WILEY | Hindawi

Research Article
A CNN-LSTM-Based Model to Forecast Stock Prices

Wenjie Lu,[1,2] **Jiazheng Li,**[3] **Yifan Li,**[3] **Aijun Sun** (iD),[1] **and Jingyang Wang**[3]

[1]*Business School, Jiangsu Second Normal University, Nanjing 210000, China*
[2]*School of Economics and Management, Hebei University of Science and Technology, Shijiazhuang 050018, China*
[3]*School of Information Science and Engineering, Hebei University of Science and Technology, Shijiazhuang 050018, China*

Correspondence should be addressed to Aijun Sun; saj1970@jssnu.edu.cn

Stock price data have the characteristics of time series. At the same time, based on machine learning long short-term memory (LSTM) which has the advantages of analyzing relationships among time series data through its memory function, we propose a forecasting method of stock price based on CNN-LSTM. In the meanwhile, we use MLP, CNN, RNN, LSTM, CNN-RNN, and other forecasting models to predict the stock price one by one. Moreover, the forecasting results of these models are analyzed and compared. The data utilized in this research concern the daily stock prices from July 1, 1991, to August 31, 2020, including 7127 trading days. In terms of historical data, we choose eight features, including opening price, highest price, lowest price, closing price, volume, turnover, ups and downs, and change. Firstly, we adopt CNN to efficiently extract features from the data, which are the items of the previous 10 days. And then, we adopt LSTM to predict the stock price with the extracted feature data. According to the experimental results, the CNN-LSTM can provide a reliable stock price forecasting with the highest prediction accuracy. This forecasting method not only provides a new research idea for stock price forecasting but also provides practical experience for scholars to study financial time series data.

1. Introduction

The change trend of the stock price has always been identified as a very important problem in the economic field [1]. Stock prices are affected by various internal and external factors, such as domestic and foreign economic environment, international situation, industry prospect, financial data of listed companies, and stock market operation. Thus, the forecasting method also has different emphasis [2, 3].

The traditional analysis method is based on economics and finance, which mainly uses the fundamental analysis method and the technical analysis method. On the one hand, the fundamental analysis method pays more attention to the intrinsic value of stocks and qualitatively analyzes the external factors that affect the stock, such as interest rate, exchange rate, inflation, industrial policy, finance of listed companies, international relations, and other economic and political factors. On the other hand, the technical analysis method mainly focuses on the direction of stock price, trading volume, and investors' psychological expectation, which primarily focuses on analyzing the stock index trajectory of individual stocks or the whole market by using K-line chart and other tools. At present, traditional fundamental analysis and technical analysis are still the most commonly employed methods for many organizations and individual investors [4, 5].

The accuracy of the traditional fundamental analysis method is difficult to be convincing. The reason is not only that the influencing factors are in a long-term cycle, but also the forecasting results are more dependent on the professional quality of analysts. As a financial time series, stock data have the characteristics of random walk [6]. Based on statistics and probability theory, some scholars use time series linear forecasting model to predict the short-term stock price with a large number of long-term data, such as vector autoregression (VAR) [7], Bayesian vector autoregression (BVAR) model [8], autoregressive integrated moving average mode (ARIMA) [9], and generalized autoregressive conditional heteroskedasticity model (GARCH) [10]. However, the accuracy of using time series

model alone is questioned due to the uncertainty and high noise characteristics of financial time series and the relationship between independent variables and dependent variables is prone to dynamic changes over time, which limits its further application and expansion [11].

It has certain limitations to predict stock price trend with single simply using the linear time series forecasting model or neural network model. At present, combining the advantages of various methods and using various best algorithms to improve the hybrid method is the development trend of financial time series deep learning [12]. Therefore, in order to make the best of the time series characteristics of data series, deeply mine the data features, and improve the accuracy of stock price forecasting, this paper proposes a stock price forecasting method based on CNN-LSTM for the stock closing price of the next day forecasting. Combining the advantages of convolutional neural networks (CNN) that can extract effective features from the data, and long short-term memory (LSTM) which can not only find the interdependence of data in time series data, but also automatically detect the best mode suitable for relevant data, this method can effectively improve the accuracy of stock price forecasting. The CNN-LSTM model uses CNN to extract the features of the input time data and uses LSTM to predict the stock closing price on the next day. In order to verify the effectiveness of the model, this paper uses the daily transaction data of 7127 trading days from July 1, 1991, to August 31, 2020, in which the first 6627 trading days data are the training set and the last 500 trading days data are the test set.

2. Related Work

At present, the financial market is a noisy, nonparametric dynamic system, and there are two main kinds of forecasting methods for stock price: traditional analysis method and machine learning method [13]. The traditional econometric methods or equations with parameters are not suitable for analyzing complex, high-dimensional, and noisy financial series data. In recent years, neural network has become a hot research direction in the field of stock forecasting because it can extract data features from a large number of high-frequency raw data without relying on prior knowledge. In 1988, White used neural network to predict IBM stock, but the experimental results were not good [14]. In 2003, Zhang used neural network and autoregressive integrated moving average model (ARIMA) to forecast stocks, respectively. The experimental results show that neural network has obvious advantages in nonlinear data forecasting, but the accuracy still needs to be improved [15]. In 2005, Sun et al. proposed a time series forecasting method based on neural network. This method combines the optimal partition algorithm (OPA) and radial basis function (RBF) neural network [16]. In 2014, Adhikari et al. proposed a method combining random walk (RW) and artificial neural network (ANN) to predict four financial time series data, and the results showed that the forecasting accuracy had a certain improvement [17]. In 2018, Zhang et al. proposed the network structure of stock price forecasting based on LM-BP neural network, which improved the traditional BP neural network training

algorithm's shortcomings of slow training speed and low precision [18]. In 2018, the experimental results of Hu et al. show that convolutional neural network can predict time series, and deep learning is more suitable for solving the problem of time series. However, because CNN is more commonly used to solve image recognition and feature extraction, the forecasting accuracy of CNN alone is relatively low [19]. In 2020, Kamalov used MLP, CNN, and LSTM to forecast the stock price of four major US public companies. Experimental results showed that these three methods showed better results compared to similar studies that forecast the direction of price change [20]. In 2020, Xue et al. established a high-precision short-term forecasting model of financial market time series based on LSTM deep neural network and compared with the BP neural network, the traditional RNN, and the improved LSTM deep neural network. The results showed that the LSTM deep neural network has high forecasting accuracy and can effectively predict the time series of the stock market [21].

The main contributions of this paper are as follows:

(1) By analyzing the correlation and time series of stock price data, a new deep learning method (CNN-LSTM) is proposed to predict the stock price. In this method, CNN is used to extract the time feature of data, and LSTM is used for data forecasting. It can make full use of the time sequence of stock price data to obtain more reliable forecasting.

(2) By comparing the evaluation indexes of CNN-LSTM with multilayer perceptron (MLP), CNN, RNN, LSTM, and CNN-RNN, it is proved that CNN-LSTM has high forecasting accuracy and is more suitable for stock price forecasting.

3. CNN-LSTM

3.1. CNN-LSTM Model. CNN has the characteristic of paying attention to the most obvious features in the line of sight, so it is widely used in feature engineering. LSTM has the characteristic of expanding according to the sequence of time, and it is widely used in time series. According to the characteristics of CNN and LSTM, a stock forecasting model based on CNN-LSTM is established. The model structure diagram is shown in Figure 1, and the main structure is CNN and LSTM, including input layer, one-dimensional convolution layer, pooling layer, LSTM hidden layer, and full connection layer.

3.2. CNN. CNN is a network model proposed by Lecun et al. in 1998 [22]. CNN is a kind of feedforward neural network, which has good performance in image processing and natural language processing [23]. It can be effectively applied to the forecasting of time series. The local perception and weight sharing of CNN can greatly reduce the number of parameters, thus improving the efficiency of model learning [24]. CNN is mainly composed of two parts: convolution layer and pooling layer. Each convolution layer contains a plurality of convolution kernels, and its calculation formula is shown in formula (1). After the convolution operation of
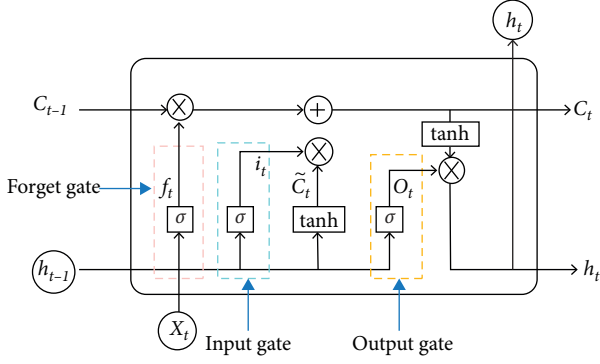
FIGURE 1: CNN-LSTM structure diagram.

the convolution layer, the features of the data are extracted, but the extracted feature dimensions are very high, so in order to solve this problem and reduce the cost of training the network, a pooling layer is added after the convolution layer to reduce the feature dimension:

$$l_t = \tanh(x_t * k_t + b_t), \tag{1}$$

where l_t represents the output value after convolution, tanh is the activation function, x_t is the input vector, k_t is the weight of the convolution kernel, and b_t is the bias of the convolution kernel.

3.3. LSTM.

LSTM is a network model proposed by Schmidhuber et al. in 1997 [25]. LSTM is a network model designed to solve the longstanding problems of gradient explosion and gradient disappearance in RNN [26, 27]. It has been widely used in speech recognition, emotional analysis, and text analysis, as it has its own memory and can make relatively accurate forecasting [28, 29]. In recent years, it has also been adopted in the field of stock market forecasting [30–32]. There is only one repeating module in a standard RNN, and its internal structure is simple. It is usually a tanh layer. However, four of the LSTM modules are similar to the standard RNN modules, and they operate in a special interactive manner [33, 34]. The LSTM memory cell consists of three parts: the forget gate, the input gate, and the output gate, as shown in Figure 2.

The LSTM calculation process is as follows:

(1) The output value of the last moment and the input value of the current time are input into the forget gate, and the output value of the forget gate is obtained after calculation, as shown in the following formula:

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

where the value range of f_t is (0,1), W_f is the weight of the forget gate, and b_f is the bias of the forget gate, x_t is the input value of the current time, and h_{t-1} is the output value of the last moment.

(2) The output value of the last time and the input value of the current time are inputted into the input gate, and the output value and candidate cell state of the
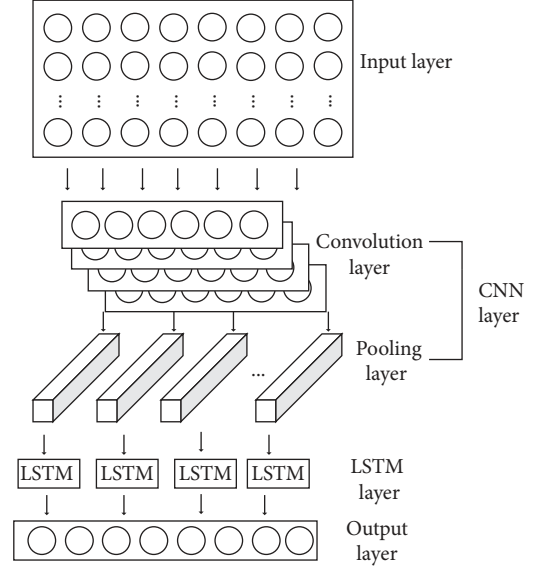

FIGURE 2: Architecture of LSTM memory cell.

input gate are obtained after calculation, as shown in the following formulas:

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

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

where the value range of i_t is (0,1), W_i is the weight of the input gate, b_i is the bias of the input gate, W_c is the weight of the candidate input gate, and b_c is the bias of the candidate input gate.

(3) Update the current cell state as follows:

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

where the value range of C_t is (0,1).

(4) The output h_{t-1} and input x_t are received as input values of the output gate at time t, and the output o_t of the output gate is obtained as follows:

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

where the value range of o_t is (0,1), W_o is the weight of the output gate, and b_o is the bias of the output gate.

(5) The output value of LSTM is obtained by calculating the output of the output gate and the state of the cell, as shown in the following formula

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

3.4. CNN-LSTM Training and Prediction Process.

The CNN-LSTM process of training and prediction is shown in Figure 3.
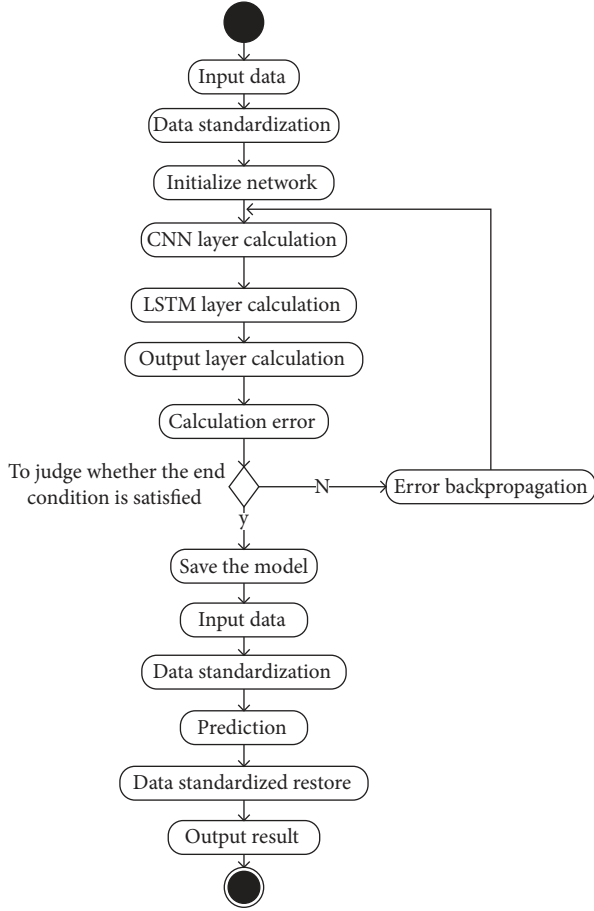
The main steps are as follows:

FIGURE 3: Activity diagram of CNN-LSTM training and prediction process.

(1) Input data: input the data required for CNN-LSTM training.

(2) Data standardization: as there is a large gap in the input data, in order to train the model better, the z-score standardization method is adopted to standardize the input data, as shown in the following formula:

$$y_i = \frac{x_i - \overline{x}}{s}, \tag{8}$$

$$x_i = y_i * s + \overline{x}, \tag{9}$$

where y_i is the standardized value, x_i is the input data, $\overline{x}$ is the average of the input data, and s is the standard deviation of the input data.

(3) Initialize network: initialize the weights and biases of each layer of the CNN-LSTM.

(4) CNN layer calculation: the input data are successively passed through the convolution layer and pooling layer in the CNN layer, the feature extraction of the input data is carried out, and the output value is obtained.

(5) LSTM layer calculation: the output data of the CNN layer are calculated through the LSTM layer, and the output value is obtained.

(6) Output layer calculation: the output value of the LSTM layer is input into the full connection layer to get the output value.

(7) Calculation error: the output value calculated by the output layer is compared with the real value of this group of data, and the corresponding error is obtained.

(8) To judge whether the end condition is satisfied: the conditions for the end are to complete a predetermined number of cycles, the weight is lower than a certain threshold, and the error rate of the forecasting is lower than a certain threshold. If one of the conditions for the end is met, the training will be completed, update the entire CNN-LSTM network, and go to step 10; otherwise, go to step 9.

(9) Error backpropagation: propagate the calculated error in the opposite direction, update the weight and bias of each layer, and go to step 4 to continue to train the network.

(10) Save the model: save the trained model for forecasting.

(11) Input data: input the input data required for the forecasting.

(12) Data standardization: the input data are standardized according to formula (8).

(13) Forecasting: input the standardized data into the trained model of CNN-LSTM, and then get the corresponding output value.

(14) Data standardized restore: the output value obtained through the model of CNN-LSTM is the standardized value, and the standardized value is restored to the original value. As shown in the following formula (9).

where x_i is the standardized restored value, y_i is the output value of the CNN-LSTM, s is the standard deviation of the input data, and $\overline{x}$ is the average value of the input data.

(15) Output result: output the restored results to complete the forecasting process.

4. Experiments

In order to prove the effectiveness of CNN-LSTM, we compared this method with MLP, CNN, RNN, LSTM, and CNN-RNN using the same training set and test set data under the same operating environment. All the experiments are carried out under the running environment of Intel i7-4700H 2.6 GHz, 12 GBs of RAM, 500 GBs of hard disk and Windows 10. According to the influence factors, including the opening price, highest price, lowest price, closing price, volume, turnover, ups and downs, and change, the next day's closing price is predicted.

TABLE 1: Partial sample data.

Date	Opening price	Highest price	Lowest price	Closing price	Volume (share)	Turnover (RMB)	Ups and downs	Change (%)
1991/7/1	136.64	138.62	136.56	136.85	2294000	12469884	−0.71	−0.5161
1991/7/2	135.91	135.96	135.69	135.96	283800	3794100	−0.89	−0.6503
1991/7/3	135.28	135.96	134.98	135.27	271500	1818504	−0.69	−0.5075
1991/7/4	136.63	136.63	134.19	136.63	1339400	8095138	1.36	1.0054
1991/7/5	136.01	137.68	135.9	135.96	1454000	9394861	−0.67	−0.4904

4.1. Data. In this experiment, the Shanghai Composite Index (000001) is selected as the experimental data. The daily trading data of 7127 trading days from July 1, 1991, to August 31, 2020, are obtained from the wind database. Each piece of data contains eight items, namely, opening price, highest price, lowest price, closing price, volume, turnover, ups and downs, and change. Some of the data are shown in Table 1. Take the data of the first 6627 trading days as training set and the data of the last 500 trading days as test set.

4.2. Model Implementation. In order to evaluate the forecasting effect of CNN-LSTM, the mean absolute error (MAE), root mean square error (RMSE), and R-square (R^2) are used as the evaluation criteria of the methods.

The MAE calculation formula is as follows:

$$MAE = \frac{1}{n} \sum_{i=1}^{n} |\hat{y}_i - t y_i|, \tag{10}$$

where $\hat{y}_i$ is the predictive value and y_i is the true value. The smaller the value of MAE, the better the forecasting.

The RMSE calculation formula is as follows:

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^{n} (\hat{y}_i - y_i)^2}, \tag{11}$$

where $\hat{y}_i$ is the predictive value and y_i is the true value. The smaller the value of RMSE, the better the forecasting.

The R^2 calculation formula is as follows:

$$R^2 = 1 - \frac{\left(\sum_{i=1}^{n} (y_i - \hat{y}_i)^2\right)/n}{\left(\sum_{i=1}^{n} (\bar{y}_i - t\hat{y}_i)^2\right)/n}, \tag{12}$$

where $\hat{y}_i$ is the predictive value, y_i is the true value, and $\overline{y_i}$ is the average value. The value range of R^2 is (0,1).

The closer the value of MAE and RMSE to 0, the smaller the error between the predicted value and the real value, the higher the forecasting accuracy. The closer R^2 is to 1, the better the fitting degree of the model is.

4.3. Implementation of CNN-LSTM. The parameter setting of the CNN-LSTM for this experiment is shown in Table 2.

According to the parameter setting of CNN-LSTM network, we can know that the specific model is constructed as follows: the input training set data is a three-dimensional data vector *(None, 10, 8)*, in which 10 is the size of the time_step and 8 is the 8 features of the input dimension. First, the data enter the one-dimensional convolution layer

TABLE 2: Parameter setting of CNN-LSTM.

| Parameters	Value
Convolution layer filters	32
Convolution layer kernel_size	1
Convolution layer activation function	tanh
Convolution layer padding	Same
Pooling layer pool_size	1
Pooling layer padding	Same
Pooling layer activation function	Relu
Number of hidden units in LSTM layer	64
LSTM layer activation function	tanh
Time_step	10
Batch_size	64
Learning rate	0.001
Optimizer	Adam
Loss function	mean_absolute_error
Epochs	100
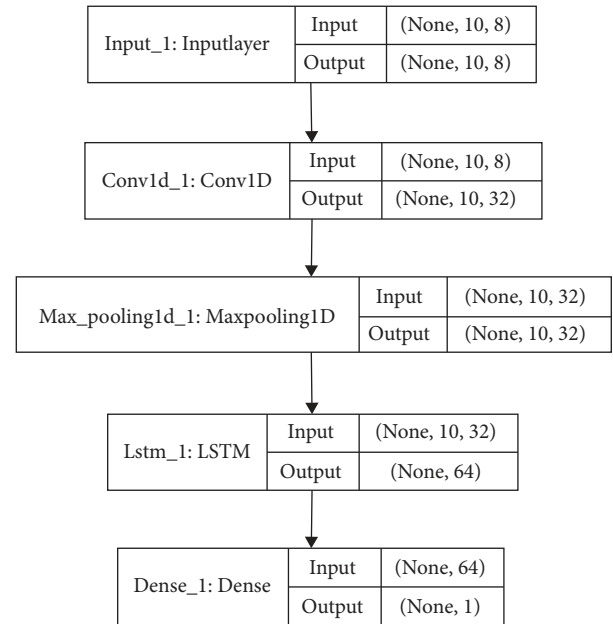
FIGURE 4: The model structure of CNN-LSTM.

to further extract features and obtain a three-dimensional output vector *(None, 10, 32)*, in which 32 is the size of the convolution layer filters. Next, the vector enters the pooling layer, and a three-dimensional output vector *(None, 10, 32)* is also obtained. And then, the output vector enters the LSTM layer for training, and the output data *(None, 64)* after training enter another layer of full connection layer to get the
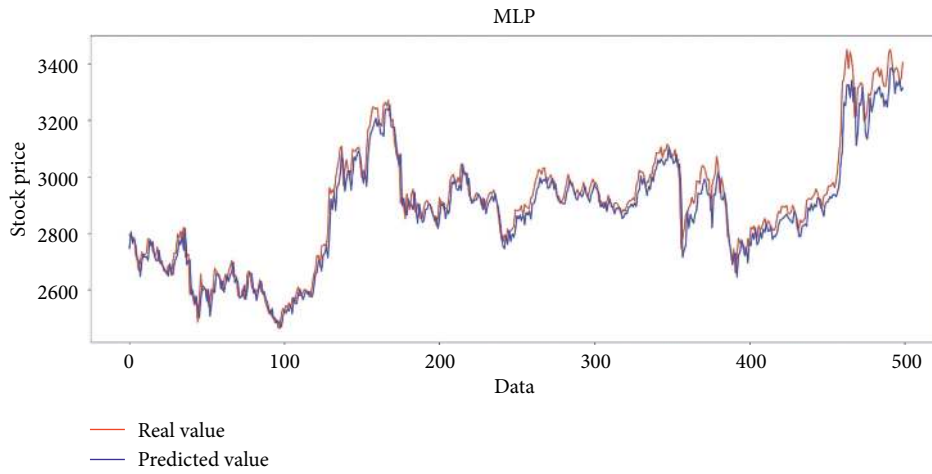
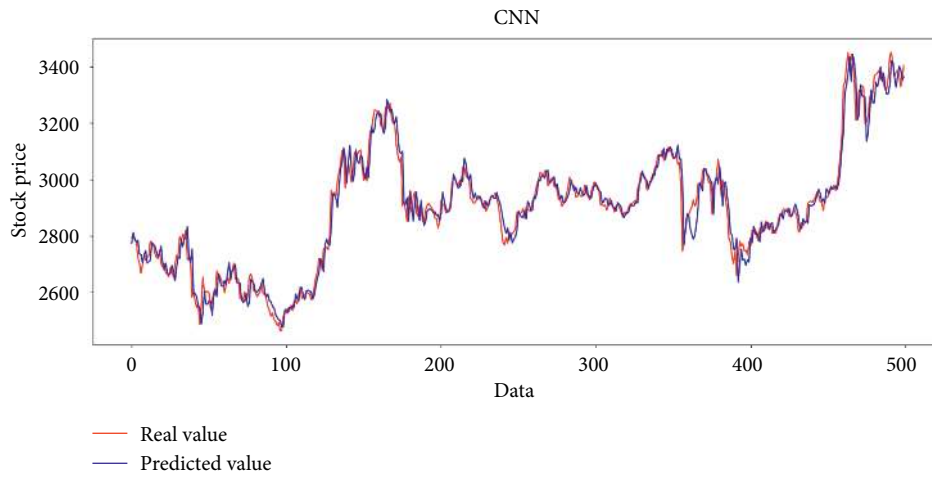Figure 5: Comparison of the predicted value and the real value for MLP.

Figure 6: Comparison of the predicted value and the real value for CNN.
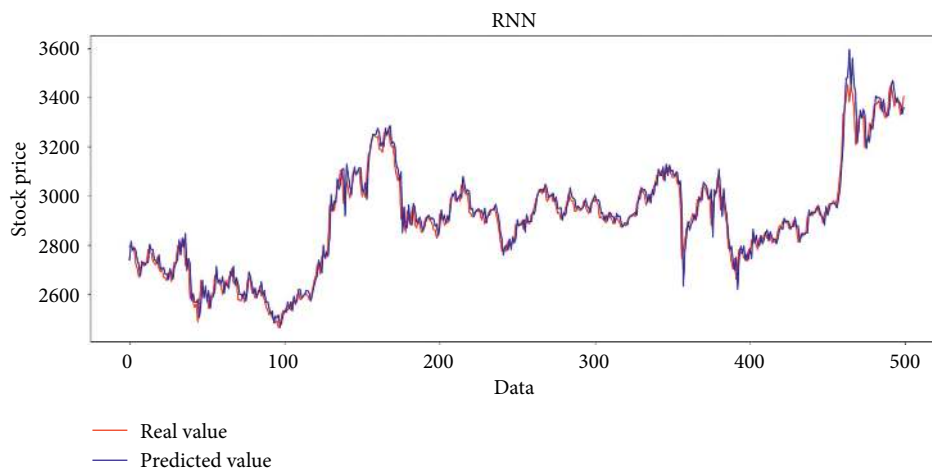

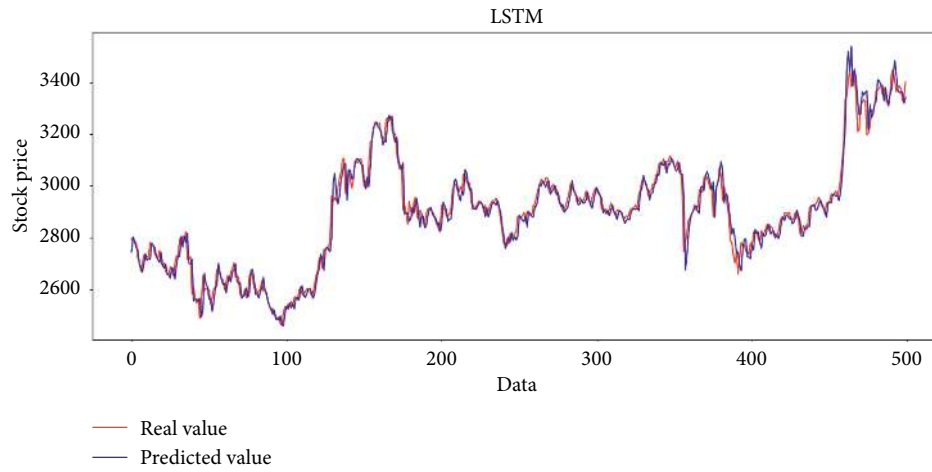Figure 7: Comparison of the predicted value and the real value for RNN.

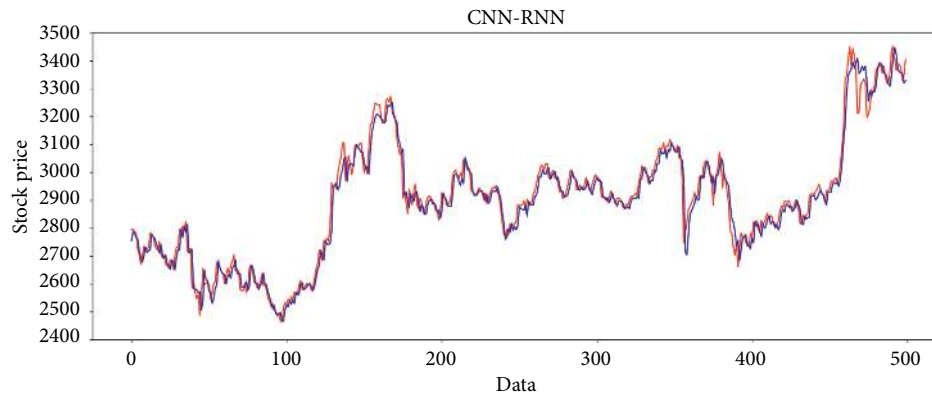FIGURE 8: Comparison of the predicted value and the real value for LSTM.

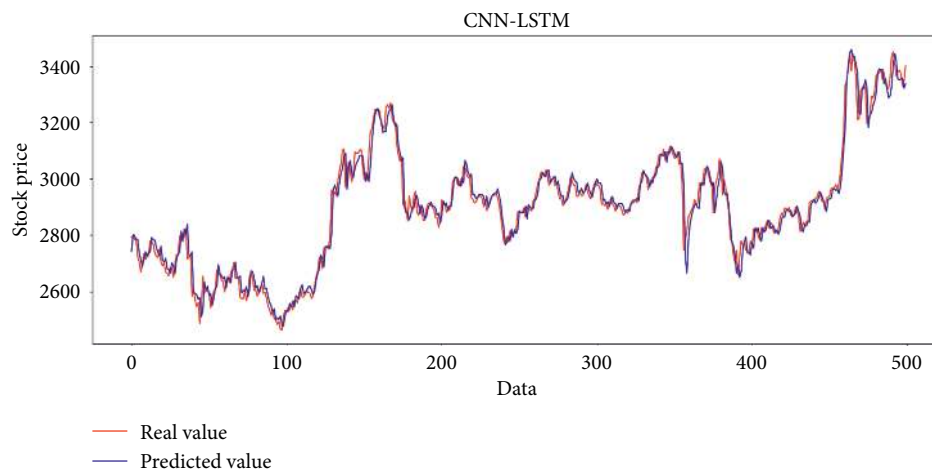FIGURE 9: Comparison of the predicted value and the real value for CNN-RNN.

FIGURE 10: Comparison of the predicted value and the real value for CNN-LSTM.

TABLE 3: Comparison of nine methods evaluation indexes.

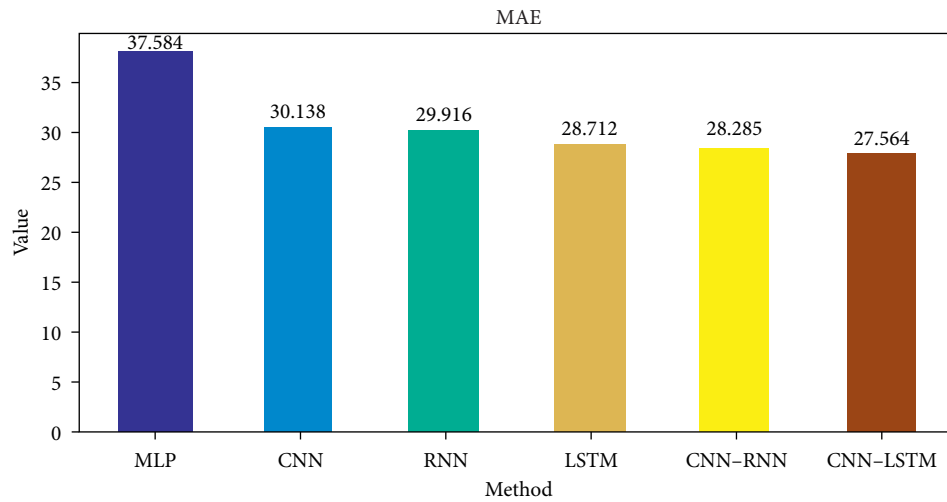| Method	MAE	RMSE	R^2
MLP	37.584	49.799	0.9442
CNN	30.138	42.967	0.9585
RNN	29.916	42.957	0.9593
LSTM	28.712	41.003	0.9622
CNN-RNN	28.285	40.538	0.9630
CNN-LSTM	**27.564**	**39.688**	**0.9646**

FIGURE 11: The result of MAE comparison among different methods.
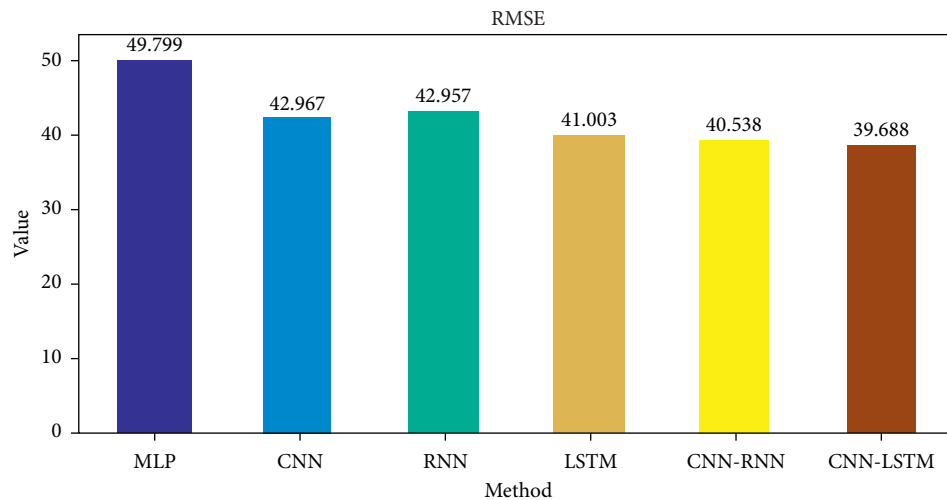

FIGURE 12: The result of RMSE comparison among different methods.

output value; 64 is the number of hidden units in the LSTM layer. The specific CNN-LSTM model structure is shown in Figure 4.

5. Results

After using the processed training set data to train MLP, CNN, RNN, LSTM CNN-RNN, and CNN-LSTM, respectively, the model completed by training is used to predict the test set data, and the real value is compared with the predicted value as shown in Figures 5–10.

In Figures 5–10, among the six forecasting methods, the broken line fitting degree of real value and predicted value is CNN-LSTM, CNN-RNN, LSTM, CNN, RNN, and MLP. CNN-LSTM has the highest degree of broken line fitting which almost coincides with each other, and MLP has the lowest degree of broken line fitting.

According to the predicted value and real value of each method, the evaluation index of each method can be
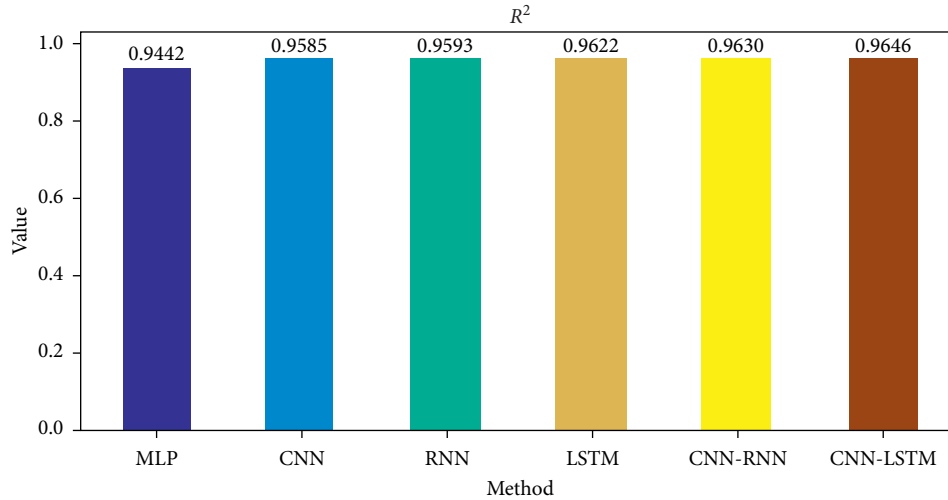
FIGURE 13: The result of R^2 comparison among different methods.

calculated, and the comparison results of the six methods are shown in Table 3 and Figures 11–13.

From Table 3 and Figures 10–12, the MAE and RMSE of MLP are the largest and R^2 is the smallest, while the MAE and RMSE of CNN-LSTM are the smallest, R^2 is the largest, and the closest is 1.

By comparing LSTM with RNN, the MAE and RMSE of LSTM decrease, R2 increases by 0.3%, MAE decreases from 29.916 to 28.712 by 4.0%, and RMSE decreases from 42.957 to 41.003 by 4.5%, so LSTM was better than RNN. However, the error measurement indexes MAE and RMSE of CNN-LSTM are the smallest, and the maximum R^2 is close to 1. Compared with LSTM, after CNN layer, MAE and RMSE of CNN-LSTM proposed in this paper are lower than those without CNN layer; R^2 has a certain improvement; MAE decreases by 4.0%, from 28.712 to 27.564; RMSE decreases by 3.2%, from 41.003 to 39.688; and R^2 increases by 0.2%. It shows that the forecasting performance of LSTM can be effectively improved by extracting data features through CNN.

The results show that the performance of CNN-LSTM is the best among the six methods. In terms of forecasting accuracy, MAE is 27.564 and RMSE is 39.688, which is the smallest among the six forecasting models and has high forecasting accuracy, in terms of forecasting performance, and the R^2 of CNN-LSTM is 0.9646, which is improved by 2.2%, 0.6%, 0.5%, and 0.2%, respectively, compared with the other four methods. Therefore, the CNN-LSTM proposed in this paper is superior to the other four comparative models in terms of fitting degree and error value. It can well predict the closing price of the next day and provide a reference for investors' investment.

6. Conclusions

According to the chronological characteristics of stock price data, this paper proposes a CNN-LSTM to predict the stock closing price of the next day. The method uses opening price, highest price, lowest price, closing price, volume, turnover, ups and downs, and change of the stock data as the input, making full use of the time sequence characteristics of the stock data. CNN is used to extract the features of the input data. LSTM is used to learn the extracted feature data and predict the closing price of the stock the next day. This paper takes the relevant data of the Shanghai Composite Index as an example to verify the experimental results. The experimental results show that the CNN-LSTM has the highest forecasting accuracy and the best performance compared with the MLP, CNN, RNN, LSTM, and CNN-RNN. MAE and RMSE are the smallest of all methods, and R^2 is close to 1. CNN-LSTM is suitable for the forecasting of stock prices and can provide a relevant reference for investors to maximize investment returns. CNN-LSTM also provides the proposal of practical experience for people's research on financial time series data. However, the model still has some shortcomings. For example, it only considers the impact of stock price data on closing prices and fails to integrate emotional factors such as news and national policy into the forecast. Our future research work is mainly to increase the sentiment analysis of stock-related news and national policies, so as to ensure the accuracy of stock forecast.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

References

[1] R. Vanaga and B. Sloka, "Financial and capital market commission financing: aspects and challenges," *Journal of Logistics, Informatics and Service Science*, vol. 7, no. 1, pp. 17–30, 2020.

[2] L. Zhang and H. Kim, "The influence of financial service characteristics on use intention through customer satisfaction with mobile fintech," *Journal of System and Management Sciences*, vol. 10, no. 2, pp. 82–94, 2020.

[3] L. Badea, V. Ionescu, and A.-A. Guzun, "What is the causal relationship between stoxx europe 600 sectors? But between large firms and small firms?" *Economic Computation And Economic Cybernetics Studies And Research*, vol. 53, no. 3, pp. 5–20, 2019.

[4] J. Sousa, J. Montevechi, and R. Miranda, "Economic lot-size using machine learning, parallelism, metaheuristic and simulation," *Journal of Logistics, Informatics and Service Science*, vol. 18, no. 2, pp. 205–216, 2019.

[5] A. Coser, M. M. Maer-Matei, and C. Albu, "Predictive models for loan default risk assessment," *Economic Computation And Economic Cybernetics Studies And Research*, vol. 53, no. 2, pp. 149–165, 2019.

[6] R. Qiao, "Stock prediction model based on neural network," *Operations Research and Management Science*, vol. 28, no. 10, pp. 132–140, 2019.

[7] C. Jung and R. Boyd, "Forecasting UK stock prices," *Applied Financial Economics*, vol. 6, no. 3, pp. 279–286, 1996.

[8] W. Bleesser and P. Liicoff, "Predicting stock returns with bayesian vector autoregressive," *Data Analysis, Machine Learning and Applications*, vol. 1, pp. 499–506, 2005.

[9] A. Adebiyi, A. Adewumi, and C. Ayo, "Stock price prediction using the ARIMA model," in *Proceedings of the 2014 UKSim-AMSS 16th International Conference on Computer Modelling and Simulation*, IEEE, Cambridge, UK, March 2014.

[10] C. Zhang, X. Cheng, and M. Wang, "An empirical research in the stock market of Shanghai by GARCH model," *Operations Research and Management Science*, vol. 4, pp. 144–146, 2005.

[11] Q. Yang and C. Wang, "A study on forecast of global stock indices based on deep LSTM neural network," *Statistical Research*, vol. 36, no. 6, pp. 65–77, 2019.

[12] K.-S. Moon and H. Kim, "Performance of deep learning in prediction of stock market volatility," *Economic Computation And Economic Cybernetics Studies And Research*, vol. 53, no. 2, pp. 77–92, 2019.

[13] J. Li, S. Pan, L. Huang, and X. Zhu, "A machine learning based method for customer behavior prediction," *Tehnicki Vjesnik-Technical Gazette*, vol. 26, no. 6, pp. 1670–1676, 2019.

[14] H. White, "Economic prediction using neural networks: the case of IBM daily stock returns," *Earth Surface Processes & Landforms*, vol. 8, no. 5, pp. 409–422, 1988.

[15] G. P. Zhang, "Time series forecasting using a hybrid ARIMA and neural network model," *Neurocomputing*, vol. 50, no. 1, pp. 159–175, 2003.

[16] Y. Sun, Y. Liang, and W. Zhang, "Optimal partition algorithm of the RBF neural network and its application to financial time series forecasting," *Neural Computing and Applications*, vol. 14, pp. 1441–1449, 2005.

[17] R. Adhikari and R. Agrawal, "A combination of artificial neural network and random walk models for financial time series forecasting," *Neural Computing and Applications*, vol. 24, pp. 305–315, 2014.

[18] L. Zhang, F. Wang, B. Xu, W. Chi, Q. Wang, and T. Sun, "Prediction of stock prices based on LM-BP neural network and the estimation of overfitting point by RDCI," *Neural Computing and Applications*, vol. 30, no. 5, pp. 1425–1444, 2018.

[19] Y. Hu, "Stock market timing model based on convolutional neural network – a case study of Shanghai composite index," *Finance& Economy*, vol. 4, pp. 71–74, 2018.

[20] E. Alibasic, B. Fazo, and I. Petrovic, "A new approach to calculating electrical energy losses on power lines with a new improved three-mode method," *Tehnicki Vjesnik-Technical Gazette*, vol. 26, no. 2, pp. 405–411, 2019.

[21] Y. Xue, C. Wang, and C. Miao, "Research on financial assets transaction prediction model based on LSTM neural network," *Neural Computing and Applications*, vol. 1, 2020.

[22] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[23] B. S. Kim and T. G. Kim, "Cooperation of simulation and data model for performance analysis of complex systems," *International Journal of Simulation Modelling*, vol. 18, no. 4, pp. 608–619, 2019.

[24] L. Qin, N. Yu, and D. Zhao, "Applying the convolutional neural network deep learning technology to behavioural recognition in intelligent video," *Tehnicki Vjesnik-Technical Gazette*, vol. 25, no. 2, pp. 528–535, 2018.

[25] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *MIT Press*, vol. 9, no. 8, pp. 1735–1780, 1997.

[26] V.-D. Ta, C.-M. Liu, and D. A. Tadesse, "Portfolio optimization-based stock prediction using long-short term memory network in quantitative trading," *Applied Sciences*, vol. 10, no. 2, pp. 437–457, 2020.

[27] O. Zarrad, M. A. Hajjaji, and M. N. Mansouri, "Hardware implementation of hybrid wind-solar energy system for pumping water based on artificial neural network controller," *Studies in Informatics and Control*, vol. 28, no. 1, pp. 35–44, 2019.

[28] T. Saric, G. Simunovic, D. Vukelic, K. Simunovic, and R. Lujic, "Estimation of CNC grinding process parameters using different neural networks," *Tehnicki Vjesnik-Technical Gazette*, vol. 25, no. 6, pp. 1770–1775, 2018.

[29] N. Gupta and A. Jalal, "Integration of textual cues for fine-grained image captioning using deep CNN and LSTM," *Neural Computing and Applications*, vol. 12, pp. 1–10, 2019.

[30] A. Yadav, C. K. Jha, and A. Sharan, "Optimizing LSTM for time series prediction in Indian stock market," *Procedia Computer Science*, vol. 167, pp. 2091–2100, 2020.

[31] H. Y. Kim and C. H. Won, "Forecasting the volatility of stock price index: a hybrid model integrating LSTM with multiple GARCH-type models," *Expert Systems with Applications*, vol. 103, pp. 25–37, 2018.

[32] N. C. Petersen, R. Christoffer, F. Rodrigues, and F. C. Pereira, "Multi-output bus travel time prediction with convolutional LSTM neural network," *Expert Systems with Applications*, vol. 120, pp. 426–435, 2019.

[33] Z. Jin, Y. Yang, and Y. Liu, "Stock closing price prediction based on sentiment analysis and LSTM," *Neural Computing and Applications*, vol. 32, no. 13, pp. 9713–9729, 2020.

[34] B. Svetlana and T. Ioannis, "An ensemble of LSTM neural networks for high-frequency stock market classification," *Journal of Forecasting*, vol. 38, no. 6, pp. 600–619, 2019.