Hindawi
Scientific Programming
Volume 2023, Article ID 9963940, 15 pages
https://doi.org/10.1155/2023/9963940

Research Article

A Hybrid Model Using PCA and BP Neural Network for Time Series Prediction in Chinese Stock Market with TOPSIS Analysis

Lei Hang, Dandan Liu, and Fusheng Xie

Business School, Shanghai Normal University Tianhua College, Shanghai 201815, China

Correspondence should be addressed to Fusheng Xie; fushengx@sina.com

Received 14 February 2023; Revised 31 May 2023; Accepted 10 June 2023; Published 29 June 2023

Academic Editor: Shah Nazir

The stock price changes rapidly and is highly nonlinear in the financial market. One of the common concerns of many scholars and investors is how to accurately predict the stock price and the trend of rising and falling in a short time. Machine learning and deep learning techniques have found their place in financial institutions thanks to the ability of time series data prediction with high precision. However, the prediction accuracy of these models is still far from satisfactory. Most existing studies use original, single prediction algorithms that cannot overcome inherent limitations. This study proposes a hybrid model using principal component analysis (PCA) and backpropagation (BP) neural networks. The historical records of China Merchants Bank are used for data collection from 2015 to 2021. PCA preprocesses the original data to reduce the dimensionality and is then adopted by the BP neural network to predict the stock closing price of China Merchants Bank. We compare and analyze the PCA–BP model with three training algorithms, and the results indicate that the Bayesian regularization algorithm performs best. Besides, we perform the stock prediction using a traditional exponential smoothing approach. The experiment results show that the predicted stock closing price is close to the actual value, and the mean absolute percentage error can reach 0.0130, which is more significant than the traditional approach. Furthermore, A TOPSIS approach is utilized to evaluate the robustness of the proposed model. Finally, we demonstrate the usability of the designed hybrid model by predicting the stock price of another selected stock.

1. Introduction

In the 1929 crash, the Dow fell 82.30%, which means the investor lost 82.30% on average. After China's stock market crash, the Shanghai Composite Index fell from 6,124.04 points to 1,664.93 points, losing 22 trillion yuan and a per capita loss of 1,30,000 yuan [1]. The stock market crash warns every investor of the importance of risk prevention and investment analysis in the stock market era. Thanks to the real-time update and disclosure of stock market and industry data, researchers can analyze and explore the operation law of stock prices through historical data to predict the stock price trend. The appropriate mathematical model for stock price prediction can reduce investment risk and improve the decision-making efficiency of investors. The traditional stock price prediction method is mainly econometrics. However, the stock price can be affected by many other factors. Consequently, it is difficult for these traditional mathematical models to consider all these factors to make accurate predictions.

The autoregressive integrated moving average (ARIMA) was widely used for time series prediction because of its statistical characteristics [2]. However, this model can only extract linear features from data. It is challenging to value stocks and estimate their future performance as long-term stock values are inherently unpredictable. Recently, various machine learning algorithms, including support vector machine (SVM), gradient-boosted regression trees, and random forecasts, have benefited from integrating statistics and learning models. These techniques may reveal complex patterns with nonlinear properties and certain relations that are difficult to find using linear algorithms. SVM is a hotspot in stock prediction as it can avoid local minima, overfitting, and dimension disasters often encountered in nonlinear models [3]. The use cases of SVM in time series analysis are also called support vector regression (SVR). However, SVR still has problems such as kernel function selection, tuning, and shallow feature extraction [4].

A deep neural network transmits more layers and has a more complex structure. It can transform shallow information in data into more abstract high-feature information [5] with solid performance and broad applicability. A recurrent neural network (RNN) can handle dependencies in time series data, so it is widely used in stock price prediction research [6]. In RNN, the previous layers may stop learning as the gradient disappears. It may forget what it sees in the long-term memory and thus has only a short-term memory [7]. Long short-term memory (LSTM) has a similar control flow to the RNN. It processes the data that transmit information during forward propagation.

The difference between these two lies in the different processing processes within the cell. LSTM comprises three gates: forget, input, and output. During training, these gates can learn what information to save or forget [8]. This is precisely what researchers seek from the vast amount of historical stock trading data. The conventional neural network (CNN) is widely used in image recognition, text recognition, target recognition, and target detection [9]. Due to its ability to extract local and in-depth features from data [10], it can investigate features for predicting the future movement of markets. In recent years, many studies combined financial news with quantitative indicators to improve the performance of stock prediction based on behavioral finance theory [11] with the development of natural language processing.

Despite the widespread use of data mining and machine learning techniques in the financial sector, foreign academics' research focuses mainly on optimizing specific algorithms and the foreign stock market. Although deep learning methods and attention mechanisms have improved feature representation, the complexity of stock data often leads to the risk of overfitting. Besides, these studies utilize standard performance metrics, such as mean squared errors, to evaluate the performance of the model predictions. There is no consensus on the accuracy of the model predictions as the dataset can be changed. Moreover, few studies investigate the robustness of the prediction models. The market complexity of China is higher, and the local stock market research is currently backward.

Financial markets are complicated, and the stock price can be affected by many inherently complex human factors, including public opinion, the political environment, and news events, which cause noise in stock data. A single model cannot cover all aspects, and the inherent attributes of the model itself have inherent limitations. This paper aims to create a reasonably accurate and reliable stock forecasting valuation model for domestic investors. This paper proposes a hybrid framework based on the combination of principal component analysis (PCA) and backpropagation (BP) neural network to predict the closing price of the stock market in China. PCA is utilized to simplify the data dimension and eliminate redundant information. PCA, however, is unable to uncover the data's nonlinear connection. The self-learning capability of the BP neural network [12], which can actualize any complicated nonlinear mapping, makes it a suitable model for stock price prediction. The historical stock data of China Merchants Bank are utilized for training and testing the proposed model. Different training algorithms are evaluated along with the PCA–BP neural network. We compare and analyze the performance of the proposed model with a traditional method called the exponential smoothing model. A TOPSIS-based approach is used to analyze the robustness of the prediction model.

The contributions of this paper are summarized as follows:

(1) This paper presents a hybrid model using PCA and BP neural networks to forecast the stock closing price of the Chinese stock market.
(2) The hybrid model is tested with different training algorithms to determine the optimal model. Except for common performance indicators such as the mean square error (MSE) and mean absolute percentage error (MAPE), this paper adopts a TOPSIS-based approach for ranking the models to evaluate the robustness of different models.
(3) The usability of the optimal model has been validated by predicting another stock's stock price. The findings of this study may be helpful to Chinese investors since they might give them the knowledge that will help them make educated choices about their investments and the diversity of their portfolios.

The rest of this paper is organized as follows: Section 2 overviews the related work of stock price prediction; Section 3 introduces the methodology of this paper; Section 4 introduces the relevant theory; Section 5 presents the experiment results; Section 6 concludes the whole paper and outlines some future research directions.

2. Related Work

The time series problem of stock market forecasting determines potential future direction or price value using historical price data. However, estimating this using typical time series techniques is difficult since the stock market data are not linear and are affected by many aspects. Numerous studies on various time series prediction methods have been carried out for decades. Time series predictions have undergone several stages, including exponential smoothing, autocorrelation, moving averages, and regression prediction [13]. These prediction methods only use autocorrelation or simple linear regression to conduct the prediction process, including the large information granularity. Hence, the accuracy is poor and has limitations by adapting various data types.

To evaluate the movement of stock prices, Ticknor [14] proposes a new method of Bayesian regularization (BR) with an artificial neural networks (ANNs) to predict financial market behavior, which reduces the possibility of overfitting and overtraining. The results indicate that the proposed model can improve the prediction and generalization ability of the ANN. Besides, the error of the prediction results is tiny, even without data preprocessing, seasonal testing, or systematic analysis. The results indicate that neural networks have advantages, such as solid learning ability, better inclusiveness to noisy data, and fine nonlinear mapping ability, which have gradually become a popular model for stock prediction. Many scholars have begun contributing insight into the application effect of neural network models, and various comparative analyses have been conducted. For example, Adebiyi et al. [15] compare and analyze the BP neural network's and ARIMA's prediction performance on New York Stock Exchange time series data. The results show that the established models using these two theories can achieve more significant performance on the stock price prediction.

Büyükşahin and Ertekin [16] propose a hybrid method of ARIMA-ANN neural network, which shows that ARIMA has better prediction accuracy in static data while ANN is more suitable for nonstationary data. Hu and Zhu [17] compare and analyze the stepwise regression with BP neural network on short-term stock price prediction. The results show that the prediction errors of these two models do not differ significantly. Besides, the model's prediction accuracy correlates with the stock denomination and fluctuation range. In contrast to statistical and machine learning methods within stock market prediction, many researchers recently utilized deep learning techniques. Al-Nefaie and Aldhyani [18] use multilayer perceptron (MLP) and LSTM models to predict fluctuations in the Saudi Stock Exchange. The results indicate that LSTM has the best model-fitting capacity among all the algorithms.

Similarly, Yadav et al. [19] propose two LSTM models using stateless or stateful models to predict the Indian stock market. The proposed model is tuned by varying the number of hidden layers. The results show that a stateless LSTM model is more suitable for time series prediction, and the network with fewer hidden layers has better prediction accuracy. Thakkar and Chaudhari [20] designed a cross-reference to an exchange-based stock trend prediction approach using LSTM to predict the stock price and movement of Wipro Limited (WIPRO) company. The usability of the proposed approach is demonstrated by the experiment with two other limited companies. Ammer and Aldhyani [21] present an LSTM algorithm that can forecast the values of four types of cryptocurrencies. The results demonstrate that the LSTM model performs better predicting all forms of cryptocurrencies than existing systems. One of the most critical issues in the field of market prediction is feature extraction from financial data, for which several solutions have been presented. In recent years, CNN has been used for automated feature selection and market forecasting. Hoseinzade and Haratizadeh [22] propose a CNN-based framework, which can collect data from various sources to extract features for predicting the future of those markets. The experiment results indicate that the proposed approach significantly improved prediction accuracy compared to the existing baseline algorithms. Alhazbi et al. [23] propose a CNN model by considering external factors such as oil prices to predict the daily movement of the Qatar Stock Exchange. The results indicate that adding external factors to the stock market data can increase the model performance. Wu et al. [24] present a graph-based CNN-LSTM model to predict the stock price with leading indicators. The experiment results show that the proposed algorithm leads to better results when compared with previous methods. Liu et al. [25] present a four-stage Central European Gas Hub model for intraday stock market forecasting. The results indicate that the proposed model could improve the forecasting performance compared with various baseline methods.

Some other researchers utilize fuzzy-based approaches for stock prediction, such as in [26, 27]. The results indicate that fuzzy-based systems can ensure interpretability and significantly improve stock profitability over traditional artificial intelligence models.

The previous existing studies overviewed in this section are summarized in Table 1. It can be seen from the table that most of the existing studies utilize deep learning methods, which may lead to the risk of overfitting due to the complexity of stock data. Besides, most existing literature uses standard performance metrics such as MSE and MAPE. Unlike these existing studies, this paper utilizes PCA to reduce the dimension of stock data. Furthermore, more performance metrics and a TOPSIS-based approach are used to evaluate the prediction model's robustness.

3. Methodology

The overall methodology of this paper is relatively straightforward. Figure 1 depicts the methodology at a high level and the flow between modules. This paper explores the hybrid model's significance for stock price prediction. This work starts with collecting stock market data used as the dataset. The dataset is then passed through the data preprocessing module, including PCA and data normalization using the max–min normalization, and the training and testing dataset is constructed. The training data serve as the prediction models' input, including the BP neural network and exponential smoothing. The BP neural network is set up by adjusting parameters such as the number of neurons and hidden layers. Then, the neural network is trained with different training algorithms. Multiple tests are performed for exponential smoothing to select the optimal damping coefficient. In the next step, six performance metrics are calculated: MSE, APE, root mean square error (RMSE), MAPE, Accuracy, and Accuracy5. To obtain the best prediction model and analyze the robustness of these models, we use TOPSIS, which ranks all the models under consideration. Finally, another stock's data are used to verify the optimal prediction model's usability.

4. Relevant Theory

4.1. PCA. PCA is a common approach used for data dimension reduction [28]. A linear transformation transforms the data into a new coordinate system. The first variance of any data projection is in the first coordinate (called the first principal component), the second variance is in the second coordinate (the second principal component), and so on. PCA is often used to reduce the dimension of a dataset while preserving the characteristics that contribute most to the variance of the dataset. This can be done by keeping the lower-order principal component and ignoring the higher-order principal component. Such lower-order components tend to retain the essential aspects of the data. The input is a dataset with m samples and n

Table 1: Comparison of the proposed approach to existing studies.

Literature	Method			Evaluation indexes					
	Algorithm	Exponential smoothing	Hybrid model	MSE	RMSE	MAPE	Accuracy	Accuracy5	TOPSIS
[14]	ANN	No	No	No	No	Yes	No	No	No
[15]	BP.	No	No	No	No	No	No	No	No
[16]	ANN	No	ARIMA-ANN	Yes	Yes	No	No	No	No
[17]	BP	No	No	Yes	Yes	No	No	No	No
[18]	MLP/LSTM	No	No	Yes	Yes	No	No	No	No
[19]	LSTM	No	No	No	Yes	No	No	No	No
[20]	LSTM	No	No	Yes	Yes	No	No	No	No
[21]	LSTM	No	No	Yes	Yes	Yes	No	No	No
[22]	CNN	No	No	No	No	No	No	No	No
[23]	CNN	No	No	No	No	No	No	No	No
[24]	CNN/LSTM	No	CNN-LSTM	No	No	No	No	No	No
[25]	CEEMD	No	CEEMD-Entropy-GRU-HA	No	Yes	No	No	No	No
Proposed approach	Yes	Yes	PCA–BP	Yes	Yes	Yes	Yes	Yes	Yes

RMSE, root mean square error.

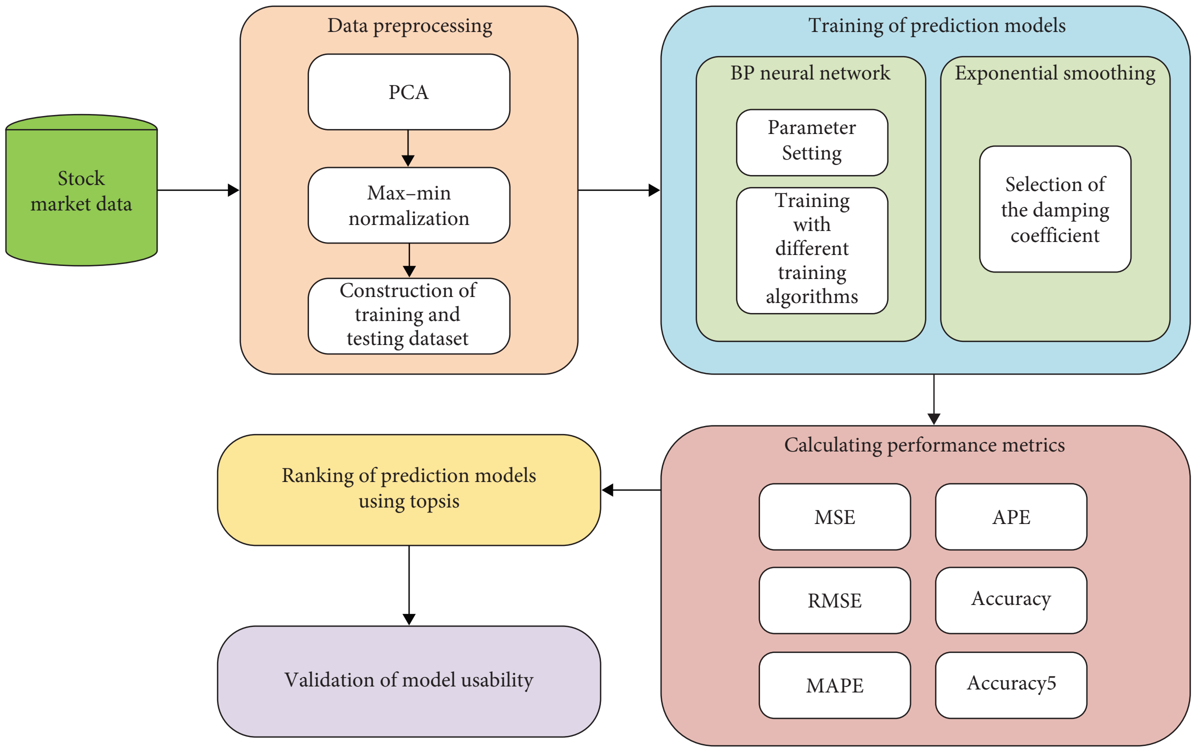

FIGURE 1: Overall methodology flow diagram of this study.

features, that is, sample data $X = |x_1, x_2, x_3, \ldots, x_m|$ Besides, the reduction to the target dimension is k. Thus, the sample data can be represented by the following matrix:

$$X = \begin{bmatrix} x_{11} & x_{12} & \ldots & x_{1n} \\ x_{21} & x_{22} & \ldots & x_{2n} \\ \ldots & \ldots & \ldots & \ldots \\ x_{m1} & x_{m2} & \ldots & x_{mn} \end{bmatrix}. \tag{1}$$

The output is the sample data after dimensionality reduction, that is, $Y = |y_1, y_2, y_3, \ldots, y_m|$.

The steps of PCA are described as follows:

(1) Decentralize the matrix to get a new matrix X, to perform zero mean normalization on each matrix column. The new matric X is represented as follows:

$$X = \begin{bmatrix} x_{11} - \overline{x}_1 & x_{12} - \overline{x}_2 & \ldots & x_{1n} - \overline{x}_n \\ x_{21} - \overline{x}_1 & x_{22} - \overline{x}_2 & \ldots & x_{2n} - \overline{x}_n \\ \ldots & \ldots & \ldots & \ldots \\ x_{m1} - \overline{x}_1 & x_{m2} - \overline{x}_2 & \ldots & x_{mn} - \overline{x}_n \end{bmatrix}. \tag{2}$$

(2) Calculate the covariance matrix of the decentralized matrix X. The covariance matrix C is obtained by using Equation (3):

$$C = \frac{1}{m-1} X^T X, \tag{3}$$

(3) Perform feature composition of the covariance matrix C to find the eigenvalue λ_k and the related feature vector v_k, that is $C_{v_k} = \lambda_k v_k$.

(4) Arrange the feature vectors in descending order according to the corresponding eigenvalues, and the first k columns are taken to form the matrix W.

(5) Calculate the sample data after k dimension reduction as described in Equation (4):

$$Y = XW. \tag{4}$$

The linear transformation can transform the sample data $x_1, x_2, x_3, \ldots, x_m$ into new synthetic variable $y_1, y_2, y_3, \ldots, y_m$, which can be represented as the following matrix:

$$\begin{cases} Y_1 = l_{11}x_1 + l_{12}x_2 + \ldots + l_{1m} \\ Y_2 = l_{21}x_1 + l_{22}x_2 + \ldots + l_{2m} \\ \qquad \ldots \\ Y_p = l_{p1}x_1 + l_{p2}x_2 + \ldots + l_{pm} \end{cases}. \tag{5}$$

The coefficient $l_i = (l_{i1}, l_{i2}, \ldots, l_{im})$ $(i = 1, 2, \ldots, m)$ is a constant vector, which must meet the following requirements:

(1) $l^2{}_{i1} + l^2{}_{i2} + \ldots + l^2{}_{im} = 1, i = 1, 2, \ldots, m$

(2) $\mathrm{cov}(Y_i, Y_j) = 0, (i \neq j, i, j = 1, 2, \ldots, m)$
(3) $\mathrm{var}(Y_1) \geq \mathrm{var}(Y_2) \ldots \geq \mathrm{var}(Y_m)$

$Y_1, Y_2, Y_3, \ldots, Y_m$ are called the principal component. The amount of information extracted from each principal component is measured by Equation (6):

$$\lambda_k / \sum_{k=1}^{p} \lambda_k. \quad (6)$$

The sum of contribution rates of the first n principal components is called cumulative contribution rate, which is calculated by Equation (7):

$$\sum_{k=1}^{n} \lambda_k / \sum_{k=1}^{m} \lambda_k. \quad (7)$$

The larger the variance contribution rate is, the stronger the ability of the corresponding principal component to reflect comprehensive information is. The principal component is generally determined if the cumulative variance contribution rate reaches 85%.

4.2. BP Neural Network. MLP network has played a significant role in developing ANNs, and it is considered an accurate model of ANNs. Its appearance has triggered an upsurge in the study of ANNs. As the original neural network, a single-layer perceptual network (M-P model) has the advantages of a transparent model, a simple structure, and a small amount of computation. However, with the deepening of the research work, people found that it still has some shortcomings, such as being unable to deal with nonlinear problems, even if the function of the computing unit does not use the valve function but other complex nonlinear functions, still can only solve the linear separable problems, cannot achieve some essential functions, thus limiting its application. The only way to enhance the classification and recognition ability of the network and solve the nonlinear problem is to adopt the multilayer feedforward network. That is, the hidden layer is added between the input layer and the output layer to form the multilayer feedforward perceptron network. In the mid-1980s, error BP training [29] was discovered to solve the connection weight learning problem of the hidden layer of a multilayer neural network and gives a complete derivation mathematically. The multilayer feedforward network which uses this algorithm for error correction is called the BP network.

As shown in Figure 2, the structure of a BP neural network generally contains three feedforward network layers: the input layer, the intermediate layer (also known as the hidden layer), and the output layer. The characteristics of the BP neural network are that each layer of neurons is only fully connected with neurons in the adjacent layer, and there is no connection between neurons in the same layer. Besides, there is no feedback connection between neurons in each layer, forming a feedforward neural network system with a hierarchical structure. BP neural network can arbitrarily complex pattern classification and excellent multidimensional function mapping. It can solve the exclusive OR and other problems that simple perceptron cannot solve. In essence, the BP algorithm takes the square of the network error as the objective function, using the gradient descent method to calculate the minimum value of the objective function.

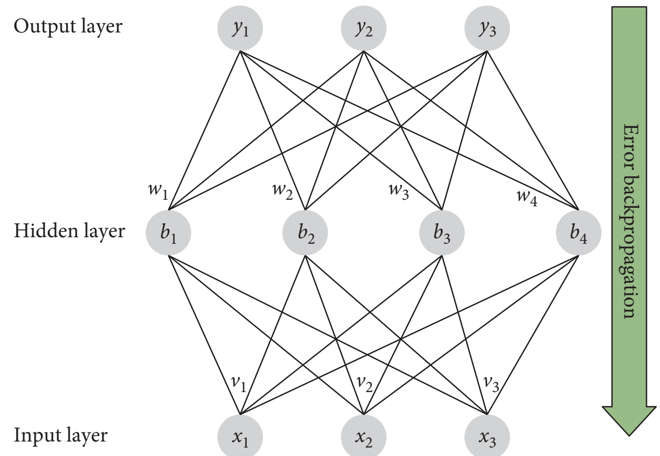

Figure 2: Structure of the BP neural network.

Three training algorithms, including LM (Levenberg–Marquardt), BR, and scaled conjugate gradient (SCG), are used in this study to train the stock market data. The LM algorithm is an iterative technique used primarily in the least squares curve fitting problem. It expresses the minimum multifunction as the sum of squares of real-value nonlinear functions. The BR algorithm can modify the mean sum of square network error to improve the network generalization ability. This algorithm is suitable for overcoming the problem of overfitting. The conjugate gradient algorithm does not require parameters but does not apply to all datasets. As a result, SCG is used since it is effective within its scope, and there is no need to set parameters. SCG can use the step size rather than the line search method in error estimation and minimize the error function.

4.3. Exponential Smoothing. Exponential smoothing is a standard method in production forecasting [30]. It is also used to forecast the middle or short-term economic development trend. Exponential smoothing is the most widely used among all the forecasting methods. Exponential smoothing is a weighted average model that uses the current state's actual value and predicted value to give different weights calculations as the predicted value of the next state. The purpose of exponential smoothing is to eliminate the irregular changes in the time series to get the general trend that reflects the changes in the time series.

The raw data sequence is presented by $\{x_t\}$ starting at time $t = 0$, and the output of the exponential smoothing is commonly written as $\{s_t\}$, which can be regarded as the best estimate of the next value of x will be. When the sequence of observations begins at time $t = 0$, the simplest form of exponential smoothing is given by Equation (8):

$$\begin{cases} s_0 = x_0 \\ s_t = ax_t + (1-a)s_{t-1}, t>0 \end{cases}, \tag{8}$$

where a is the smoothing factor, and $0<a<1$, in the market forecast, the method of determining α is generally to make a rough estimate based on experience, and the essential judgment criteria are as follows:

(1) When the time series is relatively stable, a small α value of 0.05–0.2 is selected.
(2) When the time series fluctuates, the long-term trend does not change significantly. A slightly larger α value (0.1–0.4) can be selected.
(3) When the time series fluctuates wildly, and the long-term trend changes have a significant upward or downward trend, a more considerable α value of 0.60–0.80 should be selected.
(4) When the time series is ascending or descending, the additive model is satisfied, and α takes a more significant value, 0.6–1.

This calculation process is repeated to compare the standard error of prediction under different α values and then select the optimal α value with a minor error to establish the model.

5. Experiment Results and Discussion

5.1. Performance Metrics. To perform a comprehensive judgment on the prediction ability of the prediction model, that is, the prediction accuracy, for a group of real value $x = (x_1, x_2, \ldots, x_n)$ and predicted value $\widehat{x} = (\widehat{x_1}, \widehat{x_2}, \ldots, \widehat{x_n})$, the following performance metrics are used in this study:

MSE: MSE is the average squared difference between the estimated and actual values. MSE is calculated using Equation (9):

$$\text{MSE} \frac{1}{n}\sum_{i=1}^{n}(x_i - \widehat{x_i})^2. \tag{9}$$

MSE is sensitive to outliers and varies significantly with different stock prices. Therefore, MSE cannot effectively measure the effectiveness and accuracy of the model if the data vary. As a result, RMSE is used to solve this problem by calculating the square root of MSE. The RMSE is calculated by Equation (10):

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_i - \widehat{x_i})^2}. \tag{10}$$

APE (absolute percentage error): APE is the ratio of the absolute value of the difference between the actual value x_i and the predicted value $\widehat{x}_i$ to the actual value. APE is calculated using Equation (11):

Table 2: Implementation environment of the experiment.

Component	Description
CPU	Intel(R) Core (TM) i5-8250U @ 1.60 GHz
Memory	12 G
Operating system	Windows 10 64 bit
Python	Version 3.0
SAS	Version 9.4
MATLAB	Version R 2018b
SPSS	Version 1.1.1

$$\text{APE} = |\widehat{x_i} - x_i|/x_i. \tag{11}$$

MAPE: MAPE is the average relative error APE of the n observation days. MAPE is calculated by Equation (12):

$$\text{MAPE} = \frac{1}{n}\sum_{i=1}^{n}\text{APE}. \tag{12}$$

Accuracy5: Accuracy5 is the proportion of samples with the APE within 5% of the total number of samples. Accuracy5 can be calculated by Equation (13):

$$\text{Accuracy5} = \text{count}(\text{APE} \leq 0.05)/\text{count}\,(\text{total}), \tag{13}$$

where count $(\text{APE} \leq 0.05)$ is the number of samples whose APE is within 5%, and count (total) is the total number of samples.

Accuracy: Accuracy is a comprehensive accuracy evaluation from different aspects, combining the MAPE and Accuracy5 to construct the accuracy evaluation standard, which can reflect the prediction accuracy more comprehensively. The formula for Accuracy is described in Equation (14):

$$\text{Accuracy} = 0.2 \times (1 - \text{MAPE}) + 0.8 \times \text{Accuracy5}. \tag{14}$$

TOPSIS: TOPSIS is part of the analytical multicriteria decision-making technique. The basic idea of this approach is to find a feasible scheme that is the closest to the ideal solution and the further to the negative ideal solution. TOPSIS finds the optimal and worst targets among multiple targets through the original data matrix's normalization. The clustering of an evaluation target, an ideal solution, and a negative ideal solution is calculated to obtain the degree of closeness between each target and the ideal solution. The degree of closeness between each target and the ideal solution is sorted in descending according to the degree of closeness of the ideal solution, which is used as the basis for evaluating the quality of the target. The closeness value ranges from 0 to 1, and the closer the value is to 1, the closer the corresponding evaluation target is to the optimal level. On the contrary, the closer the value is to 0, the closer the evaluation target is to the worst level.

5.2. Experiment Setup and Dataset. As shown in Table 2, all experiments in this paper are performed in a system with an

Table 3: Parameter descriptions of the dataset.

Variable	Variable name	Description
X_1	Opening price	The first transaction price in the day's transaction
X_2	Highest price	The highest unit price
X_3	Lowest price	The lowest unit price in the day's transaction
X_4	Closing price	The last transaction unit price in the day's transaction
X_5	Change amount	The closing price of the day compared with the previous day
X_6	Change rate (%)	Change amount/yesterday's closing price × 100
X_7	Turnover rate (%)	The frequency of stock buying and selling in the market
X_8	Volume	The number of shares traded, using the lot as a unit (1 lot = 100 shares)
X_9	Turnover	The amount of shares traded
X_{10}	Total market value	The total number of share capital is multiplied by the day's closing price
X_{11}	Circulation market value	The total value of tradable shares is multiplied by the closing price

Figure 3: Change curve of the stock opening price of China Merchants Bank.

Intel(R) Core (TM) i5-8250U @ 1.60 GHz processor, 12 GB memory, and a Windows 10 64 bit operating system. The implementation and evaluations of all prediction models are conducted using SAS, MATLAB, and SPSS.

The data used for the experiment are China Merchants Bank (600036) stock data from 2015 to 2021, obtained from the iFinD financial data terminal. This dataset involves various technical indicators, such as opening price, highest price, lowest price, closing price, change amount, change rate, etc. The parameters of the raw stock data are listed in Table 3.

The opening price of China Merchants Bank is plotted in Figure 3. The opening price rises with volatility, and the amplitude is significant in some periods, leading to tremendous challenges in predicting the short-term stock price trend.

PCA is performed on the stock market data, and the eigenvalues of the correlation coefficient matrix are presented in Table 4.

Table 5 presents specific feature vectors for each principal component, and the results reveal that the cumulative variance contribution of the first three principal components has reached the cumulative contribution rate of 90.31%. The following principal component score expression can be obtained (where $\widetilde{X}_i$ represents the normalized value of the variables i) according to the first three feature values of the correlation coefficient matrix. As a result, these principal components are used as input parameters of the BP neural network to simplify the prediction model.

$$\begin{aligned}F_1 = {} & 0.40\widetilde{X}_1 + 0.41\widetilde{X}_2 + 0.40\widetilde{X}_3 + 0.41\widetilde{X}_4 \\ & + 0.37\widetilde{X}_5 - 0.20\widetilde{X}_6 + 0.40\widetilde{X}_7 + 0.03\widetilde{X}_8 \\ & + 0.02\widetilde{X}_9 + 0.00\widetilde{X}_{10} + 0.09\widetilde{X}_{11}.\end{aligned} \tag{15}$$

$$\begin{aligned}F_2 = {} & -0.06\widetilde{X}_1 + 0.04\widetilde{X}_2 - 0.10\widetilde{X}_3 + 0.01\widetilde{X}_4 \\ & + 0.16\widetilde{X}_5 + 0.43\widetilde{X}_6 + 0.09\widetilde{X}_7 + 0.64\widetilde{X}_8 \\ & + 0.60\widetilde{X}_9 + 0.00\widetilde{X}_{10} + 0.00\widetilde{X}_{11}.\end{aligned} \tag{16}$$

$$\begin{aligned}F_3 = {} & -0.04\widetilde{X}_1 - 0.04\widetilde{X}_2 - 0.05\widetilde{X}_3 - 0.02\widetilde{X}_4 \\ & + 0.01\widetilde{X}_5 + 0.09\widetilde{X}_6 - 0.03\widetilde{X}_7 - 0.02\widetilde{X}_8 \\ & - 0.04\widetilde{X}_9 + 0.00\widetilde{X}_{10} + 0.99\widetilde{X}_{11}.\end{aligned} \tag{17}$$

Finding a model that correctly predicts the output from new input data is one of the critical objectives in machine learning. However, staying away from overfitting and model complexity is also crucial. A model with a high level of

Table 4: Eigenvalues of the correlation matrix.

	Eigenvalues	Difference	Scale
1	5.9439	3.8223	0.5944
2	2.1217	1.1563	0.2122
3	0.9654	0.3121	0.0965
4	0.6532	0.4369	0.0653
5	0.2163	0.1501	0.0216
6	0.0666	0.0520	0.0066
7	0.0141	0.0029	0.0014
8	0.0112	0.0032	0.0011
9	0.0080	0.0080	0.0008
10	0.0000	0.0000	0.0000
11	0.0000	0.0000	0.0000

Table 5: Specific feature vectors for each principal component.

	Prin1	Prin2	Prin3	Prin4	Prin5	Prin6	Prin7	Prin8	Prin9	Prin10	Prin11
X_1	0.403776	−0.06343	−0.04194	−0.071962	−0.030899	0.417095	0.088818	−0.321775	0.734611	0.000000	0.000000
X_2	0.405173	0.043595	−0.039633	−0.008585	0.101709	0.324936	−0.694955	0.469851	−0.112426	−0.000005	0.000000
X_3	0.402193	−0.099185	−0.045376	−0.016941	−0.11609	0.201311	0.685303	0.506562	−0.214031	0.000002	0.000000
X_4	0.408378	0.009652	−0.024476	0.081417	0.026283	−0.046728	−0.014065	−0.438702	−0.379875	0.697187	0.000000
X_5	0.365228	0.161454	0.011546	0.412207	0.191979	−0.687919	0.008424	0.207207	0.342636	0.000001	0.000000
X_6	−0.195147	0.429207	0.086416	0.742412	0.110408	0.43842	0.10275	−0.038111	−0.051416	0.000000	0.000000
X_7	0.404862	0.093624	−0.026703	0.056164	−0.070932	−0.057712	−0.021065	−0.421773	−0.362884	−0.710683	0.000000
X_8	0.032018	0.635394	−0.020326	−0.178997	−0.730224	−0.089609	−0.054917	0.064928	0.073813	0.094127	0.000000
X_9	0.017654	0.601136	−0.044516	−0.471726	0.622271	0.02987	0.158782	−0.008756	−0.020369	0.000000	0.000000
X_{10}	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	1.000000
X_{11}	0.086860	−0.001993	0.991617	−0.094984	−0.002705	0.006439	0.003375	0.008009	−0.001276	0.000000	0.000000

complexity could be able to capture more data variations, but it will also be more challenging to train and might be more prone to overfitting. In contrast, a model with a low level of complexity could be simpler to train but might not be able to extract all the pertinent information from the data. In order to avoid overfitting, it is crucial to find the ideal balance between model complexity and overfitting while creating machine learning models.

In order to find the optimal configuration structure for the BP neural network, a comprehensive experiment is performed by varying the number of neurons in the hidden layer, learning rates, and activation functions. Experiments are performed many times for every parameter configuration used for training, with the average results being recorded to investigate the random factor for initializing the weights of the BP neural network. Besides, bias in training is minimized using a fourfold cross-validation approach for each configuration across all tests. This allows the model to be trained on different data and prevents it from being overfitted to a particular dataset. For this experiment, we divide the original dataset into four pieces of equal size (375 instances in every subset). Each test round uses 75% of the data for training and 25% for testing using the predefined arrangement.

The following test evaluates each model to determine the best structure of the BP neural network. The configuration chosen and the related performance evaluated by RMSE are presented in Table 6. We set the maximum number of epochs to 100 for training the BP model in this test. The best network structure consists of 3 inputs, 10 neurons in the hidden layer, and 1 output. We apply the Sigmoid activation function with a learning rate 0.1 and the LM algorithm for training.

5.3. Comparison and Analysis of BP Neural Network with Different Training Algorithms. Figures 4–6 plot the difference between predicted values and actual values of the BP_LM, BP_BR, and BP_SCG, respectively. These figures indicate that the nonlinear fitting ability of the BP_LM and BP_BR models is excellent, which can reflect subtle local changes in the stock price and the overall trend. The BP_SCG model has poor prediction ability in the overall trend but a strong ability of the local changes prediction in the stock prices. On the other hand, this model is more suitable for feature classification, especially in predicting the rise and fall of stock prices.

The evaluation results are compared and summarized in Table 7. The BP_SCG model has the poorest prediction performance. The BP_LM model slightly outperforms the BP_BR model. The MSE is 1.7650, which shows the accuracy and effectiveness of the model. The Accuracy5 and Accuracy values are 94.33% and 95.07%, which means that the prediction accuracy is high while maintaining model stability.

Table 6: Parameter configuration for the BP neural network.

Number of neurons in hidden layer	Learning rate	Activation function	Experiment ID	Average	Experiment average
5	0.1	Sigmoid	1	1.6478	
5	0.1	Sigmoid	2	1.8413	1.6143
5	0.1	Sigmoid	3	1.5479	
5	0.1	Sigmoid	4	1.4202	
5	0.1	Sigmoid	1	1.3254	
5	0.1	Sigmoid	2	1.5648	1.4361
5	0.1	Sigmoid	3	1.7411	
5	0.1	Sigmoid	4	1.1131	
10	0.1	Sigmoid	1	1.1910	
10	0.1	Sigmoid	2	1.5585	**1.3285**
10	0.1	Sigmoid	3	1.4521	
10	0.1	Sigmoid	4	1.1124	
15	0.1	Sigmoid	1	1.6425	
15	0.1	Sigmoid	2	1.5788	1.5435
15	0.1	Sigmoid	3	1.4726	
15	0.1	Sigmoid	4	1.4801	

Bold values signify the best results.

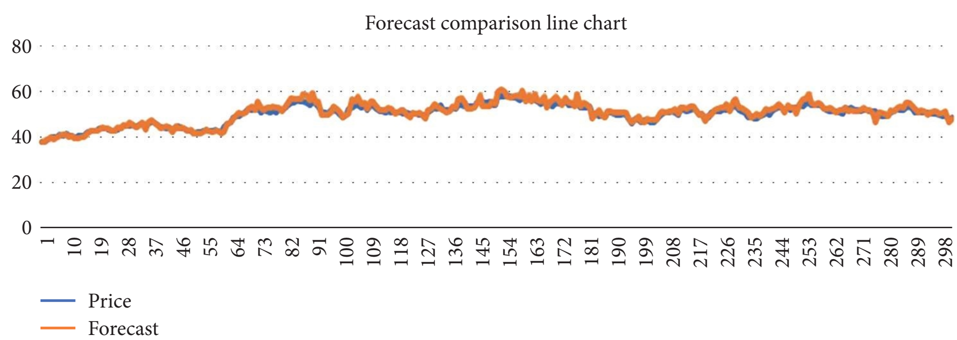

Figure 4: Comparison of predicted and actual values of BP_LM.

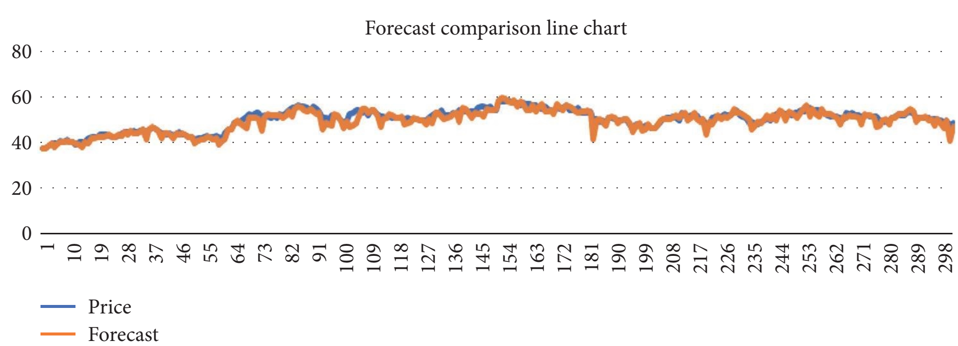

Figure 5: Comparison of predicted and actual values of BP_BR.

5.4. Comparison and Analysis of PCA–BP Neural Network with Different Training Algorithms. This subsection compares and analyzes the performance of the PCA–BP neural network with different training algorithms. Figures 7–9 plot the difference between predicted values with the actual values of the PCA–BP_LM, PCA–BP_BR, and PCA–BP_SCG, respectively. These figures show that the nonlinear fitting ability of the PCA–BP_LM and PCA–BP_BR models is excellent, and the PCA–BP_SCG model has the poorest prediction ability.

Table 8 compares and analyzes these three models in various performance metrics. The results indicate that the performance of these three models can be sorted as follows: PCA–BP_BR > PCA–BP_LM > PCA–BP_SCG. The PCA–BP_LM model has the best performance among the

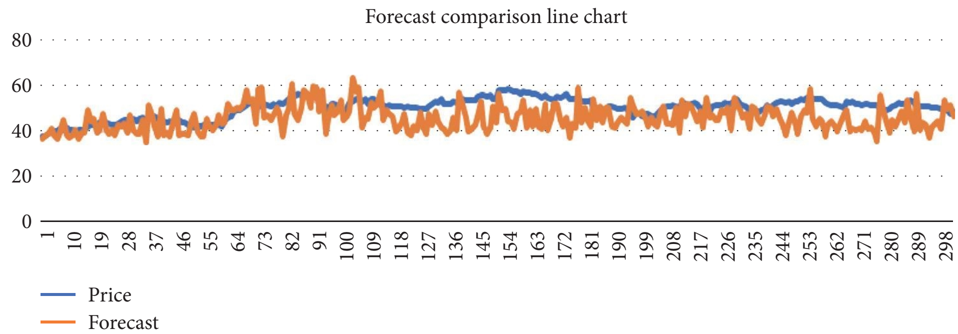

Figure 6: Comparison of predicted and actual values of BP_SCG.

Table 7: Summary of BP neural network with different training algorithms.

Performance index	BP_LM	BP_BR	BP_SCG
MSE	**1.7650**	2.8216	50.4798
RMSE	**1.3285**	1.6798	7.1049
MAPE	**0.0197**	0.0226	0.1142
Accuracy5	**0.9433**	0.8967	0.2233
Accuracy	**0.9507**	0.9128	0.3558

RMSE, root mean square error. Bold values signify the best results.

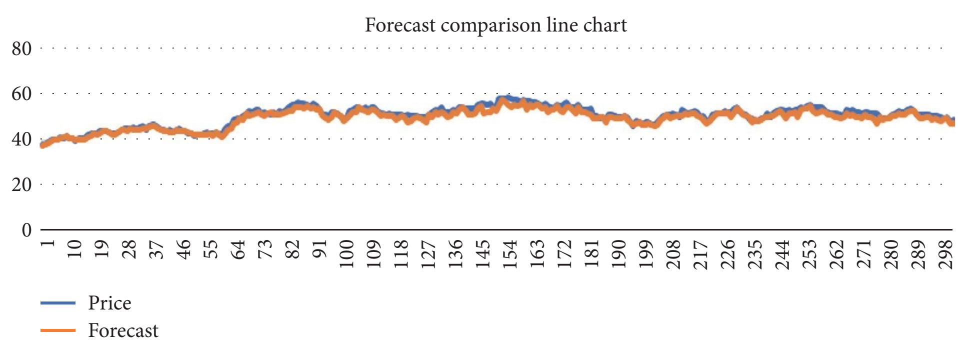

Figure 7: Predicted and actual values of PCA-BP_LM.

three models. The PCA–BP_BR model improves the prediction accuracy of MSE and MAPE by 75.8% and 34%, respectively, compared to the BP_LM model.

5.5. Comparison and Analysis of Exponential Smoothing with Different Smoothing Factors. For the exponential smoothing model, selecting the damping coefficient α is essential. The damping coefficient reflects the response speed of the model to time series changes and determines the ability to smooth random errors in prediction. In this experiment, we vary the smoothing factor α with 0.3, 0.6, and 0.9, respectively. The evaluation results of the exponential smoothing model with three different smoothing factors are represented in Table 9.

The exponential smoothing model obtains the best performance when the smoothing factor α is set to 0.3. The MAPE is 0.0154, indicating that the relative error is low on the dataset. The Accuracy5 and Accuracy are 98.67% and 98.63%, respectively, demonstrating that the prediction model performs well in both prediction ability and stability. The summary of all prediction models is represented in Table 10.

We can rank these models in terms of MSE and Accuracy. The ranking ordered by MSE is arranged as follows:

$$\begin{aligned} &M(\mathrm{PCA}-\mathrm{BP\_BR}) > M(\alpha = 0.3) > M(\alpha = 0.6) \\ &\quad > M(\mathrm{BP\_LM}) > M(\mathrm{PCA}-\mathrm{BP_{LM}}) > M(\mathrm{BP\_BR}) \\ &\quad > M(\alpha = 0.9) > M(\mathrm{BP\_SCG}) > M(\mathrm{PCA}-\mathrm{BP\_SCG}). \end{aligned} \tag{18}$$

The ranking order by Accuracy is arranged as follows:

$$\begin{aligned} &A(\mathrm{PCA}-\mathrm{BP\_BR}) > A(\alpha = 0.3) > A(\alpha = 0.6) \\ &\quad > A(\mathrm{BP\_LM}) > A(\mathrm{PCA}-\mathrm{BP\_LM}) > A(\mathrm{BP\_BR}) \\ &\quad > A(\alpha = 0.9) > A(\mathrm{BP\_SCG}) > A(\mathrm{PCA}-\mathrm{BP\_SCG}). \end{aligned} \tag{19}$$

The PCA–BP_BR is the top among these models in both MSE and Accuracy, as the BR algorithm can effectively solve the problem of data overfitting. The two BP neural network models with the SCG algorithm have the worst prediction,

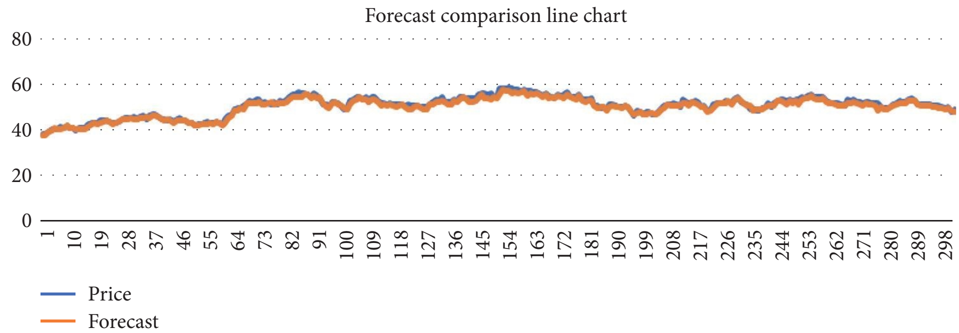

FIGURE 8: Predicted and actual values of PCA-BP_BR.

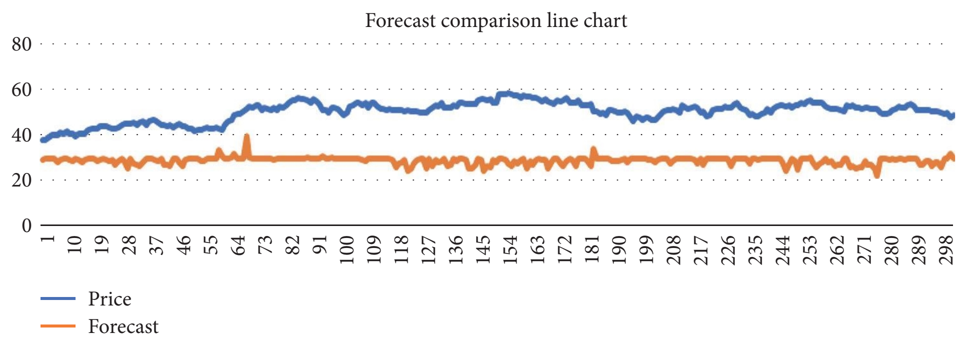

FIGURE 9: Predicted and actual values of PCA-BP_SCG.

TABLE 8: Summary of PCA-BP neural network with different training algorithms.

Performance index	PCA-BP_LM	PCA-BP_BR	PCA-BP_SCG
MSE	1.8505	**0.6835**	465.9897
RMSE	1.3603	**0.8267**	21.5868
MAPE	0.0204	**0.0130**	0.4137
Accuracy5	0.9433	**0.9967**	0.0000
Accuracy	0.9506	**0.9947**	0.1173

RMSE, root mean square error. Bold values signify the best results.

TABLE 9: Summary of the exponential smoothing with different smoothing factors.

Performance index	$\alpha = \mathbf{0.3}$	$\alpha = \mathbf{0.6}$	$\alpha = \mathbf{0.9}$
MSE	**0.9742**	1.3598	4.0850
RMSE	**0.9870**	1.1661	2.0211
MAPE	**0.0154**	0.0187	0.0336
Accuracy5	**0.9867**	0.9700	0.7867
Accuracy	**0.9863**	0.9723	0.8226

RMSE, root mean square error. Bold values signify the best results.

which is unsuitable for stock price prediction. The exponential smoothing model achieves better prediction performance with a smaller damping coefficient. Table 11 compares the proposed model with some existing studies overviewed in Section 2, and it is observed that the proposed model outperforms existing systems under different performance indexes.

5.6. Model Usability Evaluation. The PCA–BP_BR is selected as the optimal model with the best prediction performance according to the experiment results. In this subsection, another stock, "Wanxiang Denong (600371)," is selected to validate the usability of the optimal model. The process is repeated, including data preprocessing, training, and performance evaluation. The evaluation results are summarized in Table 12.

We can rank these models in terms of MSE and Accuracy. The ranking ordered by MSE is arranged as follows:

Table 10: Summary of all prediction models for China Merchants Bank.

Performance index	PCA_BP_LM	PCA_BP_BR	PCA_BP_SCG	BP_LM	BP_BR	BP_SCG	$\alpha = \mathbf{0.3}$	$\alpha = \mathbf{0.6}$	$\alpha = \mathbf{0.9}$
MSE	1.8505	**0.6835**	465.9897	1.7650	2.8216	50.4798	0.9742	1.3598	4.0850
RMSE	1.3603	**0.8267**	21.5868	1.3285	1.6798	7.1049	0.9870	1.1661	2.0211
MAPE	0.0204	**0.0130**	0.4137	0.0197	0.0226	0.1142	0.0154	0.0187	0.0336
Accuracy5	0.9433	**0.9967**	0.0000	0.9433	0.8967	0.2233	0.9867	0.9700	0.7867
Accuracy	**0.9506**	0.9947	0.1173	0.9507	0.9128	0.3558	0.9863	0.9723	0.8226

RMSE, root mean square error. Bold values signify the best results.

Table 11: Comparison of the proposed model with existing studies for predicting the stock of China Merchants Bank.

References	MSE	RMSE	MAPE	Accuracy	Accuracy5
[16]	3.0755	1.7531	0.0532	0.7426	0.7865
[17]	2.7159	1.6480	0.0421	0.7952	0.7125
[18]	2.0156	1.4197	0.0315	0.8236	0.8658
[19]	1.1435	1.0693	0.0216	0.9036	0.9385
[20]	1.5216	1.2335	0.0246	0.9012	0.9123
[21]	0.9846	0.9922	0.0189	0.9423	0.9687
[22]	0.9712	0.9855	0.0213	0.9312	0.9568
[23]	1.1256	1.0609	0.0289	0.9216	0.9487
Proposed model	0.6835	0.8267	0.0130	0.9967	0.9947

RMSE, root mean square error.

Table 12: Summary of all prediction models for Wanxiang Denong.

Performance index	PCA_BP_LM	PCA_BP_BR	PCA_BP_SCG	BP_LM	BP_BR	BP_SCG	$\alpha = \mathbf{0.3}$	$\alpha = \mathbf{0.6}$	$\alpha = \mathbf{0.9}$
MSE	0.0764	**0.0698**	1.6857	0.3494	0.0880	7.1489	0.1324	0.1744	0.4644
RMSE	0.2764	**0.2642**	1.2983	0.5911	0.2966	2.6737	1.6351	0.4176	0.6815
Mape	0.0181	**0.0169**	0.0952	0.0333	0.0201	0.2112	0.0190	0.0219	0.0375
Accuracy5	0.9833	**0.9900**	0.2267	0.8033	0.9467	0.0533	0.9367	0.9133	0.7500
Accuracy	0.9830	**0.9886**	0.3623	0.8360	0.9533	0.2004	0.9455	0.9263	0.7925

RMSE, root mean square error. Bold values signify the best results.

$$\begin{aligned} &M(PCA-BP_{BR}) > M(PCA-BP_{LM}) > M(BP_{BR}) \\ &> M(\alpha = 0.3) > M(\alpha = 0.6) > M(BP\_LM) \\ &> M(\alpha = 0.9) > M(PCA - BP\_SCG) > M(BP\_SCG). \end{aligned} \tag{20}$$

The ranking order by Accuracy is arranged as follows:

$$\begin{aligned} &A(PCA - BP\_BR) > A(PCA - BP\_LM) \\ &> A(BP\_BR) > A(\alpha = 0.3) > A(\alpha = 0.6) \\ &> A(BP\_LM) > A(\alpha = 0.9) > A(PCA - BP\_SCG) > A(BP\_SCG). \end{aligned} \tag{21}$$

In this experiment, similar results are obtained; the PCA–BP_BR model has the best prediction performance while the BP_SCG has the worst. These results prove the usability and efficiency of the proposed model.

5.7. TOPSIS Evaluation. This subsection evaluates different prediction models using TOPSIS, and the results are given in Table 13. It can be seen from the table that the PCA–BP_BR is the most robust model, followed by exponential smoothing with a smoothing factor of 0.3. Besides, the exponential smoothing model is more robust than BP_SCG and PCA–BP_SCG. Finally, the advantages and disadvantages of each model tested in this study are discussed in Table 14.

6. Conclusion and Future Research Directions

This paper proposes a hybrid PCA–BP neural network model to predict stock prices in the Chinese stock market. A comprehensive experiment is performed to compare and analyze the model performance by using different training algorithms. TOPSIS has been executed to validate the robustness of all prediction models. An exponential smoothing model is also tested and compared with the proposed model. The following conclusions can be obtained from the experiment results:

(1) The hybrid model performs better than a single model, improves prediction accuracy and operation efficiency, and reduces prediction error.

Table 13: TOPSIS evaluation of different models.

Prediction model	Positive ideal solution distance ($D+$)	Negative ideal solution distance ($D-$)	Comprehensive score index	The ranking of models
PCA-BP_LM	0.0162	0.3718	0.9582	5
PCA-BP_BR	0.0000	0.3859	1.0000	1
PCA-BP_SCG	0.3859	0.0000	0.0000	9
BP_LM	0.0161	0.3720	0.9584	4
BP_BR	0.0300	0.3606	0.9233	6
BP_SCG	0.2351	0.2073	0.4687	8
$\alpha = 0.3$	0.0032	0.3830	0.9918	2
$\alpha = 0.6$	0.0084	0.3784	0.9784	3
$\alpha = 0.9$	0.0630	0.3337	0.8413	7

Table 14: Summary of different models.

Property	PCA_BP_LM BP_LM	PCA_BP_BR BP_BR	PCA_BP_SCG BP_SCG	Exponential smoothing $\boldsymbol{\alpha = 0.3}$, $\boldsymbol{\alpha = 0.6}$, $\boldsymbol{\alpha = 0.9}$
Advantage	Fast learning rate	The training requires less data and effectively solves the problem of data overfitting	Beneficial in large-scale problems, with a fast convergence rate and small iterative computation amount	The storage requirements of the data are low, and the calculation is simple. The only need to select the appropriate damping coefficient according to the characteristics of the data will have good prediction results without training
Disadvantage	High memory consumption	Require relatively longer training time than other algorithms	More generations are required in training	Do not consider the characteristic relationship between variables, but only approximations of mathematical expressions
Summary	Good prediction performance can be achieved without dimensionality reduction	The prediction accuracy is better with the PCA method	The data type, size, or parameter setting can cause poor prediction performance. Overall, this algorithm is not suitable for stock price prediction	The algorithm does not consider that the stock price is affected by multiple factors, and the prediction effect is excellent and convenient. Besides, the stock price prediction performs well when selecting an appropriate smoothing factor

(2) The PCA–BP model with the BR training algorithm has the best prediction accuracy.
(3) The selection of the damping coefficient is tested in many rounds. The results indicate that the exponential smoothing approach has good prediction performance in time series prediction, exceeding some neural network models.

The novelty of this study is compared with some previously published papers in the same subject area. However, this study has some limitations. While the PCA–BP model, as observed, has provided a quite good prediction ability, it would be interesting to find out which of these algorithms is more accurate for stock price prediction. In future work, the results obtained in this study will be applied in a production environment with a more extensive dataset. Besides, deep learning algorithms instead of BP neural networks will be used for stock price prediction.

Data Availability

All data or codes used to support the findings of this study are available from the corresponding author.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Authors' Contributions

LH designed the research and wrote the original manuscript, DL managed the data and conducted the empirical analysis, and FX provided guidance and revised the manuscript.

Acknowledgments

This research was supported by the Shanghai Chenguang Plan (under grant number 21CGB08) and the Yang Fan Project of the Shanghai Science and Technology Committee (under grant number 23YF1431200).

References

[1] F. Nie and M. Bagheri, "A comparison of investors' protection in different securities holding systems and the legal implications of direct and indirect holding: a focus on China's central securities depository legal position," *Capital Markets Law Journal*, vol. 16, no. 2, pp. 187–202, 2021.

[2] B. Zhu and J. Chevallier, "Carbon price forecasting with a hybrid ARIMA and least squares support vector machines methodology," in *Pricing and Forecasting Carbon Markets*, pp. 87–107, Springer, Cham, 2017.

[3] D. Aggarwal, S. Chandrasekaran, and B. Annamalai, "A complete empirical ensemble mode decomposition and support vector machine-based approach to predict Bitcoin prices," *Journal of Behavioral and Experimental Finance*, vol. 27, Article ID 100335, 2020.

[4] R. Ren, D. D. Wu, and T. Liu, "Forecasting stock market movement direction using sentiment analysis and support vector machine," *IEEE Systems Journal*, vol. 13, no. 1, pp. 760–770, 2019.

[5] W. Ting, X. Yang-yu-xin, and C. Tie-ming, "Short-term trend forecasting of stocks based on multi-category feature system," *Computer Science*, vol. 47, no. 11A, pp. 491–495, 2020.

[6] U. Ugurlu, I. Oksuz, and O. Tas, "Electricity price forecasting using recurrent neural networks," *Energies*, vol. 11, no. 5, Article ID 1255, 2018.

[7] A. S. Saud and S. Shakya, "Analysis of look back period for stock price prediction with RNN variants: a case study on banking sector of NEPSE," *Procedia Computer Science*, vol. 167, pp. 788–798, 2020.

[8] X. Pang, Y. Zhou, P. Wang, W. Lin, and V. Chang, "An innovative neural network approach for stock market prediction," *The Journal of Supercomputing*, vol. 76, pp. 2098–2118, 2020.

[9] J. Cao and J. Wang, "Stock price forecasting model based on modified convolution neural network and financial time series analysis," *International Journal of Communication Systems*, vol. 32, no. 12, Article ID e3987, 2019.

[10] H. Zhao and L. Xue, "Research on stock forecasting based on LSTM-CNN-CBAM model," *Computer Engineering and Applications*, vol. 57, no. 3, pp. 203–207, 2021.

[11] Y. Ma, R. Mao, Q. Lin, P. Wu, and E. Cambria, "Multi-source aggregated classification for stock price movement prediction," *Information Fusion*, vol. 91, pp. 515–528, 2023.

[12] Y. A. Arabyat, A. A. AlZubi, D. M. Aldebei, and S. A. Z. Al-oqaily, "An efficient method for pricing analysis based on neural networks," *Risks*, vol. 10, no. 8, p. 151, 2022.

[13] A. S. Weigend, *Time Series Prediction: Forecasting the Future and Understanding the Past*, Routledge, 1st edition, 1994.

[14] J. L. Ticknor, "A Bayesian regularized artificial neural network for stock market forecasting," *Expert Systems with Applications*, vol. 40, no. 14, pp. 5501–5506, 2013.

[15] A. A. Adebiyi, A. O. Adewumi, and C. K. Ayo, "Comparison of ARIMA and artificial neural network models for stock price prediction," *Journal of Applied Mathematics*, vol. 2014, Article ID 614342, 7 pages, 2014.

[16] Ü. Ç. Büyükşahin and Ş. Ertekin, "Improving forecasting accuracy of time series data using a new ARIMA-ANN hybrid method and empirical mode decomposition," *Neurocomputing*, vol. 361, pp. 151–163, 2019.

[17] L. Y. Hu and J. M. Zhu, "Comparative analysis of stock price prediction based on stepwise regression and BP neural network," *Journal of Liaoning University of Technology (Natural Science Edition)*, vol. 39, no. 3, pp. 201–205, 2019.

[18] A. H. Al-Nefaie and T. H. H. Aldhyani, "Predicting close price in emerging Saudi Stock Exchange: time series models," *Electronics*, vol. 11, no. 21, p. 3443, 2022.

[19] 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.

[20] A. Thakkar and K. Chaudhari, "CREST: cross-reference to exchange-based stock trend prediction using long short-term memory," *Procedia Computer Science*, vol. 167, pp. 616–625, 2020.

[21] M. A. Ammer and T. H. H. Aldhyani, "Deep learning algorithm to predict cryptocurrency fluctuation prices: increasing investment awareness," *Electronics*, vol. 11, no. 15, Article ID 2349, 2022.

[22] E. Hoseinzade and S. Haratizadeh, "CNNpred: CNN-based stock market prediction using a diverse set of variables," *Expert Systems with Applications*, vol. 129, pp. 273–285, 2019.

[23] S. Alhazbi, A. B. Said, and A. Al-Maadid, "Using deep learning to predict stock movements direction in emerging markets: the case of Qatar stock exchange," in *2020 IEEE International Conference on Informatics, IoT, and Enabling Technologies (ICIoT)*, pp. 440–444, IEEE, Doha, Qatar, 2020.

[24] J. M.-T. Wu, Z. Li, N. Herencsar, B. Vo, and J. C.-W. Lin, "A graph-based CNN-LSTM stock price prediction algorithm with leading indicators," *Multimedia Systems*, vol. 29, pp. 1751–1770, 2023.

[25] Y. Liu, X. Liu, Y. Zhang, and S. Li, "CEGH: a hybrid model using CEEMD, entropy, GRU, and history attention for intraday stock market forecasting," *Entropy*, vol. 25, no. 1, Article ID 71, 2023.

[26] W. Wang, W. Lin, Y. Wen et al., "An interpretable intuitionistic fuzzy inference model for stock prediction," *Expert Systems with Applications*, vol. 213 Part A, Article ID 118908, 2023.

[27] M.-E. Wu, J.-H. Syu, J. C.-W. Lin, and J.-M. Ho, "Effective fuzzy system for qualifying the characteristics of stocks by random trading," *IEEE Transactions on Fuzzy Systems*, vol. 30, no. 8, pp. 3152–3165, 2022.

[28] M. Waqar, H. Dawood, P. Guo, M. B. Shahnawaz, and M. A. Ghazanfar, "Prediction of stock market by principal component analysis," in *2017 13th International Conference on Computational Intelligence and Security (CIS)*, pp. 599–602, IEEE, Hong Kong, China, 2017.

[29] I. Goodfellow, Y. Bengio, and A. Courville, "6.5 back-propagation and other differentiation algorithms," in *Deep Learning*, pp. 200–220, MIT Press, 2016.

[30] E. S. Gardner Jr., "Exponential smoothing: the state of the art," *Journal of Forecasting*, vol. 4, no. 1, pp. 1–28, 1985.