Prediction of Shanghai Stock Market Based on CNN-LSTM Model with GA Optimized Attention

Hanyu Hu*

* Corresponding author: 312706@whut.edu.cn

College of Science, Wuhan University of Technology, Wuhan 430063, China.

Abstract: Investment in stocks is a means of maximizing the benefits of assets through the purchase and sale of stocks. For equity investment, accurate forecasting of stock prices, and ups and downs are essential for asset return management. Currently, there are not only classical Markov chain models, but also many models with deep learning for predicting time-series outcomes. By studying the current relevant work on stock forecasting, it is recognized that the existing forecasting models have a low number of feature extractions from stock data and insufficient training data, resulting in a lack of forecasting accuracy of the models. For improvement of the prediction accuracy, this paper proposes a CNN-LSTM model with GA-optimized Attention, which is a combined prediction model based on the traditional LSTM model with a convolutional module to extract stock features, an Attention mechanism to improve accuracy, and a GA algorithm to find the optimal fully connected layer weights. In addition, this paper uses the Qlib[1] platform to calculate the stock Alpha158 factor in data processing, and the RFE and PCA methods for feature reduction and extraction to obtain the processed variables. Finally, 500 SSE stocks are used to conduct comparison experiments and ablation experiments between the model proposed in this paper and the four classical models. The experiments can prove that the model proposed in this paper outperforms the other four models in terms of prediction accuracy, i.e.the model has the smallest MAE(0.00612) and MAPE(0.04391), and the largest R^2(0.98714).

Keywords: Stock predictions, CNN-LSTM, Attention mechanism, Genetic algorithm

1. INTRODUCTION

Stocks are securities issued by a joint-stock company to raise funds and are used by individual shareholders as evidence of their shareholding and to receive dividends and bonuses. Shareholders buy and sell stocks to maximize the benefits of their assets. As the Efficient Markets Hypothesis[1], accurately predicting stock prices and ups and downs, and making portfolio decisions that maximize benefits and minimize risks, becoming an important topic in the current investment field. For the problem of stock prediction, we not only predict the price of a stock (regression problem), but also the rise or fall of a stock in the following days (classification problem). By means of deep learning, stock prediction is designated as a supervised learning task of regression and classification. The models are trained and tested using

[1]https://qlib.readthedocs.io/

ICBBEM 2023, May 19-21, Hangzhou, People's Republic of China

DOI 10.4108/eai.19-5-2023.2334242

stock characteristic factors to predict stock prices and ups and downs over the next 3 trading days.

In order to forecast the trend of stock prices and ups and downs more accurately, scholars first consider the application of mathematical statistics and econometric methods to the study of stock markets, such as ARIMA models[2] and GARCH models[3], etc., while these models are based on the premise of linear or stable series, requiring a high degree of stability or linearity in the stock data. However, the large number of stock characteristics make it difficult to meet the requirements of traditional mathematical models for stability or linearity of data. Subsequently, some scholars proposed methods such as using SVM to forecast stock price movements[4]. Although these methods address the impact of the complexity of stock data on forecasting, the characteristic that stocks are time-series in nature is ignored, and temporal forecasting cannot be realized. Further then the LSTM Model[5] was used for stock price prediction, which successfully achieved the prediction of time-series data, but the LSTM model is poor in its parallelism capability when dealing with more data. It also has inaccurate extraction of data features and high data loss, which reduces the accuracy of the model predictions. The Transformer model was later developed to predict stocks, where the Transformer model incorporated the Self-Attention mechanism, which could greatly improve the parallelism of model training. However, it was found through experiments that when this model handled a large amount of time series data, the model gradient flow would be blocked by the layer normalization module, resulting in a decrease in prediction accuracy.

Therefore, this paper proposes to introduce the Self-Attention mechanism into the CNN-LSTM model, while using the Genetic Algorithm (GA) to perform global parameter optimization, and finally obtain the final stock prediction results. This model is referred to as the CNN-LSTM model with GA algorithm added to optimize Attention. The model not only uses Self-Attention mechanism to improve the parallelism of the model, but also takes into account the complexity and diversity of the stock factor data, which involves multi-dimensional time series, by adding the Convolutional Neural Network (CNN) to extract and process the temporal eigenvalues of the input layer data, and then input to the LSTM module, for reducing the confusion of the multi-dimensional time series and improving the stability of the model. At the same time, the GA algorithm is used to optimize the parameters of each Attention mechanism and LSTM, and to adjust the relevant parameters of the fully connected layer of the model, so as to ensure the accuracy of the model prediction.

Main contributions to this article

- Used data on the top 500 SSE stocks (.SH) obtained from the Tushare[2] platform, calculating the Alpha158 factor for stocks using the Qlib database to extract more comprehensive feature values as data of stocks.

- The RFE algorithm and PCA algorithm were used to do data dimensional reduction to obtain the final feature set and improve the model's ability to calculate feature values.

- The Self-Attention mechanism is introduced into the CNN-LSTM model, and the GA is used to find and discriminate the model network parameters to obtain the model in this paper, so that it not only focus on extracting more valuable stock characteristics, but also to eliminate

[2] https://tushare.pro/

as much as possible the influence of differences between stock factors in order to obtain more accurate prediction results

- Through comparison tests with 3 other models, and ablation tests with no GA optimized model, it is verified that the model proposed in this paper has good applicability and higher accuracy in stock prediction situations.

2. RELATED WORK

According to the amount of data and the requirements for prediction accuracy, the main models studied by related academics in stock prediction can be divided into three categories. The first category uses classical mathematical statistical models for prediction, the second category uses deep learning models for prediction, and the third category uses intelligent algorithms to optimized models for prediction.

2.1 Work of Using Classical Mathematical Probability Statistical Model

Traditional methods of mathematical statistics (e.g. Markov chain, ARIMA, etc.) are extremely widely used in stock forecasting. Deneshkumar Venugopal etc. used Hidden Markov chain to develop a forecasting model for stock movements[6], predicting stock movements based on probability values obtained from a steady-state probability distribution. Adebiyi A. Ariyo etc. used the ARIMA model to construct a stock price forecasting model[7],which shows that the model has strong short-term forecasting potential. However, these models are computationally intensive and not very feasible with large and complex data.

2.2 Work of Using Classical Deep Learning Model

The RNN based models are popular in stock forecasting as they are specifically designed to capture temporal patterns in sequential data. David M. Q. Nelson etc. proposed the use of LSTM networks in stock forecasting scenarios[8] to improve the accuracy of stocks prediction. In 2019, Japanese scholars Yasuhiko Morimoto etc. proposed using a hybrid neural network incorporating an attention mechanism for stock price prediction[9], i.e., they proposed using a CNN-LSTM hybrid neural network with multiple factors to predict stock prices. But these models do not optimize the prediction results according to the importance of the stock factors, resulting in still less than ideal results in accuracy of prediction.

2.3 Work of Use Intelligent Algorithm to Optimize The Model

Because a large number of parameters in neural network related models need to be optimized, it has become common to use intelligent algorithms (e.g. genetic algorithms, annealing algorithms, etc.) to perform parameter search for models. Kim and Han proposed a model with a combination of ANN and GA[9], which was used to predict stock price indices with discrete features to improve prediction accuracy. Chinese scholars SONG G. etc. proposed a long LSTM stock price forecasting model based on adaptive particle swarm optimization[11]. The key parameters of the LSTM model are optimized by the PSO algorithm with adaptive learning strategy to match the stock data features and improve the accuracy of stock price prediction. These intelligent algorithms must be combined with existing models and are selected according to the purpose of the optimization of each model.

Since all these related models do not use enough stock factors and do not incorporate intelligent algorithms into a more complex model to further improve prediction accuracy, this paper proposes a CNN-LSTM Model with GA optimized Attention that uses the stock Alpha158 factor as the data-set, which has higher prediction accuracy compared to previous related works.

3. METHODOLOGY

The framework has a data processing module on the left and a model calculation module on the right. Then the main steps of the data processing module are: (1) inputting the acquired stock data into the Qlib platform to calculate the Alpha158 factor of the stock (2) dimensionality reduction using the RFE algorithm (3). Then use PCA to do initial feature extraction of the data; the key technical aspects of the model calculation module are: (1) use CNN module for further feature extraction (2) then use LSTM model to train each pooling layer (3) use GA optimised Attention to find the optimal weights in the fully connected layer for calculation and finally get the prediction results.

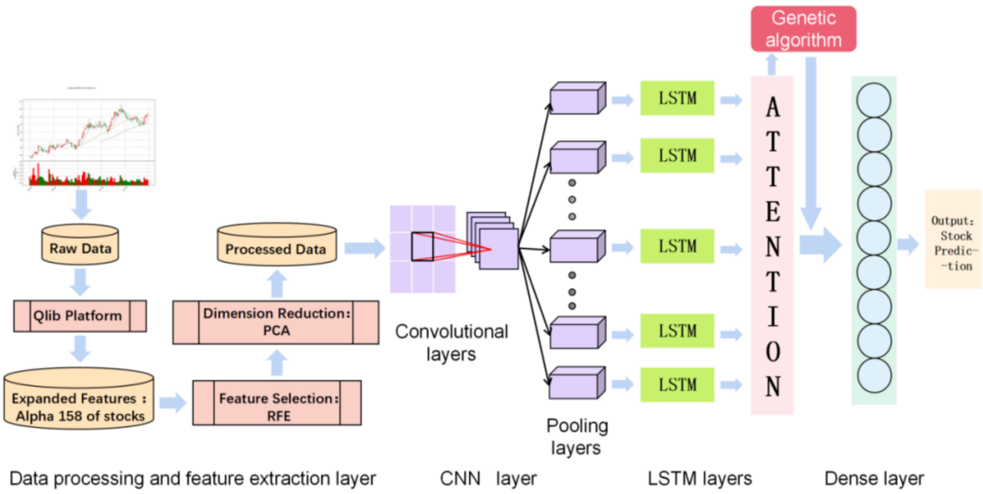

Figure 1 The basic framework of CNN-LSTM model with the Attention of GA optimization.

3.1 Stock data processing and feature extraction

Using the collected stock data $\mathrm{S} = \{\mathrm{s}_1, \mathrm{s}_2, \ldots, \mathrm{s}_\mathrm{n}\}$, each s_n contains the opening price o_{tk}, closing price c_{tk}, highest price h_{tk}, daily volume t_{tk}, etc. for t trading days.158 individual stock volume (technical) indicators ALPHA_{158} are obtained with the Alpha158 factor system of the Qlib platform.

As the 158 technical indicators were too redundant and strongly correlated, the RFE was used to extract the main features in order to select the top 20 technical indicators that could effectively reflect the stock prediction results.

For the RFE operation, the c_{tk} of the indicator is chosen as the dependent variable of the RFE, and the rest of the indicator is entered into the model as a feature variable for training, where we estimate and extract features. The limit of features removed from the pool is also set to 1, which means that this method will remove one feature at each step and retain all relevant features. The final model outputs the 20 technical indicators that have the most influence on c_{tk} price changes, noting these 20 main technical indicators as X=$\{x_1, x_2, \ldots, x_{20}\}$.

Then, a PCA model was used to further obtain a set of linearly uncorrelated stock impact factors ST=$\{st_1, st_2, \ldots, st_k\}(k \leq 20)$ in order to further reduce the dimension, to serve as the final data set for data processing.

For the PCA operation, X=$\{x_1, x_2, \ldots, x_{20}\}$ was entered into the PCA as a variable and the parameter n_components=0.98 was set to represent the output of the principal component with a cumulative contribution of 98% as the stock impact factor in the final datasets, denoted as ST in the final datasets.

3.2 CNN module to extract stock time series features

The processed ST=$\{st_1, st_2, \ldots, st_k\}(k \leq 20)$ was first processed by normalizing layers, then collated into a 3-dimensional tensor input format ([[sample],[time],[features]]) and input to a 1-dimensional CNN model for training, setting k convolutional neural network channels, setting the time window of 90, i.e. a window is added once for every 90 data, and the ST is input to a 2-layer 1D convolutional layer for training, with each layer containing 20 convolutional kernels, and a nonlinear transformation of the feature variables using the relu activation function. That is, each layer is computed as $a^j = relu(a^{j-1} \otimes W^j + b^j)$, where $relu = \max(0, w^T x + b)$,□ denotes convolution and W is the convolution kernel.Max pooling is then performed to obtain a pooling layer, which contains k channels, i.e. representing k pooled stock features. Each pooling layer is denoted as Pooling_ST$_i$=$\{pooling\_st^i_{t_1}, pooling\_st^i_{t_2}, \cdots\cdots, pooling\_st^i_{t_n}\}$ (i=1,2,...,k).

3.3 Attention mechanism optimized LSTM model trained for each channel

In this paper, the k pooling layers obtained after the processing of 3.2 are each used as the input layer of the LSTM , and $pooling\_st^i_{t_j}$ at the j^th moment inPooling_ST$_i$ is used as the j^th LSTM neuron input element, which is activated using the function (sigmoid) and then entered into the hidden layer for computation. Meanwhile a rolling prediction is applied to each pooling layer, continuously updating the training set of the LSTM to obtain a set of output sets Output$_i$=$\{output^i_{t_1}, output^i_{t_2}, \cdots\cdots, output^i_{t_n}\}$, each $output^i_{t_j}$ of which is output by each LSTM neuron. LSTM layers, each hidden layer neuron contains input gates, forgetting gates, cell states, and output gates.

The input doors, the status of input doors consisting of the output values $output^i_{t_{j-1}}$ at moment j-1 and the input $pooling\_st^i_{t_j}$ at moment t, are obtained by means of an activation function σ (sigmoid) with a given weight vector W and a bias b, the status values of input doors at the t moment ($q^i_{t_j}$).

$$q_{t_j}^i = \sigma(W_q \cdot [\text{ output } t_{t_{j-1}}^i, \text{pooling\_st}_{it_j}^i] + b_q) \tag{1}$$

The forgetting gate, where the value of part $q_t(f_{t_j}^i)$ is forgotten, i.e. rounded off and not computed into the network, is a different structure from the RNN neural network, avoiding "vanishing gradient" and "exploding gradient".

$$f_{t_j}^i = \sigma(W_f \cdot [\text{ output}t_{t_{j-1}}^i, \text{ pooling\_}st_{t_j}^i] + b_f) \tag{2}$$

Cell state, the inner product of $q_{t_j}^i$, cell update values $\tilde{c}_{t_j}^i$ and $f_{t_j}^i$ with vector summation to obtain Values of update cell status at the t moment($c_{t_j}^i$).

$$\tilde{c}_{t_j}^i = \tanh(W_C \cdot [\text{ output}_{t_{j-1}}^i, \text{ pooling } \_st_{t_j}^i] + b_C) \tag{3}$$

$$c_{t_j}^i = f_{t_j}^i \otimes c_{t_{j-1}}^i \oplus q_{t_j}^i \otimes \tilde{c}_{t_j}^i \tag{4}$$

The output doors, the $c_{t_j}^i$ and status values of output doors at the t moment ($o_{t_j}^i$) are calculated by vector multiplication to obtain $output_{t_j}^i$.

$$o_{t_j}^i = \sigma(W_o \cdot [\text{ output}_{t_{j-1}}^i, \text{ pooling\_}st_{t_j}^i] + b_o) \tag{5}$$

$$\text{output}_{t_j}^i = o_{t_j}^i \otimes \tanh\left(c_{t_j}^i\right) \tag{6}$$

On the basis of the LSTM processing, an Attention mechanism was added to improve the prediction accuracy and reduce the error. The each Output_i was input to the Attention mechanism respectively, and weights of each hidden layer vector are derived by training with Attention and normalization using the softmax function. The output is connected at the fully connected layer according to the weights. The calculation formula is

$$M_i = \tanh(W_S * \text{ Output}_i + \text{b}_S) \tag{7}$$

$$a_t^i = \text{soft}\max(M_i) \tag{8}$$

A weighted average sum of the hidden layer output vectors using the trained weights.

$$\text{result}_i = a_t^i \cdot \text{ Output}_i = a_{t_1}^i \cdot \text{ output}_{t_1}^i + a_{t_2}^i \cdot \text{ output}_{t_2}^i + \cdots + a_{t_n}^i \cdot \text{ output}_{t_n}^i \tag{9}$$

The final output of this pooling layer is obtained, and the set of outputs of the k pooling layers is Result={$result_1$,$result_2$,⋯， $result_k$}.

3.4 GA optimized parameters for constructing fully connected layers

The Result is input into the GA algorithm module, and the weight of each is set to $\frac{1}{k}$. Then the crossover operation is performed by the multi-point intersection method, and the chromosomes between two individuals are exchanged, with the crossover probability set to 0.8 and the variation rate set to 0.003. Global search is performed to obtain the optimal fully connected coefficient $\{b_i\}$, the final fully connected layer is obtained, to output the prediction result *Prediction*.

$$\text{Prediction}=b_1 \bullet result_1 + b_2 \bullet result_2 + \cdots + b_k \bullet result_k \tag{10}$$

Algorithm 1 : CNN- LSTM with Attention optimized by GA for prediction

Input: Original stock data collected S = $\{s_1, s_2, \cdots, s_n\}$. RFE module:
features−to−select = 20. PCA module: components = 0.98. CNN—LSTM with Attention module: Number of convolution: size = k, Number of convolution layers: layers = 2, These are important parameters in model training: window = 5, lstm_units = k, dropout = 0.001, epoch = 200 . GA: crossover probability_pc = 0.8, mutation probability_pm = 0.003.

Output: stock prediction *Prediction*

1: Enter S = $\{s_1, s_2, \cdots, s_n\}$ into the Qlib platform ▷ Alpha158 of stock;

2: Alpha158 of stock input to RFE module ▷ X = $\{x_1, x_2, \cdots, x_{20}\}$;

3: Enter X = $\{x_1, x_2, \cdots, x_{20}\}$ into PCA module ▷ ST = $\{s_{t1}, s_{t2}, \cdots, s_{tk}\}$;

// k is number of principal components

4: Enter ST = $\{s_{t1}, s_{t2}, \cdots, s_{tk}\}$ into CNN—LSTM with Attention module;

PoolingST$_i$ ⇐ Maxpooling (Con1 D) ;

LSTM layer train PoolingST$_i$ respectively ▷ Output$_i$;

every Output$_i$ input into Attention ▷ Result$_i$;

5: Enter Result$_i$ into the GA module ▷ weight $\{b_i\}$;

6 : Connect each Result$_i$ of pooling into Dense layer based on weights b_i; Prediction ⇐$\sum_{i=1}^{k} b_i$ · Result$_i$

7 : **return** Prediction.

4. EXPERIMENTS AND RESULTS ANALYSIS

4.1 Datasets and Features

In this paper, in order to test the proposed model and verify the stability and accuracy of the model, we selected 500 stocks from the Shanghai Stock Exchange (.SH) to train and test the model. Stocks from 600000.SH to 600499.SH were obtained with the help of the Tushare open data platform (Table 1), where '*Datasets*' is the exchange code of the stock, '*Market*' is the market in which the stock is traded, '*Frequency*' is the trading frequency of the stock, '*Stocks*' is the number of stocks acquired, '*Days*' is the trading day interval of the stock under study, and '*Features*' is the basic characteristics factor of the stock.

Table 1 Stock's features, where, volume is the trading volume, open is the opening price of the trading day, close is the closing price of the trading day, high is the highest trading price, and low is the lowest trading price.

Datasets	Market	Frequency	Stocks	Days	Features
SH	China	1 day	500	From 01/01/2013 to 12/12/2022	Volume, money, open, close, high, low, change

Each stock data acquired above was saved as a separate .csv file with the name of the stock code and the folder named sh_data. data was obtained from 1 January 2013 to 12 December 2022, for a total of 500 stocks. The data in the sh_data folder was then converted to .bin and .txt format by calling the dump_all command in the Qlib platform and imported into the Alpha158 factor database in the Qlib platform[12] to calculate the Alpha158 factor for these 500 stocks (Table 2).

Table 2 Partial Alpha360's factor, where KMID is the intra-day increase or decrease, KLEN is the intro-day amplitude, and VWAP is the Volume Weighted Average Price.

Alpha158	e.g., KMID=(close-open)/open
	e.g., KLEN=(high-low)/open
	e.g.,VWAP=$(\sum volume_i \times ((high_i + low_i + close_i)/3)/(\sum volume_i)$
	…………

The Alpha 158 Factor is a set of 158 technical indicators that reflect some of the characteristics of a stock by adding, subtracting, multiplying and dividing basic stock data, taking absolute values, finding logarithms, finding means and variances, etc. It contains technical indicators such as KMID, KLEN and VWAP. KMID is the intra-day ups and downs, reflecting the ups and downs of the stock in a day, KLEN is the intra-day amplitude, reflecting the vibration of the stock in a day, and VWAP is the volume-weighted average price, which is a comprehensive consideration of the impact of volume on the stock price.

These 158 indicators are more comprehensive and more reflective of the relationships between raw stock data, and they can be used to reflect more information about the stock, which can be used to make stock predictions more accurate. So these indicators are fed into the RFE and PCA modules instead of raw data for feature reduction and extraction.

4.2 Experiment setup

In this paper, we compare this model(baseline) with LSTM model, CNN-LSTM model and LSTM with Attention model for comparison experiments and with CNN-LSTM with Attention for ablation experiments.

4.3 Comparative experiment to predict the price of a single stock

600000.SH stock was used to compare the CNN-LSTM model with GA optimized Attention mechanism proposed in this paper, with the traditional LSTM model, CNN-LSTM model, and LSTM model with Attention. First we took the basic basic daily average information of this stock during the trading period, import the data into the Qlib platform, calculate the Alpha158 factors, and use RFE and PCA modules to downscale these 158 factors and extract the factors

(Figure 2 and Figure 3), and then divide these data into training_data and test_data.(Figure 4) to train the 4 models, and get Figure 5, it is easy to find that the overall error of the baseline model proposed in this paper is smaller than the other models in terms of the error of the model test (Table 3), The MAE was 0.00612, the RMSE was 0.08891 and the MAPE was 0.04391, all of which significantly indicated the best predictive accuracy of the baseline model. And the prediction of the price of the future trading day of the model (Table 4), the baseline predicts a closing price of 7.407576 for the first trading day that follows and 7.363454 for the third trading day, which are higher than the other models. Therefore, it can be shown that the baseline has better prediction effect and accuracy in SSE stock price prediction.

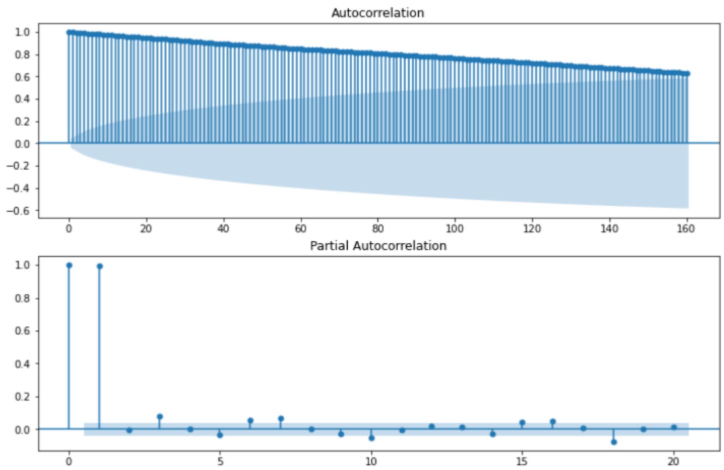

Figure 2 The autocorrelation coefficient and partial correlation coefficient of Alpha158 factor.

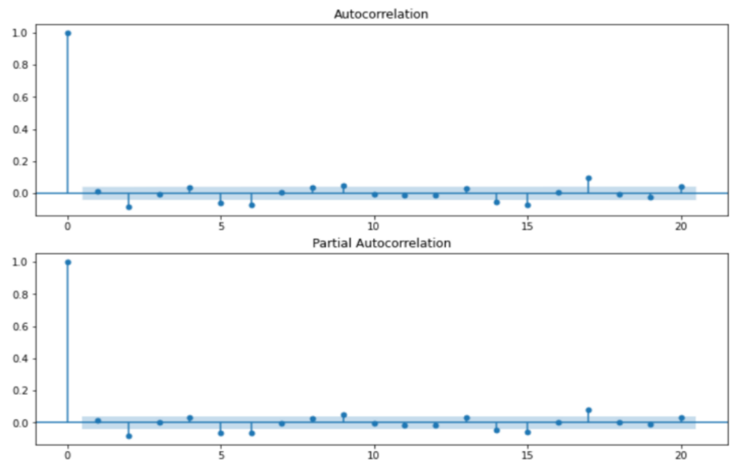

Figure 3 After RFE and PCA treatment, the autocorrelation coefficient and partial correlation coefficient of the extracted features.

Table 3 Compare the experimental error results,pink number represents the minimum error of the four models, and red number represents the maximum of R^2.

Model	MAE	RMSE	MAPE	R^2
baseline	0.00612	0.08891	0.04391	0.98714
CNN-LSTM	0.01003	0.07621	0.05413	0.93758
LSTM with Attention	0.01806	0.19257	0.09921	0.85727
LSTM	0.00803	0.08963	0.05697	0.91224

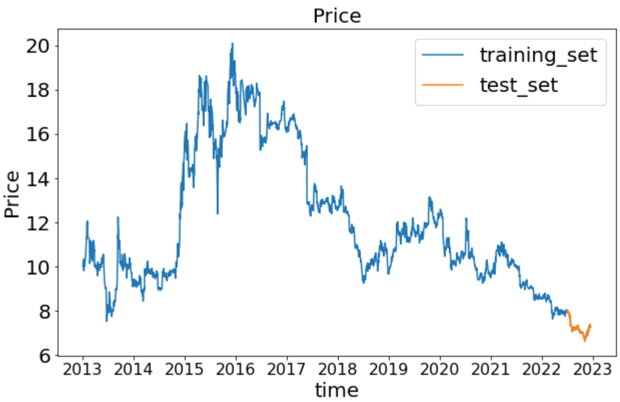

Figure 4 Distribution diagram of model training_set and test_set, where blue is the training set, orange is the test_ set.

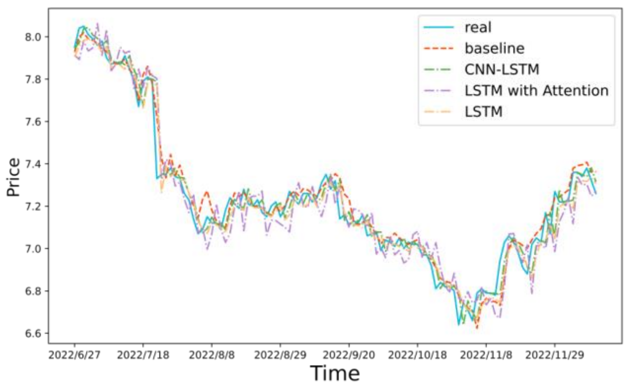

Figure 5 Comparison of experimental prediction results. Among them, the green '--' is the predicted result of the model proposed in this paper. The pre-test proposed in this paper. The pre-test and back-test result is close to the true value, and the predicted price in the last three trading days is also the highest.

Table 4 The predicted price of four models over the next three trading days, pink number is the highest price in each trading day.

Model	Forecast prices for the last three trading days Day1	Day2	Day3
baseline	7.407576	7.360251	7.363454
CNN-LSTM	7.338636	7.380911	7.314984
LSTM with Attention	7.304593	7.247571	7.273001
LSTM	7.316823	7.336634	7.304962

4.4 Model ablation experiment

The GA algorithm is added to the model to optimize the calculation of the coefficients in Attention, so the baseline model is compared with the CNN-LSTM with Attention model which is not optimized using GA (ablation test).With Figure 6, it can be seen that the stability of the baseline model outperforms model of ablation. Combined with the four errors in Table 5, the prediction accuracy of baseline also outperforms model of ablation, which can indicate that the results are more accurate by adding GA to the coefficient optimization of Attention for full connectivity. Moreover, the control variable method is used in the experiment, and the initial parameters of each model are the same. it can indicate that adding GA algorithm components in improving the prediction results accuracy is effective.

Table 5 Error analysis of ablation experiment,pink number represents the minimum error of the four models, and red number represents the maximum of R^2, It shows that the prediction accuracy of the model can be improved by using GA to optimize Attention.

Model	MAE	RMSE	MAPE	R^2
baseline	0.00612	0.08891	0.04391	0.98714
CNN-LSTM with Attention	0.03003	0.22832	0.10437	0.75758

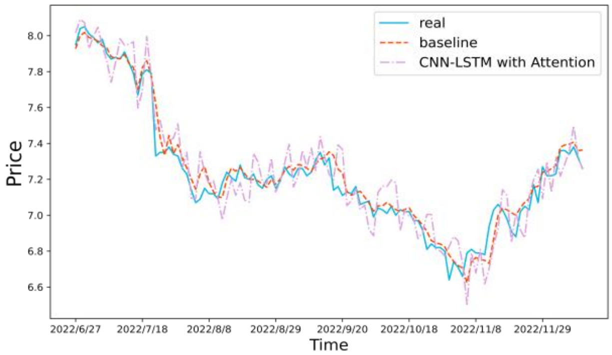

Figure 6 The graph of predicted results of ablation experiment, The red '--' is the predicted result of the model mentioned in this paper. Compared with the Attention model without GA optimization, the predicted result is more stable.

The above comparison and ablation experimental results show that the CNN-LSTM model with GA-optimized Attention has a smaller prediction error than the other four models and a higher R^2 than the other models, indicating that the model has the best prediction accuracy and the Attention model optimized by adding GA has the best effect on stock price prediction. It is the most optimal model in terms of prediction accuracy.

5. CONCLUSION

In this paper, the model for predicting future stock prices is investigated again. Based on the existing model,

CNN-LSTM model with GA optimized Attention is proposed, and the neural network are optimized by the GA algorithm to improve the accuracy of the model. In addition, the Alpha158 is used as the original stock data, and the RFE and PCA modules are used for dimension reduction and feature extraction as the input data for the model. The final experiments show that the proposed model has better accuracy and outperforms other models, and the GA-optimized Attention can further improve the accuracy of the model prediction according to the ablation tests.

REFERENCES

[1] Eugene F Fama. 1970. Efficient capital markets: A review of theory and empirical work. The Journal of Finance 25, 2 (1970), 383–417.

[2] Al-Shiab M. The predictability of the Amman stock exchange using the univariate autoregressive integrated moving average (ARIMA) model[J]. Journal of Economic and Administrative Sciences, 2006, 22(2): 17-35.

[3] Franses P H, Dijk D V. Forecasting stock market volatility using (non-linear) GARCH models[J]. Journal of Forecasting, 1996, 15(3): 229-235.

[4] Kim K. Financial time series forecasting using support vector machines[J]. Neurocomputing, 2003, 55(1-2): 307-319.

[5] Hochreiter, S, and J. Schmidhuber. “Long short-term memory.” Neural Computation 9.8(1997):1735-1780.

[6] D. Venugopal, S. K. Kaliyaperumal, and S. M. Niraikulathan, "Stock Market Trend Prediction Using Hidden Markov Model", in Forecasting in Mathematics - Recent Advances, New Perspectives and Applications. London, United Kingdom: IntechOpen, 2020 ,doi: 10.5772/intechopen.93988

[7] A. A. Ariyo, A. O. Adewumi and C. K. Ayo, "Stock Price Prediction Using the ARIMA Model," 2014 UKSim-AMSS 16th International Conference on Computer Modelling and Simulation, 2014, pp. 106-112, doi: 10.1109/UKSim.2014.67.

[8] D. M. Q. Nelson, A. C. M. Pereira and R. A. de Oliveira, "Stock market's price movement prediction with LSTM neural networks," 2017 International Joint Conference on Neural Networks (IJCNN), 2017, pp. 1419-1426, doi: 10.1109/IJCNN.2017.7966019.

[9] C. Li, X. Zhang, M. Qaosar, S. Ahmed, K. M. R. Alam and Y. Morimoto, "Multi-factor Based Stock Price Prediction Using Hybrid Neural Networks with Attention Mechanism," 2019 IEEE Intl Conf on Dependable, 2019,pp.961-966,doi:10.1109/DASC/PiCom/CBDCom/CyberSciTech.2019.00176.

[10] Kim,Han,"Genetic algorithms approach to feature discretization in artificial neural networks for the prediction of stock price index",Expert Systems with Applications,ISSN 0957-4174,
[11] SONG G, ZHANG Y F, BAO F X, et al. Stock prediction model based on particle swarm optimization LSTM. Journal of Beijing University of Aeronautics and Astronsutics, 2019, 45(12): 2533-2542.
[12] DING L M,Wang C,Temporal FusionTransformer stock picking model based on QLIBALPHA360[R],Hong Kong,CITIC Construction (International) Securities Co., Ltd.2022.05.24