test set, the prices are inversely transformed to their actual values to check the model performance. The predicted series is presented in Figure 1.

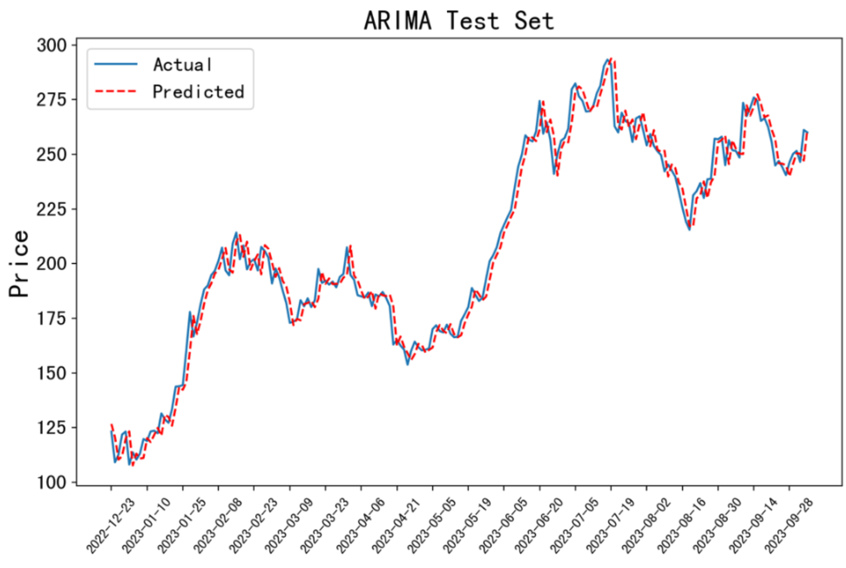

Figure 1. ARIMA test results

4.2. Neural Network

A 2-layer neural network model is adopted, with the first layer of size 20*100,000 and the second layer of size 100,000*1. No activation function is deployed to maintain the linearity between the input and output. The number of epochs is 50 and the batch size is 50. The predicted series is shown in Figure 2.

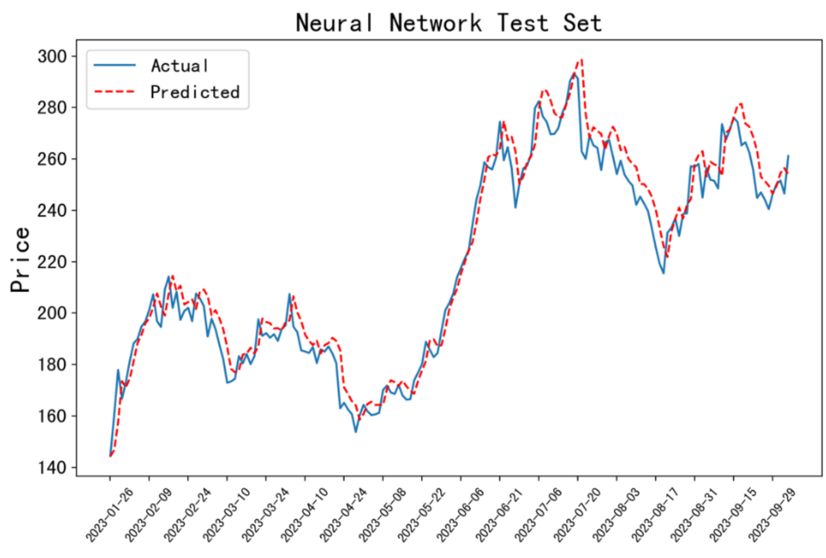

Figure 2. Neural network test results

4.3. LSTM

The LSTM model is concatenated to a linear layer. Its result is the predicted value. The LSTM model has 100 nodes. The prediction is shown in Figure 3.

DOI: 10.54254/2755-2721/55/20241572

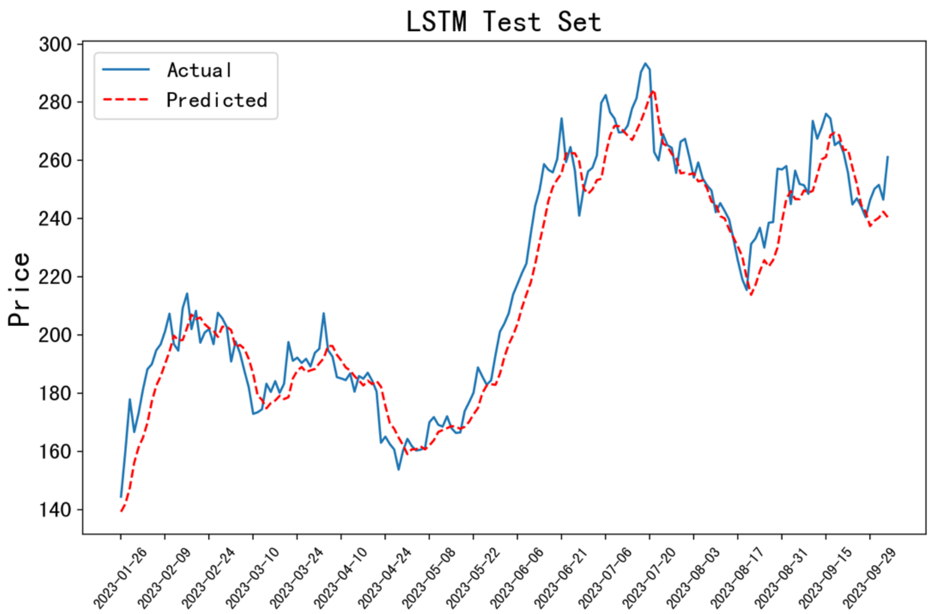

Figure 3. LSTM test results

4.4. Decomposition Linear

Each input is decomposed into a seasonal part and its residual. Each part is connected to a two-layer neural network and the two networks share the same hyperparameters. This study determines that the size of the first linear layer is 20*10,000. The size of the second linear layer is 10,000*1. The epoch number is 60 and the batch size is 50. The prediction is shown in Figure 4.

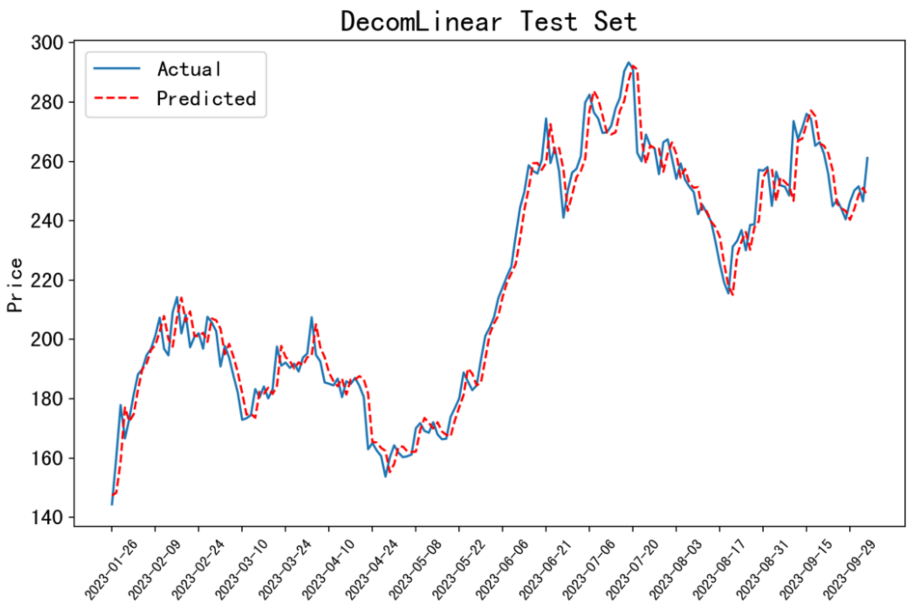

Figure 4. DL test results

4.5. Comparison

In this article, the model performance is assessed using the mean absolute error (MAE). The MAE is calculated as follows.

$$MAE = \frac{1}{n}\Sigma_{i=1}^{n}|\widehat{y_i} - y_i| \tag{16}$$

Where $\hat{y_i}$ denotes the forecasted value and y_i denotes the actual value at time i. After the experimentation of all the models, the comparison outcome is presented in Table 3. The result shows the decomposition linear model performs the best in terms of MAE.

Table 3. Result comparison

	MAE
ARIMA	7.2400
NN	5.8770
LSTM	6.8390
DL	5.5890

5. Conclusion

The stock market forecasting has long been an intriguing topic for researchers and investors. Tesla is one of the corporations with the biggest capitalization, thus representative for the whole stock market. This research signifies a comprehensive evaluation of methodologies employed for stock price forecasting, with a particular focus on the autoregressive integrated moving average model (ARIMA), the decomposition linear model (DL), the neural network model (NN) and the long short-term memory (LSTM). The decomposition linear model attains the lowest MAE, according to the experimentation results. For this reason, the decomposition linear model performs the best when it comes to forecasting the price of Tesla stock.

References

[1] Adebiyi, A. A., Adewumi, A. O., & Ayo, C. K. (2014). Comparison of ARIMA and artificial neural networks models for stock price prediction. Journal of Applied Mathematics.

[2] Xu, K., Yu, J., Hu, Y., & Ai, X. (2019, April). Security monitoring data fusion method based on ARIMA and LS-SVM. In IOP Conference Series: Earth and Environmental Science (Vol. 252, No. 4, p. 042104). IOP Publishing.

[3] Yeze, Z., & Yiying, W. (2019, March). Stock price prediction based on information entropy and artificial neural network. In 2019 5th International Conference on Information Management (ICIM) (pp. 248-251). IEEE.

[4] Gülmez, B. (2023). Stock price prediction with optimized deep LSTM network with artificial rabbits optimization algorithm. Expert Systems with Applications, 227, 120346.

[5] Bennett, A., & Kallus, N. (2023). Proximal reinforcement learning: Efficient off-policy evaluation in partially observed markov decision processes. Operations Research.

[6] Pramod, B. S., & Pm, M. S. (2020). Stock price prediction using LSTM. Test Engineering and Management, 83, 5246-5251.

[7] Li, C., & Qian, G. (2022). Stock Price Prediction Using a Frequency Decomposition Based GRU Transformer Neural Network. Applied Sciences, 13(1), 222.

[8] Altché, F., & de La Fortelle, A. (2017, October). An LSTM network for highway trajectory prediction. In 2017 IEEE 20th international conference on intelligent transportation systems (ITSC) (pp. 353-359). IEEE.

[9] Zhang, K., Zhong, G., Dong, J., Wang, S., & Wang, Y. (2019). Stock market prediction based on generative adversarial network. Procedia computer science, 147, 400-406.

DOI: 10.54254/2755-2721/55/20241572

[10] Zeng, A., Chen, M., Zhang, L., & Xu, Q. (2023, June). Are transformers effective for time series forecasting?. In Proceedings of the AAAI conference on artificial intelligence (Vol. 37, No. 9, pp. 11121-11128).