



Research on financial assets transaction prediction model based on LSTM neural network

Xue Yan¹ · Wang Weihan² · Miao Chang¹

Received: 30 January 2020 / Accepted: 2 May 2020 / Published online: 16 May 2020
© Springer-Verlag London Ltd., part of Springer Nature 2020

Abstract

In recent years, with the breakthrough of big data and deep learning technology in various fields, many scholars have begun to study the stock market time series by using deep learning technology. In the process of model training, the selection of training samples, model structure and optimization methods are often subjective. Therefore, studying these influencing factors is beneficial to provide scientific suggestions for the training of recurrent neural networks and is beneficial to improve the prediction accuracy of the model. In this paper, the LSTM deep neural network is used to model and predict the financial transaction data of Shanghai, and the three types of factors affecting the prediction accuracy of the model are systematically studied. Finally, a high-precision short-term prediction model of financial market time series based on LSTM deep neural network is constructed. In addition, this paper compares BP neural network, traditional RNN and RNN improved LSTM deep neural network. It proves that the LSTM deep neural network has higher prediction accuracy and can effectively predict the stock market time series.

Keywords LSTM · Financial model · Transaction forecasting · Model analysis · Financial market

1 Introduction

Since the mid-19th century, the limited liability company system has gradually been replaced by joint-stock companies. The degree of prosperity of financial transactions represents the economic status of a country. According to statistics, the total market capitalization of China's stock market in 2015 has reached 146% of GDP, ranking first among developing countries. Moreover, in 2015, the total market capitalization of the US stock market accounted for 174% of GDP, leading the developed countries. The stock market, as a high-risk investment market with both risks and returns, has been closely watched by investors. Stock exchanges in various countries generate huge amounts of transaction data every day. Investors and investment

institutions are increasingly using data as their primary reference when trading securities and investing in stocks. Real-time transaction data, such as K-line and time sharing, are often an objective reflection of the market. Investors and investment institutions often select these historical data for analysis and forecasting in order to achieve higher profits [1].

In an attempt to predict the changing trend of financial prices, financial investors are constantly observing the subtle changes in financial markets. Researchers at home and abroad have applied various statistical and econometric methods to financial market research, such as exponential smoothing, multiple regression, ARIMA models (autoregressive moving average models) and GARCH models (autoregressive conditional heteroscedastic models) [2]. However, due to the many factors affecting financial prices, the mechanism of influence is also very complicated, and it is difficult to simply use mathematical models to explain. Traditional statistical models and econometric models can usually simulate linear or stationary sequences, and the amount of data introduced by the model should not be too large and requires preprocessing (usually differential) and the need to change non-stationary sequences to

✉ Wang Weihan
bqt1700501009@student.cumtb.edu.cn

¹ School of Management, Heilongjiang University of Science and Technology, Harbin 150022, Heilongjiang, China

² School of Information Management and Engineering, Shanghai University of Finance and Economics, Shanghai 200433, China

stationary sequences. Even if a new generalized autoregressive model can be used to describe a time series of non-stationary nonlinearities, the calculation results of the model are not ideal.

Neural network machine learning is a highly complex nonlinear artificial intelligence system that is an artificial simulation of the human brain's abstract and figurative capabilities. It has distributed storage, self-organization, parallel processing and self-tuning capabilities. The above characteristics of the neural network make it more suitable for dealing with complex nonlinear problems with multiple influencing factors, instability and random types. Stock price forecasting is such a problem. Therefore, in the past ten years, scholars from various countries have used neural networks for data analysis and prediction in the stock market and have obtained a lot of research results. For example, the "Stock Forecasting and Testing System for Wavelet Neural Networks" developed by Google Finance and Economics has conducted in-depth mining and forecasting of the net value of equity funds. Haitong Securities has also conducted in-depth research on the weekly average of the Shanghai Composite Index. The selected quantitative model is a multilayer BP neural network. Recently, Ali Research Institute and Baidu Big Data Research have also used the integrated environment neural network model to mine user needs, stock forecasting and image recognition [3].

Compared to traditional statistics and econometric prediction methods, neural network machine learning is better. However, the traditional BP neural network prediction model is not applicable to the problem of time series training prediction in theory. At the same time, there are problems; for example, the number of input data is difficult to determine, it is easy to fall into local minimum and the network structure is difficult to determine. As a new type of neural network, RNN has a timing concept, which can realize multiple input and output in time series and realize self-connection between hidden layers, and it is equivalent to the interaction between timings after expansion. However, the RNN also has serious problems such as rapid gradient degradation and inability to converge to the optimal solution. Recently, the implementation of the LSTM neural network has further answered these questions [4]. However, in the analysis of the application of neural networks to the stock market, these problems still need to be explored and solved.

Since the beginning of the financial market, scholars and economic financiers around the world have been exploring and researching the ever-changing financial market and have achieved many successes. It is proved by practice that the change of financial price in the short term is predictable. As mentioned above, due to the complexity and variety of various influencing factors and the ambiguity of

the influencing mechanism, it is extremely difficult to truly achieve a completely accurate stock price forecast. However, for the European and American markets with financial value orientation, the neural network of machine learning has played its role. This supervised learning simulates the memory of the human brain and can abstractly describe the random time-history process. This kind of machine learning is worthy of research and experiment by relevant experts and scholars, and the prospects are promising. The LSTM neural network is the latest achievable neural network model. Its selective memory and temporal internal influence characteristics are very suitable for financial price time series, which are random non-stationary sequences. Therefore, the author believes that in the future, the expansion and improvement of the LSTM neural network will be endless and widely used.

2 Related work

The earliest research on financial markets began in foreign countries, and research on financial forecasting in the 1990s has received widespread attention from foreign academic circles. The widely used forecasting techniques have been divided into three categories: investment analysis, time series analysis and artificial intelligence.

Hien LV uses neural network technology for financial forecasting, and Hien LV uses neural networks to predict the daily rate of return on IBM stocks [5]. However, because the traditional BP neural network is very easy to gradient explosion, it only converges to the local minimum. Therefore, the trained model cannot get accurate prediction results. Khemakhem S and other scholars developed a TOPIX prediction system using neural network technology [6], which is used to predict the weighted average index of the Tokyo Stock Exchange (similar to the Dow Jones index). The empirical results show that the neural network model is more suitable for the prediction of the TOPIX weighted average index.

Maciel L developed a forward artificial neural network model [7], and the empirical data of the model are the historical data of the 7-day moving average of the Dow Jones Industrial Average. Empirical evidence shows that the predictive power of neural network models is better than the simple moving average statistical algorithm.

Ahmed M and other research scholars established a comprehensive model of moving average rules and forward artificial neural networks [8] and empirically analyzed the Spanish stock market. The analysis results show that the profit from the stock trading conducted under the same trading time according to the prediction results of the neural network model is far higher than the profit obtained

by holding. This study shows that the neural network model has a high guiding significance for investors' investment strategies.

Kim S compared the neural network model with the ARIMA model in time series prediction [9]. The research results show that the neural network model has obvious advantages in nonlinear data analysis and processing, and the prediction accuracy is much higher than the ARIMA model. However, the article does not further improve and discuss the neural network model.

Hu R believes that stock price data have high white noise and certain randomness, so the data are preprocessed by a complete independent component analysis method to reduce noise, and then brought into the neural network and established a stock trend forecasting model [10]. The model first uses independent component analysis and then performs noise reduction processing to predict the variables. The preprocessed parameters will contain low-noise data as input parameters for the model. The empirical analysis uses the opening price of the Nikkei 225 Index. The results show that the neural network model is superior to the wavelet analysis technique. However, the article does not analyze or explain the internal structure of the neural network model in detail or discuss the problem.

Tsantekidis A compares Bayesian estimates with neural network predictions in the stock market [11]. They selected historical data of 70 stocks from 215 stocks for 5 years and compared the data with different criteria. The experimental results show that both methods can be applied in the stock trading market, but the effect of the neural network method is more prominent. Demidov R A uses neural network models to predict large-cap data of TKC stocks [12] and brings various types of parameters into models such as trends, patterns, dynamics and volatility. At the same time, he uses historical data to compare and predict the predicted values. The experimental results show that the neural network model is superior to other predictive models and can be used to establish Markowitz's most investment portfolio. However, the article does not explain in detail the selection basis and rationality of the input parameters of the neural network model.

Researchers such as Hong-Yan R used a multilayer BP neural network to analyze stock prices in the Jordanian stock market [13]. Through empirical analysis, it is found that BP neural network has the characteristics of strong predictive ability and high precision compared with statistical and econometric analysis methods. However, the article does not mention the problem that the traditional BP neural network is easy to fall into the local minimum. Kraus M innovatively combines genetic algorithm, rough set classification and neural network to establish a comprehensive prediction model [14], so as to make full use of the advantages of each method, make up for the

shortcomings of each method and realize data analysis and prediction. However, the article does not compare the individual prediction effects of the three methods with the prediction effect of the comprehensive model. Therefore, the paper does not verify the comprehensive model established by it, and it is not objective.

Fasoli and other scholars combined the ARCH/GARCH model with the traditional BP neural network model [15] and used the 30-year stock data of the Istanbul stock market as the research target to train the neural network and make predictions. It can be seen from the empirical that the comprehensive prediction model of ANN-APGARH effectively makes the prediction results more accurate and reliable, but it is not feasible to process the massive data.

Throughout the relevant research literature abroad, the stock market in Europe and the USA or other small Southeast Asian countries is basically the subject of research, or the statistical model is compared with the neural network, or the model algorithm is simply combined with various model algorithms. The stock market of developing countries represented by the Chinese stock market is not involved in the research of foreign scholars because of its particularity. At the same time, there is no comparative study between the stock market of developed countries and the stock market of developing countries in the experiment of neural network stock price forecasting. Moreover, the use of RNN and LSTM neural networks by foreign scholars only resides in image recognition and artificial intelligence, such as Wang R's handwritten text generation [16], Baiges J's generation of description text according to pictures [17] and so on. In addition, foreign scholars have used the LSTM neural network for the estimation of ore grade and recovery rate [18]. Therefore, from this point of view, at present, no one has applied LSTM to stock price forecasts.

3 LSTM definition and network structure

Based on the improved recurrent neural network (RNN), the LSTM deep neural network replaces the hidden layer neurons of the general recurrent neural network with a special LSTM structure. The LSTM structure consists of an input gate, an output gate, a forgetting gate and a memory unit (Cell). Moreover, the input gate, the output gate and the forgetting gate are all logical units. They do not send their output to other neurons but are responsible for setting the weights at the edges of the other parts of the neural network connected to the memory unit to correct the error function of the selective memory feedback with the gradient [19]. Its specific structure is as follows (Fig. 1):

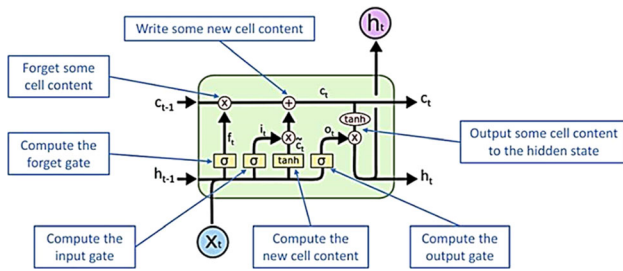


Fig. 1 Hidden layer neuron structure of LSTM deep neural network

Input gate, and output gate is used to receive and output parameters and correction parameters, which are, respectively, recorded as I, ω . The forget gate indicates whether to retain the history information stored in the current hidden layer node, denoted as ϕ . The memory cell (Cell) represents the memory of the neuron state and is denoted as s_c^t . The design of three-gate s and a separate Cell unit allows the LSTM unit to have the ability to save, read, reset and update long-distance history information [20].

In the recurrent neural network (RNN), the input values of the hidden layer neurons are composed of two parts. However, the input of the LSTM input gate consists of three parts, including the output vector of the input layer neurons, the output vector of the Cell of the previous hidden layer and the reserved information of the Cell at the previous moment. The input vector of the input gate at time t and the output vector formula obtained by the excitation function f are as follows:

$$a_i^t = \sum_{i=1}^I W_{i1}X_i^t + \sum_{h=1}^H W_{h1}b_h^{t-1} + \sum_{c=1}^C W_{c1}s_c^{t-1} \quad (1)$$

$$b_i^t = f(a_i^t) \quad (2)$$

Like the input gate, the input of the forget gate is composed of the same three parts. The input vector of the forget gate at time t and the output vector formula obtained by the excitation function f are as follows:

$$a_\phi^t = \sum_{i=1}^I W_{i\phi}X_i^t + \sum_{h=1}^H W_{h\phi}b_h^{t-1} + \sum_{c=1}^C W_{c\phi}s_c^{t-1} \quad (3)$$

$$b_\phi^t = f(a_\phi^t) \quad (4)$$

The input to the Cell unit consists of two parts, including the input vector of the input layer and the output of the output gate of the previous hidden layer. The input formula for this unit is as follows:

$$a_c^t = \sum_{i=1}^I W_{ic}X_i^t + \sum_{h=1}^H W_{hc}b_h^{t-1} \quad (5)$$

After that, the Cell unit decides whether to retain the past information according to the judgment of the forget

gate. The formula is as follows. In the formula, g is the excitation function:

$$a_c^t = b_\phi^t s_c^{t-1} + b_\phi^t g(a_c^t) \quad (6)$$

The output of the output gate consists of three parts, including the output vector of the input layer neurons, the output vector of the Cell of the previous hidden layer and the reserved information of the Cell at the current time. Therefore, the input vector of the output gate at time t and the output vector formula obtained by the excitation function are as follows:

$$a_\omega^t = \sum_{i=1}^I W_{i\omega}X_i^t + \sum_{h=1}^H W_{h\omega}b_h^{t-1} + \sum_{c=1}^C W_{c\omega}s_c^t \quad (7)$$

Finally, the output vector formula for the Cell unit is as follows. In the formula, h is the excitation function:

$$b_c^t = b_w^t h(s_c^t) \quad (8)$$

Then, like the traditional RNN, the output vector of the Cell unit is used as the input vector of the output layer of the neural network, and the final output is obtained through the excitation function.

For the backpropagation process of LSTM, the error function is denoted as L . First, the following three formulas are defined. δ_j^t represents the error signal of the j node at time t , ϵ_c^t represents the error signal output by the Cell unit at time t , and ϵ_s^t represents the error signal of the hold value in the Cell unit at time t :

$$\delta_j^t = \frac{\partial L}{\partial a_j^t} \epsilon_j^t = \frac{\partial L}{\partial b_c^t} \epsilon_c^t = \frac{\partial L}{\partial s_c^t} \quad (9)$$

Further, the output error signal formula of the Cell unit is as follows:

$$\begin{aligned} \epsilon_c^t &= \frac{\partial L}{\partial b_c^t} = \sum_{k=1}^K \frac{\partial L}{\partial a_k^t} \frac{\partial a_k^t}{\partial b_c^t} + \sum_{h=1}^H \frac{\partial L}{\partial a_h^{t+1}} \frac{\partial a_h^{t+1}}{\partial b_c^t} \\ &= \sum_{k=1}^K \delta_k^t w_{ck} + \sum_{h=1}^H \delta_h^t w_{ch} \end{aligned} \quad (10)$$

The error signal of the output gate is as follows:

$$\begin{aligned} \delta_w^t &= \frac{\partial L}{\partial a_w^t} = \frac{\partial L}{\partial b_w^t} \frac{\partial b_w^t}{\partial a_w^t} = \frac{\partial b_w^t}{\partial a_w^t} \sum_{c=1}^C \frac{\partial L}{\partial b_c^t} \frac{\partial b_c^t}{\partial b_w^t} \\ &= f'(a_w^t) \sum_{c=1}^C \epsilon_c^t h(s_c^t) \end{aligned} \quad (11)$$

The error signal formula of the Cells unit is as follows:

$$\begin{aligned} \epsilon_s^t &= \frac{\partial L}{\partial s_c^t} = \epsilon_c^t b_w^t h(s_c^t) + w_{cw} \delta_w^t + b_\phi^t \epsilon_s^{t+1} + w_{c\phi} \delta_\phi^{t+1} \\ &\quad + w_{cl} \delta_l^{t+1} \end{aligned} \quad (12)$$

$$\delta_c^t = \frac{\partial L}{\partial a_c^t} = \frac{\partial L}{\partial s_c^t} \frac{\partial s_c^t}{\partial a_c^t} = \varepsilon_s^t b'_i g'(a_c^t) \tag{13}$$

The error signal formula of forget gates is as follows:

$$\begin{aligned} \delta_\phi^t &= \frac{\partial L}{\partial a_\phi^t} = \frac{\partial L}{\partial b_\phi^t} \frac{\partial b_\phi^t}{\partial a_\phi^t} = \frac{\partial b_\phi^t}{\partial a_\phi^t} \sum_{c=1}^C \frac{\partial L}{\partial s_c^t} \frac{\partial s_c^t}{\partial b_\phi^t} \\ &= f'(a_\phi^t) \sum_{c=1}^C \varepsilon_s^t s_c^{t-1} \end{aligned} \tag{14}$$

The error signal formula of input gates is as follows:

$$\begin{aligned} \delta_i^t &= \frac{\partial L}{\partial a_i^t} = \frac{\partial L}{\partial b_i^t} \frac{\partial b_i^t}{\partial a_i^t} = \frac{\partial b_i^t}{\partial a_i^t} \sum_{c=1}^C \frac{\partial L}{\partial s_c^t} \frac{\partial s_c^t}{\partial b_i^t} \\ &= f'(a_i^t) \sum_{c=1}^C \varepsilon_s^t g(a_c^t) \end{aligned} \tag{15}$$

According to the error signal obtained above, the weight is updated, and the weight update formula is as follows:

$$w_{ij} = w_{ij} - \mu \nabla L(w_{ij}) \tag{16}$$

In the formula, $\nabla L(w_{ij}) = \frac{\partial L}{\partial w_{ij}} = \frac{\partial L}{\partial a_j^t} \frac{\partial a_j^t}{\partial w_{ij}} = \delta_j^t b_j^t$.

In summary, the training steps based on LSTM improved recurrent neural network are as follows [21]:

- (1) The t-time data feature is input to the input layer, and the result is output through the excitation function.
- (2) The input layer outputs the result, the hidden layer output at time $t - 1$ and the information stored in the Cell at time $t - 1$ are entered into the node of the LSTM structure. After that, the data are output to the next hidden layer or output layer through the processing of the input gate, output gate, forget gate and Cell units.
- (3) Output the LSTM structure node to the output layer neurons and output the result.
- (4) The error is backpropagated, and the individual weights are updated.

4 Optimization method of neural network training

A major difficulty in the training of deep neural networks is the optimization of the training process. There are many algorithms for solving optimization problems. (The most common is gradient descent.) The focus of the optimization problem is on which iterative method is used to iterate to optimize the learning rate, allowing the network to optimize with the fastest number of trainings and preventing over-fitting.

First, for sample i , we assume that the system parameter is w and its cost function is $C_i(w)$. On the training set of n samples, the overall cost function is:

$$C(w) = \sum_{i=1}^n C_i(w) \tag{17}$$

The core of the optimization method is to find w by iterative method, so that the cost function is the smallest.

This article focuses on the following three optimization methods:

At present, stochastic gradient descent (SGD) generally represents mini-batch gradient descent. SGD is the gradient of mini-batch calculated every iteration, and then, the parameters are updated, which is the most common optimization method. Its formula is as follows:

$$w_{t+1} = w_t - \eta \sum_{j=1}^m \nabla C_{ij}(w) \tag{18}$$

In the formula, η represents the learning rate, and $C_{ij}(w)$ represents the cost of the j th sample in the i th mini-batch.

From the above, SGD uses the same learning rate for all parameter updates, and it is more difficult to select the appropriate learning rate for all parameters. Therefore, scholars have further developed some adaptive learning rate optimization methods based on SGD.

RMSprop sets different learning rates for each dimension parameter and does not need to set the basic learning rate, but adaptively constrains the learning rate.

First, the expected G is maintained, and the square of the gradient in the previous iteration is described:

$$G_t = \gamma(G_{t-1}) + (1 - \gamma)\nabla C(w)_t^2 \tag{19}$$

The learning rate is constrained by G , and the parameter update formula is as follows. In the formula, d represents the dimension:

$$w_{t+1}^d = w_t^d - \frac{\eta}{G_{t+1}^d} \nabla C(w) \tag{20}$$

Adam (adaptive moment estimation) is essentially an RMSprop with a momentum term, which dynamically adjusts the learning rate of each parameter by using the first moment estimation and the second moment estimation of the gradient. The main advantage of Adam is that after the offset correction, each iteration learning rate has a certain range, which makes the parameters relatively stable. The formula is as follows:

A first-order momentum is maintained:

$$m_t = \alpha m_{t-1} + (1 - \alpha)\nabla C(w) \tag{21}$$

A second-order momentum is maintained:

$$v_t = \beta v_{t-1} + (1 - \beta)\nabla C(w)^2 \tag{22}$$

Since both m and v are initialized to 0, we use the power of t to make it a bit larger in the first few iterations:

$$\hat{m}_t = \frac{m_t}{1 - \alpha^t}, \hat{v}_t = \frac{v_t}{1 - \beta^t} \tag{23}$$

Finally, the parameter update formula is as follows:

$$w_{t+1} = w_t - \eta \frac{1}{\sqrt{\hat{v}_t}} \hat{m}_t \tag{24}$$

5 Over-fitting solution

In deep neural networks, as the number of layers and the number of neurons in each layer increase, the number of parameters in the model will increase at an extremely fast rate. When the number of parameters is large, there is a problem of over-fitting. The so-called over-fitting means that the model has a good fitting effect on the training data, but its effect on the verification set is very poor, that is, the model generalization ability is poor. Deep neural network models are prone to over-fitting. At present, the following two methods are mainly used to solve the over-fitting problem.

Dropout is a technique that Hinton proposed in 2012 to prevent over-fitting of deep neural network models. Dropout allows the model to randomly “discard” hidden layer neurons with probability p at each training, as shown in Fig. 2b. The weights of these discarded neurons will not be updated during this training, but all neurons will be used when the model is used.

L2 regularization is also known as weight attenuation. For deep learning, over-fitting due to excessive features often occurs. To do this, we can reduce the weight of features or punish unimportant features. The L2 regularization method prevents over-fitting and improves the generalization of the model by adding an extra item to the cost function:

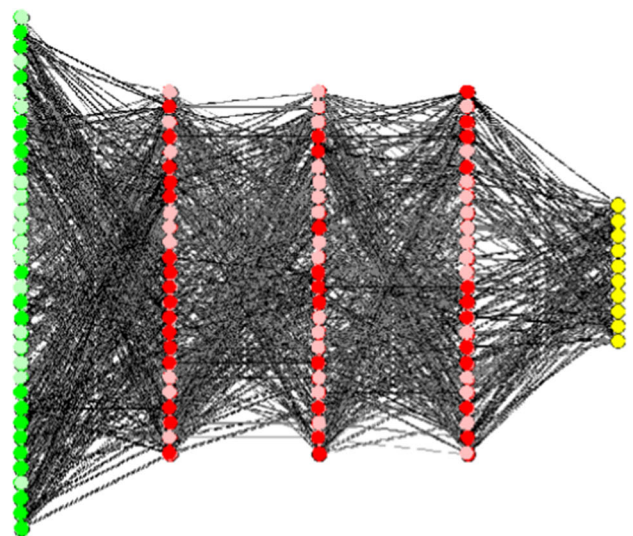
$$C = c_0 + \frac{\lambda}{2n} \sum_w w^2 \tag{25}$$

The following formula is obtained by performing the partial derivative:

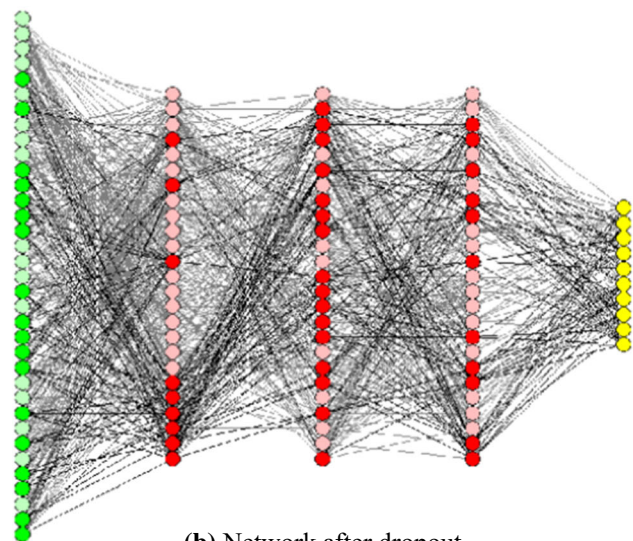
$$\frac{\partial C}{\partial w} = \frac{\partial c_0}{\partial w} + \frac{\lambda}{n} w \tag{26}$$

Furthermore, the expression of the gradient descent is as follows:

$$\begin{aligned} w &= w - \eta \frac{\partial c_0}{\partial w} - \frac{\eta \lambda}{n} w \\ &= \left(1 - \frac{\eta \lambda}{n} \right) w - \eta \frac{\partial c_0}{\partial w} \end{aligned} \tag{27}$$



(a) Network before dropout



(b) Network after dropout

Fig. 2 Dropout technology

The L2 regularization has the effect of making w “small.” In a sense, a smaller weight w means that the network is less complex and the data fit just fine.

In summary, it can be seen that the improved LSTM deep neural network based on recurrent neural network (RNN) can effectively consider the timing of input data and can solve the problem that gradient recurrence and gradient explosion are easy to occur in traditional recurrent neural networks. However, due to the complexity of the deep neural network structure, its training often requires various optimization techniques to speed up its training process and solve the often over-fitting problems.

6 Model design

When constructing the LSTM deep neural network prediction model, we need to consider the following parameters:

- (1) Input layer and output layer selection: This article uses the daily data of the previous N days to predict the average price increase and decrease in the next 3 days (a big rise, a small increase of 1 and a big drop of 0). There are four types of input variables, and a total of 28 input variables are available for selection. In the experiment, the input variables are filtered and set to X ($X \leq 28$). The number of neurons in the output layer is three, which is the category of the stock price increase and decrease next day. Therefore, the number of input layer neurons is the number of features X input, and the number of neurons in the output layer is 3.
- (2) Hidden layer selection: There is currently no universally accepted method for selecting the hidden layer of the network, that is, how to select the number of hidden layers and the number of nodes. Different network structures need to be determined for different problems. It is generally believed that obtaining a lower error by increasing the number of hidden layer neurons is easier to achieve than increasing the number of hidden layers. In addition, considering the length of the time series itself, two hidden layers are used in this paper. On the other hand, the number of hidden layer nodes has a direct relationship with the complexity of solving the problem and the number of input and output neurons. If the number of hidden layer nodes is too small, sufficient number of connection weights cannot be generated to satisfy the sample learning. If the number of hidden layer nodes is too large, the generalization ability of the network becomes poor. The example shows that the network training effect is the best when the number of two hidden layer nodes in the dual hidden layer network is similar. This paper sets the initial hidden layer node number to 136. At the same time, based on this, the number of hidden layer nodes is increased or decreased to study the influence of the model structure on the accuracy of the model.

First, the initial model is assigned the necessary parameters: The time series length is set to 30, and the number of hidden layer nodes is set to 136. At the same time, RMSprop is selected as the optimization method, and categorical cross-entropy is selected as the loss function. The batch size is set to 32, and the epoch is set to 30. The

model is a multi-classification model, and the model accuracy evaluation uses the default accuracy index (accuracy). Moreover, accuracy measures the correct proportion of the classification and represents the performance of the model. If \hat{y}_i is assumed to be the prediction category of the n th sample and y_i is the real category of the n th sample, the accuracy rate on n samples is:

$$\text{accuracy} = \frac{1}{n} \sum_{i=1}^n 1(\hat{y}_i = y_i) \quad (28)$$

In the formula, $1(x)$ means that the value is 1 when the prediction result is consistent with the real result, and the other case is 0. On this basis, this paper intends to establish multiple comparison models to study the influence of training samples, network structure and optimization methods on the accuracy of the model, and then establish an optimal prediction model.

- (1) Training samples: First, for the input features, the transaction basic data, technical indicator data, transaction basic data + market data, technical indicator data + market data, transaction data + market data + technical indicators and all indicators are input. Moreover, the four models are compared to select the best input features. The principal component analysis was used to reduce the dimensionality of the 28 indicator data, and the processing result was used as the input variable of the LSTM deep neural network. It is compared to the corresponding non-dimensionality reduction model to illustrate the need for dimensionality reduction and ultimately to select the best input feature data. Furthermore, for the length of the time series sequence, the sequence length of the time series is changed to study the influence of the length of the time series on the accuracy of the model. Finally, for the number of samples, the number of training samples is changed to perform model training separately to study the effect of the length of the time series on the accuracy of the model.
- (2) Model structure: On the basis of the optimal model selected in the previous step, by increasing and decreasing the number of hidden layer neurons, the influence of the number of different hidden layer neurons on the accuracy of the model is studied.
- (3) Optimization method: The SGD, RMSprop and Adam methods are used to optimize the network training process, so as to study the influence of different optimization methods on the accuracy of the model and finally select the optimization method most suitable for LSTM deep neural network.
- (4) The data are subdivided into bullish data and bear market data and input into the model for training to

compare the effects of the models in different market conditions. Since the sample data in each state are reduced compared to before in the market segment state, it is expected that over-fitting may occur. Therefore, the model is optimized by dropout and L2 regularization techniques.

- (5) Using the BP neural network, the traditional RNN is used for training and compared with the results of the LSTM deep neural network to illustrate the higher prediction accuracy of the LSTM deep neural network.

7 Model verification

The raw data in this paper are collected from the CSMAR database, and the example of data is shown in Table 1. The raw data mainly include basic trading indicators, financial indicators and the Shanghai and Shenzhen 300 Index. The constituent stocks included in the Shanghai and Shenzhen 300 Index are the main constituent stocks of the Shanghai and Shenzhen Stock Exchanges, reflecting the basic operation of the Shanghai and Shenzhen stock markets. In view of the large differences in China's stock market system around 2005, this paper selects the Shanghai and Shenzhen 300 Index and its constituent stocks from January 1, 2008, to January 19, 1919.

This article uses the data from the previous 30 days to predict the rise and fall of the average closing price of the stock in the next three days (a big rise of 2, a small increase of 1 and a big fall of 0). First, 300 stock data were initially screened. Among them, ten stocks listed in 2005, the company's financial indicators data are missing, so the stocks with missing data are removed, and 290 stock data are retained. Secondly, since the original data only have basic transaction data and financial indicator data, it is necessary to calculate technical indicators and forecast targets based on the basic transaction data, that is, the average price of the stocks in the next three days. In addition, the CSI 300 index needs to be consolidated based

on the date and individual stock data. Furthermore, the max–min method is used to perform dimensionless processing on 28 types of feature data, and the data are segmented into standard input data with a sequence length of 30, and finally 592,756 samples are obtained.

First, the average price of each stock in the next three days is calculated, and the average price of the next three days on the t th day is set to 3 t_c . Furthermore, in order to make the number of each type of training samples similar, the front and rear tertiles of all stock ups and downs (AC) are selected, which are recorded as 0.33AC and 0.67AC.

If $\bar{c}_{t3} < 0.33AC$, the sample is marked as 0. If $0.33AC \leq \bar{c}_{t3} < 0.67AC$, the sample is marked as 1. If $0\bar{c}_{t3} \geq 0.67AC$, the sample is marked as 2. Finally, the data are randomly scrambled, and then, 80% of them are taken as training data, and 75% of them are further used as train_data (355653) to train the model, and 25% of them are used as verification data val_data (118551) to select a comparison model. The remaining 20% is used as the final test data test_data (118552) to test the accuracy and stability of the model.

There are many factors that affect stock prices. However, in summary, they can be broadly divided into four categories: The one category is transaction data related to stock trading, such as opening price and closing price. The second category is statistical technical indicators derived from transaction data, such as MACD, KDJ and turnover rate. The third category is the stock price index related to the overall situation of the stock market. This paper selects the Shanghai and Shenzhen 300 Index which can reflect the overall trend of the Shanghai and Shenzhen stock markets. The fourth category is the company's financial indicators, such as price–earnings ratio, price-to-book ratio and other indicators. Based on these four types of indicators, the following six comparison models are established to select the best input characteristics.

It can be seen from Table 2 that the model accuracy rate of the verification set of the model M1 with the transaction basic indicator as the input variable is 51.94%, which is 19 percentage points higher than the accuracy of the random model 33.33%. Therefore, it can be concluded that it is effective to predict the rise and fall of the stock price based on the basic indicators of the transaction. The model accuracy rate of the verification set of the model M2 with the technical index as the input variable is 44.36%. Comparing M1 and M2, it can be found that the accuracy of the model M1 with the transaction basic indicator as the input variable is 7 percentage points higher than the accuracy of the model M2 with the technical indicator as the input variable. Moreover, these technical indicators are calculated through various trading basic indicators. It can be concluded that although the current technical indicators have a certain value for predicting the stock price, the

Table 1 Raw data samples

Code	Date	Open	High
sh601016	2019/1/19	8.16	8.17
Low	Close	Change	Volume
7.95	8.07	– 0.02063	6,465,962
Turnover	PE	PS	PC
0.005785	100.2883	12.3349	18.14478
PB			
2.66363			

Table 2 Influences of input characteristics on model accuracy

Model	Dimension	Feature selection	Accuracy (%)
M1	6	I Basic trading indicators	51.94
M2	12	II Technical indicators	44.36
M13	12	I Basic trading indicators, III Shanghai and Shenzhen 300 Index	62.46
M23	18	II Technical indicators, III Shanghai and Shenzhen 300 Index	60.58
M123	24	I Basic trading indicators, II Technical indicators, III Shanghai and Shenzhen 300 Index	61.00
M1234	28	I Basic trading indicators, II Technical indicators, III Shanghai and Shenzhen 300 Index, IV Financial indicator	60.70

technical indicators have loss information for the original trading indicators. From Figs. 3 and 4, it can be concluded that the accuracy of the M2 model has been oscillating and unstable compared to M1, which may be related to the information noise contained in the selected technical indicators. On the other hand, China’s stock market has a 10% ups and downs limit, which leads to a smoother basic trading indicator data, while technical indicators have no such restrictions. Therefore, it can be inferred that the volatility of the model M2 may be related to the large variation of the technical indicator data itself. In addition, it can be seen from Figs. 3 and 4 that the model M1 has a slight over-fitting phenomenon, and the model M2 has no over-fitting. Considering that the training samples of the two models are all 470,000, the number of samples is enough, the model M1 has six dimensions, and the model M2 has 12 dimensions, it can be explained that when the

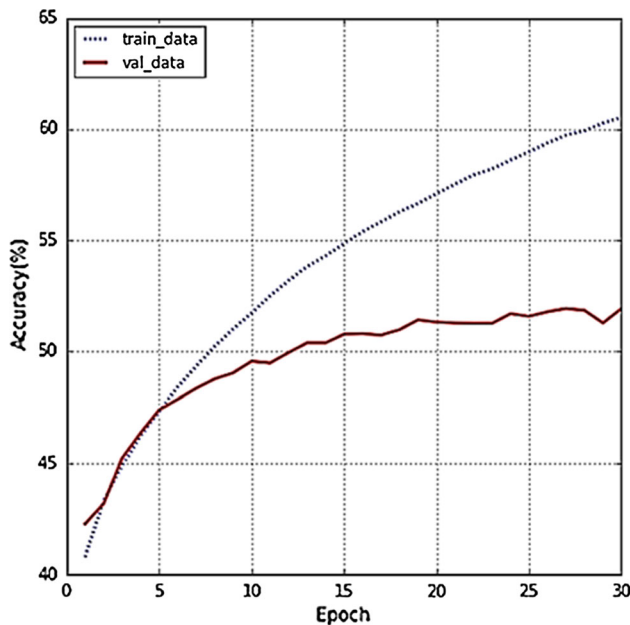


Fig. 3 Results of model M1

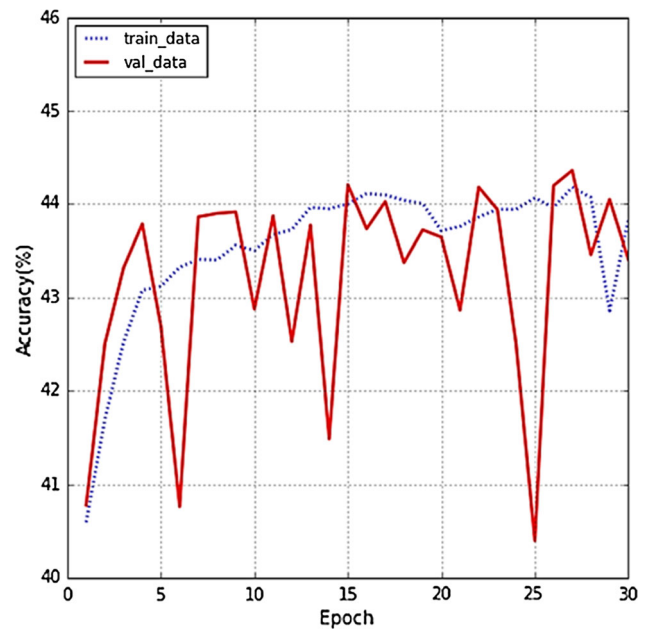


Fig. 4 Results of model M2

input data dimension is too small, the over-fitting problem is easy to occur.

As can be seen from Table 2 and Fig. 5, the accuracy of the verification set of the model M13 with the transaction basic indicator and the CSI 300 index as input variables is 62.46%. It is 11% higher than the accuracy of the model M1 with the transaction basic indicator as the input variable. Therefore, it can be explained that the CSI 300 index has a great influence on the stock price trend of individual stocks. In addition, as can be seen from Fig. 6, the accuracy of the verification set of the model M23 with the technical index and the CSI 300 index as input variables is 60.58%. Compared with the accuracy of the model M2 with technical indicators as input variables, the accuracy rate is increased by 16 percentage points. It again shows that the Shanghai and Shenzhen 300 Index has a great impact on the stock price trend of individual stocks. By comparing the model M13 with the model M1 and comparing the

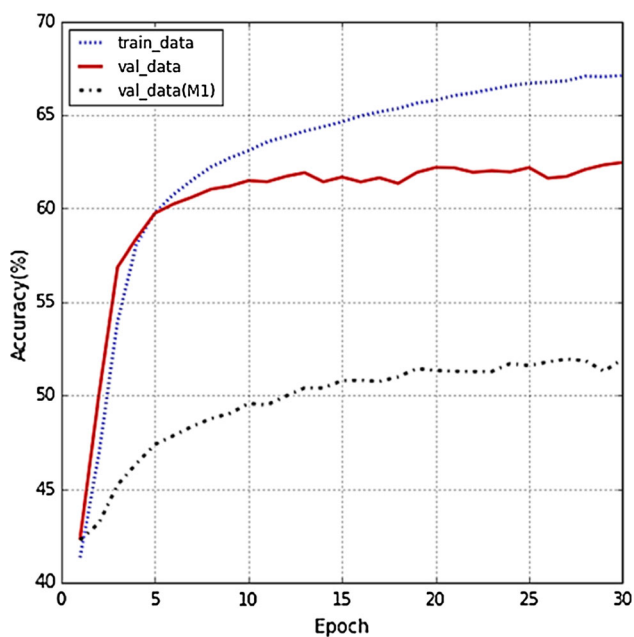


Fig. 5 Results of model M1

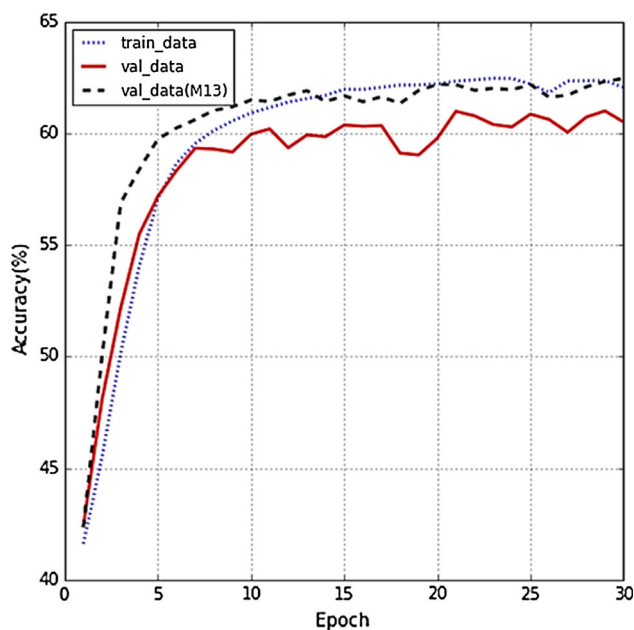


Fig. 7 Results of model M123

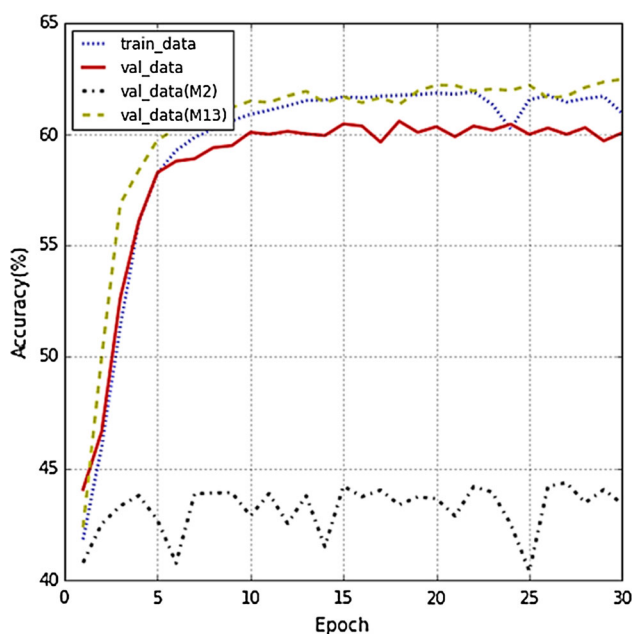


Fig. 6 Results of model M23

model M23 with the model M2, it can be found that an appropriate increase in the amount of information of the input variable helps to improve the accuracy of the model.

As can be seen from Table 2 and Fig. 7, the accuracy of the verification set of the model M123 with the transaction basic indicators, technical indicators and the CSI 300 index as input variables is 61.00%, which is lower than the accuracy of the model M13. Therefore, it can be concluded that the redundancy of the input information leads to a

decrease in the accuracy of the model. It can be seen from Table 2 and Fig. 8 that the accuracy of the verification set of the model M1234 with the transaction basic index, technical index, CSI 300 index and business surface index as input variables is 60.70%, which is slightly lower than that of M123. Therefore, it can be concluded that the company’s financial indicators have no reference value for the short-term forecast of the stock price. In addition, it can be seen that with the increase in epoch, the model M1234 with a large input data dimension has a rapid decline in the

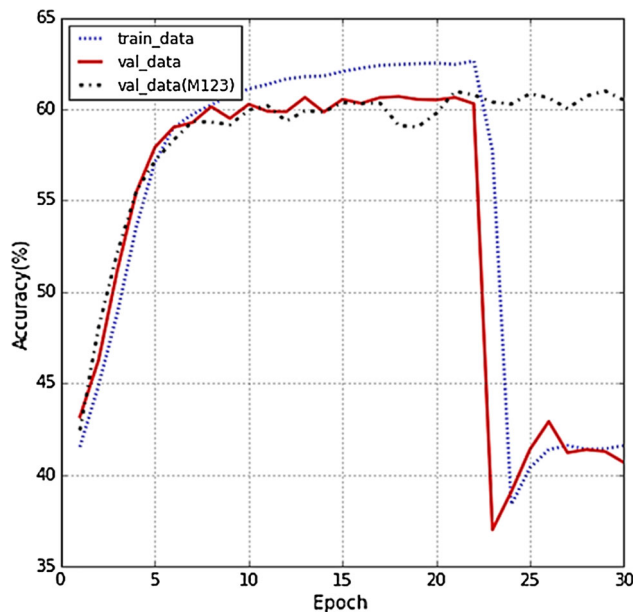


Fig. 8 Results of model M1234

accuracy of the training set and the verification set. Therefore, it can be concluded that when the input data dimension of the model is large and the input data information is redundant, a gradient explosion is likely to occur.

In summary, when the dimension of the input data is small, appropriately increasing the effective information can improve the accuracy of the model. However, when the input data dimension itself is large, continuing to increase the information may result in a decrease in the accuracy of the model. Therefore, it can be concluded that for deep neural networks, when there is a difference in the amount of information of the input data, the accuracy of the input vector group model containing more information is not necessarily higher. Generally speaking, the more information is obtained, the more accurate the judgment is. However, due to the characteristics of the neural network, more input variables are more likely to cause the network to learn confusion due to too many variables and reduce the learning effect.

The optimization of the training process is a problem that will be encountered in the training process of each neural network model. There are many ways to solve optimization problems. The most common is the gradient descent. The focus of the optimization problem is on which iterative method is used to iterate to optimize the learning rate, which allows the network to optimize with the fastest number of trainings and prevent over-fitting. In this paper, the SGD optimization method (M13-SGD) and Adam (M13-Adam) optimization method are compared with the RMSprop optimization method adopted by the basic model.

It can be seen from Table 3 that the model M13-SGD with the SGD optimization method has a model accuracy lower than the model M13 with the RMSprop optimization method by 1 percentage point. However, the model accuracy of the model M13-Adam using the Adam optimization method is not significantly different from the accuracy of the model M13 using the RMSprop optimization method. In addition, as can be seen from Fig. 9, the model M13-SGD converges from the 20th epoch model, whereas the model M13 converges from the fifth epoch model. Therefore, it can be concluded that for the LSTM deep neural

Table 3 Influences of optimization method on model accuracy

	Optimization method	Accuracy (%)
<i>Model</i>		
M13-SGD	SGD	61.40
M13-Adam	Adam	62.08
<i>Contrast model</i>		
M13	RMSprop	62.46

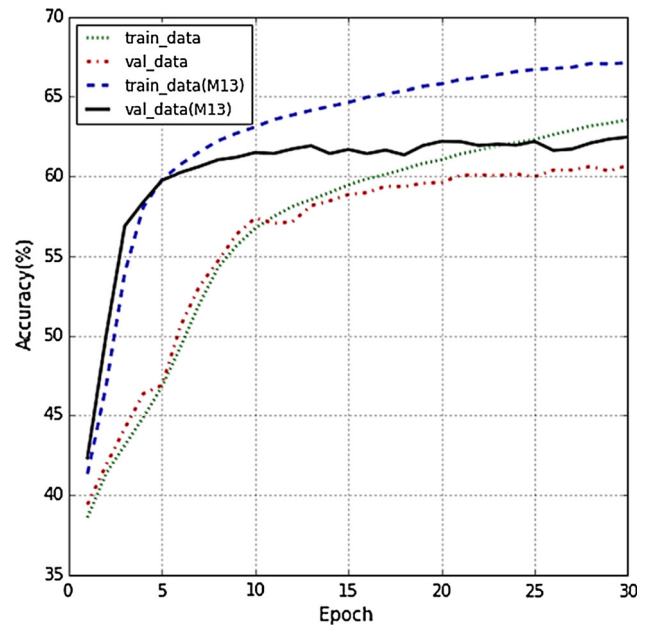


Fig. 9 Results of the model M13-SGD

network, the RMSprop optimization method converges faster than the SGD optimization method. It can be seen from Fig. 10 that the Adam optimization method and the RMSprop optimization method are used for training, and the model has little difference in convergence speed and model accuracy. However, when using the Adam method for training, the model tends to over-fitting as the epoch increases. Therefore, for the LSTM deep neural network, when the sample size is small, the RMSprop optimization method is suitable.

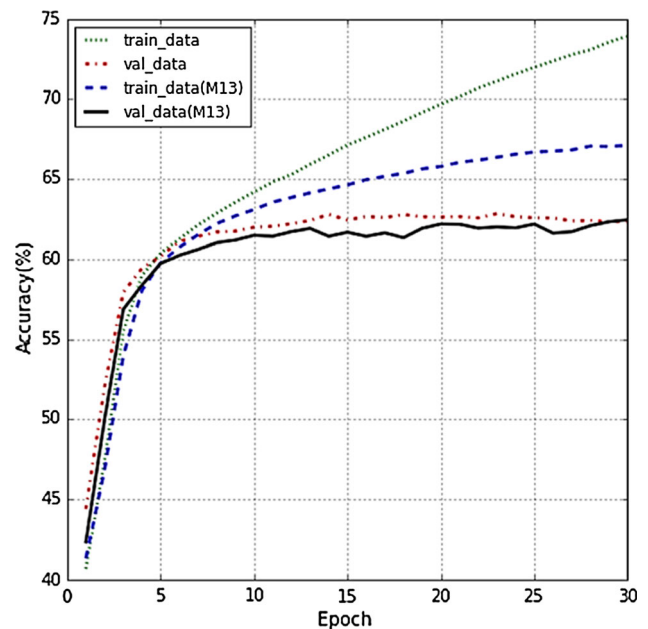


Fig. 10 Results of the model M13-Adam

After a comparative study of training samples, model structure and optimization methods, the optimal model M13 is selected according to the accuracy of the verification set, and the test set data are used to finally verify the effect of the model. After verification, the accuracy of the test set was 62.31%, which was only slightly lower than the accuracy of the verification set of 62.46. The stability of the optimal model is further proved.

From the above empirical evidence, it can be concluded that the LSTM deep neural network is effective for predicting the stock market time series. Further, we compare it with the traditional RNN and BP neural networks. As can be seen from Table 4 and Fig. 11, the accuracy of the traditional RNN model is only 37.38%, which is 25 percentage points lower than the accuracy of the LSTM model. In addition, as the number of training increases, the RNN model hardly improves, and the model disappears. However, the LSTM model is rapidly improving. It can be concluded that LSTM can solve the problem that the traditional RNN easily disappears. As can be seen from Table 4 and Fig. 12, the accuracy of the traditional BP model is only 42.41%, which is 20% lower than the accuracy of the LSTM model but 5 percentage points higher than the traditional RNN. Further, it can be concluded that although the RNN is theoretically time series, when the timing increases, the RNN is prone to the gradient disappearing problem, and the RNN does not have any advantage at this time. From the literature, BP neural network has achieved good results when BP neural network is used to predict single stocks or small samples. However, in the case of the 590,000 large sample of this paper, the prediction accuracy of BP neural network is lower than that of LSTM deep neural network, which also shows that deep neural network has advantages for large sample training learning.

In summary, compared with the traditional RNN and BP networks, LSTM deep neural network can effectively solve the gradient disappearance problem and improve the prediction accuracy, which has obvious advantages for the prediction of stock market time series.

Table 4 Results of different network models

	Network type	Accuracy (%)
<i>Model</i>		
M13-RNN	Traditional RNN	37.38
M13-BP	Traditional BP	42.41
<i>Contrast model</i>		
M13	LSTM	62.46

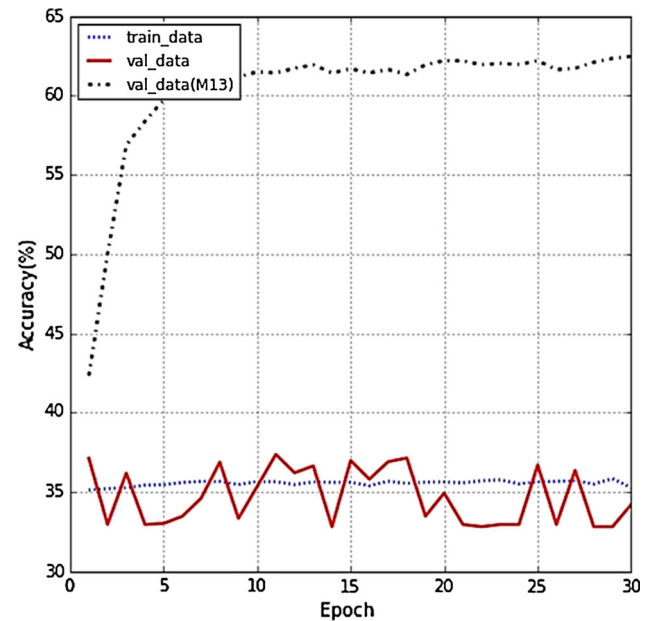


Fig. 11 Results of the model M13-RNN

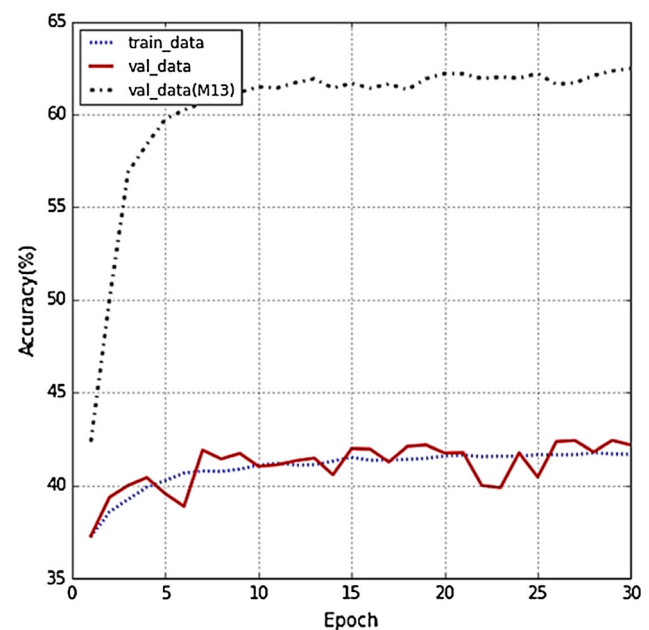


Fig. 12 Results of model M13-BP

8 Conclusion

In this paper, the LSTM deep neural network is used to model and predict the data of the Shanghai and Shenzhen 300 Index constituents from 2008 to 2019, and the three types of factors affecting the prediction accuracy of the model are systematically studied. Finally, a high-precision stock market time series short-term prediction model based on LSTM deep neural network is constructed.

In terms of training samples, this paper examines the choice of input features, the sequence length of time series and the influence of the number of samples on the prediction accuracy of the model. According to the input characteristics, this paper comprehensively considers the factors affecting the stock price rise and fall and extracts four types of factors: basic trading indicators, technical indicators, CSI 300 index and financial indicators, and uses various characteristics and combinations as input variables of the network. The study found that the basic trading index and the CSI 300 index as the input variable model have the highest prediction accuracy, and the CSI 300 index has a great influence on the trend of individual stocks. In addition, the principal component analysis method is used to reduce the dimension and establish a contrast model. It is found that for data with high data dimension, the principal component dimensionality reduction can be improved under the premise of ensuring the preservation of complete information, which can improve the accuracy of the model, accelerate the convergence speed of the model and solve the gradient explosion phenomenon caused by high-dimensional input. However, when the dimension of the input variable is low, even if the principal component is reduced in the case of retaining complete information, the accuracy of the model cannot be improved, and the fitting phenomenon may occur because the dimension is too small. According to the length of the sequence, the length of the sequence is changed to train the model. The length of the sequence is too short. The deep neural network model cannot completely learn the hidden rules in the time series, which leads to the decrease in the accuracy of the model. If the sequence length is too long, the accuracy of the model will not increase, and the gradient explosion phenomenon will easily occur, and the training time of the model will be greatly increased. According to the number of samples, the model is trained by increasing and decreasing the number of samples. It is found that the influence of sample size on the convergence speed of the model is obvious, the sample size is too small, the model convergence speed is slow, and the model accuracy is also reduced. When the number of samples reaches a certain amount, continuing to increase the number of samples does not increase the prediction accuracy of the model.

Finally, this paper compares BP neural network, traditional RNN and RNN improved LSTM deep neural network. It proves that LSTM deep neural network has higher prediction accuracy and can effectively predict stock market time series.

In summary, although the LSTM deep neural network is effective for predicting time series, it can effectively solve the problem of gradient disappearance of traditional RNN, but the current prediction accuracy still needs to be

improved. There are many noises in the stock market time series, and there are many factors affecting its trend. However, this paper only selects indicators that are easy to quantify as input data. Adding more valid information may further improve the accuracy of the model, such as quantifying the policy and message surface data as input to the model. In addition, the stock market can be further broken down by industry.

Compliance with ethical standards

Conflict of interest The authors have no competing interests.

References

- Liu S, Borovykh A, Grzelak LA et al (2019) A neural network-based framework for financial model calibration. *J Math Ind* 9(1):9
- Cao J, Wang J (2019) Stock price forecasting model based on modified convolution neural network and financial time series analysis. *Int J Commun Syst* 32(1):e3987
- Liu S, Oosterlee CW, Bohte SM (2019) Pricing options and computing implied volatilities using neural networks. *Risks* 7(1):16
- Sezer OB, Ozbayoglu AM (2018) Algorithmic financial trading with deep convolutional neural networks: time series to image conversion approach. *Appl Soft Comput* 70:525–538
- Hien LV, Hai-An LD (2019) Exponential stability of positive neural networks in bidirectional associative memory model with delays. *Math Methods Appl Sci* 42(18):6339–6357
- Khemakhem S, Boujelbene Y (2018) Predicting credit risk on the basis of financial and non-financial variables and data mining. *Rev Account Finance*. <https://doi.org/10.1108/RAF-07-2017-0143>
- Maciel L, Ballini R, Gomide F (2017) Evolving possibilistic fuzzy modeling for realized volatility forecasting with jumps. *IEEE Trans Fuzzy Syst* 25(2):302–314
- Ahmed M, Afzal H, Majeed A et al (2017) A survey of evolution in predictive models and impacting factors in customer churn. *Adv Data Sci Adapt Anal* 9(3):1750007
- Kim S, Kang M (2019) Financial series prediction using Attention LSTM. arXiv preprint [arXiv:1902.10877](https://arxiv.org/abs/1902.10877)
- Hu R (2019) Deep learning for ranking response surfaces with applications to optimal stopping problems. *Quant Finance*. <https://doi.org/10.1080/14697688.2020.1741669>
- Tsantekidis A, Passalis N, Tefas A, Kannianen J, Gabbouj M, Iosifidis A (2018) Using Deep Learning for price prediction by exploiting stationary limit order book features. arXiv preprint [arXiv:1810.09965](https://arxiv.org/abs/1810.09965)
- Demidov RA, Zegzhda PD, Kalinin MO (2018) Threat analysis of cyber security in wireless adhoc networks using hybrid neural network model. *Autom Control Comput Sci* 52(8):971–976
- Hong-Yan R, Wei WU, Qiao-Xuan LI et al (2018) Prediction of dengue fever based on back propagation neural network model in Guangdong, China. *Chinese J Vector Biol Control* 29(03):221–225
- Kraus M, Feuerriegel S (2017) Decision support from financial disclosures with deep neural networks and transfer learning. *Decis Support Syst* 104:38–48

15. Fasoli D, Panzeri S (2017) Optimized brute-force algorithms for the bifurcation analysis of a spin-glass-like neural network model. arXiv preprint [arXiv:1705.05647](https://arxiv.org/abs/1705.05647)
16. Wang R, Liu H, Feng F et al (2017) Bogdanov-Takens bifurcation in a neutral BAM neural networks model with delays. *IET Syst Biol* 11(6):163–173
17. Baiges J, Codina R, Castañar I et al (2019) A finite element reduced order model based on adaptive mesh refinement and artificial neural networks. *Int J Numer Methods Eng* 121(4):588–601
18. Blien U, Lindner HG (1993) Neuronale Netze-Werkzeuge für empirische Analysen ökonomischer Fragestellungen Neural/Networks-Tools for Empirical Economics. *Jahrbücher für Nationalökonomie und Statistik* 212(5–6):497–521
19. Yang Z, Zhang K, Liang Y, et al Single Image super-resolution with a parameter economic residual-like convolutional Neural Network. 2017
20. Zhongfu W, Yanhong F (2018) Evaluation model of economic competitiveness based on multi-layer fuzzy neural network. *Cluster Comput* 39:1–8
21. Li C, Yu X, Huang T et al (2017) Distributed optimal consensus over resource allocation network and its application to dynamical economic dispatch. *IEEE Trans Neural Netw Learn Syst* 99:1–12

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.