Predicting Stock Prices Using The RF-LSTM Combination Model

Yao Yao *

School of Civil and Resources Engineering, University of Science and Technology of Beijing, Beijing, 100083, China

* Corresponding Author Email: U202140035@xs.ustb.edu.cn

Abstract. Because of the nonlinearity and complexity of stock data, a single model's prediction performance is insufficient. A combined RF-LSTM model is suggested to forecast the closing price of stocks in order to solve this problem. Firstly, stock data is obtained from the Kaggle platform, feature sets are constructed, and data is normalized. Secondly, considering the high nonlinearity and information redundancy among multiple features, the optimal feature set is selected using random forest (RF) to reduce data dimensionality and training complexity. Ultimately, a long-short term memory (LSTM) network—which is capable of processing time series data in deep learning—is used to forecast stock prices, and the prediction model is modified accordingly. Comparing the proposed RF-LSTM combined model to the single LSTM neural network model, the results indicate a reduction in root mean squared error (RMSE) and mean squared error (MSE) of 4.14% and 1.00%, respectively. The prediction accuracy of stock prices can be increased by using this combined model.

Keywords: Machine learning; Stock price prediction; Long-short term memory; Random Forest.

1. Introduction

Stocks are a vital component of the capital market, and investors, investment institutions, and the nation's macroeconomic development all place a high value on stock prices. Financial research is hindered by the high volatility and uncertainty of stock prices [1]. Stocks are volatile investments. By predicting stock prices, investors can prevent risks and enhance the safety and profitability of stock investments. It's critical to research stock price prediction techniques in order to optimize investment risk avoidance and enhance investment profits [2].

In order to forecast stock prices in the BANKEX index of Standard & Poor's (S&P), Four deep learning models were used by A. Balaji et al.: Extreme Learning Machine (ELM), Convolutional Neural Network (CNN), Gated Recurrent Unit (GRU), and Long Short-Term Memory (LSTM). According to the results, all deep learning models were capable of generating good prediction accuracy [3]. In recent years, a lot of research has been done on artificial neural networks and associated technologies for stock price prediction techniques. Because stock price predictions effectively solve the drawbacks of previous prediction methods, their accuracy and dependability have increased [4]. L. O. Orimoloye et al. investigated the stock-price prediction capabilities of shallow neural networks (SVMs and single-layer neural networks) and deep neural networks (DNNs). The outcomes demonstrated that DNN outperformed shallow neural networks in terms of prediction [5]. Additionally, CHENK et al. anticipated stock returns using the LSTM network model and assessed how different input variables affected the prediction accuracy of the model [6]. LSTM neural network has the characteristics of selective memory and internal influence of time series, and performs outstandingly in quasi-random non-stationary sequences such as stock prices. TAN Z et al. forecasted stock prices using random forest (RF), analyzed the features' importance using RF, and chose the right parameters based on the features' significance to develop workable techniques [7]. The stock market is dynamic, non-linear, and non-parametric, making the RF approach a better fit for analyzing these properties than previous methods. Complex features can be analyzed by RF and its learning speed is fast. High-dimensional data can be selected features using this tool. Various predictions, classifications, and feature selections have been extensively used in recent years [8]. Combining the advantages and disadvantages of the above research, this paper Combining the two

methods, the RF-LSTM stock price trend prediction method is used to forecast the stock's closing price. Stock price prediction requires careful consideration of both the best algorithm to use and the best feature extraction technique. In order to make predictions in this article, the first step is to create a feature set using 16 technical indicators that are often utilized in the stock market. To decrease data dimension and increase prediction accuracy, the best technical indicators are selected from the 16 stock technical indicators using the RF feature selection approach. Prediction of stock prices and analysis of prediction results is done using the LSTM algorithm.

2. Stock market technical indicators

Technical indicators are specific results obtained by processing raw data in a time sequence according to a certain indicator algorithm, resulting in a data sequence. In stock market forecasting analysis, technical indicators are widely used and possess characteristics that make them intuitive and easy to apply when determining market trends [9]. The application range and constraints of different technical indicators vary. The comprehensiveness and accuracy of feature representation cannot be guaranteed by using only one technical indicator when representing stock features. The accuracy of feature representation and complementarity of data can be improved by selecting multiple representative and quantifiable technical indicators. The stock market's 16 commonly used technical indicators are used to construct a predictive feature set, as illustrated in Table 1. It encompasses all the important factors that influence stock prices and provides a complete reflection of their trends, making it ideal for stock price prediction.

Table 1. chart of feature set

Feature name	description
AROONOSC	Aaron Oscillation Line, the further it deviates from the zero line, the stronger the trend.
BOP	Balance of Power, used to observe when prices push to a certain extreme value, buy The balance of power between the seller and the seller.
MFI	The Money Flow Index indicator is an instrument for measuring purchasing and selling pressure in technical analysis.
APO	Measurement of a stock's price movement is achieved using the Absolute Price Oscillator indicator, a technical analysis indicator.
ADOSC	Chaikin A/D Oscillator is a volume and price indicator that is used to measure the changing trend of the accumulation/distribution line (A/D Line).
ADX	Average Directional Index is a technical indicator for trend judgment. The average trend index is a technical indicator for trend judgment.
CCI	To determine if the stock price is over the range of the normal distribution, the Commodity Channel Index was created.
WILLR	The William Index uses swing points to measure overbought and oversold conditions in the market.
ATR	A moving average of stock price changes over a predetermined length of time is called Average True Rage, and it is mostly used to assess chances for purchases and sales.
ULTOSC	The Ultimate Oscillator is a multi-functional indicator. In addition to its role in trend confirmation and overbought and oversold, its "breakthrough" signal can not only provide the most appropriate trading opportunities, but also further enhance the reliability of the indicator.
RSI	Relative Strength Index determines overbought/oversold conditions based on price changes over a certain period.
HT_DCPERIOD	Hilbert Transform - Dominant Cycle Period uses price as an information signal and calculates the dominant cycle of the price by transforming the price into Hilbert space. The dominant cycle is the main cycle in the market and can be used as a basis for timing.
ROC	The rate of change indicator determines the percentage of changes in the stock price's closing price within a given time frame, compares price movement to gauge price momentum, and assesses the stock price's trend to see if there is a willingness to buck the trend. Among the counter-trend indicators is this one.
STDDEV	Standard deviation is a way of measuring price volatility by relating a price range to its moving average.
OBV	The OBV indicator is a technical indicator that discovers popular stocks and analyzes trends in stock price movement by utilizing the factor of "volume" as a breakthrough.
BETA	A risk indicator called the beta coefficient compares the price changes of individual companies or stock funds to the stock market as a whole.

3. Model Introduction

3.1. Random Forest Algorithm

In the bagging class, one kind of ensemble learning technique is the Random Forest algorithm [10]. The method combines many decision tree classifiers $\{h(x,\ \theta_k),\ k = 1, 2, \ldots, K\}$, where K is the random forest's total number of decision trees and $\{\theta_k\}$ is a random vector that follows an

independent identical distribution. The bootstrap resampling technique is used to recover several samples from the original sample, as seen in Figure 1. After then, a decision tree is built for every sample, and the voting result that shows up the most often across all decision trees is used to decide the final prediction outcome. Random Forests typically exhibit better performance than individual decision trees on the vast majority of datasets.

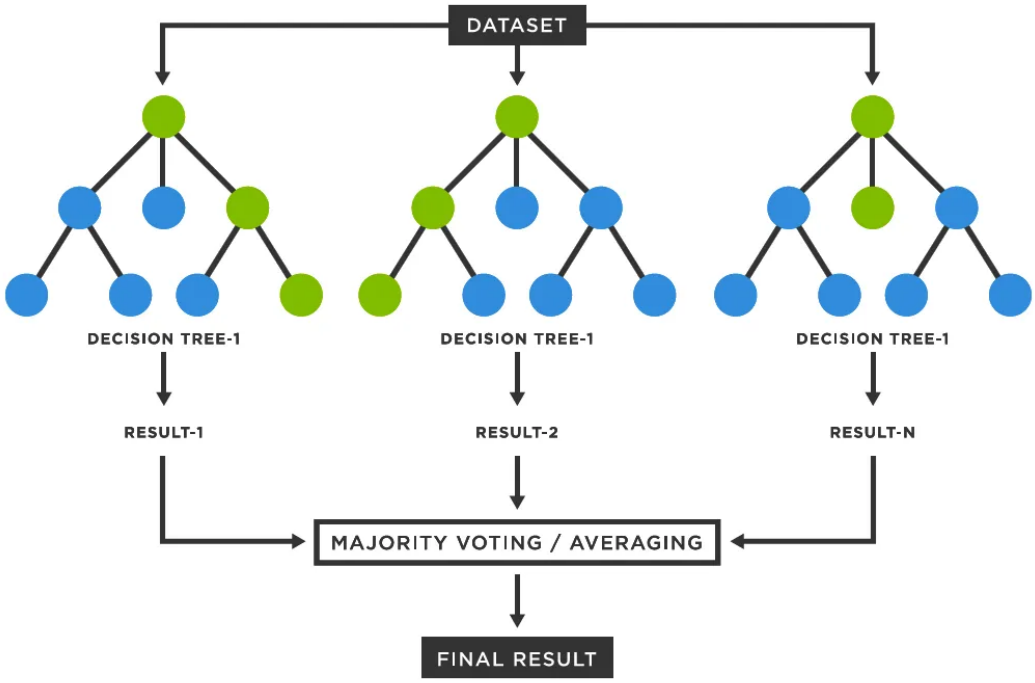

Fig. 1 Schematic diagram of random forest algorithm [10]

Random Forest may be used as a stand-alone feature selection method. The Random Forest algorithm's variable importance measure is used to rank the features. Subsequently, the least significant feature (with the lowest relevance score) is eliminated from the feature set each time a sequential backward search technique is applied. The feature set with the fewest variables and the best classification accuracy is the one that is selected as the feature set after the classification accuracy is determined repeatedly. By optimizing input variable combinations according to Random Forest's variable importance ranking, one can greatly improve prediction accuracy, simplify the RF model, and improve the forecasting capability of the model.

The Random Forest algorithm's steps are as follows, assuming that the decision tree $h(\theta)$, with leaf nodes represented by $l(x,\theta)$, corresponds to the random parameter vector θ.

(1) Create k training sets at random θ_1, θ_2, ..., θ_k, using the bootstrap resampling technique. Then, for each training set, create the associated decision tree sets $\{h(x,\theta_1)\}$, $\{h(x,\theta_2)\}$, ..., $\{h(x,\theta_k)\}$.

(2) Assuming the number of attributes in the training sample is M, randomly select m (0 < m < M) features from the M features as the current node's splitting feature set. By calculating the information content implied by each feature, choose the best splitting method from the m features to split the node.

Each decision tree is grown to its maximum extent without pruning during this process.

For the test sample Z, apply each decision tree to test and obtain the corresponding classes $\{h(z,\theta_1)\}$, $\{h(z,\theta_2)\}$, ..., $\{h(z,\theta_k)\}$.

The test sample Z is classified into the category to which the class with the greatest number of outputs among the k decision trees, represented as $H(x)$, according to the voting technique. The following is the formula for calculating $H(x)$:

$$H(x) = arg\max_{y} \sum_{k} I(h_k(x) = y) \quad (1)$$

In the equation: The target variable is y, the indicator function is I, any decision tree is indicated by $h(i)$, and the ensemble classification model is represented by $H(x)$.

The margin function calculates how much the average correct categorization differs from the misclassification. For a classifier model, a higher margin function value indicates higher confidence in its predictive capability. The formula for the margin function is as follows:

$$mg(X,Y) = av_k I[h_k(X) = Y] - \max_{j \neq y} av_k I[h_k(X) = j] \tag{2}$$

In the equation: X is the input vector; Y is the correct classification; j is the incorrect classification; av is the average value; n is the number of trees.

3.2. LSTM

A unique kind of RNN called LSTM deals with the problems of gradient expansion and disappearing gradients that arise while training lengthy sequences in RNNs. This means that as the network layers increase, the ability of later nodes to perceive information from earlier nodes weakens, leading to a phenomenon where past information is forgotten over time [11]. Building upon the basic RNN, LSTM incorporates memory cells in each neuron of the hidden layer, allowing for controlled memory information along the time series. This enables a deeper exploration of potential patterns in the data, leading to more accurate and reliable predictions. Figure 2 displays an LSTM memory cell's structural diagram.

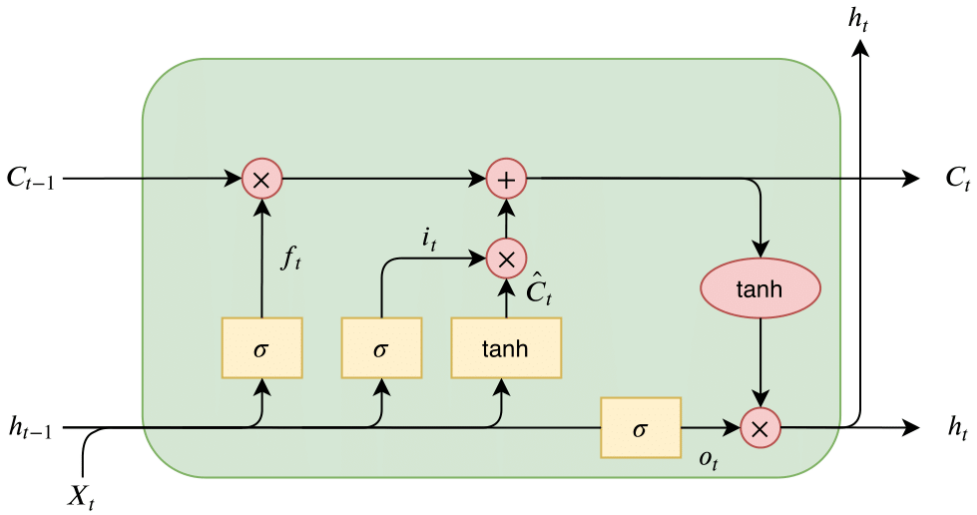

Fig. 2 Structure diagram of LSTM memory unit [11]

The covert stratum and states of memory cells at the previous time step are represented by C_{t-1} and h_{t-1} , respectively; the current hidden layer and memory cell states are represented by S_t and h_t , respectively. Each time information is passed between units, the forgotten entrance f_t, input entrance X_t, and output entrance o_t are employed to regulate the recall and forgetting of both recent and past knowledge. The following are the formulae for the forget entrance f_t, input entrance X_t, and output entrance o_t:

$$f_t = \sigma(W_{Xf}X_t + W_{hf}h_{t-1} + b_f) \tag{3}$$

$$i_t = \sigma(W_{Xi}X_t + W_{hi}h_{t-1} + b_i) \tag{4}$$

$$o_t = \sigma(W_{Xo}X_t + W_{ho}h_{t-1} + b_o) \tag{5}$$

In the equations: X_t represents the input at time step t; σ is the sigmoid activation function; W_{Xf} is the weight distribution matrix that the forget entrance uses as input X. The weight distribution matrix W_{hf} links the state of the covert stratum to the forget entrance; The linear bias of the forget entrance is shown by b_f; The weight distribution matrix that links input X to the input entrance is

represented by W_{Xi}; The weight distribution matrix that links the input entrance to the covert stratum state is represented by W_{hi}; The input entrance's linear bias is indicated by b_i; The weight distribution matrix that links input X to the output entrance is represented by W_{Xo}; The weight distribution matrix that links the output entrance and the covert stratum state is represented by W_{ho}; The output entrance's linear bias is represented by b_o.

At time step t, The input entrance input for updating information $\hat{C}_t$ and the state value of the memory cell C_t from the previous time step make up the state value of the memory cell C_{t-1}. The present the state of the memory cell, C_t, is the consequence of these two components being managed by the input entrance and forget entrance. The formulas are as follows:

$$\hat{C}_t = tanh(W_{Xc}X_t + W_{hc}h_{t-1} + b_c) \tag{6}$$

$$C_t = i_t \cdot \hat{C}_t + f_t \cdot C_{t-1} \tag{7}$$

In the equations: The weight distribution matrix from input X to the memory unit is represented by W_{Xc}; The weight distribution matrix from the covert stratum to the memory unit is represented by W_{hc}; The memory unit's bias in linear terms is represented by b_c.

Lastly, the hidden layer status as of right now h_t is determined by the output entrance o_t and the status of the memory cell right now s_t. The formula for h_t is:

$$h_t = o_t \cdot tanh(s_t) \tag{8}$$

3.3. RF-LSTM

Figure 3 displays the RF-LSTM prediction model's flowchart, with specific steps outlined as follows.

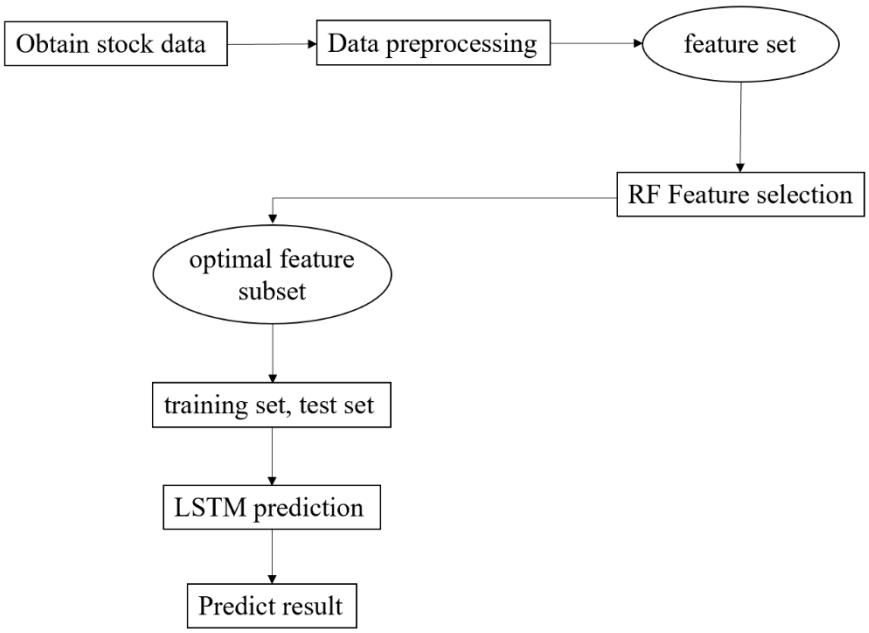

Fig. 3 Flow chart of RF-LSTM (Photo/Picture credit: Original)

Compile stock data, normalize the feature set, and create technical indicators for forecasting.

(1) Utilizing the bootstrap resampling approach, train a random forest model with the out-of-bag (OOB) data, which is the likelihood that each sample from the original data will not be picked, expressed as $(1 - 1/N)^N$.

(2) To find the OOB data error, or $errOOB1$, for each decision tree, choose the corresponding OOB data.

(3) All sample characteristics x in the OOB data should be subjected to noise interference; the OOB data error ($errOOB2$) should be updated correspondingly. The formula to determine the significance of feature x, assuming N trees in the forest, is:

$$FIM_x = (errOOB2 - errOOB1)/N \quad (9)$$

In the equation: FIM stands for feature importance measures. After adding random noise, if the OOB data accuracy drops significantly, it suggests that the trait has a considerable influence on the samples' prediction outcomes, indicating a reasonably high level of significance.

(4) Determine the significance of each aspect and order them accordingly.

(5) Based on the feature sets obtained and the corresponding OOB error rates, Choose the feature set with the least amount of OOB errors.

(6) Input the selected feature set into LSTM for prediction, compare the predicted results with the sample labels, calculate the loss function for cross-entropy.

(7) To get the final prediction results, test the trained model using test data, compare it to actual stock prices, and make inferences.

4. Experiment

4.1. Experimental data

To validate the effectiveness of the method in this study, the Netflix raw data studied in this paper is sourced from Kaggle. Kaggle is a free, open-source platform for AI datasets. The dataset includes trading information from 2018-02-05 to 2022-02-04, including the following five types of daily trade data: volume, closing price of the stock, opening price of the stock, highest price of the stock, and lowest price of the stock. Twenty percent of the samples from the subset are utilized as an independent test set, while the remaining eighty percent are used as the feature selection sample set. The independent test set is utilized to test the chosen features, and the classification predictions made without feature selection are employed in a comparison study.

Construct a feature set from the selected data. The characteristic quantities built using technical indicators have distinct value ranges, and these value ranges differ significantly because various calculation methods are used for each indication. Huge quantitative differences will cause the optimization of algorithm model parameters to become complicated and make it easy to overfit, hence negatively influencing the outcome of the final prediction. As a result, every one-dimensional feature component in this article is converted to [-1, 1] and the feature amounts are normalized.

$$x_d = \frac{v_d - v_{d\_min}}{v_{d\_max} - v_{d\_min}} \quad (10)$$

In the equation: The smallest value of the d-th dimensional feature component is denoted by v_{d_min}; The maximum value is denoted by v_{d_max}.

4.2. Lab environment

This article uses Anaconda development software to build the model. Sklearn and Keras modules are used for model analysis, whereas pandas and numpy modules are mostly used for data processing. There are 10,000 decision trees (n_estimator), four decision tree depths (max_depth), and one hundred iterations. The input variable for the time series is the optimal feature subset that was extracted; the duration of the input is 100d; and the output variable is the current day. Three LSTM layers and two dense layers make up the total of five layers in the LSTM model. To prevent overfitting, regular terms and dropout parameters are incorporated during LSTM model training. The model's efficacy is assessed using the root mean square error (RMSE) and mean square error (MSE) as assessment metrics.

4.3. Forecast results and analysis

4.3.1 LSTM

In Figure 4, the prediction impact is displayed.

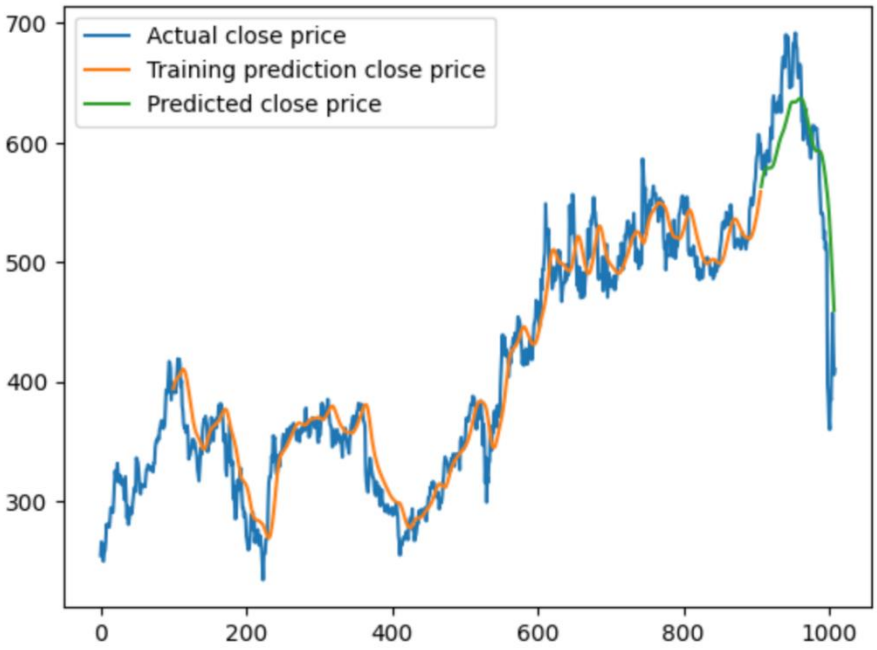

Fig. 4 Result chart of LSTM (Photo/Picture credit: Original)

Since MSE is the average sum of squared differences of each data deviation from the real value, the MSE of the LSTM is 1.51%. The prediction impact gets worse with increasing value. The sample standard deviation of the residual, or actual value, as compared to the expected value is known as the RMSE. This may be used to show how dispersed the sample is. During nonlinear fitting, the lesser the root mean square error (RMSE), the better. LSTM has an RMSE of 11.26%.

4.3.2 RF-LSTM

Figure 5 displays the feature significance ranking that is achieved after features are retrieved using the RF-LSTM method. The feature set obtained from 16 technical indicators for prediction is shown by the vertical axis, while the horizontal axis shows the relevance of the feature quantity.

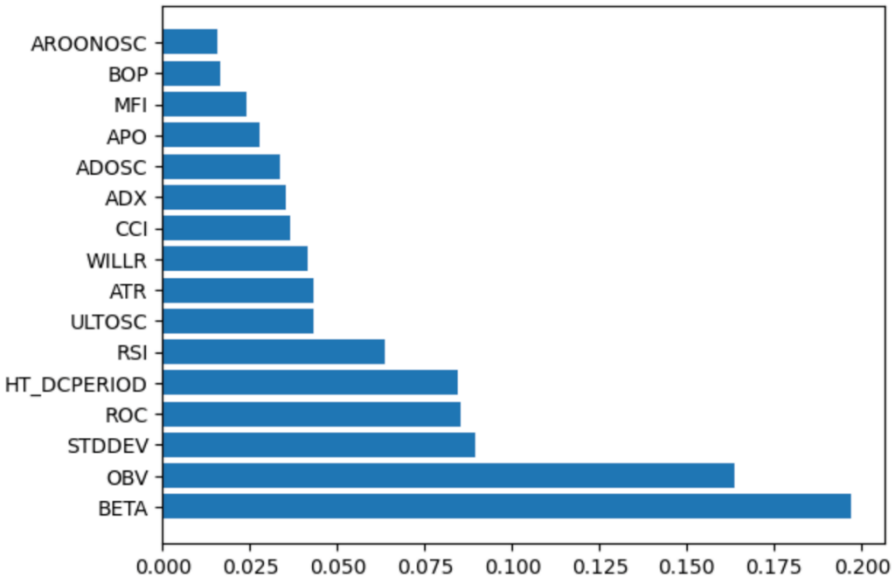

Fig. 5 Sorting graph of feature importance (Photo/Picture credit: Original)

The significance of the remaining features—all of which are less than 0.1—is minimal, whereas that of OBV and BETA is significant. The optimal feature subset finally obtained is BETA. The original basic data's dimensionality can be decreased while maintaining the majority of its information by substituting BETA for it as the training data for modeling. The figure 6 illustrates the prediction effect.

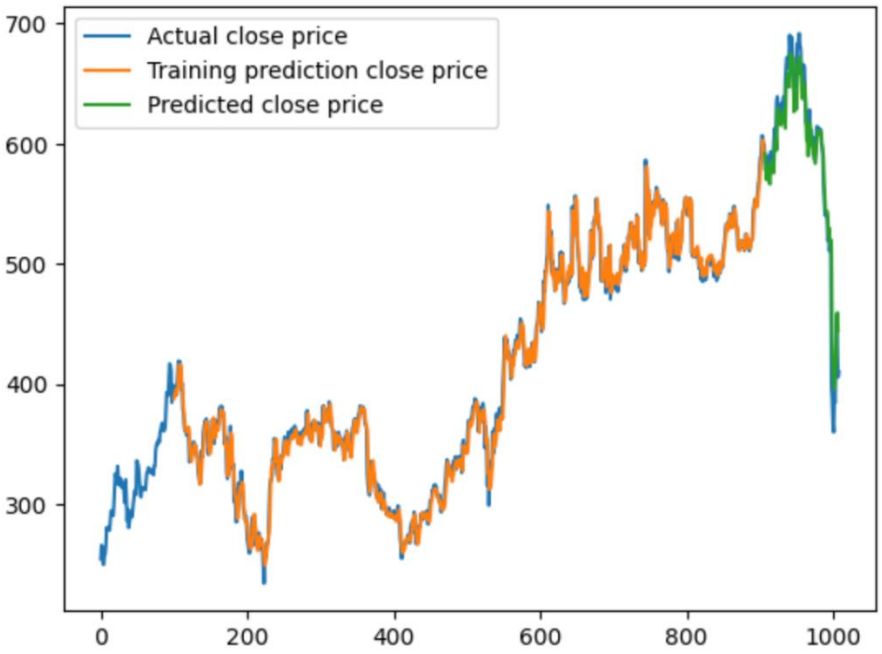

Fig. 6 Result chart of RF-LSTM (Photo/Picture credit: Original)

The mean square error (MSE) of each data point's divergence from the real value is 0.51% for the RF-LSTM. The prediction impact gets worse with increasing value. The sample standard deviation of the residual, or actual value, as compared to the expected value is known as the RMSE. This may be used to show how dispersed the sample is. During nonlinear fitting, the lesser the root mean square error (RMSE), the better. RF-LSTM has an RMSE of 7.12%.

4.3.3 Comparison

The general trend of stock prices may be predicted using both RF-LSTM and LSTM alone. The genuine value and the LSTM prediction results, however, differ significantly. This is primarily because insufficient feature extraction from a single model results in a poor prediction impact of a single model for financial time series with noise in the features, which also contain noise and information redundancy. The RF-LSTM model predicts a value that is more accurate, fits the curve more closely, and is closer to the genuine value. The model fits the data better overall. It shows how choosing the right amount of variables may simplify the network design, improve model performance, and improve the fit between the actual and projected value curves by reducing the size of the input layer and the number of concealed layer nodes.

The prediction value error of the RF-LSTM model is lower than that of the conventional LSTM model. Model's improved generalizability and dependability, as well as its superior impact on stock price and forecast, are demonstrated by the decreased prediction error.

The model's superior impact on stock price and forecast, together with its improved generalizability and dependability, are demonstrated by the decreased prediction error.

5. Conclusion

This work builds a predictive feature set by including technical indicators linked to stocks into the RF-LSTM model, which is used to predict stock prices. These indicators can handle the dimensionality reduction of the model input data, remove the correlation of input features cut down

on the LSTM neural network's layer count, thoroughly examine the data's latent information, and comprehensively reflect the trend of stock prices. This improves the simplicity of the input data while simplifying the network structure as a whole. A comparison of the simulation results shows that Compared to the LSTM model, the RF-LSTM model offers a higher prediction effectiveness. The prediction findings are more steady in addition to having an increased level of accuracy. There may be variations between the results and the actual numbers due to the impact of outside variables and the intrinsic volatility of stock market patterns, but the general trend of the forecasts stays the same.

Even though this study has produced some initial findings, there are still some problems. The stock market is complicated and unpredictable; in addition to pertinent indicator data, variables including global events, country legislation, industry advancements, and human engagement can impact stock patterns. In the stock market, volume and price indicators show delayed responses, and models are unable to swiftly forecast unforeseen occurrences. It is still necessary to validate the model's generalizability on further datasets. The only way to increase the model's speed and accuracy and make more accurate stock forecasts is to carry out additional research and optimize the model in different ways.

References

[1] Su Z, Fang T, Yin L. Understanding stock market volatility: What is the role of US uncertainty. The North American Journal of Economics and Finance, 2019, 48: 582-590.

[2] Nikou M, Mansourfar G, Bagherzadeh J. Stock price prediction using DEEP learning algorithm and its comparison with machine learning algorithms. Intelligent Systems in Accounting, Finance and Management, 2019, 26(4): 164-174.

[3] Balaji A J, Ram D S H, Nair B B. Applicability of deep learning models for stock price forecasting an empirical study on BANKEX data. Procedia computer science, 2018, 143: 947-953.

[4] Kumar G, Jain S, Singh U P. Stock market forecasting using computational intelligence: A survey. Archives of computational methods in engineering, 2021, 28(3): 1069-1101.

[5] Orimoloye L O, Sung M C, Ma T, et al. Comparing the effectiveness of deep feedforward neural networks and shallow architectures for predicting stock price indices. Expert Systems with Applications, 2020, 139: 112828.

[6] Chen K, Zhou Y, Dai F. A LSTM-based method for stock returns prediction: A case study of China stock market. 2015 IEEE international conference on big data (big data). IEEE, 2015: 2823-2824.

[7] Tan Z, Yan Z, Zhu G. Stock selection with random forest: An exploitation of excess return in the Chinese stock market. Heliyon, 2019, 5(8).

[8] Lohrmann C, Luukka P. Classification of intraday S&P500 returns with a Random Forest. International Journal of Forecasting, 2019, 35(1): 390-407.

[9] Oriani F B, Coelho G P. Evaluating the impact of technical indicators on stock forecasting. 2016 IEEE Symposium Series on Computational Intelligence (SSCI). IEEE, 2016: 1-8.

[10] Schonlau M, Zou R Y. The random forest algorithm for statistical learning. The Stata Journal, 2020, 20(1): 3-29.

[11] Bansal M, Goyal A, Choudhary A. A comparative analysis of K-nearest neighbor, genetic, support vector machine, decision tree, and long short-term memory algorithms in machine learning. Decision Analytics Journal, 2022, 3: 100071.