



Research article

Stock price forecasting based on Hausdorff fractional grey model with convolution and neural network

Wenhua Dong and Chunna Zhao*

School of information science and engineering, Yunnan University, Kunming 650500, China

* **Correspondence:** Email: zhaochunna@ynu.edu.cn.

Abstract: Forecast of stock prices can guide investors' investment decisions. Due to the high-dimensional and long-memory characteristics of stock data, it is difficult to predict. The fractional grey model with convolution (FGMC (1, m)) can be used to predict time series, because of its memory and ability to process high-dimensional data. However, the FGMC (1, m) model has some disadvantages, including complex calculation, loss of information, and approximate background values. In this paper, Hausdorff fractional derivative and Newton-Cotes formula are used to optimize these shortcomings and can get a Hausdorff fractional grey model with convolution (HFGMC (1, m)) model. The HFGMC (1, m)-LLE-BP model is proposed in this paper. HFGMC (1, m) provides a solution that can reduce the complexity of the cumulative generator matrix calculation and preserve the global information of the sequence. Newton-Cotes formula is used to calculate the background value, which can solve the shortcomings of approximate background values. The HFGMC (1, m) model is used to predict the linear component of the sequence, and the BP neural network is used to predict the nonlinear component of the sequence. In addition, because of the high-dimensional and nonlinear characteristics of stock data, a local linear embedding (LLE) algorithm is used to remove redundant information in high-dimensional non-linear data. The experimental results show that the HFGMC (1, m)-LLE-BP model is effective for predicting the stock price in different trends.

Keywords: stock price forecasting; FGMC (1, m); Hausdorff fractional derivative; Newton-Cotes formula; LLE algorithm; BP neural network

1. Introduction

The stock price is the primary factor affecting investors. Short-term stock price forecasts have

important significance for reference to investors. The factors affecting stock price are very complex, which makes it difficult to analyze and predict stock price. The stock market can be seen as a nonlinear dynamic system under the action of multiple factors. Time series forecasting models are usually adopted for stock price forecasts. There are multiple regression models, exponential smoothing models, ARIMA models, neural network models, combined forecasting models, and grey forecasting models.

Many scholars have conducted a series of studies on stock price prediction. Shi [1] used ARMA and BP neural networks to obtain the linear and non-linear components of the stock price sequence and then used the Markov model to correct the error of the result. Vijn [2] used the random forest algorithm to integrate the indicators of multiple stocks, and then used BP neural network to output the predicted observations. Liu [3] proposes a new effective metric, Mean Profit Rate (MPR). The effectiveness of metrics is measured based on the correlation between the metric value and profit of the model. And, the relationship between the features of Bitcoin and the next day change in the price of Bitcoin using an Artificial Neural Network ensemble approach called Genetic Algorithm based Selective Neural Network Ensemble [4] is explored by Sin. Zhu [5] proposes an intelligent trading system using support vector regression optimized by genetic algorithms (SVR-GA) and multilayer perceptron optimized with GA (MLP-GA). And, an improved one-step ahead forecasting system and an improved multi-step ahead prediction [6] are proposed by Dong. Zhang [7] uses BP neural network to train stock prices in the form of spline division, and then output the predicted observations with data independent of the training samples. Teo [8] train the wavelet packet multi-layer perceptron neural network (WP-MLP) by backpropagation for time series prediction. Summarizing the above literature, using deep learning and machine learning methods to predict stock prices can achieve good prediction results. However, most methods do not consider the memory of the sequence when processing the original price sequence. Sang [9] studied the characteristics of long-term memory of fluctuations in the Chinese stock market, which means that the historical data of the stock market have an impact on the stock price at the current time. Since fractional calculus has power-law memory [10], the idea of fractional calculus can be used to obtain information about historical data. Based on the characteristics of fractional calculus, some models that combined the idea of fractional calculus with traditional time series forecasting methods have attracted the attention of many scholars. For example, the ARFIMA model, fractional grey model, etc. are the models that combine fractional ideas. Because the grey model has good performance in the uncertainty system, the prediction of the stock price is very consistent with the grey model.

The Grey model is an important part of the grey theory, and the most famous one is GM (1, 1) model. However, GM (1, 1) has many shortcomings. The background value is based on approximation, and it cannot handle high-dimensional data. And the ability of nonlinear fitting is insufficient. Because of the shortcomings, many scholars had improved the GM (1, 1) model [11–25]. Tien [13] proposed the GMC (1, m) model to solve the shortcomings of traditional GM (1, 1) that cannot handle high-dimensional data. Literature [14–19] proposed a variety of grey Bernoulli prediction models and grey models combined with kernel methods, which solved the shortcomings of insufficient nonlinear fitting capabilities. Literature [20,21] had solved the shortcoming that the background value is based on approximation. Literature [22–25] combined fractional calculus with the grey model and used historical data to solve many practical application problems.

Since most time series including stock price series usually contain linear and non-linear factors, a single forecasting model cannot achieve good results. According to the high-dimensional, memorable, and non-linear characteristics of stock data, a Hausdorff fractional grey model with convolution,

locally linear embedding, and BP neural network (HFGMC (1, m)-LLE-BP) model is proposed in this paper. HFGMC (1, m) model is improved from FGMC (1, m) in literature [25]. The FGMC (1, m) model has several disadvantages. There are factorial operators in the elements of the accumulation generation matrix in the FGMC (1, m) model. When the data length is long, it will have the disadvantages of complex calculation and information loss. The calculation of background value in the FGMC (1, m) model is approximately obtained by using the trapezoidal formula between the two values in the sequence, which will calculate background value ignore the local change trend of the sequence. These shortcomings degrade the prediction performance of FGMC (1, m). The operator derived from HFGMC (1, m) model combined with Hausdorff fractional derivative does not contain a factorial operator, which solves the shortcomings of complex calculation and information loss. Because of the disadvantages of the background value based on approximation, the HFGMC (1, m) model uses the Newton-Cotes discrete formula and the Newton mean difference interpolation polynomial to optimize. Because the Newton-Cotes discrete formula and the Newton mean difference interpolation polynomial consider the changing trend of local data, it can solve the shortcomings of approximate background values. HFGMC (1, m) is used to predict the linear component of stock price. The BP neural network is used to predict the nonlinear component of stock price. The prediction result is obtained by adding a linear component and a nonlinear component.

This paper uses the stock closing price as the forecast target to verify the performance of HGFMC (1, m)-LLE-BP. Compared with FGMC (1, m), the prediction accuracy of HFGMC (1, m) is much higher than that of FGMC (1, m). This is because FGMC (1, m) only retains local information, while HGFMC (1, m) retains global information. Therefore, the prediction accuracy of HGFMC (1, m) will be higher. Besides, thanks to the excellent nonlinear fitting ability of the BP neural network, it can fit the nonlinear residuals in the time series. The prediction results show that the BP neural network improves the performance of HGFMC (1, m). Finally, compare the HGFMC (1, m)-LLE-BP model with the FGMC (1, m)-LLE-BP model, BP neural network model, LSTM model and the ARIMA-BP hybrid model. The performance of the HGFMC (1, m)-LLE-BP model is far better than other models.

The rest of this paper is organized as follows. In Section 2, the specific framework of the proposed model is given. In Section 3, the FGMC (1, m) is described. In Section 4, the HFGMC (1, m) is given. In Section 5, the LLE algorithm and BP neural network are introduced. In Section 6, experiments have verified the effectiveness of the proposed model. In Section 7, the work of this paper is summarized.

2. Specific frameworks of the proposed model

In this paper, the HFGMC (1, m) is proposed to predict the linear component of the stock price, and then the BP neural network is used to predict the nonlinear component of the stock price. Because stock price is affected by nonlinear high-dimensional stock data, this paper used the LLE algorithm to reduce the dimensionality of nonlinear high-dimensional stock data. The flow chart of the model proposed in this paper is shown in Figure 1.

The text description of the specific steps is as follows.

Step 1: Input parameters r, K . The parameter r is used to calculate the r -order accumulative generator matrix. The parameter K represents the number of neighboring points of each sample point, which is used for the calculation of locally linear embedding.

Step 2: Input the original data matrix. Suppose the original data matrix $X^{(0)}$, $X' \in R^{n \times m}$. n is the length of the sequence, and m is the m stock indicators including the sequence to be predicted. Select the sequence to be predicted (usually the first column of the original data matrix), and the remaining

sequences are used as related factors that affect the sequence to be predicted (usually other columns in the original data matrix except for the first column, and each column represents a related factor).

Step 3: Form the correlation factor matrix that affects the sequence to be predicted, and apply the locally linear embedding algorithm to reduce the dimension to d dimension. The sequence to be predicted and the correlation factor matrix after dimensionality reduction form the input matrix $X^{(0)}$. Then normalize the input matrix.

Step 4: Perform Hausdorff fractional accumulative generation operation (HFAGO) on the input matrix to obtain the Hausdorff fractional accumulative matrix.

Step 5: Combine the Newton-Cotes formula and use the Hausdorff fractional accumulation matrix to calculate the background value.

Step 6: After obtaining the optimized background value, combine the least square estimation to solve the parameters.

Step 7: Solve the whitening equation and get the time response function. Solve the in-sample time response sequence.

Step 8: Perform the inverse Hausdorff fractional accumulation generation operation (I-HFAGO) on the time response sequence within the sample to obtain the in-sample prediction sequence.

Step 9: Subtract the in-sample prediction sequence and the in-sample true value sequence to obtain the in-sample residual sequence. The mean of the sum of the residual sequence elements is the mean absolute error (MAE). Update r and K , and then judge whether the parameters r and K exceed the search range. If it exceeds the search range, select r , K that minimize MAE to solve the in-sample prediction sequence, and then go to step 10. If it does not exceed the search range, repeat Step 2 to Step 9.

Step 10: Input the out-of-sample correlation factor matrix R , and then use the time response function to solve the out-of-sample time response sequence. Then the out-of-sample time response function obtains the out-of-sample prediction sequence through I-HFAGO as a linear component.

Step 11: The in-sample residual sequence uses BP neural network to do nonlinear fitting, and then the out-of-sample residual sequence is output as the nonlinear component.

Step 12: Add the linear component and the non-linear component to get the combined prediction sequence.

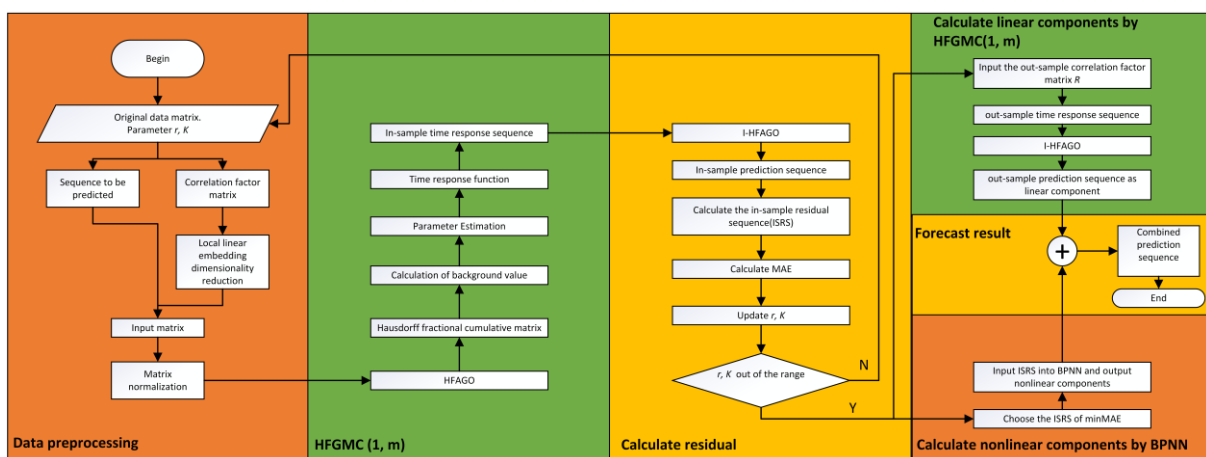


Figure 1. The flow chart of the model proposed in this paper.

The above steps can be divided into 6 modules. Step 1 to Step 3 is the data preprocessing part.

GM (1, m) model can be obtained. But the GM (1, m) had been proved inaccurate in [13]. When τ is equal to 0 and m is equal to 1, it can degenerate into a GM (1, 1) model.

By integrating the Eq (5) and using the trapezoidal formula in the interval $[k-1, k]$, the discrete form of the existing fractional grey model can be obtained as

$$\left(x_1^{(r)}(\tau+t) - x_1^{(r)}(\tau+t-1)\right) + b_1 Z_1^{(r)}(\tau+t) = \sum_{j=2}^m b_j Z_j^{(r)}(t) + u \quad (6)$$

where $Z_j^{(r)}$ is called the background values, which are defined as

$$Z_1^{(r)}(\tau+t) = \frac{1}{2} \left[X_1^{(r)}(\tau+t) + X_1^{(r)}(\tau+t-1) \right] \quad (7)$$

$$Z_j^{(r)}(t) = \frac{1}{2} \left[X_j^{(r)}(t) + X_j^{(r)}(t-1) \right], \quad j = 2, 3, \dots, m \quad (8)$$

The parameter b_1, b_2, \dots, b_m, u of Eq (5) can be obtained by least square estimation. Then substitute it into the whitening equation of Eq (4), and solve the differential equation to obtain the time response function. The form of the time response function is

$$\hat{x}_1^{(r)}(\tau+t) = x_1^{(r)}(\tau+1) e^{-b_1(t-1)} + \int_1^t e^{-b_1[t-1-(\nu-1)]} \left[\sum_{j=2}^m b_j x_j^{(r)}(\nu) + u \right] d\nu \quad (9)$$

The convolution operation of the second term on the right side of the Eq (9) can be discretized using the Gaussian formula to obtain the following discrete equation. The specific mathematical form is

$$\hat{x}_1^{(r)}(\tau+t) = x_1^{(r)}(\tau+1) e^{-b_1(t-1)} + \sum_{\nu=2}^t \left\{ e^{-b_1\left(t-\nu+\frac{1}{2}\right)} \cdot \frac{1}{2} \left[f(\nu) + f(\nu-1) \right] \right\} \quad (10)$$

where $f(\nu) = \sum_{j=2}^m b_j x_j^{(r)}(\nu) + u$. Finally, perform the inverse r -order accumulative generation operation

(r -IAGO) to obtain the predicted sequence

$$\hat{X}_1^{(0)} = (A^r)^{-1} \hat{X}_1^{(r)} \quad (11)$$

4. The HFGMC (1, m)

Accumulate generation operation (AGO) is the core step of the grey model, which can eliminate the instability caused by fluctuations in the original data. From the r -order accumulation matrix in Eq (4), it can be seen that the original sequence needs to be multiplied by a factorial multiplier when the accumulation operation is performed. When the length of the sequence is long, the calculation in the r -order accumulation matrix will be very complicated. Besides, it can be seen from Eqs (7) and (8)

that the calculation of the background value is only obtained by using a simple trapezoidal calculation formula. However, most systems do not grow linearly, which leads to calculation errors when using FGMC (1, m) models. The error is shown in Figure 2.

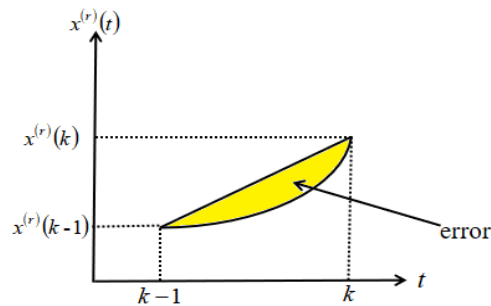


Figure 2. The flow chart of the model proposed in this paper.

4.1. Derivation of Hausdorff Fractional Accumulate Generation Operation (HFAGO)

The Hausdorff fractional derivative is defined as [26].

$$\frac{df(x)}{dx^r} = \lim_{x' \rightarrow x} \frac{f(x) - f(x')}{x^r - x'^r} \quad (12)$$

Since the first difference can usually be approximately defined by the first derivative as

$$\Delta f(t) \approx \lim_{h \rightarrow 1} \frac{f(x) - f(x-h)}{h} \Big|_{x=t} = f(t) - f(t-1) \quad (13)$$

Therefore, the difference of Hausdorff fractional derivative can be approximately defined by Hausdorff fractional derivative as

$$\Delta^r f(t) \approx \lim_{x' \rightarrow (x-1)} \frac{f(x) - f(x')}{x^r - x'^r} \Big|_{x=t} = \frac{f(t) - f(t-1)}{t^r - (t-1)^r} = [t^r - (t-1)^r]^{-1} \Delta f(t) \quad (14)$$

Let $f(t) = x^{(0)}(1), f(t-1) = x^{(0)}(2), \dots, f(1) = x^{(0)}(t)$ (This is the reverse mapping of the sequence when doing AGO operations), then 1-order reverse accumulate operation (1-RAGO) can be defined as

$$\nabla f(t) = \sum_{j=1}^t f(j) \quad (15)$$

1-RAGO is essentially the same as 1-AGO because it just does a simple inverse mapping of the sequence.

$$\nabla f(t) = \sum_{j=1}^t f(j) = x^{(1)}(t) = \sum_{i=1}^t x^{(0)}(i)$$

Eq (15) combined with Eq (13) to get

$$\Delta \nabla f(t) = \Delta \left(\sum_{j=1}^t f(j) \right) = \sum_{j=1}^t f(j) - \sum_{j=1}^{t-1} f(j) = f(t) \quad (16)$$

In the background of the grey prediction model, the difference and accumulate operation are reversible [24]. Eq (16) can be also written as

$$\nabla \Delta f(t) = f(t) \quad (17)$$

Without loss of generality, HFAGO is expressed as ∇^r and satisfies the following relationship

$$\Delta^r \nabla^r f(t) = f(t) \quad (18)$$

Combine Eq (17) with Eq (14), then move the term to the right to get

$$\Delta(\nabla^r f(t)) = f(t) \cdot [t^r - (t-1)^r] \quad (19)$$

Combining Eq (17), Eq (15) is simplified to

$$\nabla^r f(t) = \sum_{j=1}^t f(j) [j^r - (j-1)^r] \quad (20)$$

Equation (20) is HFAGO. Compared with Eq (2), HFAGO does not contain a factorial multiplier. To understand the difference between r-AGO and HFAGO more intuitively, the next step is to visually compare r-AGO and HFAGO. Assuming

$$\phi(t) = \frac{\Gamma(r+t-1)}{\Gamma(t) \cdot \Gamma(r)} \quad (21)$$

$$\omega(t) = t^r - (t-1)^r \quad (22)$$

where t is equal to $1, 2, \dots, n$. The $\Gamma(\bullet)$ in Eq (21) is the gamma function. Using Eqs (21) and (22), Eq (4) can be rewritten as

$$A^r = \begin{bmatrix} \phi(1) & & & & \\ \phi(2) & \phi(1) & & & \\ \phi(3) & \phi(2) & & & \\ \vdots & \vdots & \ddots & \phi(1) & \\ \phi(n) & \phi(n-1) & \cdots & \phi(2) & \phi(1) \end{bmatrix}_{n \times n} \quad (23)$$

$$A_H^r = \begin{bmatrix} \omega(1) & & & & \\ \omega(2) & \omega(1) & & & \\ \omega(3) & \omega(2) & & & \\ \vdots & \vdots & \ddots & \omega(1) & \\ \omega(n) & \omega(n-1) & \cdots & \omega(2) & \omega(1) \end{bmatrix}_{n \times n} \quad (24)$$

Assuming that $r = 0.5$, visualized Eqs (21) and (22). The visual comparison is shown in Figure 3.

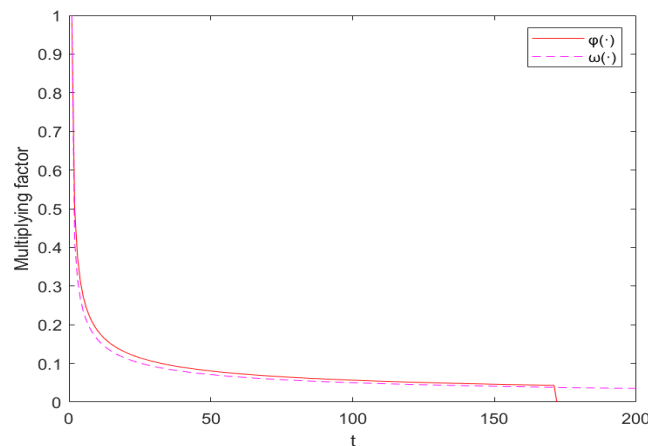


Figure 3. Visual comparison chart of Eqs (21) and (22).

In Figure 3, the solid line represents the change of the multiplier overtime when r -AGO is performed. The dotted line represents the change of the multiplier over time when performing HFAGO. The analysis shows that the r -AGO is 0 when $t > 172$. This is because the factorial number is too large, so calculate the value of the denominator in matlab2020a will get INF. It will result in the loss of subsequent information and make the calculation result inaccurate. The visualization curve of HFAGO is very similar to the traditional r -AGO. On this basis, HFAGO can retain global information and reduce the amount of calculation.

4.2. Optimization of background value based on Newton-Cotes formula

Integrate both sides of Eq (5) on the interval $[k-1, k]$ at the same time.

$$\left(x_1^{(r)}(\tau+t) - x_1^{(r)}(\tau+t-1) \right) + b_1 \int_{k-1}^k x_1^{(r)}(\tau+t) dt = \sum_{j=2}^m b_j \int_{k-1}^k x_j^{(r)}(t) dt + u \quad (25)$$

To optimize the error caused by the trapezoidal formula, it is necessary to use the Newton-Cotes formula to calculate the optimized background value for the integral term in Eq (25). Assume the integration area $[a, b]$ is divided into l equal parts. Select node $x_i = a + ih$ to construct a discrete integral formula, where h is the step size. Then, the constructed discrete integral formula is called the Newton-

Cotes formula. The mathematical form is defined as

$$I_l = (b-a) \sum_{k=0}^l C_i^{(l)} f(x) \quad (26)$$

The $C^{(l)}$ in Eq (26) is called the Cotes coefficient. When $l=1$, it is a trapezoidal formula. when $l=4$, the corresponding quadrature formula is

$$I_4 = \frac{b-a}{90} \left[7f(a) + 32f\left(a + \frac{1}{4}\right) + 12f\left(a + \frac{1}{2}\right) + 32f\left(a + \frac{3}{4}\right) + 7f(b) \right] \quad (27)$$

This paper divided the integration area into 4 equal parts. Then, the integral term in Eq (25) is constructed using Eq (27) to obtain the optimized background value. The optimized background value is

$$Z^{(r)}(k+1) = \frac{1}{90} \left[7x^{(r)}(k) + 32x^{(r)}\left(k + \frac{1}{4}\right) + 12x^{(r)}\left(k + \frac{1}{2}\right) + 32x^{(r)}\left(k + \frac{3}{4}\right) + 7x^{(r)}(k+1) \right] \quad (28)$$

It can be noticed that $x^{(r)}\left(k + \frac{1}{4}\right)$, $x^{(r)}\left(k + \frac{1}{2}\right)$, $x^{(r)}\left(k + \frac{3}{4}\right)$ do not exist for discrete data, and the Newton means difference interpolation polynomial can be used to solve this problem. The form of the Newton means difference interpolation polynomial is

$$N_n(x) = f(x_0) + f[x_0, x_1](x-x_0) + \dots + f[x_0, x_1, \dots, x_n](x-x_0)(x-x_1)(x-x_n) \quad (29)$$

where $f[x_0, x_1, \dots, x_n]$ is the $(n-1)$ order mean difference of the function $f(x)$, and n represents the data points involved in the construction of the interpolation polynomial. However, the more points involved in the construction of the polynomial are not the better. When $n > 8$, the Runge effect usually appears [27]. This paper will use every 4 data points to construct a Newton mean difference interpolation polynomial.

5. Local linear embedding algorithm and BP neural network

In this section, the LLE algorithm [28] and BP neural network [29] are introduced.

5.1. Local linear embedding (LLE) dimensionality reduction algorithm

The LLE algorithm was proposed by S.T. Roweis and L.K. Saul in 2000. The LLE algorithm is an important dimensionality reduction algorithm in flow learning. The principle is to assume that the data in a high-dimensional space is linear in a region, and certain data can be expressed linearly using data samples in its neighborhood. Then map the data to a low-dimensional space, and try to keep the weight coefficients of the linear relationship before and after the projection the same.

The algorithm steps are as follows.

Step 1: Find the K neighboring points of each sample point X_i in the high-dimensional space,

and then calculate the Euclidean distance between them.

Step 2: Solve the covariance matrix between neighboring points.

Step 3: Solve the weight coefficient vector of each sample point.

Step 4: The weight coefficient vector forms the weight coefficient matrix W , and then calculates the matrix M that can minimize the reconstruction error function.

Step 5: Solve $d+1$ smallest eigenvalues in matrix M . The eigenvector corresponding to the eigenvalues from 2 to $d+1$ are the original data matrix X corresponding to the mapping space with the smallest reconstruction error.

5.2. BP neural network

BP neural network was proposed by Hinton and McClelland in 1986. The core idea is based on the multi-layer forward neural network combined with the error loss gradient descent direction. The error loss gradient descent direction is combined with the error backpropagation algorithm to update the weights of the multilayer neural network. After enough iterations, the error loss converges to the minimum and finally can approximate any nonlinear function.

6. Experiment analysis

Through the above method and combined with the process in Figure 1, experiments are carried out to verify the performance of the model proposed in this paper. The first step is to select experimental examples. The second step is to select parameters. The third step is to compare the predicted value with the real value. The last step is to evaluate the performance of the model proposed in this paper through error indicators.

6.1. Experimental examples

This paper selected the stock closing price as the forecast target and forecasted the closing price for the next 10 days to verify the performance of the model proposed in this paper. This paper divides the predicted stocks into two categories. The first category is stocks with large fluctuations in the closing price of the stock. For example, within the range of a given trading day (excluding public holidays), The stocks with the highest closing price minus the lowest closing price greater than 5 are the first type of stocks. The second category is stocks with small fluctuations in the closing price of the stock. For example, within the range of a given trading day, the stocks with the highest closing price minus the lowest closing price less than 2 are the second type of stocks. In order to verify the effectiveness of this model for the above two types of stocks, this paper selects 6 stocks from the Shanghai Stock Exchange (code prefix is SH) and Shenzhen Stock Exchange (code prefix is SZ) as prediction objects. The 6 stocks are, SZ002607, SZ002567, SZ000796, SH600586, SZ002770 and SZ000573. SZ002607, SZ002567 and SZ000796 belong to the first category of stocks. SH600586, SZ002770 and SZ000573 belong to the second category of stocks. Stock data were crawled on NetEase Finance and Economics website using a web crawler program.

Figure 4 shows the stock closing price trend chart for a total of 256 trading days from 2019-1-16 to 2020-2-10. The closing price of SH600586, SZ002770 and SZ000573 has a small variation range, while the closing price of SZ002607, SZ002567 and SZ000796 has a large variation range.



Figure 4. Daily closing prices of 6 stocks.

Table 1. Modeling data of SH600586.

No.	Date	Close	Date	Top_price	Low_price	Opening_price	Pre_price	Floor_price
1	19/1/16	2.95	19/1/2	3.01	2.94	2.96	2.95	-0.01
					⋮			
256	20/2/10	2.52	20/1/17	2.87	2.83	2.86	2.86	-0.02
			20/1/20	2.85	2.82	2.83	2.84	0.01
			20/1/21	2.86	2.82	2.84	2.85	-0.01
					⋮			
			20/2/10	2.53	2.43	2.46	2.47	0.05

No.	Date	Close	Date	Floor	Exchange	Volume	Amount	Total value	Circulation value
1	19/1/16	2.95	19/1/2	-0.339	0.5625	8,122,601	24,196,597	4,287,408,174	4,245,758,958
						⋮			
256	20/2/10	2.52	20/1/17	-0.6993	0.3202	4,574,400	13,015,088	4,057,706,800	4,057,706,800
			20/1/20	0.3521	0.3255	4,650,641	13,176,695	4,071,994,500	4,071,994,500
			20/1/21	-0.3509	0.3737	5,339,971	15,129,108	4,057,706,800	4,057,706,800
						⋮			
			20/2/10	2.0243	0.8219	11,743,314	29,233,159	3,600,500,400	3,600,500,400

Table 1 shows the data used for modeling in SH600586, and the modeling data of the remaining 5 stocks are shown in the Supplementary. In this paper, the highest price, lowest price, opening price, previous closing price, floor change, floor, exchange, trading volume, trading amount, total market value and circulating market value are selected as indicators that affect the closing price of stocks.

Indicators such as the highest price, lowest price, opening price, previous closing, floor change, floor and the exchange can reflect the changing trend of the closing price of stocks in the recent period. Indicators such as trading volume, trading amount, total market value and circulating market value can reflect the financial status of the stock company corresponding to the stock. Imitating the method of literature [30], the above indicators are reduced by dimensionality reduction algorithm, and then input into the HGFMC (1, m) model.

6.2. Discussion of hyperparameters

First, the delay period parameter τ is discussed. It can be noted that there are 10 trading days' indicator data after the serial number 256. The indicator data of these 10 trading days is used to predict the closing price of the next 10 days. Since the HGFMC (1, m) model is a time series forecasting model that processes high-dimensional data, the model needs a priori data of the next 10 days to predict the stock closing price in the next 10 days. Considering that investors are not sensitive to changes in stock data [31] (except for market crises), this makes investors have a certain delay in responding to stock data. In other words, the stock data at the moment will have a time delay to affect investor sentiment, and then investor sentiment will affect the stock price. It can be seen from Eq (10) that HGFMC (1, m) integrates the delay function and shows the relationship between the stock price and various indicators. Therefore, when used Eq (10) to solve the time response function, the delay period is set to 10. The setting of the delay period τ is equal to the amount of prior data required.

Then the dimension d of the low-dimensional mapping space of the LLE algorithm is discussed. Literature [30,32,33] shows that setting the dimension d of the low-dimensional space to 2, 3, and 4 can well express the feature space of the data. Therefore, this paper uses dimension 3 as the dimension of the low-dimensional mapping space of the LLE algorithm.

6.3. Discussion on the stability of stock prices

The stationary test of stock prices is discussed in this section. This paper uses the ADF test to test the stability of stock prices.

The ADF test can check whether there is a unit root in the sequence. If there is a unit root, the sequence is not stationary. First, ADF test is performed on the original price series.

Table 2. ADF test of the original sequence.

Stock code	Test statistic	Significance level		
		1%	5%	10%
SZ000573	-1.34	-3.45	-2.87	-2.57
SZ000796	-2.22	-3.45	-2.87	-2.57
SZ002567	-1.62	-3.45	-2.87	-2.57
SZ002607	-2.10	-3.45	-2.87	-2.57
SZ002770	-0.75	-3.45	-2.87	-2.57
SH600586	-1.08	-3.45	-2.87	-2.57

As shown in Table 2, the original sequence of 6 stocks cannot reject the existence of unit roots at the significance levels of 1, 5, and 10%. Therefore, the original sequence of 6 stocks can be considered

unstable. Since the stock price series is not stationary, it is necessary to stabilize the series before forecasting.

From Eqs (1) and (20), it can be known that after the original sequence uses HFAGO, the random walk of the price appears as the increment between data points on the accumulative generation sequence. Therefore, the original sequence can be considered stationary after using HFAGO.

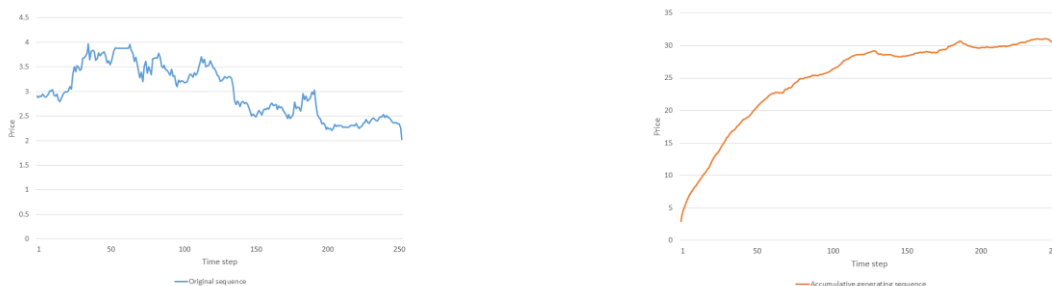


Figure 5. Stabilization of original series.

Figure 5 shows the stabilization process of SZ002770's stock price. The left image of Figure 5 shows the original sequence. The right figure of Figure 5 shows the accumulative generation sequence. The stationary test of the accumulative generation sequence of 6 stocks also uses the ADF test for further verification.

Table 3. ADF test of the accumulative generation sequence.

Stock code	Test statistic	Significance level		
		1%	5%	10%
SZ000573	-5.46	-3.45	-2.87	-2.57
SZ000796	-2.87	-3.45	-2.87	-2.57
SZ002567	-6.52	-3.45	-2.87	-2.57
SZ002607	-6.31	-3.45	-2.87	-2.57
SZ002770	-3.66	-3.45	-2.87	-2.57
SH600586	-4.93	-3.45	-2.87	-2.57

It can be seen from Table 3 that SZ000573, SZ002567, SZ002607, SZ002770 and SH600586 reject the hypothesis at the significance levels of 1, 5, and 10%, while SZ00796 rejects the hypothesis at the significance levels of 5% and 10%. Therefore, it can be considered that the accumulative generation sequence obtained by the original sequence after using HFAGO is stationary.

6.4. Select parameters

It can be seen from Figure 1 that the establishment of the model requires the selection of appropriate parameters r and K . This paper established the contour diagram between the parameters r , K and the mean absolute error (MAE), and then found the parameter that could minimize the MAE.

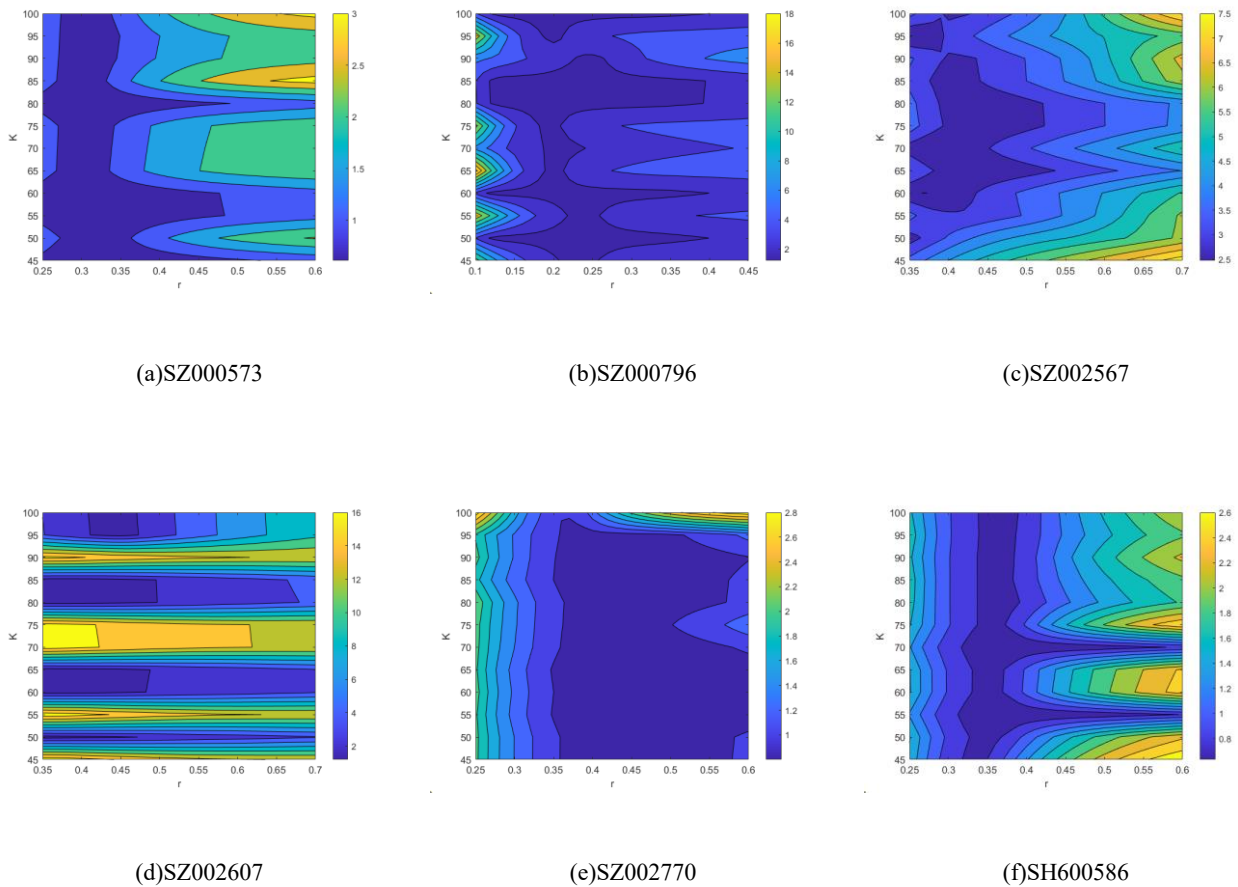


Figure 6. Contour diagram of parameters r , K and MAE.

Figure 6 is a contour diagram between the parameters r , K and MAE. In the figures, the horizontal axis represents the order r , and the vertical axis represents the number of K neighboring points of each sample point. The bluer the color, the smaller the MAE. The yellower the color, the larger the MAE. This paper enumerated the order r in the range of 0 to 1 in steps of 0.1 and enumerated the number of neighboring points K of the sample points from 0 to 100 in steps of 5. Then, select the search range that minimizes the MAE. Finally, use the exhaustive method within the range to get the relative optimal parameters.

Table 4. Parameter range.

Stock code	r	K
SZ000573	$r \in [0.25, 0.6]$	
SZ000796	$r \in [0.1, 0.45]$	
SZ002567	$r \in [0.35, 0.7]$	$K \in [45, 100]$
SZ002607	$r \in [0.35, 0.7]$	
SZ002770	$r \in [0.25, 0.6]$	
SH600586	$r \in [0.25, 0.6]$	

Table 4 shows the optimal parameter range of each stock. The search step for parameter r is 0.01, and the search step for parameter K is 5. For all stocks, the optimal search range of parameter K is 45 to 100. For SZ000573, SZ002770 and SH60586, the search range of parameter r is 0.25 to 0.6. For SZ000796, the search range of parameter r is 0.1 to 0.45. For SZ002567 and SZ002607, the search range of parameter r is 0.35 to 0.7.

Table 5. Result of parameters.

Stock code	r	K	MAE
SZ000573	0.30	60	0.6120
SZ000796	0.14	50	1.2227
SZ002567	0.35	95	2.4767
SZ002607	0.35	65	1.1929
SZ002770	0.44	65	0.8013
SH600586	0.41	55	0.6336

Table 5 shows the result of parameters. For SZ000573, when the order r is 0.30 and K is 60, the MAE is 0.6120. For SZ000796, when the order r is 0.14 and K is 50, the MAE is 1.2227. For SZ002567, when the order r is 0.35 and K is 95, the MAE is 2.4767. For SZ002607, when the order r is 0.35 and K is 65, the MAE is 1.1929. For SZ002770, when the order r is 0.44 and K is 65, the MAE is 0.8013. For SH600586, when the order r is 0.41 and K is 55, the MAE is 0.6336.

6.5. Discussion between HFGMC (1, m) and FGMC (1, m)

To make the distribution of the data input to FGMC (1, m) the same as HGFMC (1, m), the parameters of the data input to FGMC (1, m) also need to be set according to Table 5. The comparison of HFGMC (1, m) and FGMC (1, m) uses the mean error between the in-sample predicted sequence and the real sequence as an indicator. Table 6 shows the MAE comparison between HFGMC (1, m) and FGMC (1, m).

Table 6. The MAE comparison.

Stock code	FGMC (1, m)	HFGMC (1, m)
SZ000573	16.3341	0.6120
SZ000796	10.2270	1.2227
SZ002567	11.1814	2.4767
SZ002607	8.4681	1.1929
SZ002770	1.9685	0.8013
SH600586	3.0270	0.6336

It can be seen from Table 6 that the improvement of FGMC (1, m) by HGFMC (1, m) is very obvious, and the average MAE is reduced by 7.3778.

6.6. Discussion of BP Neural Network

In this section, the improvement of the model by the BP neural network is discussed. It can be known from Eq (10) that the out-sample forecast data can be regarded as a linear combination of various indicators, so a single HFGMC (1, m) can only predict the linear part of the future data. Due to the strong nonlinear fitting ability of the BP neural network, it can fit the residual function in the in-sample prediction sequence. Therefore, using BP neural network can get the nonlinear part of future data. Next, compare the model performance of HFGMC (1, m) and HFGMC (1, m) combined with the BP neural network.

Table 7. Comparison of MAE of out-sample forecast sequence.

Stock code	HFGMC (1, m) combined BP	HFGMC (1, m)
SZ000573	0.12	0.64
SZ000796	0.42	0.37
SZ002567	0.20	3.66
SZ002607	0.78	1.12
SZ002770	0.03	0.89
SH600586	0.21	0.21

As shown in Table 7, except for the slight decrease and no change in the forecast accuracy of SZ000796 and SH600586, the forecast accuracy of other stocks has been significantly improved.

6.7. Forecast result

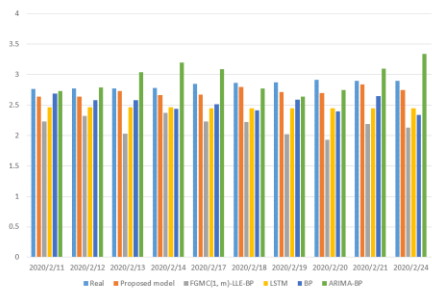
This paper chose FGMC (1, m)-LLE-BP, long short-term memory neural network (LSTM), BP neural network (BP), ARIMA-BP [34] as the comparison model. BPNN and LSTMNN used 10 lagging data of the stock closing price to input the neural network in the form of a vector, and output the current value. For example, the input and output samples of SH600586 are shown in Table 8.

Table 8. Input and output samples of SH600586.

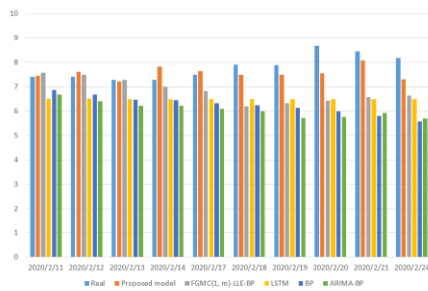
Sample	x_1	x_2	...	x_9	x_{10}	y
1	2.94	2.98	...	3.05	3.00	2.98
2	2.98	3.00	...	3.00	2.98	2.99
			⋮			

The vector x in Table 8 represents the input vector, and y represents the output. This paper used a sliding window method to obtain all training samples. The forecast results of each model are shown in Figure 7.

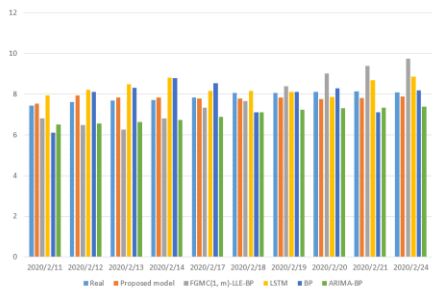
From the analysis of the prediction results in Figure 7, it is concluded that both SZ000573, SZ002770 and SH600586 with a small change in stock trend or SZ000796, SZ002567 and SZ002607 with a large change in stock trend predicted by the proposed model are very close to the true value. However, there are some errors between the predicted results of the comparison models and the true values. The specific model performance analysis is given in the error analysis section.



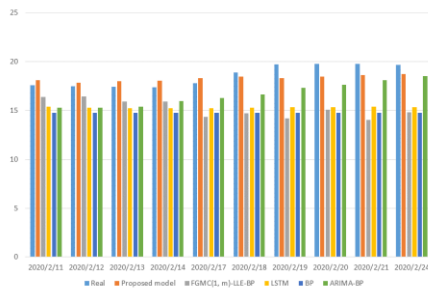
(a)SZ000573



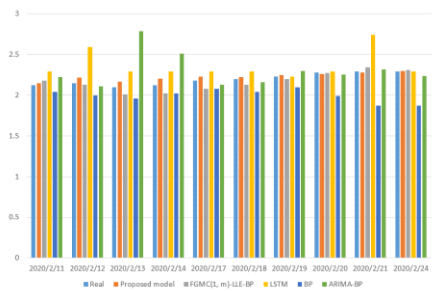
(b)SZ000796



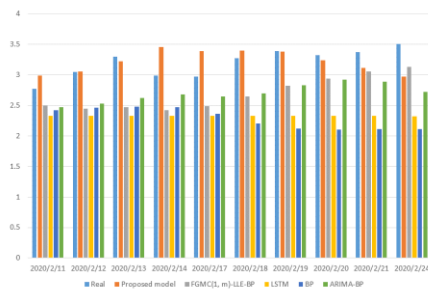
(c)SZ002567



(d)SZ002607



(e)SZ002770



(f)SH600586

Figure 7. Forecast results.

6.8. Error analysis

The forecast error analysis in this paper used mean absolute error (MAE), root mean square error (RMSE) and mean absolute percentage error (MAPE) as evaluation indicators.

$$MAE = \frac{1}{n} \sum_{i=1}^n |\hat{x}(i) - x(i)| \quad (30)$$

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (\hat{x}(i) - x(i))^2} \quad (31)$$

$$MAPE = \frac{1}{n} \sum_{i=1}^n \left| \frac{\hat{x}(i) - x(i)}{x(i)} \right| \times 100\% \quad (32)$$

where $\hat{x}(i)$ is the predicted value, and $x(i)$ is the true value.

Table 9. Error comparison of SZ000573.

Indicator	Proposed model	FGMC (1, m)-LLE-BP	LSTM	BP	ARIMA-BP
MAE	0.12	0.66	0.37	0.31	0.21
RMSE	0.13	0.68	0.38	0.34	0.23
MAPE	4.3%	23%	13.2%	11%	7.4%

Table 10. Error comparison of SZ000796.

Indicator	Proposed model	FGMC (1, m)-LLE-BP	LSTM	BP	ARIMA-BP
MAE	0.37	1.01	1.29	1.53	1.72
RMSE	0.48	1.30	1.38	1.73	1.87
MAPE	4.5%	12.5%	16.3%	19.0%	21.6%

Table 11. Error comparison of SZ002567.

Indicator	Proposed model	FGMC (1, m)-LLE-BP	LSTM	BP	ARIMA-BP
MAE	0.20	0.91	0.51	0.65	0.91
RMSE	0.22	1.01	0.60	0.78	0.90
MAPE	2.6%	11.6%	6.5%	8.0 %	11.4%

Table 12. Error comparison of SZ002607.

Indicator	Proposed model	FGMC (1, m)-LLE-BP	LSTM	BP	ARIMA-BP
MAE	0.78	3.36	3.23	3.79	1.90
RMSE	0.86	3.81	3.39	3.93	1.95
MAPE	4.1%	17.6%	17.1%	20.0%	10.3%

From the analysis of Tables 9–14, it can be seen that when the model proposed in this paper predicted SZ000573, SZ002770 and SH600586 with small changes in trend, the error between the predicted value and the true value is on average below 0.21. When predicted SZ000796, SZ002567

and SZ002607 with large changes in trend, the average error between the predicted value and the true value is less than 0.78. The performance is stable.

Table 13. Error comparison of SZ002770.

Indicator	Proposed model	FGMC (1, m)-LLE-BP	LSTM	BP	ARIMA-BP
MAE	0.03	0.05	0.16	0.19	0.14
RMSE	0.04	0.06	0.22	0.22	0.25
MAPE	1.7%	2.4%	7.6%	9.0%	6.9%

Table 14. Error comparison of SH600586.

Indicator	Proposed model	FGMC (1, m)-LLE-BP	LSTM	BP	ARIMA-BP
MAE	0.21	0.49	0.86	0.90	0.48
RMSE	0.28	0.52	0.89	0.97	0.51
MAPE	6.9%	15.5%	26.6%	28.0%	15.1%

When FGMC (1, m)-LLE-BP predicted SZ000573, SZ002770, and SH600586 with small changes in trend, the error between the predicted value and the true value is on average below 0.66. When predicted SZ000796, SZ002567 and SZ002607 with large changes in trend, the average errors are 1.01, 0.91 and 3.36 respectively. The performance is not stable.

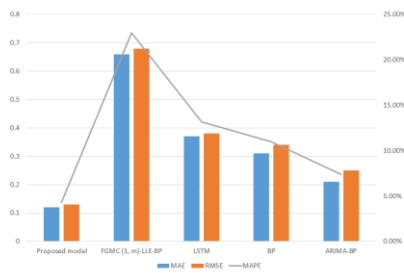
When LSTM neural network model predicted the SZ000573, SZ002770, and SH600586 with small trend change, the average errors are 0.37, 0.16 and 0.49 respectively. When predicted the SZ000796, SZ002567 and SZ002607 with large changes in trend, the average errors are 1.29, 0.51 and 3.23 respectively. This shows that LSTMNN has volatility in stocks with large changes in predicted trends and stocks with small changes in predicted trends.

When BP neural network model predicted the SZ000573, SZ002770, and SH600586 with small trend change, the average errors are 0.31, 0.19 and 0.90 respectively. When predicted the SZ000796, SZ002567 and SZ002607 with large changes in trend, the average errors are 1.53, 0.65 and 3.79 respectively.

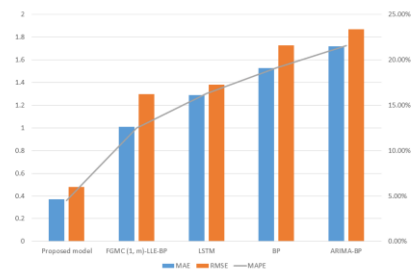
When ARIMA-BP predicted the SZ000573, SZ002770, and SH600586 with small trend change, the average errors are 0.21, 0.14 and 0.48 respectively. When predicted the SZ000796, SZ002567 and SZ002607 with large changes in trend, the average errors are 1.72, 0.91 and 1.90 respectively.

Combining the analysis of three error indicators, the model proposed in this paper performs best, followed by ARIMA-BP and FGMC (1, m)-LLE-BP. However, neither the LSTM nor the BP neural network model of a single prediction model can achieve the desired effect. This shows that when predicting time series with linear and non-linear factors, the combined model can achieve good results.

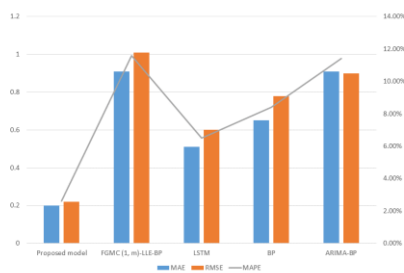
From Figure 8, it can be seen that the model proposed in this paper has stable and excellent performance for 6 stocks with different trends, while the performance of the other 4 comparison models is only good in some stocks. Combining the analysis of three evaluation indicators, the model proposed in this paper performed best on the MAE, RMSE, and MAPE indicators when predicted SZ002607, SZ002567, SZ000796, SH600586, SZ002770 and SZ000573. Besides, the model proposed in this paper performs better than FGMC (1, m)-LLE-BP, which shows that the HFGMC (1, m) has a certain improvement in the accuracy of prediction.



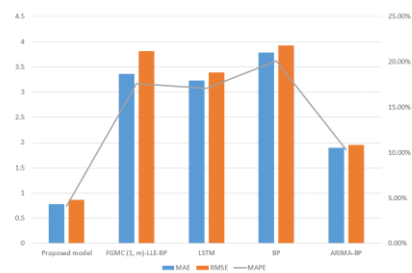
(a)SZ000573



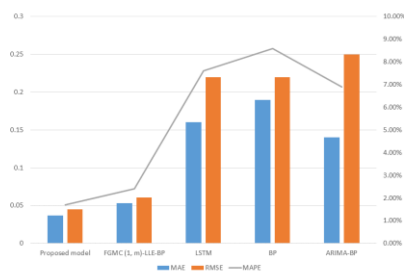
(b)SZ000796



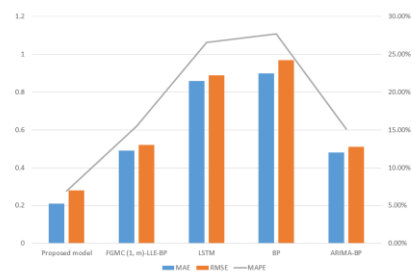
(c)SZ002567



(d)SZ002607



(e)SZ002770



(f)SH600586

Figure 8. Error comparison of 6 stocks.

6.9. Limitations and future works

In the discussion in Section 6.2, it is pointed out that a model that deals with high-dimensional data requires certain prior data. However, it is very difficult to obtain real-time a priori data for forecasting objects with very high real-time requirements such as stock data. Generally, the high-dimensional grey model obtains the prior data by staggering the time axis of the prediction object (closing price) and prior data (indicators) (as shown in Figure 9). Therefore, the practice of staggering

the time axis determines that the model cannot be used for long-term forecasting. When the time axis is staggered longer, the error may increase significantly. In addition to staggering the time axis, the use of sliding time windows to obtain prior data is also widely used in deep learning time series forecasting [2,3,30] (as shown in Table 8). In addition, the price of individual stocks (the experimental case of this paper) is different from stock indexes (Dow Jones, Nasdaq, Shanghai Composite Index, etc.). The price of individual stocks is usually predicted for about 10 days or 10 units of time [4,30,35], and the stock index reflects the general laws of the market, so long-term forecasts are feasible [36–39]. However, whether it is individual stock prices or stock indexes, the problem of prior data cannot be solved when using high-dimensional models. This is a limitation of high-dimensional models.

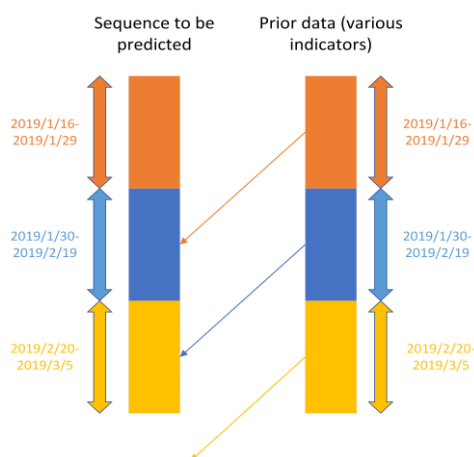


Figure 9. The relationship between the sequence to be predicted and the prior data.

Since directly staggering the time axis may lose the information of various indicators in the time axis, the next step can consider using the maximum entropy model to integrate all the information in the time axis as the prior information of the prediction object. Through the above work, the accuracy of the model may be more accurate.

7. Conclusions

Because the stock price is affected by many factors, the stock market can be regarded as a non-linear dynamic and grey system under the action of many factors. It is very difficult to predict stock price. There are many time series forecasting methods, but many methods do not consider that the stock market has long-term memory.

According to the characteristics of the stock market, the HFGMC (1, m)-LLE-BP is proposed. The HFGMC (1, m) is based on the FGMC (1, m) model, and combined the Hausdorff fractional derivative and Newton-Cotes formula. The HFGMC (1, m) model is used to solve the linear component in the prediction sequence. Then, the BP neural network is used to extract the non-linear component in the prediction sequence. Besides, in order to compress the redundant information in high-dimensional data, the HFGMC (1, m) model combined the locally linear embedding algorithm to reduce the dimensionality of factors that affect the stock price. Finally, the linear component and the non-linear component are added to obtain the combined prediction sequence. In the experimental part, the model proposed in this paper predicts the closing price of 6 stocks with different trends in the next

10 days. Based on the above experimental results and discussion, the following conclusions are drawn:

(i) FGMC (1, m) model has some shortcomings. Hausdorff fractional derivative and Newton-Cotes formula can improve FGMC (1, m) model. The HFGMC (1, m) model improves the prediction accuracy of time series.

(ii) Because the stock price is affected by a variety of linear and nonlinear factors, a single time prediction model cannot accurately predict the stock price. A combined forecasting model has advantages in forecasting time series affected by complex factors.

(iii) The model proposed in this paper is very stable and accurate in predicting the stock price with different historical trends. Results show that the model proposed in this paper can be widely used in the prediction of simple or complex time series.

Acknowledgments

This work is supported by National Nature Science Foundation under Grant 61862062, 61104035.

Conflict of interest

The authors declared that they have no conflict of interest.

References

1. S. Z. Shi, W. L. Liu, M. L. Jin, Stock price forecasting based on a combined model of ARMA and BP neural network and Markov model, *Inf. Process. Manage.*, **4** (2013), 215–221.
2. M. Vijh, D. Chandola, V. A. Tikkiwal, A. Kumar, Stock closing price prediction using machine learning techniques, *Proc. Comput. Sci.*, **167** (2020), 599–606.
3. G. Liu, X. J. Wang, A new metric for individual stock trend prediction, *Eng. Appl. Artif. Intell.*, **82** (2019), 1–12.
4. E. Sin, L. Wang, Bitcoin price prediction using ensembles of neural networks, in *13th International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery*, (2017), 666–671.
5. M. Zhu, L. Wang, Intelligent trading using support vector regression and multilayer perceptrons optimized with genetic algorithms, in *The 2010 International Joint Conference on Neural Networks*, (2010), 1–5.
6. G. Q. Dong, K. Fataliyev, L. Wang, One-step and multi-step ahead stock prediction using backpropagation neural networks, in *The 9th International Conference on Information, Communications & Signal Processing*, (2013), 1–5.
7. D. H. Zhang, S. Lou, The application research of neural network and BP algorithm in stock price pattern classification and prediction, *Futur. Gener. Comp. Syst.*, **115** (2021), 872–879.
8. K. K. Teo, L. Wang, Z. Liu, Wavelet packet multi-layer perceptron for chaotic time series prediction: effects of weight initialization, in *International Conference on Computational Science*, Springer, Berlin, Heidelberg, (2001), 310–317.
9. H. K. Sang, C. Cheong, S. M. Yoon, Long memory volatility in Chinese stock markets, *Phys. A Stat. Mech. Appl.*, **389** (2020), 1425–1433.

10. S. L. Gao, Q. Zhao, The memory effect of fractional calculus, *J. Leshan Normal University*, **30** (2015), 1–4.
11. W. Y. Wu, S. P. Chen, A prediction method using the grey model GMC (1, n) combined with the grey relational analysis: a case study on Internet access population forecast, *Appl. Math. Comput.*, **169** (2005), 198–217.
12. X. Ma, Z. B. Liu, The GMC (1, n) model with optimized parameters and its application, *J. Grey Syst.*, **29** (2017), 122–138.
13. T. L. Tien, A research on the grey prediction model GM (1, n), *Appl. Math. Comput.*, **218** (2012), 4903–4916.
14. W. Q. Wu, X. Ma, B. Zeng, W. Y. Lv, Y. Wang, W. P. Li, A novel Grey Bernoulli model for short-term natural gas consumption forecasting, *Appl. Math. Model.*, **84** (2020), 393–404.
15. Q. Z. Xiao, M. Y. Gao, X. P. Xiao, M. Goh, A novel grey Riccati-Bernoulli model and its application for the clean energy consumption prediction, *Eng. Appl. Artif. Intell.*, **95** (2020), 103863.
16. X. Ma, Z. B. Liu, Y. Wang, Application of a novel nonlinear multivariate grey Bernoulli model to predict the tourist income of China, *J. Comput. Appl. Math.*, **347** (2019), 84–94.
17. W. Q. Wu, X. Ma, B. Zeng, Y. Wang, W. Cai, Forecasting short-term renewable energy consumption of China using a novel fractional nonlinear grey Bernoulli model, *Renew. Energy*, **140** (2019), 70–87.
18. U. Şahin, T. Şahin, Forecasting the cumulative number of confirmed cases of COVID-19 in Italy, UK and USA using fractional nonlinear grey Bernoulli model, *Chaos Solitons Fractals*, **138** (2020), 109948.
19. X. Ma, Z. B. Liu, The kernel-based nonlinear multivariate grey model, *Appl. Math. Model.*, **56** (2018), 217–238.
20. L. Z. Wu, S. H. Li, R. Q. Huang, Q. Xu, A new grey prediction model and its application to predicting landslide displacement, *Appl. Soft Comput.*, **95** (2020), 106543.
21. X. Y. Zeng, S. L. Yan, F. L. He, Y. C. Shi, Multi-variable grey model based on dynamic background algorithm for forecasting the interval sequence, *Appl. Math. Model.*, **80** (2020), 99–114.
22. W. Q. Wu, X. Ma, B. Zeng, Y. Wang, W. Cai, Application of the novel fractional grey model FAGMO (1, 1, k) to predict China's nuclear energy consumption, *Energy*, **165** (2018), 223–234.
23. W. L. Xie, C. X. Liu, W. Z. Wu, W. D. Li, C. Liu, Continuous grey model with conformable fractional derivative, *Chaos Solitons Fractals*, **139** (2020), 110285.
24. X. Ma, W. Q. Wu, B. Zeng, Y. Wang, X. X. Wu, The conformable fractional grey system model, *ISA Trans.*, **96** (2020), 255–271.
25. X. Ma, M. Xie, W. Q. Wu, B. Zeng, Y. Wang, X. X. Wu, The novel fractional discrete multivariate grey system model and its applications, *Appl. Math. Model.*, **70** (2019) 402–424.
26. J. Wang, Y. Liang, L. Qiu, X. U. Yang, Improved machine learning technique for solving Hausdorff derivative diffusion equations, *Fractals*, **28** (2020), 160–169.
27. J. Wang, H. Shan, W. Q. Wang, X. F. Chen, P. Y. Li, Research on wind speed prediction based on hybrid gray theory, *Acta Energetica Solaris Sinica*, **39** (2018), 3544–3549.
28. S. T. Roweis, L. K. Saul, Nonlinear dimensionality reduction by locally linear embedding, *Science*, **290** (2000), 2323–2326.
29. G. E. Hinton, D. E. Rumelhart, J. L. McClelland, Distributed representations, Parallel distributed processing: explorations in the microstructure of cognition, *Language*, **63** (1986), 45–76.

30. Z. X. Yu, L. Qin, Y. J. Chen, M. D. Parmar, Stock price forecasting based on LLE-BP neural network model, *Phys. A Stat. Mech. Appl.*, **553** (2020), 124197.
31. J. Miśkiewicz, Economy with the time delay of information flow-The stock market case, *Phys. A Stat. Mech. Appl.*, **391** (2012), 1388–1394.
32. J. Wang, R. K. W. Wong, T. C.M. Lee, Locally linear embedding with additive noise, *Pattern Recognit. Lett.*, **123** (2019), 45–52.
33. S. Kadoury, M. D. Levine, Face detection in gray scale images using locally linear embeddings, *Comput. Vis. Image Underst.*, **105** (2007), 1–20.
34. E. Cadenas, R. Rivera, Wind speed forecasting in three different regions of Mexico, using a hybrid ARIMA-ANN model, *Renew. Energy*, **35** (2010), 2732–2738.
35. C. L. Xiao, W. L. Xia, J. J. Jiang, Stock price forecast based on combined model of ARI-MA-LS-SVM, *Neural Comput. Appl.*, **32** (2020), 5379–5388.
36. H. L. Niu, K. L. Xu, A hybrid model combining variational mode decomposition and an attention-GRU network for stock price index forecasting, *Math. Biosci. Eng.*, **17** (2020), 7151–7166.
37. Y. L. Wang, L. W. Wang, F. J. Yang, W. X. Di, Q. Chang, Advantages of direct input-to-output connections in neural networks: The Elman network for stock index forecasting, *Inf. Sci.*, **547** (2021), 1066–1079.
38. Y. Lin, Y. Yan, J. L. Xu, Y. Liao, F. Ma, Forecasting stock index price using the CEEMDAN-LSTM model, *N. Am. Econ. Financ.*, **57** (2021), 101421.
39. S. Feuerriegel, J. Gordon, Long-term stock index forecasting based on text mining of regulatory disclosures, *Decis. Support Syst.*, **112** (2018), 88–97.



AIMS Press

2021 the Author(s), licensee AIMS Press. This is an open access article distributed under the terms of the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/4.0>)