Proceedings of the 3rd International Conference on Signal Processing and Machine Learning
DOI: 10.54254/2755-2721/6/20230531

Integrative forecasting and analysis of stock price using neural network and ARIMA model

Jin Li[1,2]

[1]Department of International Education, New Oriental School, Guangzhou, 510000, China

[2]limaxjin@gmail.com

Abstract. The volatilities of stock prices have a crucial effect on financial decision-making worldwide. With a reliable and accurate forecast model, investors could gain insights into stock price fluctuations and market trends, thus maximizing the opportunity to make profits. In this work, two models were proposed for stock price forecasting. A neural network based on exploiting the abilities of convolutional neural network and bi-directional long short-term memory is proposed and implemented for forecasting the Nasdaq-100 daily closing price. For long-term stock price forecast, we proposed a hybrid model that combined the autoregressive integrated moving average procedure and a neural network layer for modeling the linear and nonlinear features of the Nasdaq Composite monthly closing price. The proposed models produced promising experiment results, indicating the models' capability of making practical forecasts and analyses based on different data scales and volumes. This work also proposed further considerations of indicators related to the stock price in financial time-series forecasting.

Keywords: deep learning, CNN, BiLSTM, ARIMA-LSTM, stock price forecasting.

1. Introduction

Investors in finance and risk management are interested in forecasting direct investment returns from financial assets, including stocks, bonds, and commodities [1]. Among all, the stock index is one of the essential indicators for investors. Investors rely heavily on the stock index to make optimal decisions to minimize loss and maximize their return [2]. However, precise forecasting of the stock index has been exacting since the financial market is inherent with its non-stationary nature in time series analysis [3]. Besides, factors lying on the macro level, such as political and public events, company performance, consumer behavior, and national well-being, can significantly affect the stock price [4]. Once the listed factors become public knowledge, the stock market will adjust in response, making the prediction seem impossible. However, since the market is not efficient, we can still predict future patterns based on historical market fluctuations.

In the financial market analysis and predictions, studies have concerned short-term stock price forecasting on a scale of days to months and long-term forecasting on a scale of years. Both short-term and long-term predictions are valuable in decision-making. Short-term forecasting can capture the details of changing stock prices, while long-term forecasting reflects the market trend under specific changes and events in a macro-level concern. Therefore, various investors are concerned about the short-

DOI: 10.54254/2755-2721/6/20230531

term and long-term performance of the stock market. Thus, combining short-term and long-term forecasting makes summarizing financial patterns more effective and comprehensive. However, most studies focus on implementing forecasting models without regard to financial forecasting as an integrated subject. The short-term and long-term forecasting researches are usually done separately. This study proposes an integrated short-term and long-term forecasting method to solve the emerging problem. So far as we are concerned, this study is the first to integrate short-term and long-term financial data analysis. The autoregressive integrated moving average (ARIMA) model is proposed for long-term data to fit the stock price pattern and market trend. For short-term patterns, we use a hybrid neural network model of Convolutional Neural Network (CNN) and Long Short-Term Memory (LSTM). The proposed approach is detailed in Figure. 1. Our major contributions to this study are as follows.

(1)A neural network model CNN-BiLSTM is presented to forecast the stock's next-day closing price by analyzing the time series data correlation.

(2) Compared with four deep learning models and demonstrated that the CNN-BiLSTM model presented is the most likely to produce satisfying results in short-term stock forecasting.

(3) By utilizing other long-term features, a hybrid ARIMA-LSTM model is proved to be more accurate in long-term stock forecasting.

(4) The variation trend and specific changes in the stock market are shown by integrating forecast results from both short-term and long-term perspectives.

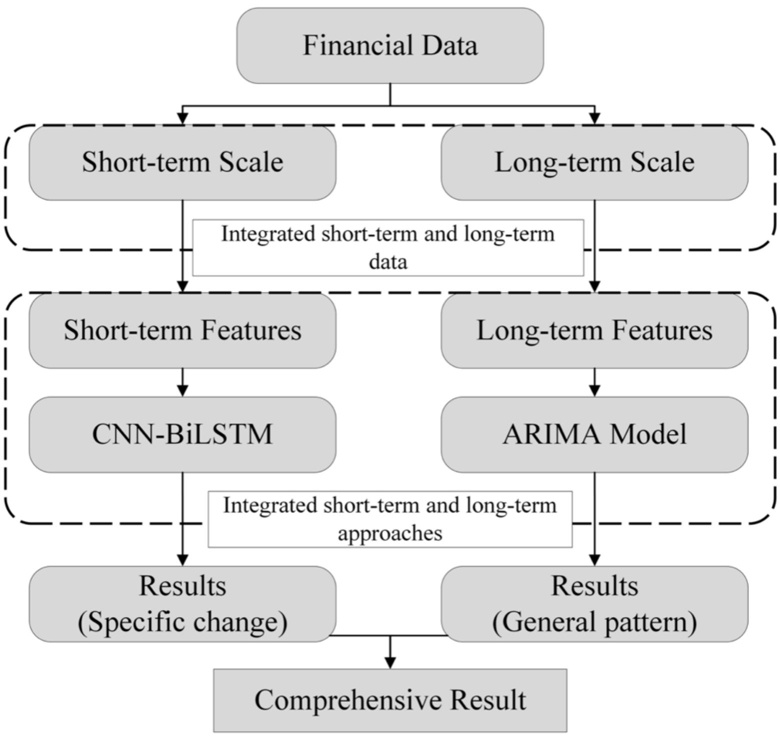

Figure 1. Process summary.

For the remaining sections of this work, we have organized the related works in the following section II. Section III presents models of short-term hybrid neural networks and long-term hybrid ARIMA in detail. In Section IV, the results and performance of our proposed models are presented. Finally, we concluded in Section V.

2. Related work

As accurate stock price forecasting continued to gain research interest, many studies have influenced and contributed to market analysis over the past two decades. Studies have extensively applied statistics, Machine Learning (ML), and Deep Learning (DL) in forecasting in diverse disciplines. Such applications have revealed the great potential of ML and DL applications to the stock market [5].

Primary stock prediction techniques fall into several categories, including statistical methods and machine learning [6]. One of the most eminent statistical approaches for stock market analysis is the Auto-Regressive Integrated Moving Average (ARIMA) model [7]. Ariyo et al used the ARIMA model to predict stock prices from New York Stock Exchange (NYSE) and Nigeria Stock Exchange (NSE) [8]. By comparing the Bayesian Information Criteria (BIC), Adjusted R-square (R2), and Standard Error of Regression, the well-fitted results concluded that the ARIMA model could compete with numerous techniques in short-term prediction. To enhance the forecasting performance of the ARIMA model, Wang et al presented a hybrid approach that utilizes the advantage of the Exponential Smoothing Model (ESM), ARIMA, and Back Propagation Neural Network (BPNN). They used the monthly SZII and DJIAI opening indices to assess the suggested model. The results showed that the hybrid model outperformed the traditional ones. However, the statistical models in general cases assumed that time series data are linear correlated [9].

Moreover, the ARIMA model failed to predict the stock price correctly when an extreme market collapse occurs, as Majumder and Hossain pointed out the failure of the ARIMA model in a collapsed market [10]. They proposed the resolution by adding the APC feature to train the model with 20 collapsed stocks from DSE data. Such a proposed method increased the accuracy with the lowest MAPE, but the performance is not guaranteed other than the collapsed market.

In recent decades machine learning approaches have been expanded to forecast stock market trends [11]. Waqar et al investigated the high dimensionality of the stock exchange to predict the market by applying principal component analysis (PCA) with linear regression [12]. The experiment has shown that the PCA does not necessarily improve the result accuracy unless there is a correlation among input features. The implementation of a model combination with Genetic Algorithms (GA) and Support Vector Regression (SVR) was proposed by Huang for stock selection [13]. In the model, GA performed features selection and SVR generated predicted returns on a stock portfolio. The experiment result showed that the investment return outperformed the benchmark. However, machine learning methods have noticeable limitations. The prediction results largely depended on the input representation and the prediction methods. Using different features as input can primarily affect the forecast performance.

Researchers have developed deep neural network architectures in financial forecasting to overcome the limitations mentioned. The primary branches are the Convolutional Neural Network (CNN) and Recursive Neural Network (RNN). Long Short-Term Memory (LSTM) network is an improved method base on RNN, which successfully addressed the vanishing gradient problem [14]. With the deep learning approach, Selvin et al identified a critical feature of CNN: it relies on current information and is not subject to previous information [15]. Also, they concluded that CNN could capture hidden dynamics of the stock market and give accurate predictions. As financial forecasting with deep learning architectures is becoming increasingly popular, research has continued to build hybrid models with different concerns in time series forecasting. Lu et al proposed a CNN-BiLSTM-AM method that is composed of CNN, Bi-directional Long Short-Term Memory (BiLSTM), and Attention Mechanism (AM) [16]. They used the Shanghai Composite Index data to train the model and predict the next day's closing price. They measure several parameters, such as mean absolute error (MAE), root-mean-square deviation (RMSE), and R-square (R2), to conclude that the proposed model is more suitable for stock prediction.

3. Materials and methodology

3.1. Data representation

Given a sequence$X = \{x_1,\ x_2,\ \cdots, x_T\}$ where $x_T \in R^n$, T defined the time window length. Let $x_t = \{x_t^{f1},\ x_t^{f2},\ \cdots,\ x_t^{fn}\}$ be the set of features at time t.n is the number of features. For example, x_1 denotes all the features at time $t = 1$, and x_t^{f1} represents a specific feature (feature 1) at time $t = 1$. Let the sequence $Y = \{y_1,\ y_2,\ \cdots, y_{T-1}\}$ be a set of target values, in this case, time-series data of stock closing prices. The input to this model consists of pairs of X and Y, which is $\{x_1,\ x_1,\ \cdots, x_T,\ y_1,\ y_2,$

DOI: 10.54254/2755-2721/6/20230531

$\cdots, y_{T-1}\}$. We aim to learn the pattern between X and Y to give a prediction to the target value at time $t = T$., which is y_T per the denotations.

3.2. Forecasting of the daily stock price using neural network

Time-series data of the stock market forecasting can be generalized as a mixture of short-term and long-term patterns. There are numerous factors that affect the market and cause short-term fluctuations. In order to accurately predict the cyclical pattern and capture the short-term features, we propose a neural network approach built on CNN-BiLSTM to forecast the next-day stock closing price. The suggested method incorporates CNN and long short-term memory (LSTM). The CNN layer can extract the local features of the stock time-series data, whereas the BiLSTM (forward LSTM layer and reverse LSTM layer) can find connections from the feedback of stock time-series data. Finally, the dense layer can fully receive outputs from the preceding BiLSTM layer for generating the final output. The model setup is given in Figure 2.

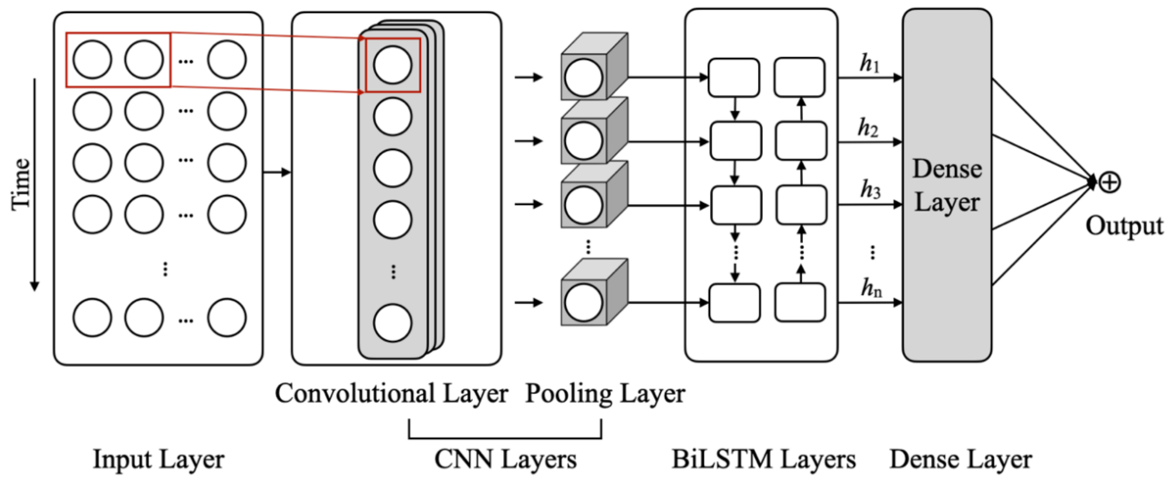

Figure 2. CNN-BiLSTM model setup diagram.

3.2.1. Convolutional neural network. Lecun et al. firstly proposed the CNN model in 1998 [17]. CNN has excellent results with image classification and natural language processing as a feed-forward neural network. The advantages of CNN have also been exploited extensively in time-series data forecasting. With the weight-sharing characteristic, CNN increases learning efficiency by significantly reducing the number of parameters. CNN includes two major parts: convolutional layers and pooling layers. The convolutional layers identify and extract features from the input data, while the pooling layers summarize the features into one or more dense layers to perform further classification or regression tasks. We use the 1-D CNN as it was widely used in the sequential data processing. The calculation of the convolutional layer is given by Equation (1).

$$l_t = \tanh(x_t \odot w_t + b_t) \tag{1}$$

In Formula 1, l_t is the convolutional layer processed output value, $tanh$ is the hyperbolic tangent activation function, x_t is the input, w_t is the convolution filter weight, and b_t is the convolutional filter bias.

3.2.2. Long short-term memory. Long short-term memory (LSTM) was firstly proposed by Schmidhuber et al. in 1997 as a developed recurrent neural network [14]. As shown in Figure 3, the LTSM layer has a unique cell structure comprising a cell state along with three gates: a forget gate, an input gate, and an output gate. LSTM is used to solve the vanishing gradient problem and exploding gradient problem that a traditional RNN model has. The forget gate can choose to discard some information from the past state, and then, the input gate can determine the information to retain.

In Figure 3, C_{t-1} is the past moment cell state, h_{t-1} is the past moment output of the LSTM neural cell unit, x_t is the current input, and σ is the activation function, C_t is the current cell state, and h_t is the current output. The LSTM process formulae are given from Equation (2) to (7).

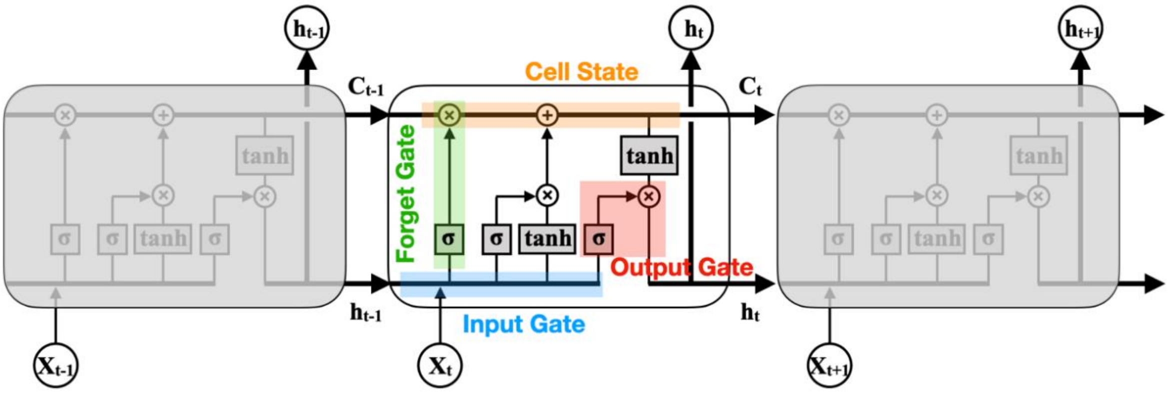

Figure 3. LSTM cell diagram.

$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] + b_f \right) \tag{2}$$

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \tag{3}$$

$$\tilde{C}_t = \tanh(W_c \cdot [h_{t-1}, x_t] + b_c) \tag{4}$$

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t \tag{5}$$

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \tag{6}$$

$$h_t = o_t * \tanh(C_t) \tag{7}$$

Equations (2) to (7) show the calculation of the LSTM process. W_f, W_i, and W_c represent weight of the forget gate, input gate, and the candidate input gate, respectively. b_f, b_i, and b_c represent the bias of the forget gate, input gate, and the candidate input gate, respectively. In (2), the output value from the last moment and the current input entered the forget game. The forget gate output f_t is given as a value from 0 to 1. In (3) and (4), the last moment output and the current input go through the input gate. The input gate result i_t is given as a value from 0 to 1 and the candidate cell state $\tilde{C}_t$ is obtained. Then in (5), the current cell state is updated and given as C_t. In (6), the output gate receives the last moment output and the current input. The output gate result o_t is given as a value from 0 to 1. The final output of LSTM h_t is obtained by merging the output o_t and the cell state C_t, as shown in (7).

*3.2.3. CNN-BiLSTM process of prediction. T*he main steps of CNN-BiLSTM model for stock forecasting are shown below:

I. Input: The historical data of the stock are input.

II. Standardization: The input data are standardized using Min-max feature scaling with Equation (8). Where x_{std} is the standardized input, x_t is the input at t, x_{max} is the maximum input, and x_{min} is the minimum input.

$$x_{std} = \frac{x_t - x_{min}}{x_{max} - x_{min}} \tag{8}$$

DOI: 10.54254/2755-2721/6/20230531

III. Prediction: The standardized input date is input into the CNN-BiLSTM model to generate the prediction output.

IV. Restoration: The prediction output is transformed back to the original value using Equation (9). Where y_t is the output at t, y_{std} is the standardized output, y_{max} is the maximum output, and y_{min} is the minimum output.

$$y_t = y_{std} \cdot (y_{max} - y_{min}) + y_{min} \tag{9}$$

V. Output: The final output from the restoration is used for later prediction.

3.3. Forecasting of the monthly stock price using hybrid ARIMA

As the volume of the financial data become extensive, we need a colossal amount of data to perform training, which does not fit the most satisfactory conditions for the neural network. Generally, the long-term data reveal the financial market trend, and it is considered to be more linear. Therefore, we propose using a hybrid ARIMA model to forecast the monthly stock price with the same dataset listed.

Box and Jenkins introduced the ARIMA model in 1970 [18,19]. In the ARIMA model, the future value of a variable, in this case, stock price, is a linear combination of past values and past errors. The Equation is shown in (10).

$$Y_t = \phi_0 + \phi_1 Y_{t-1} + \phi_2 Y_{t-2} + \cdots + \phi_p Y_{t-p} + \varepsilon_t - \theta_1 \varepsilon_{t-1} - \theta_2 \varepsilon_{t-2} - \cdots - \theta_q \varepsilon_{t-q} \tag{10}$$

Where Y_t is the actual value, ε_t is the error at t, ϕ_i and θ_j are coefficients of A.R. and M.A. respectively, p and q are integers represent autoregressive and moving average, respectively. However, the ARIMA model is not designed with multiple nonlinear features and other variables. In order to integrate the features with the model, we introduce a layer of LSTM. The ARIMA and LSTM models are proposed to address the linearity and non-linearity of the data, respectively, to improve the forecast performance and accuracy. The detailed process is shown in Figure 4, where y_t is the stock price data in month t. Firstly, the raw data of monthly stock closing price fit into the ARIMA model. Then, the residual of the ARIMA model is calculated and standardized. We use the Min-max feature scaling, as represented with Equation (11), for the data standardization. Where $(r_t)', r_{max}, r_{min}$ represent the standardized residual at time t, maximum residual, and minimum residual, respectively.

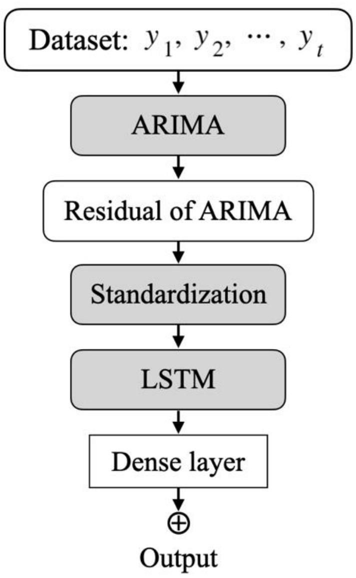

Figure 4. ARIMA model structure diagram.

$$(r_t)' = \frac{r_t - r_{min}}{r_{max} - r_{min}} \tag{11}$$

Then, the standardized residual of the ARIMA will fit into the LSTM model to train the nonlinear tendency. Finally, a dense layer is used to integrate the output. The final forecasting result is produced subsequently.

4. Experiment

4.1. Data summarization Table 1 Details of the data used

We used data from Yahoo Finance and FRED Economic Research. Specifically, we used the daily Nasdaq-100 Index data and monthly Nasdaq Composite Index from Yahoo Finance. The corresponding monthly data of Effective Federal Funds Rate (EFFR) was obtained from FRED Economic Research as exogenous variables to be evaluated for long-term stock forecasting purposes. The details of the data are as follows in Table 1.

Table 1. Details of the data used.

Source	Data length	Data type	Time unit	URL
Yahoo Finance	2182	Nasdaq-100 Index	Daily	https://finance.yahoo.com/quote/%5ENDX/history?p=%5ENDX
Yahoo Finance	151	Nasdaq Composite Index	Monthly	https://finance.yahoo.com/quote/%5EIXIC?p=%5EIXIC
FRED Economic Research	151	EFFR	Monthly	https://fred.stlouisfed.org/series/FEDFUNDS

The execution code was written in Python 3.7 on a PC (AMD Ryzen 5 5600X 6-Core Processor 3.70 GHz, 16 GB RAM) running Windows 10 operating system. The machine learning models were executed using the TensorFlow-based Keras library.

4.2. Evaluation Criteria

To assess the effectiveness of the forecasting of all models, we introduced the mean absolute error (MAE), root mean square error (RMSE), and R-square (R^2) as the error measure criteria. The equations are given from (12)-(14).

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \tag{12}$$

In (12), y_i denotes the actual observation, and $\hat{y}_i$ denotes the predicted value. A lesser MAE value indicates a more precise forecast result.

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \tag{13}$$

In (13), y_i is denotes actual observation, and $\hat{y}_i$ denotes the predicted value. A lesser RMSE value indicates a more precise forecast result.

$$R^2 = 1 - \frac{SS_{RES}}{SS_{TOT}} = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y}_i)^2} \tag{14}$$

In (14), SS_{RES} denotes the sum of squares of residuals, SS_{TOT} denotes the total sum of squares, y_i denotes the actual observation, and $\hat{y}_i$ denotes the predicted value. A lesser RMSE value indicates a more precise forecast result. R^2 can take the value ranging from 0 to 1. The closer the R^2 value to 1, the more precise is forecast result.

DOI: 10.54254/2755-2721/6/20230531

4.3. Experiment Detail

4.3.1. Neural networks model implementation. For this part of the experiment, the historical dataset ranging from January 02, 2014, to August 30, 2022, is used for short-term forecasting purposes. The dataset contains a series of data, including opening price, highest price, lowest price, closing price, volume, and percentage change. The dataset is split into two periods, 85% to 15%. The first period of the daily Nasdaq-100 ranges from January 02, 2014, to May 13, 2021, and the second period ranges from May 14, 2021, to August 30, 2022. The data obtained from the first period is used for training the model, while those obtained from the second period are used for testing the model performance and making predictions. Since the trading volumes are large in volume, we take the natural logarithm and denote it as Log Volume. The preview of the daily Nasdaq-100 Index data is shown in Table 2.

Table 2. Sample daily Nasdaq-100 Index data.

Date	Opening price	Highest price	Lowest price	Closing price	Log Volume	Percentage change
2014-01-02	3575.6001	3577.0300	3553.6499	3563.5701	21.5394	-0.003365
2014-01-03	3564.9399	3567.5100	3537.6101	3538.7300	21.6816	-0.007352
2014-01-06	3539.0200	3542.5200	3512.4500	3526.9600	21.6709	-0.003408
2014-01-07	3539.2900	3562.9900	3535.5000	3557.8501	21.6426	0.005244
2014-01-08	3558.3000	3575.1499	3551.1201	3567.5400	21.5896	0.002597

The settings of the parameters of the proposed CNN-BiLSTM model in the experiment are shown in Table 3. The LSTM, BiLSTM, and CNN–LSTM were also trained to compare the effectiveness of the proposed CNN-BiLSTM model. Details are shown in Table 4. All model epochs and batch size were set to 20 and 70, respectively, with adaptive moment estimation (ADAM) as the optimizer. The learning steps in the training process can be made as scale-invariant to parameter gradients with the ADAM algorithm.

Table 3. Parameter specification of the proposed CNN-BiLSTM model.

Parameters	Value
Convolutional layer filters	64
Convolutional layer kernel size	1
Convolutional layer padding	Same
Convolutional layer activation function	ReLU
Max pooling layer pool size	1
BiLSTM layer units	40

Table 4. Parameter specification of all neural network models.

Models	Details
$LSTM_1$	LSTM layer with 100 units
$LSTM_2$	LSTM layer with 200 units
$BiLSTM_1$	BiLSTM layer with 100 units
$BiLSTM_2$	BiLSTM layer with 200 units
CNN-LSTM	Convolutional layer with 64 filters Max pooling layer with size 1 LSTM layer with 100 units
CNN-BiLSTM	Convolutional layer with 64 filters Max pooling layer with size 1 BiLSTM layer with 40 units

4.3.2. Hybrid ARIMA model implementation. For the experiment of long-term stock forecasting, we used the historical datasets of the monthly Nasdaq Composite from January 01, 2010, to June 01, 2022. The dataset contains a series of the stock closing price. The dataset is split into two periods, 70% to 30%. The first period of the daily Nasdaq Composite Index and other indicators ranges from January 01, 2010, to August 01, 2020, and the second period ranges from September 01, 2020, to June 01, 2022. The data obtained from the first period is used for training the model, while those obtained from the second period are used for testing the model performance and making predictions. The preview of the daily Nasdaq Composite Index data is shown in Table 5.

Table 5. Sample monthly Nasdaq Composite data.

Date	Closing price
2010/01/01	2147.3501
2010/02/01	6292.6802
2010/03/01	6445.2002
2010/04/01	6521.4399
2010/05/01	6574.7300

The settings of the parameters of the proposed ARIMA-LSTM model in the experiment are shown in Table 6. The parameters are generated using the Python package Pmdarima with the lowest Akaike information criterion (AIC). The single ARIMA and single LSTM models were also trained to compare the effectiveness of the proposed ARIMA-LSTM model.

Table 6. Parameters specification of the proposed ARIMA-LSTM model.

Parameters	Value
Order of the autoregressive: p	0
Degree of differencing: d	4
Order of the moving average:	5
LSTM layer units	128

DOI: 10.54254/2755-2721/6/20230531

4.4. Experiment Results

4.4.1. Neural networks model outcomes. The training dataset is fed to the algorithms to train the models. The predicted values generated by the models are used to compare with the actual observations in the test set of the data. The comparison of the different models is shown in Figure 5. The orange curve in each model is the forecast result while the blue curve represents the actual stock price. The x-axis denotes the time, and the y-axis denotes the stock price in United States Dollars. The degree of fitting of the broken-line graphs between actual observation values and predicted values are ranked from low to high. More specifically, the LSTM1 model has achieved the lowest degree of fitting of the broken-line graphs between actual observation values and predicted values, then BiLSTM2, LSTM2, BiLSTM1, CNN-LSTM, and CNN-BiLSTM. The CNN-BiLSTM model showed the predicted values almost perfectly fitting with the actual observations.

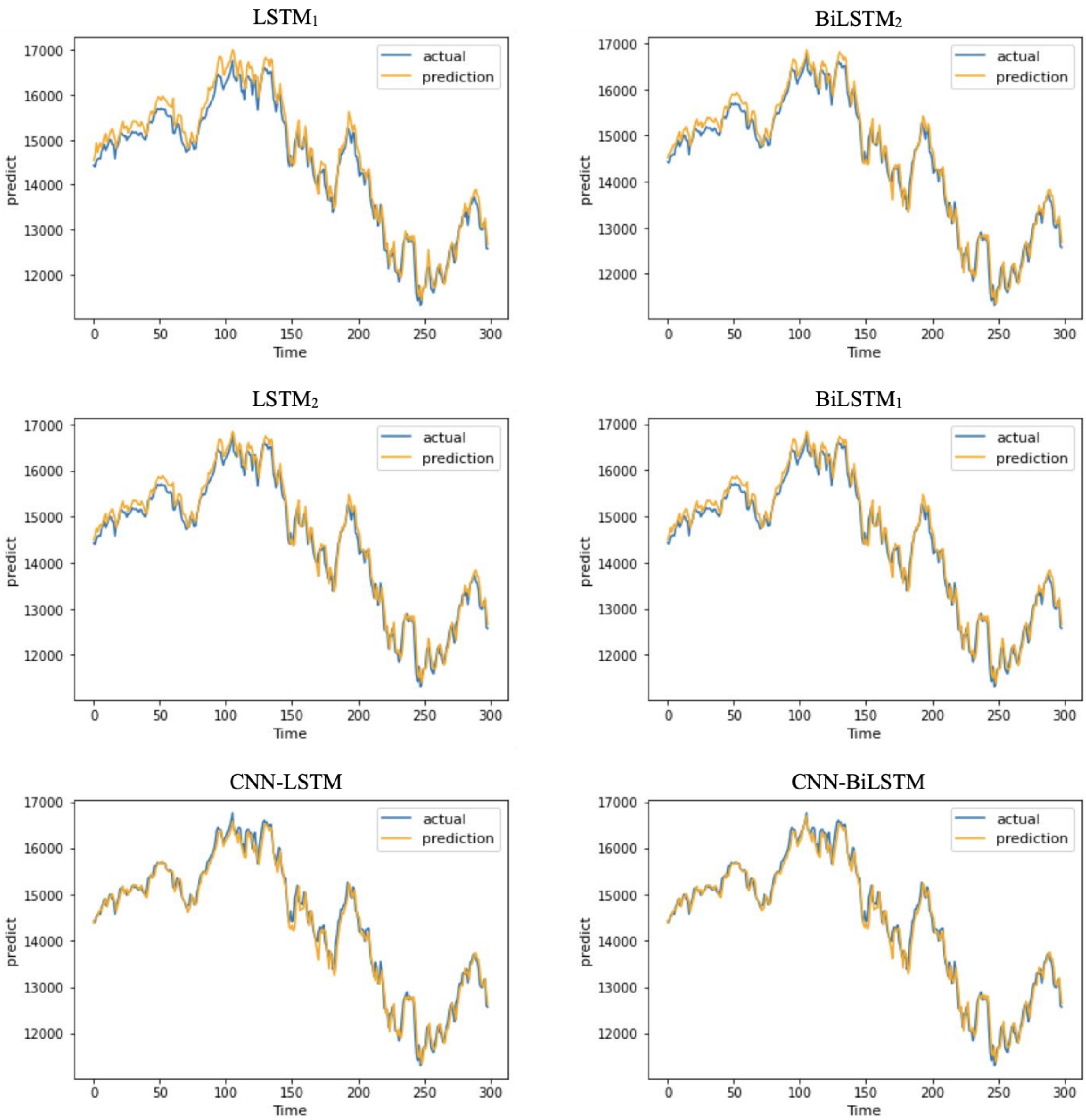

Figure 5. Comparison of the outputs of neural network models.

To quantify the degree of fitness between actual observation and predicted values, the evaluation criteria of each of the seven models are calculated. The details are shown in Table 7.

From the results in Table 3, the LSTM1 model had the largest MAE and RMSE, while the R2 was the smallest. The CNN-BiLSTM model had the minimum MAE and RMSE, while the R2 was the largest. Comparing the LSTM1 model, the increase of hidden units in LSTM2 improved the forecast performance with a reduction in MAE from 211.158 to 162, RMSE from 253.405 to 199.605, and an increase in R2 from 0.96749 to 0.97983. Comparing the LSTM1 model, the BiLSTM1 model reduced the MAE significantly from 211.158 to 159.250, RMSE from 253.405 to 196.269, and R2 increased from 0.96769 to 0.98049. The BiLSTM can have an improvement in forecast accuracy compared with LSTM to some degree, but the improvement was not necessarily guaranteed with an increase of hidden units in BiLSTM. In comparing the BiLSTM1 model, the BiLSTM2 model increased the MAE from 159.250 to 166.655, RMSE from 196.269 to 204.117, and R2 reduced from 0.98049 to 0.97890. The introduced CNN layer can increase the performance of both LSTM and BiLSTM while other parameter stays the same. Among all the models being compared, the CNN-BiLSTM has the best forecast performance with MAE 129.252, RMSE 173.724, and R2 0.98472. Therefore, the CNN-BiLSTM is the best suitable model for short-term stock price forecasting.

Table 7. Comparison of the evaluation criteria of neural network models.

Models	MAE	RMSE	R^2
$LSTM_1$	211.158	253.405	0.96749
$BiLSTM_2$	166.655	204.117	0.97890
$LSTM_2$	162.994	199.605	0.97983
$BiLSTM_1$	159.250	196.269	0.98049
CNN-LSTM	130.299	174.116	0.98465
CNN-BiLSTM	129,252	173.724	0.98472

4.4.2. Hybrid ARIMA model results. The comparison of the different models is shown in Figure 6. The orange curve in each model is the forecast result, while the blue curve represents the actual stock price. The x-axis denotes the time, and the y-axis denotes the stock price in United States Dollars. From the degree of fitting of the broken-line graphs between actual observation values and predicted values, the optimal forecast was achieved by using the ARIMA-LSTM model. The comparison of the ARIMA model and the LSTM model for the long-term forecast with the RMSE criteria is shown in Table 8. From the experiment result, although the ARIMA model produced a lower RMSE, the degree of fitting in the graph displayed that the ARIMA model had failed in capturing nonlinear variations of the time series. Moreover, the LSTM model was used only for monthly stock closing price forecasting, and the neural network model also failed to perform an accurate forecast with a small amount of data, despite the discoverable trend displayed in the graph. The ARIMA-LSTM model surpasses the competing two methods by achieving the best fitting with actual and predicted stock price and the lowest RMSE. Despite the fact that individual stock price predictions are way far from the actual values, the forecast result can still fit the general trend of the actual data.

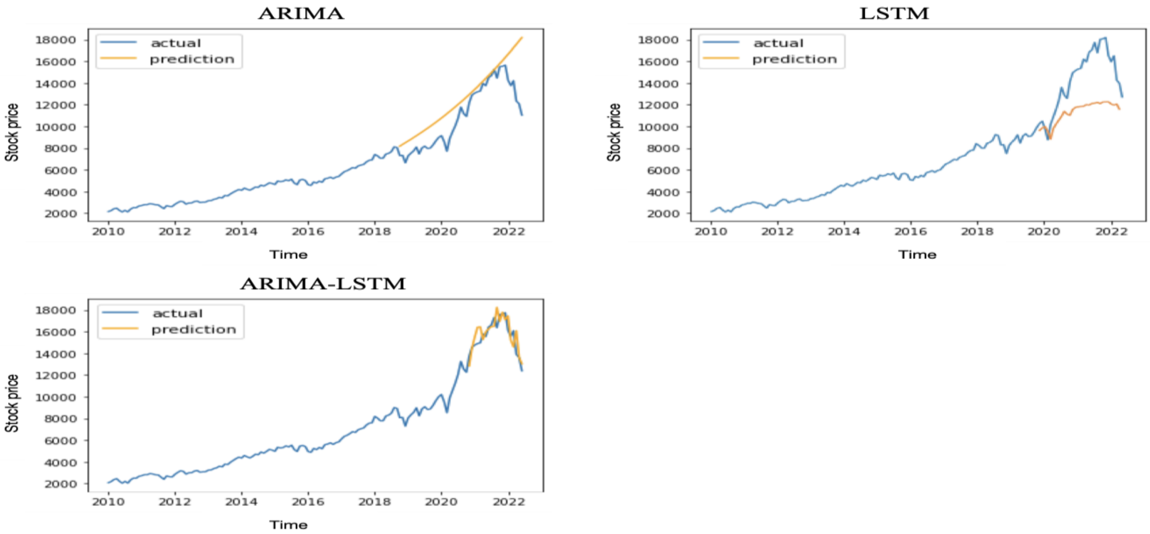

Figure 6. Comparison of the outputs of ARIMA models.

Table 8. Comparison of the evaluation criteria of ARIMA models.

Models	RMSE
LSTM	2688.831
ARIMA	1487.514
ARIMA-LSTM	878.532

5. Conclusion

In this study, two major models were used in Nasdaq index stock price forecasting. We proposed a neural network model to predict the daily price of Nasdaq-100 on a short-term scale, represented the daily stock price of the most representative Nasdaq-listed stocks, and an improved ARIMA model was proposed to study the stock price forecasting from the Nasdaq Composite in long-term scale, represented the monthly performance of all stocks listed on the Nasdaq stock exchange. The attainability of the integrated models on stock price forecasting is shown by the result, which showed a comparatively satisfying accuracy. Moreover, the integrated forecasting model can provide results that reflect the pattern from different perspectives, which helps investors better understand the market trend and guide their decisions. The experimental analysis indicated that although the neural network models are broadly used in financial forecasting, the utilization of feature processing with neural networks in statistical models also showed noticeable potential. It is worth noting that due to the market's volatility, more economic indicators could be added as the training features for the model to improve the forecasting ability further.

Our future work will improve the proposed model's forecasting accuracy by adding different features with long-term and short-term data, such as major stock market indices, gold prices, and other economic indicators. Furthermore, we plan to closely inspect the short-term fluctuations and long-term trends of financial data to explore the significance of the correlation between them. Finally, more optimized configurations of the proposed CNN-BiLSTM will be another primary concern, as parameters should be continually adjusted to achieve a forecast performance.

DOI: 10.54254/2755-2721/6/20230531

References

[1] Montgomery, D. C., Jennings, C. L., & Kulahci, M. (2015). Introduction to time series analysis and forecasting. John Wiley & Sons.

[2] Iyiola, O., Munirat, Y., & Nwufo, C. (2012). The modern portfolio theory as an investment decision tool. Journal of Accounting and Taxation, 4(2), 19-28.

[3] Manuca, R., & Savit, R. (1996). Stationarity and nonstationarity in time series analysis. Physica D: Nonlinear Phenomena, 99(2-3), 134-161.

[4] Zhong, X., & Enke, D. (2017). Forecasting daily stock market return using dimensionality reduction. Expert Systems with Applications, 67, 126-139.

[5] Sezer, O. B., Gudelek, M. U., & Ozbayoglu, A. M. (2020). Financial time series forecasting with deep learning: A systematic literature review: 2005-2019. Applied soft computing, 90, 106181.

[6] Shah, D., Isah, H., & Zulkernine, F. (2019). Stock market analysis: A review and taxonomy of prediction techniques. International Journal of Financial Studies, 7(2), 26.

[7] Hiransha, M., Gopalakrishnan, E. A., Menon, V. K., & Soman, K. P. (2018). NSE stock market prediction using deep-learning models. Procedia Computer Science, 132, 1351-1362.

[8] Ariyo, A. A., Adewumi, A. O., & Ayo, C. K. (2014). Stock price prediction using the ARIMA model. In 2014 UKSim-AMSS 16th international conference on computer modelling and simulation, March 26-28, Cambridge, United Kingdom, pp. 106-112.

[9] Wang, J. J., Wang, J. Z., Zhang, Z. G., & Guo, S. P. (2012). Stock index forecasting based on a hybrid model. Omega, 40(6), 758-766.

[10] Majumder, M. M. R., & Hossain, M. I. (2019). Limitation of ARIMA in extremely collapsed market: A proposed method. In 2019 International Conference on Electrical, Computer and Communication Engineering, February 07-09, Cox's Bazar, Bangladesh, pp. 1-5.

[11] Nabipour, M., Nayyeri, P., Jabani, H., Shahab, S., & Mosavi, A. (2020). Predicting stock market trends using machine learning and deep learning algorithms via continuous and binary data; a comparative analysis. IEEE Access, 8, 150199-150212.

[12] Waqar, M., Dawood, H., Guo, P., Shahnawaz, M. B., & Ghazanfar, M. A. (2017). Prediction of stock market by principal component analysis. In 2017 13th International Conference on Computational Intelligence and Security, December 15-18, Hong Kong, China, pp. 599-602.

[13] Huang, C. F. (2012). A hybrid stock selection model using genetic algorithms and support vector regression. Applied Soft Computing, 12(2), 807-818.

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

[15] Selvin, S., Vinayakumar, R., Gopalakrishnan, E. A., Menon, V. K., & Soman, K. P. (2017). Stock price prediction using LSTM, RNN and CNN-sliding window model. In 2017 international conference on advances in computing, communications and informatics, September 13-16, Manipal, India, pp. 1643-1647.

[16] Lu, W. J., Li, J. Z., Wang, J. Y., & Qin, L. L. (2021). A CNN-BiLSTM-AM method for stock price prediction. Neural Computing and Applications, 33(10), 4741-4753.

[17] LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. Proceedings of the IEEE, 86(11), 2278-2324.

[18] Box, G. E., Jenkins, G. M., Reinsel, G. C., & Ljung, G. M. (2015). Time series analysis: forecasting and control. John Wiley & Sons.

[19] Alam, M. D., & Uddin, G. (2009). Relationship between interest rate and stock price: empirical evidence from developed and developing countries. International Journal of Business and Management, 4(3), 43-51.