Predicting Stock Prices Using Hybrid LSTM and ARIMA Model

Chi Ma, Jie Wu*, Hui Hu, YueNai Chen, JingYan Li

Abstract—**With the continuous development of China's financial market and the gradual improvement of the financial system, investors are increasingly interested in participating in investments. At the same time, there is a strong demand for accurate and effective financial information services. In recent years, many researchers have begun to pay attention to the financial field and try to use a variety of artificial intelligence technologies to solve the problem of financial data prediction in this field. Aiming at the problem of stock trend prediction, we present a hybrid model based on Long Short-Term Memory (LSTM) and Auto Regressive Integrated Moving Average (ARIMA) approach to predict stock prices. Through sample training and experiments in multiple data sets and different market stages, it is shown that the proposed Mixture model has higher prediction accuracy and stability than the single deep learning model, ARIMA model after wavelet transform and the models in the references, and has a stronger market adaptability.**

Index Terms—**LSTM, ARIMA, predicting stock prices, prediction model, time series**

I. Introduction

FORECASTING future stock trends is one of the most attractive research questions in the current academic and investment fields[1]. The vigorous development of computer technology and the increasingly perfect technology system have laid the foundation for meeting the demand of this kind of financial information. With the continuous popularization of computer technology applications, methods and models based on artificial intelligence and machine learning are gradually applied to the research of this problem. Researchers have constantly put forward some good performance methodological models to try to predict the future trend of individual stocks or certain groups of specific portfolios, such as index, fund, etc[2], [3].

However, the domestic stock market is huge and there are many kinds of stocks. Therefore, we construct a stock data set with certain universality and market representativeness as the research object, and the research of model method is carried out on the basis.

In recent years, shallow machine learning methods such as Neural Network (NN) and Support Vector Machine (SVM) have been widely used in the classification and regression of financial time series data. Since the excellent performance, both NN and SVM models have been successfully applied to the problem of finding trends in financial products. Among them, Sureshkumar and Elango used Artificial Neural Network (ANN) to predict stock prices and evaluate the performance of the neural network system in stock forecasting [4]. Cao and Tay used support vector machine to forecast the prices of financial products such as stocks and bonds in the financial market [5].

In order to further improve the accuracy of trend prediction of financial products, some improved algorithms and learning strategies have been proposed and applied to the actual environment [6]–[10]. At the same time, it has been pointed out in [11] that using more complex machine learning methods can achieve higher accuracy than shallow machine learning. With the development of deep learning theory and technology, more and more researchers are trying to combine deep learning with stock trend prediction [12]-[15]. Compared with shallow learning, deep learning has higher accuracy, stronger learning ability and more complex and comprehensive description ability of abstract ideas[16]-[18]. These excellent characteristics indicate that the deep learning methods and models will perform better in predicting the future trend changes of stocks.

In addition, among many tools in the field of technical analysis, ARIMA as a traditional time series analysis and prediction model, is also well used in the analysis and prediction of stock time series data. Some research results show that ARIMA model can achieve good prediction accuracy under the premise of high processing efficiency [19], [20].

Aiming at the prediction of stock time series data, based on the research of traditional prediction models, this paper proposes a hybrid model combines depth learning and the DB-ARIMA model which combines the wavelet denoising method and ARIMA. We have constructed stock time series data set through screening and verification historical trading data of representative stock and index samples. We collected 7 years of historical data of Shanghai Stock Exchange Index, Shenzhen Stock Exchange Index, Shanghai-Shenzhen 300

Manuscript received March 16, 2023; revised December 29, 2023.

This paper is supported by Foundation of Guangdong Educational committee under the Grant No. 2022ZDZX4052, 2021ZDJS082, No.2019KQNCX148.

Chi Ma is an associate professor of School of Computer Science and Engineering, Huizhou University, Huizhou, 516007, Guangdong, China (e-mail: machi@hzu.edu.cn).

Jie Wu is an associate professor of the School of Computer and Software Engineering, University of Science and Technology Liaoning, Anshan, 114051, Liaoning, China (*corresponding author to provide e-mail: wujieaa@163.com).

Hui Hu is an associate professor of School of Computer Science and Engineering, Huizhou University, Huizhou, 516007, Guangdong, China (e-mail: Huhui@hzu.edu.cn).

YueNai Chen is a lecturer of School of Mathematics and Statistics, Huizhou University, Huizhou, 516007, Guangdong, China (e-mail: chenyuenai@163.com).

JingYan Li is a lecturer of School of Computer Science and Engineering, Huizhou University, Huizhou, 516007, Guangdong, China (e-mail: LJY@hzu.edu.cn).

Index and ten stocks with the highest weight in the Shanghai-Shenzhen 300 Index.

The rest of this paper is organized as follows: In section 2, the construction of the hybrid LSTM and DB-ARIMA model is introduced in detail. Section 3 and 4 present the experimental work and results of the proposed model. Finally, conclusions and comments on further research are given in Section 5.

II. Methodology

A. Stock Price Prediction Model Based on LSTM

Based on LSTM, this article constructs a stock price and index trend prediction model. The "closing price" of the index and individual stocks is regarded as the representative attribute of the stock trend and the characteristics of the time series constructed. The modeling process based on LSTM network is essentially the process of designing the structure of LSTM neural network. Therefore, the specific design process of modeling mainly consists of the following five parts.

Constructing training input and output

The training process of LSTM model follows the supervised learning process. Thus, we need to design the corresponding training set. Fig. 1 is a training set data structure for financial time series prediction.

In the Fig. 1, t represents time steps, which representing a lag period. When the value of t is 1, it means that the training output is one day after the training input. For example, if the training inputs the closing price values of the first day, the second day, the third day and the nth day, then the training output is a sequence of training inputs with a lag of one day, that is, the data of the second day, the third day, the fourth day and the nth + 1 day. Again, for example, if the training input is the closing price sequence of samples in a certain period of time, then the corresponding training output is the closing price sequence of samples after a certain time lag (that is, the future closing price sequence of samples). Such input and output training are intended to train the correlation and regularity before and after LSTM network learning sample sequence. In this paper, we use the closing price of the sample as the prediction target, and the input and output of the training model are all one-dimensional matrices.

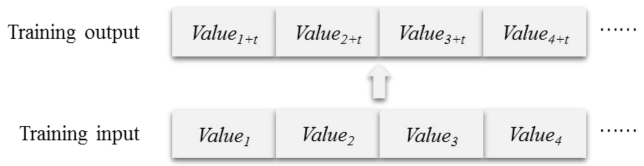

Fig. 1. Schematic diagram of LSTM model training set

Design of LSTM Network Structure

The LSTM network structure for stock timing prediction is shown in Figure 2. the "gate" structure of LSTM network is in the hidden layer. In this paper, the dimension of input and output layer is consistent with training input and output, and they are all one-dimensional.

Selection of activation function for LSTM network

In the traditional LSTM network construction, tanh function is usually chosen as the activation function of the network. However, it has the characteristics of gradually increasing or decreasing with the independent variables, and the corresponding function values will gradually become flat and infinitely close to 1 or -1. This situation often leads to the gradual disappearance of the gradient in the training process of LSTM network, which leads to the network parameters can’t be updated and the network can’t continue to be effectively trained.

In order to solve this problem, the ReLU (modified linear unit) function is selected as the activation function of LSTM network in the model construction. The functional sketches of ReLU and tanh function is shown in Figure 3. It is not difficult to see that when the independent variable is greater than 0, the ReLU can guarantee the stability of gradient and make the training of LSTM network more stable.

LSTM network structure design to avoid overfitting problem

Deep learning model can fit and describe the sequence well, but with the training of model and the increase of the scale of parameters, it often leads to over-fitting problems. In this paper, LSTM network is adjusted and optimized to avoid over-fitting problems.

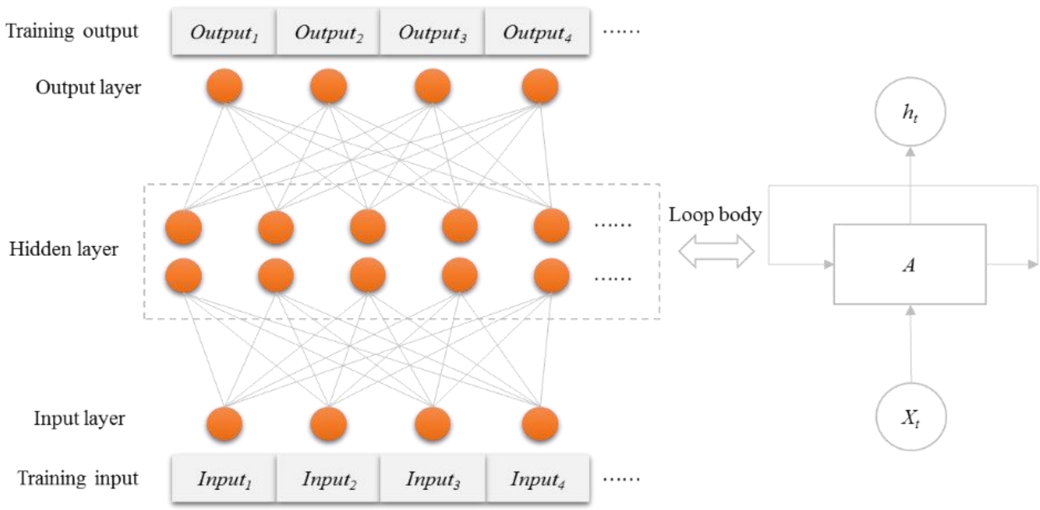

Fig. 2. A brief sketch of LSTM network structure

1) Design of appropriate loss function

The deep LSTM network model adopted in this paper needs to solve the regression problem. Therefore, MSE (Mean Squared Error) is chosen as the loss function. The definition of this function is as formula (1):

$$MSE(Y,Y') = \frac{\sum_{i=1}^{n}(y_i - y_i')^2}{n} \tag{1}$$

In the formula, y_i——The real output corresponding to the ith data;

y'_i——The output value of the predicted value of the neural network corresponding to the ith data.

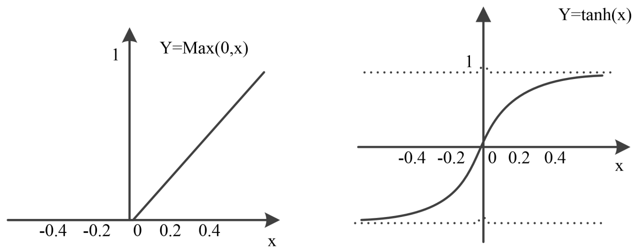

Fig. 3. Comparison diagram of ReLU (left) and tanh function (right)

2) $L2$ regularization for the loss function

Loss function can effectively describe the performance of the model on training data. $L2$ regularization can prevent the invalid random noise in the excessive simulation training data of the model by limiting the weight, which is defined as follows:

$$R(w) = \|w\|_2^2 = \sum_i |w_i|^2 \tag{2}$$

In the formula, w——The weight of the model, and it is the parameter that needs to be calculated regularization loss.
By choosing the appropriate loss function and Regularizing the loss function, the over-fitting of the model in the training process can be effectively avoided, so that the model can focus on the real fluctuation law of the stock closing price sequence rather than random noise.

Optimization of LSTM network model training method

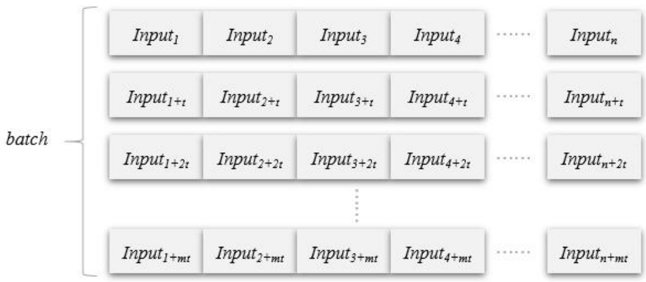

Fig. 4. Model training input after batch partitioning

Gradient descent is the most important way to adjust the parameters of the model. It ensures that the loss function of the neural network model on the training data can be as small as possible. The quality of gradient descent directly determines the quality of the model predictions. In this paper, we synthesize the advantages and disadvantages of gradient descent and random gradient descent, and adopt batch gradient descent algorithm. The algorithm divides the training data into batches, and only optimizes the loss function of the current batch at a time. The structure of batch is shown in Fig. 4. The design adopted in this article not only ensures parameter updates, but also reduces the number of iterations required for model convergence.

In the figure above, m represents the size of a batch. When the value of m is large, the complete sequential data can be covered by a small number of batches, but each batch contains a large amount of data, and the training time of the model on a single batch is longer. But we can adjust the network parameters from more data of the model, and considering the time-consuming and training effectiveness of the model training, therefore, it is necessary to reasonably select the value of m. Through continuous experimental testing and comparison, the m value is set to 25 in proposed network model.

In addition, learning rate needs to be further adjusted to optimize the updating speed of model parameters when training model. In this article, in order to optimize the learning rate in the construction of deep LSTM network models and ensure more stable parameter updates, we adopted the Adam (Adaptive Moment Estimation) algorithm[21]. The formulas of the Adam algorithm are as follows.

$$m_t = \mu * m_{t-1} + (1-\mu) * g_t \tag{3}$$

$$n_t = v * n_{t-1} + (1-v) * g_t^2 \tag{4}$$

$$\Delta\theta_t = -\frac{\hat{m}_t}{\sqrt{\hat{n}_t + \varepsilon}} * \eta \tag{5}$$

In the formula, m_t——The first moment estimation of gradient;

n_t——Estimation of the second moment of the gradient;

$\hat{m}_t$ ——The correction of the first moment estimation of gradient, and is approximate to the unbiased estimation of expectation;

$\hat{n}_t$ ——The correction of the second moment estimation of the gradient, and is approximate to the unbiased estimation of the expectation.

η——The learning rate of the model.

Adam algorithm forms a dynamic constraint on learning rate (as shown in formula (5)), and it has a clear scope. In this paper, the algorithm is used to realize the smooth iteration of learning rate and the effective updating of model parameters.

B. Stock Price Prediction Model Based on DB-ARIMA

The process of constructing DB-ARIMA model includes three steps.

Data stationary processing

For time series, stationarity is a very important attribute. It reflects whether there are a large number of random process components which can’t be quantified and predicted in the time series, and whether the statistical attributes of the time series will change with the passage of time. Before using ARIMA model, we need to check and smooth the financial time series.

In the experimental data set, financial time series is constructed with date as index, opening price, closing price and maximum price as attribute characteristics, and closing price as prediction target. Due to the fact that the ARIMA model is a single feature model, this article takes the closing price as a characteristic attribute of the time series, and the financial time series can be represented as follows:

$$TimeSeries = \{day_1, day_2, day_3, ..., day_n\}$$

$$day_n = \{index : t_n, value : close\}(t_n : time, eg : 2016.01.04) \quad (6)$$

Taking the stock 601318 as an example, some financial time series are constructed as shown in TABLE I.

The time series diagram of the stock 601318 is shown in Figure 5.

Through observation and ADF Test (Augmented Dcikey-Fuller Test), we can see that the time series data is non-stationary (ADF Test is not introduced here). We stabilize the time series by first-order differential processing. The sequence of the stock 601318 after differential stationary is shown in Figure 6.

After ADF test, it can be seen that the stock time series has good stationarity after first-order differential treatment, which can be used to construct ARIMA model.

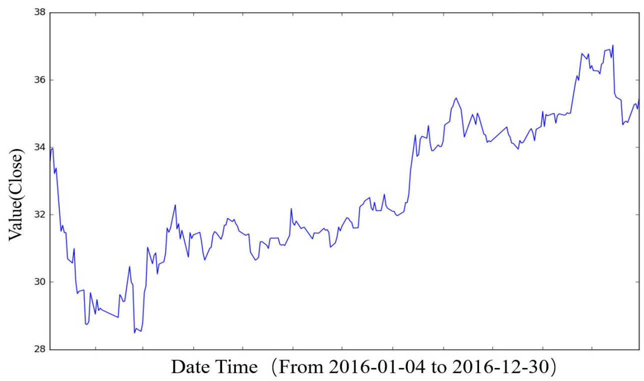

Fig. 5. Primitive time series diagram of 601318

TABLE I
PARTIAL FINANCIAL TIME SERIES OF 601318

Sequence Point（day）	Date	Close price [a]
day1	2016-01-04	33.461
day2	2016-01-05	33.933
day3	2016-01-06	33.962
day4	2016-01-07	33.215
day5	2016-01-08	33.382
……	……	……

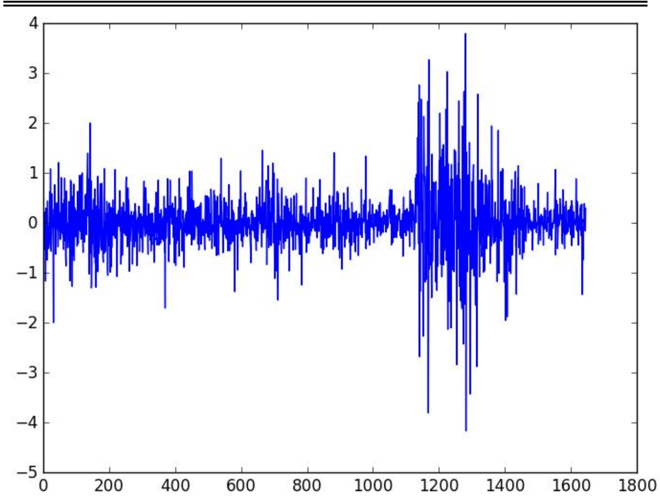

Fig. 6. Smoothed first-order difference sequence diagram of 601318

Design of ARIMA Model

ARIMA (Autoregressive Integrated Moving Average) model is a classical model for time series analysis and prediction. It is essentially a mixed model. It mainly consists of three parts: AR (Autoregressive)、MA (Moving Average) and Difference， so ARIMA model can be described as follows:

$$ARIMA(p, d, q) = AR(p) + difference(d) + MA(q) \quad (7)$$

Among them, p, d and q are the order of AR, Difference and MA models respectively. The construction of ARIMA model focuses on the order determination of three models. Since the first-order difference has been processed, the order of difference is 1. Then, the order of AR and MA components needs to be determined. This paper uses ACF (Autocorrelation Function) and PACF (The Partial Autocorrelation Function). ACF function describes the linear correlation between current and historical values of time series, that is, the correlation between time series and its own lag sequence. The PACF function describes the linear correlation of the time series before and after given intermediate observations. Taking the stock 601318 as an example, ACF and PACF are constructed for the close price time series after stabilization, and results are shown in figure 7 and 8.

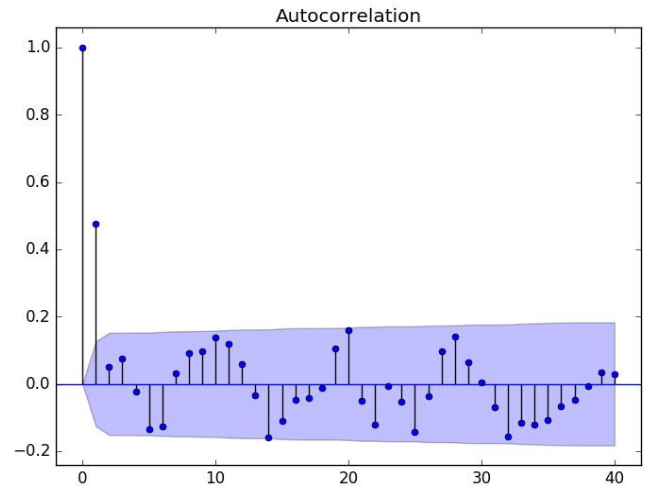

Fig. 7. ACF diagram of Smoothed first-order difference sequence of 601318

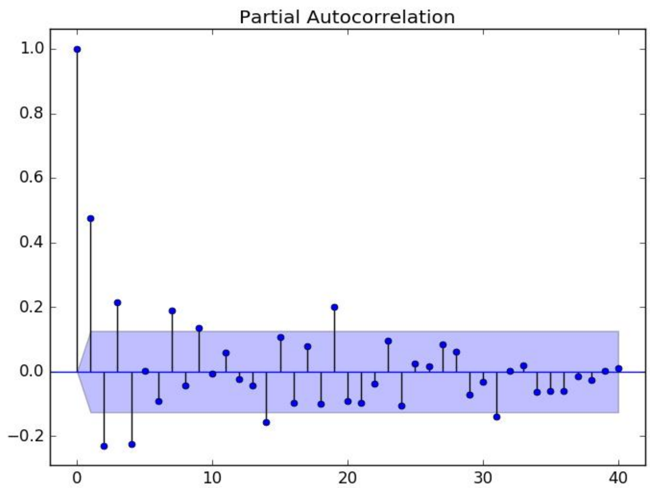

Fig. 8. PACF diagram of Smoothed first-order difference sequence of 601318

Through the observation of figure7 and 8, we can see that the autocorrelation coefficient of ACF graph decreases gradually to 0 after the lag of first order. The partial autocorrelation coefficient of ACF graph exceeds the confidence boundaries when the delay is 1, 2, 3, 4, 7 and 8 orders (19 orders can be regarded as errors), and it can be considered that the partial autocorrelation coefficient gradually reduces to 0 after 8 orders.

In conclusion, the order *p* of AR component is 8 and *q* of MA component is 1. Then the ARIMA model is constructed as shown in Formula 8.

$$ARIMA(8,1,1) = AR(8) + Difference(1) + MA(1) \quad (8)$$

DB-ARIMA model

In the experiment, we found that there is a certain amount of noise in the stock time series data, which has a certain impact on prediction. However, in traditional ARIMA models, noise data is not processed. Therefore, a DB-ARIMA model combined with wavelet transform is proposed in this paper. We regard stock time series data as a special signal, which conforms to the continuity of time domain and loads the corresponding information content such as stock closing price.

$$f(i) = s(i) + e(i) \quad i = 0,1,2,3,...,n-1 \quad (9)$$

In the formula, $f(i)$ denotes the stock signal with noise, $s(i)$ denotes the real signal of the stock, and $e(i)$ denotes the noise signal of the stock. We clean the stock time series data by referring to the way of signal de-noising, which makes it easier for the model to distinguish the changing rules of stock prices. In fact, signal denoising also enhances the signal, making the sequence law more prominent, which is helpful to the prediction of stock time series.

Considering the stock market, the real and effective signals of the stock are usually low-frequency and stable signals, while the noise is high-frequency signals. Therefore, the denoising of one-dimensional stock signal based on wavelet transform can be divided into three steps:

1)Wavelet decomposition signal and determine the wavelet basis and decomposition layer number *N*;

2)Threshold quantization of wavelet decomposition;

3)Wavelet reconstruction signal.

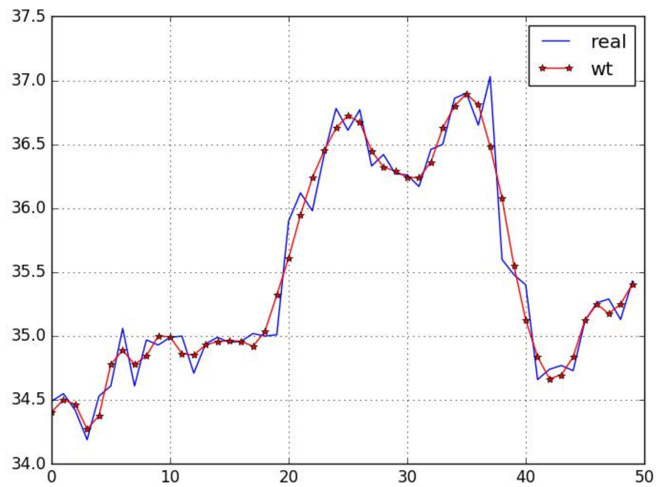

Fig. 9. Contrast diagram of stock time series signals before and after wavelet denoising processing

In this paper, the most common wavelet-based Daubechies (DB) wavelet is used to denoise the corresponding wavelet transform, which can provide more effective signal analysis and synthesis capabilities. DB4 is used as the basis of wavelet transform to denoise the stock time series signal. The contrast diagram of stock time series signals before and after wavelet denoising processing is shown in Figure 9.

As can be seen from the above figure, the original stock timing becomes smoother and the overall change rule becomes more prominent after the wavelet denoising treatment. That can help improve the accuracy of predictions.

C. The hybrid LSTM and DB-ARIMA model

Deep learning model tends to be conservative, and the prediction of the value of each time node tends to predict a lower fluctuation range. The ARIMA model, on the contrary, is more prone to predict the results of a larger fluctuation range in each time node, which seems to be relatively "radical".

Due to the complexity of stock data samples, both conservative and aggressive predictions cannot fully cover the situation of stock samples. Therefore, this paper proposes a hybrid model, which integrates the deep learning model and ARIMA model, and fuses the prediction results of the two models in order to achieve more accurate prediction results and avoid the limitations of the single model. The overall architecture of the hybrid model is shown in Figure 10.

In this paper, the fusion strategy of prediction results in the hybrid model is the weighted average method, which is defined by formula 10.

$$Value_{hybrid} = \frac{Weight_{LSTM} * Value_{LSTM} + Weight_{ARIMA} * Value_{ARIMA}}{2} \quad (10)$$

In the formula, the hybrid model can accumulate the predicted results from the deep learning model and ARIMA model by weighting, so that the predicted results are closer to the real values. Because of the different situation of each sample, the weights of the depth learning model and ARIMA model are also different in the prediction tasks of different samples.

In this paper, standard deviation measurement method is used to determine the weight of the model, and the calculation formula is as follows.

$$Weight_{LSTM} = (1 - \frac{LSTM_{diff}}{LSTM_{diff} + ARIMA_{diff}}) \times 2 \quad (11)$$

$$Weight_{ARIMA} = 2 - Weight_{LSTM} \quad (12)$$

$$LSTM_{diff} = Sample_{std} - LSTM_{std} \quad (13)$$

$$ARIMA_{diff} = Sample_{std} - ARIMA_{std} \quad (14)$$

In the formula, $Sample_{std}$——Standard deviation of samples;

$LSTM_{std}$——Standard deviation of predicted results of deep learning model;

$ARIMA_{std}$——Standard deviation of predicted results of DB-ARIMA.

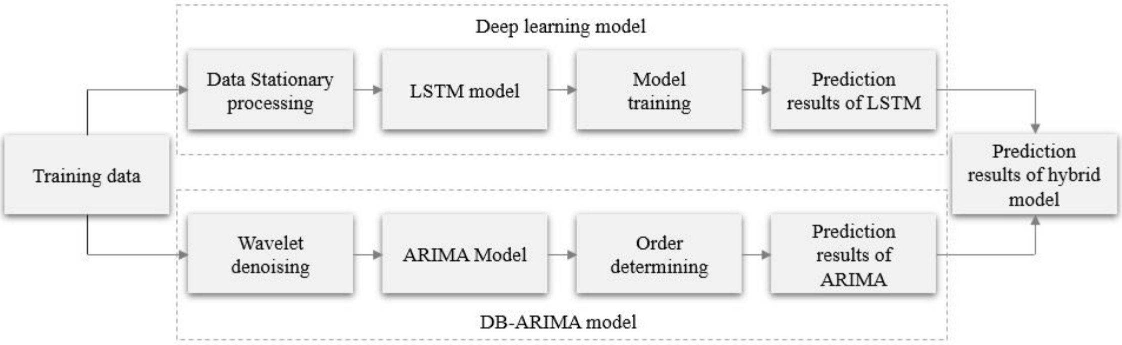

Fig. 10. Overall schematic diagram of hybrid model

The idea of this weighting method is to introduce the standard deviation as a measure of the fluctuation of the series. In econometrics, standard deviation can also well characterize the sharp rise and fall of financial data and the size of risk index.

Through the corresponding statistical analysis, the weight settings of 12 stock samples are shown in TABLE II.

TABLE II
WEIGHT SETTINGS FOR HYBRID MODEL

Stock code	SAMPLE TYPE	WEIGHT OF DEEP LEARNING	Weight of DB-ARIMA
000001	Index	1.48	0.52
399001	Index	1.80	0.20
399300	Index	1.54	0.46
600000	Stock	1.84	0.16
600016	Stock	1.58	0.42
600030	Stock	0.69	1.31
600036	Stock	1.90	0.10
600519	Stock	1.94	0.06
600837	Stock	1.69	0.31
601166	Stock	1.83	0.17
601318	Stock	1.23	0.77
601328	Stock	1.51	0.49

III. EXPERIMENTAL WORK

A. Experimental environment

The experimental environment used in this paper includes Windows 10 and Linux，3 PC and 1 Sever, the programming language is Python, and the main toolkits include Tensorflow, Statsmodels and fbprophet. Data is stored in the form of CSV and MySQL.

B. Experimental date

Because of the particularity of stock timing prediction problem, it is necessary to establish corresponding models for each sample. In order to verify the stability of the model and its adaptability to the market, we selected two sets of datasets for experiments. We divide the historical data of 12 stocks and index samples in the stock time series data set. In the first dataset, the historical data of each sample from 4 January 2016 to 23 December 2016 were used as model training data, and the historical data from December 26 to December 30, 2016 (a total of five trading days) are used as the test data. In the second dataset,the historical data of each sample from 3 January 2022 to 23 December 2022 were used as model training data, and the historical data from December 26 to December 30, 2022 (a total of five trading days) are used as the test data to evaluate the effectiveness of the model. After obtaining the training data, the deep learning model and DB-ARIMA model are trained and constructed respectively.

C. Experimental evaluation

For the experimental evaluation of the hybrid model, RMSE, MAPE and MAD are used as the criteria to measure the prediction effect.

IV. EXPERIMENTAL RESULTS

A. Comparison experiment of the hybrid model and single model

After training samples in two datasets to obtain corresponding models, prediction experiments were conducted on the data of each sample. Taking the prediction experiment of 00000 1 as an example, the specific prediction results of each model in different datasets are shown in Table III.

TABLE III
000001 PREDICTION RESULTS OF MODELS

Data	TRUE VALUE	LSTM	DB-ARIMA	Hybrid model
2016-12-26	3120.58	3115.16	3088.78	3122.61
2016-12-27	3117.63	3121.58	3127.46	3119.92
2016-12-28	3109.43	3105.06	3087.41	3110.04
2016-12-29	3102.68	3097.83	3080.91	3102.61
2016-12-30	3102.36	3105.64	3114.73	3103.07
2022-12-26	3065.56	3060.23	3017.54	3067.62
2022-12-27	3095.57	3081.02	3060.17	3097.69
2022-12-28	3087.4	3062.53	3047.55	3089.92
2022-12-29	3073.7	3065.48	3041.83	3075.86
2022-12-30	3089.26	3071.52	3064.74	3091.57

The comparison results of index 00000 1 predicted in the first dataset are shown in Figures 11. The comparison results of index 00000 1 predicted in the second dataset are shown in Figures 12. Seen from the figures, for the prediction task of 000001, the predicted value of the proposed hybrid model is closer to the real value than that of the deep learning model and the DB-ARIMA model. However, the performance of a single sample does not represent the overall performance of each model on stock timing prediction. Therefore, we conducted comparative experiments using multiple models and multiple sets of samples. TABLE IV shows the predictions of mixed model, deep learning model and ARIMA model on 12 indexes and stock samples.

As shown in the TABLE IV, the RMSE, MAPE and MAD of the proposed hybrid model are the smallest. It is proved that the hybrid model has the best performance in the prediction task of 12 samples. At the same time, it is found that the deep learning model performs better than the DB-ARIMA model in the prediction task, especially in the prediction of exponential samples. On the other hand, it is found that the predicted values of DB-ARIMA model at some nodes are closer to the real values than those of deep learning model. After weighted averaging, the predictive advantages of these nodes can be transferred to the hybrid model, further improving the overall prediction accuracy of the model.

B. Comparison experiment of the hybrid model and Prophet model

The Prophet model proposed by Facebook is an additive regression model. Prophet model is often used in time series prediction. It is robust in preventing numerical missing, steep trend change and maximum outlier. This article uses the Prophet model to predict the same 12 index and stock samples, and compares the predicted results with the hybrid model proposed in this article. The RMSE results of the predicted results and the RMSE results of the hybrid model are shown in TABLE V.

Prophet model is more inclined to rely on a large number of manual adjustment parameters. In the absence of human factors, we can see that the Prophet model has poor prediction effect on samples, especially in the prediction of exponential samples. By comparing with Prophet model, we can see that the hybrid model proposed in this paper has better performance in stock time series prediction.

C. Comparison experiment of the hybrid model and Multi feature ARIMA

Because the sample size and data size of financial products studied in many literatures are different on the issue of stock time series prediction, and the sample data involved in some literatures are difficult to obtain, two representative literatures [22], [23] are selected through the selection of literatures.

On the sample and data studied in the literature, the hybrid model is used to perform the same financial time series forecasting task, and the results are compared.

Reference [22] proposes a multi-feature ARIMA model. The research sample is 399300 (Shanghai-Shenzhen 300 Index). The training data of the ARIMA model is the closing price sequence of 243 trading days before 2011, and the test data is the closing price sequence of the last 8 trading days in 2011.The predicted results of the reference and the results of the hybrid model presented in this paper are shown in TABLE VI. Comparison diagram as shown in Figure 13.

TABLE IV
PREDICTION RESULTS OF THE MODELS UNDER DIFFERENT SAMPLES

Stock code	DEEP LEARNING MODEL			DB-ARIMA MODEL			Hybrid model		
	RMSE	MAPE	MAD	RMSE	MAPE	MAD	RMSE	MAPE	MAD
000001	4.707	0.128	3.916	8.869	0.221	6.825	4.004	0.110	3.432
399001	13.986	0.106	10.754	77.527	0.567	57.816	12.771	0.093	9.941
399300	5.501	0.147	4.812	10.482	0.267	8.728	4.841	0.138	4.530
600000	0.037	0.028	0.038	0.119	0.667	0.107	0.034	0.203	0.032
600016	0.019	0.159	0.014	0.035	0.337	0.031	0.018	0.145	0.013
600030	0.035	0.179	0.029	0.093	0.488	0.078	0.030	0.167	0.023
600036	0.072	0.286	0.050	0.103	0.472	0.083	0.056	0.252	0.044
600519	3.441	0.781	2.574	2.519	0.682	2.223	3.182	0.722	2.377
600837	0.153	0.553	0.087	0.103	0.569	0.088	0.135	0.495	0.078
601166	0.064	0.199	0.032	0.145	0.651	0.085	0.047	0.146	0.031
601318	0.118	0.300	0.106	0.086	0.233	0.082	0.107	0.273	0.096
601328	0.009	0.111	0.006	0.033	0.402	0.023	0.006	0.098	0.006

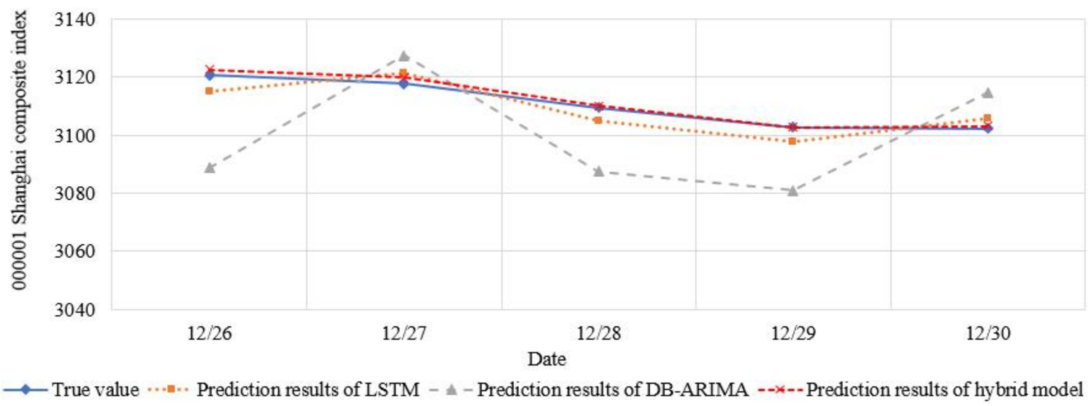

Fig. 11. Comparison of the predicted results in the first dataset

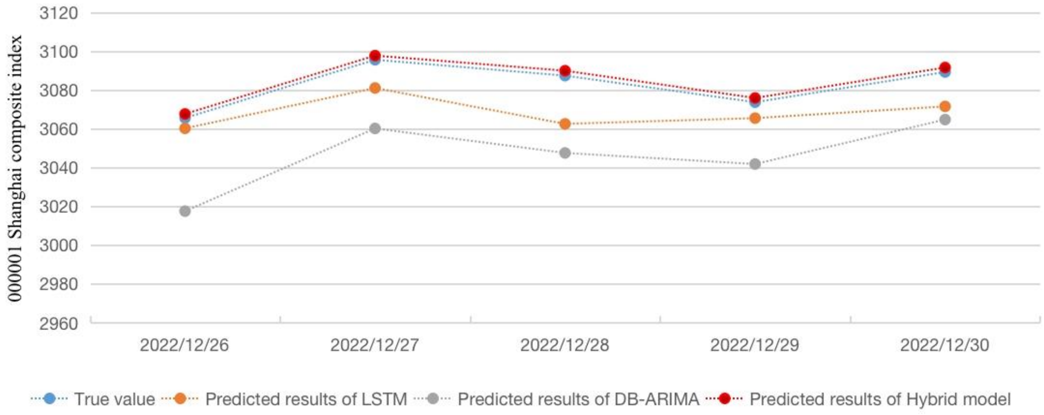

Fig. 12. Comparison of the predicted results in the second dataset

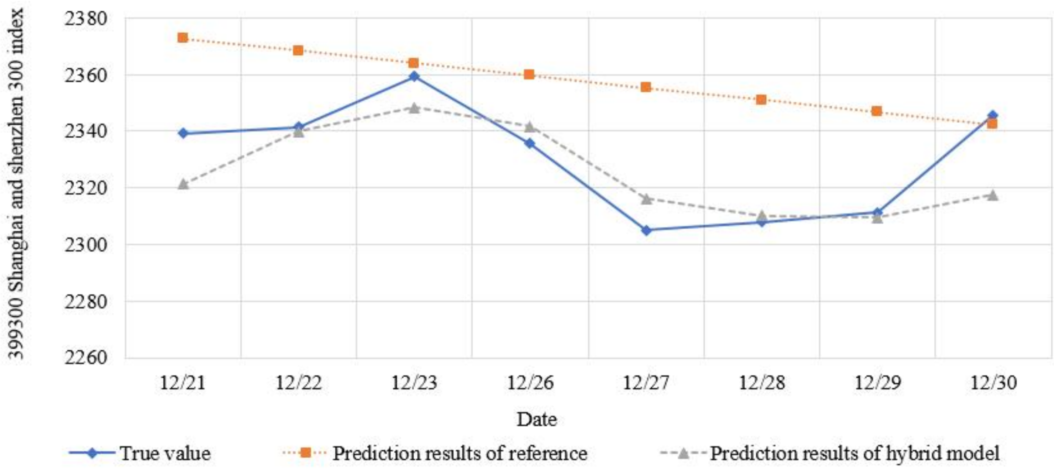

Fig. 13. Comparison of predicted results

TABLE V
PREDICTION RESULTS OF PROPHET MODEL AND HYBRID MODEL

Stock code	SAMPLE TYPE	RMSE OF PROPHET	RMSE OF HYBRID MODEL
000001	Index	330.186	4.005
399001	Index	417.426	12.773
399300	Index	244.255	4.845
600000	Stock	1.041	0.035
600016	Stock	0.292	0.018
600030	Stock	0.314	0.030
600036	Stock	0.612	0.056
600519	Stock	15.809	3.182
600837	Stock	0.585	0.135
601166	Stock	0.384	0.047
601318	Stock	0.880	0.107
601328	Stock	0.457	0.006

TABLE VI
COMPARISON OF MULTI-FEATURE ARIMA AND HYBRID MODEL

Date	TRUE VALUE	MULTI-FEATURE ARIMA	HYBRID MODEL
2011.12.21	2339.11	2372.73	2321.32
2011.12.22	2341.33	2368.37	2339.89
2011.12.23	2359.16	2364.02	2348.39
2011.12.26	2335.70	2359.68	2341.54
2011.12.27	2305.03	2355.34	2316.17
2011.12.28	2307.93	2351.02	2310.05
2011.12.29	2311.36	2346.70	2309.44
2011.12.30	2345.74	2342.39	2317.53

From above is obvious, the prediction results of the hybrid model are closer to the real values than that of the ARIMA model combined with multiple features proposed in [21]. Through further calculation, the RMSE of the model proposed in the references is 31.874, while the RMSE of the hybrid model is only 13.211. It can be seen that the hybrid model proposed in this article has superior performance and higher accuracy in predicting stock time series.

Reference [23] proposes a hybrid ARIMA model combining the characteristics of index data. The sample is 000001 (Shanghai Composite Index). The training data are monthly data from January 2005 to October 2016, and the test data are monthly data for the last three months. The predicted results of the reference and the results of the hybrid model presented in this paper are shown in TABLE VII.

Through further calculation, the RMSE of the model proposed in [23] is 77.143, while the RMSE of the hybrid model is only 30.702, so the hybrid model proposed in this paper is more accurate.

TABLE VII
COMPARISON OF HYBRID ARIMA AND HYBRID MODEL

Date	TRUE VALUE	HYBRID ARIMA	HYBRID MODEL
2016-08	3085.49	3010.56	3049.88
2016-09	3004.70	3060.73	3044.01
2016-10	3100.49	3005.10	3104.36

V. CONCLUSION AND FUTURE STUDIES

Stock time series prediction is a highly challenging scientific problem, which is of great significance for both the academic field of financial prediction and the application field of market investment decision-making. In this work, a hybrid model combining deep learning and DB-ARIMA is proposed, and the prediction accuracy is further improved by taking advantage of the complementary advantages of the two single models. After training and testing on multiple datasets and different market conditions, it shows that the Mixture model method proposed in this paper has good prediction performance, and to some extent, it proves the feasibility of the Mixture model method. At the same time, the proposed model has better market adaptability.

In future work, we will study the deep fusion of hybrid models, not just weighted average. In addition, the generalization ability of the model is further improved.

REFERENCES

[1] R. Mahanta, T. N. Pandey, A. K. Jagadev, and S. Dehuri, "Optimized Radial Basis Functional neural network for stock index prediction," in *International Conference on Electrical, Electronics, and Optimization Techniques, ICEEOT 2016*, pp1252–1257.

[2] Y. Zhong, L. Luo, X. Wang, and J. Yang, "Multi-factor stock selection model based on machine learning," *Engineering Letters*, vol. 29, no. 1, pp177–182, 2021.

[3] Melina, Sukono, H. Napitupulu, A. Sambas, A. Murniati, and V. A. Kusumaningtyas, "Artificial Neural Network-Based Machine Learning Approach to Stock Market Prediction Model on the Indonesia Stock Exchange During the COVID-19," *Engineering Letters*, vol. 30, no. 3, pp988–1000, 2022.

[4] Dr. K.K.Sureshkumar and Dr.N.M.Elango, "Performance Analysis of Stock Price Prediction using Artificial Neural Network," *GJCST*, vol. 12, no. 1, pp19–25, 2012.

[5] L. Cao and F. E. H. Tay, "Financial Forecasting Using Support Vector Machines," *Neural Computing & Applications*, vol. 10, no. 2, pp184-192, 2001.

[6] J. L. Ticknor, "A Bayesian regularized artificial neural network for stock market forecasting," *Expert Systems with Applications*, vol. 40, no. 14, pp5501–5506, 2013.

[7] W. Dai, J.-Y. Wu, and C.-J. Lu, "Combining nonlinear independent component analysis and neural network for the prediction of Asian stock market indexes," *Expert Systems with Applications*, vol. 39, no. 4, pp4444–4452, 2012.

[8] M. Qiu, Y. Song, and F. Akagi, "Application of artificial neural network for the prediction of stock market returns: The case of the Japanese stock market," *Chaos, Solitons and Fractals*, vol. 85, pp1–7, 2016.

[9] H. Yu and H. Liu, "Improved stock market prediction by combining Support Vector Machine and Empirical Mode Decomposition," in *Proceedings - 2012 5th International Symposium on Computational Intelligence and Design*, *ISCID 2012*, in vol. 1. pp531–534.

[10] R. P. Schumaker and H. Chen, "Textual analysis of stock market prediction using breaking financial news: The AZFin text system," *ACM Transactions on Information Systems*, vol. 27, no. 2, pp1-19, 2009.

[11] R. Dash and P. Dash, "Efficient stock price prediction using a Self Evolving Recurrent Neuro-Fuzzy Inference System optimized through a Modified technique," *Expert Systems with Applications*, vol. 52, pp75–90, 2016.

[12] M. Roondiwala, H. Patel, and S. Varma, "Predicting Stock Prices Using LSTM," *International Journal of Science and Research (IJSR)*, vol. 6, 2017.

[13] Q. Zhuge, L. Xu, and G. Zhang, "LSTM neural network with emotional analysis for prediction of stock price," *Engineering Letters*, vol. 25, no. 2, pp167–175, 2017.

[14] Z. Ge, W. Wang, and D. Chen, "Predicting index returns from the market structure disagreement: Evidence from china," *Engineering Letters*, vol. 28, no. 4, pp1063–1074, 2020.

[15] E. Alshabeeb et al., "Intelligent Techniques for Predicting Stock Market Prices: A Critical Survey," *Journal of Information and Knowledge Management*, vol. 22, no. 3, 2023.

[16] J. Wang, Q. Cui, X. Sun, and M. He, "Asian stock markets closing index forecast based on secondary decomposition, multi-factor analysis and attention-based LSTM model," *Engineering Applications of Artificial Intelligence*, vol. 113, 2022.

[17] Y. Chen, J. Wu, and Z. Wu, "China's commercial bank stock price prediction using a novel K-means-LSTM hybrid approach," *Expert Systems with Applications*, vol. 202, 2022.

[18] S. Banik, N. Sharma, M. Mangla, S. N. Mohanty, and S. Shitharth, "LSTM based decision support system for swing trading in stock market," *Knowledge-Based Systems*, vol. 239, 2022.

[19] N. Pahuja, A. Oturkar, K. Sharma, D. Bohra, and J. Shrivastava, "Stock Market Prediction Using the ARIMA Model," *International Journal for Scientific Research and Development*, vol. 3, no. 11, pp159–160, 2016.

[20] X. Yang, T. Yu, Y. Lu, and Z. Chu, "An empirical study on stock price based on ARIMA mode," in *International Conference on Logistics, Engineering, Management and Computer Science*, *LEMCS 2014*, pp273–276.

[21] K. Chen, Y. Zhou, and F. Dai, "A LSTM-based method for stock returns prediction: A case study of China stock market," in *Proceedings - 2015 IEEE International Conference on Big Data*, *IEEE Big Data 2015*, pp2823–2824.

[22] S. Shen and Y. Shen, "ARIMA Model in the Application of Shanghai and Shenzhen Stock Index," *Applied Mathematics*, vol. 7, no. 3, Art. no. 3, 2016.

[23] C. Li, B. Yang, and M. Li, "Forecasting analysis of Shanghai stock Index based on ARIMA model," in *MATEC Web of Conferences*, vol. 100, 2017.