

Stock Price Prediction Based on ARIMA-GM Hybrid and LSTM Model

Xiaochuan Huang

*School of Mathematics and Statistics, Beijing Institute of Technology, Beijing, China
1120221527@bit.edu.cn*

Abstract. The stock market has grown to be an essential component of China's economic market as the country's economy has developed. And stock prices are closely watched by investors and managers. This paper uses Python to crawl the monthly stock prices of Ping An of China from January 2016 to December 2024 and decides to estimate its closing price using the ARIMA, GM, and LSTM models. Furthermore, this paper mixes the ARIMA model with the GM model to build a hybrid model. The empirical results show that the ARIMA model's mean square errors is 35.79, while the GM model's is 125.63. The MSE loss function of the LSTM model is 14.186, and the MSE of the ARIMA-GM hybrid model is 7.575. The results of this paper show that the hybrid model's prediction accuracy surpasses not only that of a single model but also that of the LSTM model in machine learning, providing a scientific and effective guide for investor and manager behavior.

Keywords: Stock Price Prediction, ARIMA, GM, LSTM, Hybrid Forecast

1. Introduction

As one of the core elements of the economic market, stocks closely reflect the overall operation trend of the macro-economy. Especially in the domestic stock market, the stock fluctuations of some large financial groups reflect the state of the national economy to a certain extent. Therefore, both investors and managers are paying close attention to the trend of stock prices, and stock price prediction has become a popular financial study topic.

In recent years, time series analysis and machine learning have received extensive attention from researchers in stock price prediction. Cui used the ARIMA, ARCH and GARCH models in time series analysis to study stock returns and found that the ARIMA model had the best fitting effect [1]. Huang explained the modeling process and model evaluation of the ARIMA model in detail, pointing out that the ARIMA model is appropriate for short-term predictions, but further research is needed for long-term prediction [2]. Many researchers have begun to combine ARIMA models with machine learning models. The prediction effects of Fu's ARIMA-RF combined model [3] and Guan's ARIMA-RNN hybrid model were better than those of a single model, which effectively improved the prediction accuracy [4]. Xiao forecasted stock values using the LSTM and ARIMA models and found that ARIMA had a good short-term prediction effect, while LSTM was more effective in predicting all stock prices, but no model combination was performed [5]. Gu successfully increased the model's prediction accuracy after mixing by feeding the ARIMA model's prediction error into

LSTM [6]. Due to the large volatility of stock prices, some researchers also use gray models to predict them. Tian et al. proposed a similar gray model based on the entanglement theory, which effectively improved the prediction accuracy of oscillation sequences [7]. On this basis, Guo et al. established the ARIMA-GM multivariate regression model based on the grey theory and optimized the fitting effect [8].

The above studies mostly use a single method to select ARIMA model parameters. This paper uses a variety of methods to select parameters and compare them, then establishes a residual-corrected GM (2,1) model, mixes the GM model with the ARIMA model by weighted average, and forecasts stock prices using it, and compares the prediction results with those of the LSTM model.

2. Data and model methods

2.1. Source and description of data

The source of the data is the AKShare data interface. Python crawled the monthly stock prices of Ping An of China from January 2016 to December 2024 on AKShare as the research object. The data includes date, opening price, closing price, top price, bottom price, number of traded stocks and transaction amount, a total of 108 data.

2.2. Selection and description of indicators

This paper only uses closing prices for empirical research. The closing price is the price at which a stock was last traded at the end of a trading day. Investors closely monitor it, which might show the market's supply and demand balance at the end of the trading day. When evaluating the model performance, this paper contrasts meaning absolute error (MAE), mean relative error (MSE), and root mean squared error (RMSE), with the priority of $MSE > RMSE > MAE$. The calculation formula for the three indicators are as follows:

$$MSE = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2 \quad (1)$$

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2} \quad (2)$$

$$MAE = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i| \quad (3)$$

The sample size is n , and the actual value is y_i , whereas the anticipated value is $\hat{y}_i$,

2.3. Methods introduction

2.3.1. ARIMA model

ARIMA model is commonly used in time series analysis. It is made up of three parts: AR (autoregressive model), I (difference process) and MA (moving average model). The time series' historical values are used by the AR part to forecast the present value. The I part transforms the non-stationary series into a stationary series by performing differential processing on the initial data. The MA part uses the past prediction errors to correct the prediction [9]. The ARIMA model has three parameters. 'p' represents the autoregressive order, 'd' represents the difference order, and 'q' represents the moving average order. The ARIMA(p, d, q) may be stated mathematically as follows:

$$AR : y_t = c + \emptyset_1 y_{t-1} + \emptyset_2 y_{t-2} + \cdots + \emptyset_p y_{t-p} + \epsilon_t \quad (4)$$

$$I : \Delta^d y_t = \Delta^{d-1} y_t - \Delta^{d-1} y_{t-1} \quad (5)$$

$$MA : y_t = c + \epsilon_t + \theta_1 \epsilon_{t-1} + \theta_2 \epsilon_{t-2} + \cdots + \theta_q \epsilon_{t-q} \quad (6)$$

$$ARIMA : \Delta^d y_t = c + \emptyset_1 \Delta^d y_{t-1} + \cdots + \emptyset_p \Delta^d y_{t-p} + \epsilon_t + \theta_1 \epsilon_{t-1} + \cdots + \theta_q \epsilon_{t-q} \quad (7)$$

In which t is the current time point, y_t is the value at the current time point, $\emptyset_i$ is the autoregressive coefficient ($i = 1, 2, \dots, p$), θ_i is the moving average coefficient ($i = 1, 2, \dots, q$), ϵ_t represents the error at the current time point, whereas c is the time series' long-term mean.

The modeling procedure for the ARIMA model involves stationarity test (divided into Augmented Dickey-Fuller test and Ljung-Box test), model order determination, model prediction and other parts. Figure 1 shows the exact flow chart:

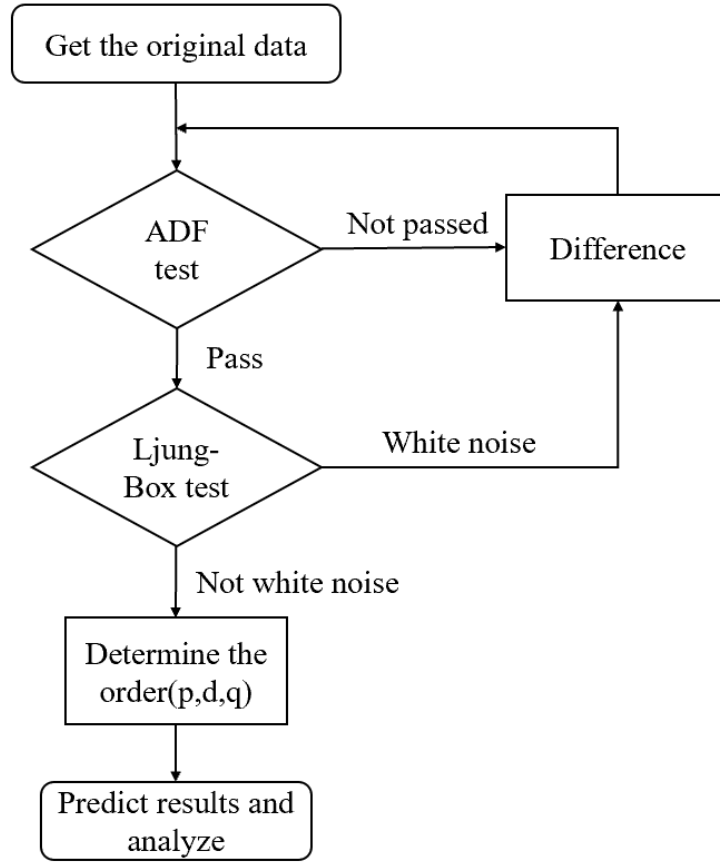


Figure 1. ARIMA modeling flow chart (photo credit: original)

2.3.2. Grey Theory Model: GM (2,1)

Famous professor Deng Julong proposed the grey system hypothesis, and it is used to deal with systems with small samples, strong uncertainty, and incomplete information. The simplest model in grey system theory, GM (1,1), is frequently employed for time series data. GM (2,1) introduces a second-order differential equation based on GM (1,1), which can better handle nonlinear and volatile data. The modeling process is as follows.

Assume the original time series is:

$$X^{(0)} = \{x^{(0)}(1), x^{(0)}(2), \dots, x^{(0)}(n)\} \quad (8)$$

In which $x^{(0)}(k)$ represents the observed value at the k th time point. Then the original sequence is accumulated once to obtain the cumulative sequence $X^{(1)}$:

$$x^{(1)}(k) = \sum_{i=1}^k x^{(0)}(i), k = 1, 2, \dots, n \quad (9)$$

According to the cumulative sequence, the second-order differential equation is obtained:

$$\frac{d^2 x^{(1)}}{dt^2} + a_1 \frac{dx^{(1)}}{dt} + a_2 x^{(1)} = b \quad (10)$$

In which a_1 , a_2 and b are all parameters to be estimated, and the equation is solved using the least squares method:

$$\begin{aligned} \hat{a} &= [a_1 \ a_2 \ b]^T = (B^T B)^{-1} B^T Y \\ \text{Observation Matrix : } Y &= [x^{(0)}(2) \ x^{(0)}(3) \ \dots \ x^{(0)}(n)]^T \\ \text{Coefficient Matrix : } B &= \begin{bmatrix} -z^{(1)}(2) & -x^{(1)}(2) & 1 \\ -z^{(1)}(3) & -x^{(1)}(3) & 1 \\ \vdots & \vdots & \vdots \\ -z^{(1)}(n) & -x^{(1)}(n) & 1 \end{bmatrix} \end{aligned} \quad (11)$$

The equation's solution is the expected value of the cumulative sequence $\hat{x}^{(1)}(k)$, which is then subtracted to get the anticipated value of the original sequence $\hat{x}^{(0)}(k)$:

$$\hat{x}^{(0)}(k) = \hat{x}^{(1)}(k) - \hat{x}^{(1)}(k-1), k = 2, 3, \dots, n \quad (12)$$

2.3.3. ARIMA-GM hybrid model

The mixed mode of the model is weighted average mixing, that is, the predicted value of the mixed model is expressed as:

$$\hat{y}_t = w \hat{y}_t^{ARIMA} + (1-w) \hat{y}_t^{GM(2,1)} \quad (13)$$

In which w is the weight coefficient, which is determined by grid search method, optimization algorithm, cross validation method and Bayesian optimization method in this paper.

To increase the hybrid model's predictive accuracy, this paper uses the moving average approach to smooth the prediction outcomes of the GM (2,1). Here's the specific method:

$$e(k) = x^{(0)}(k) - \hat{x}^{(0)}(k) \quad (14)$$

$$e = [e(1), e(2), \dots, e(n)] \quad (15)$$

In which $e(k)$ is the residual at the k th time point. Then the residual sequence e is moved averaged to obtain the smoothed residual sequence:

$$e_{smooth}(k) = \frac{e(k) + e(k-1) + \dots + e(k-h+1)}{h} \quad (16)$$

In which h is the number of time points of the moving average. Add the smoothed residual to the original forecast value to get the revised forecast value:

$$\hat{x}^{(0)}_{corrected}(k) = \hat{x}^{(0)}(k) + e_{smooth}(k) \quad (17)$$

2.3.4. LSTM model

Long Short-Term Memory(LSTM) is a type of improved Recurrent Neural Networks. Long-term dependencies are hard for the model to learn with typical RNNs since the gradient either decays or climbs exponentially as the number of time steps increases during training. LSTM adds memory units and gating mechanisms, this successfully overcomes the gradient disappearing and gradient bursting concerns of RNN while processing lengthy sequence data. [10]. LSTM working diagram is given in Figure 2:

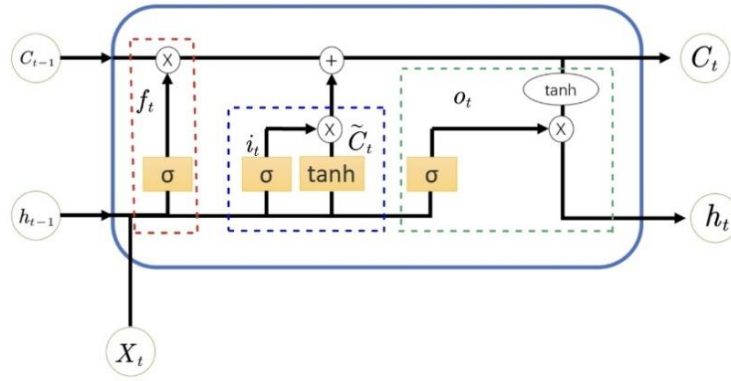


Figure 2. LSTM network structure diagram (by unit) (photo credit: original)

In figure 2, $\{X_1, X_2, \dots, X_t\}$ is the data input, $\{C_1, C_2, \dots, C_t\}$ is the memory cell state, and $\{H_1, H_2, \dots, H_t\}$ is the hidden state.

Forget Gate f_t : It determines which information is discarded. The hidden state H_{t-1} of the previous time step and the input X_t of the current time step are concatenated into a vector, which is then linearly transformed and input into the Sigmoid activation function to obtain a number in the interval $[0,1]$. If $f_t = 0$, the information in the memory cell is completely retained; if $f_t = 1$, it is completely retained; if $0 < f_t < 1$, it is partially retained. The specific expression is as follows:

$$f_t = \sigma(W_f \bullet [H_{t-1}, X_t] + b_f) \quad (18)$$

In which W_f is the weight matrix of the forget gate, b_f represented the bias term, and σ is the activation function.

Input Gate i_t : It determines which new information is stored in the memory cell. The Sigmoid activation function is used to output a number between $[0,1]$, indicating the retention ratio of the new information, in addition, the tanh activation function is employed to construct the candidate memory cell state simultaneously. The output result is the sum of the forget gate's product with the memory cell state at the previous time step, as well as the new information added under the control of the input gate. The specific expression is as follows:

$$i_t = \sigma(W_i \bullet [H_{t-1}, X_t] + b_i) \quad (19)$$

$$\tilde{C}_t = \tanh(W_C \bullet [H_{t-1}, X_t] + b_C) \quad (20)$$

$$C_t = f_t \bullet C_{t-1} + i_t \bullet \tilde{C}_t \quad (21)$$

In which $\tilde{C}_t$ denotes the candidate memory cell, whereas W_i and W_C represent the weight matrices for the input gate and candidate memory unit respectively, b_i and b_C are both bias terms.

Output Gate o_t : It determines what data the memory cell C_t sends to the hidden state H_t . After compressing the memory cell state to $[-1,1]$ using the tanh activation function and creating a number in the $[0,1]$ interval with the Sigmoid activation function, the output is the outcome of the two previously stated components. Here are the precise expressions:

$$o_t = \sigma(W_o \bullet [H_{t-1}, X_t] + b_o) \quad (22)$$

$$H_t = o_t \bullet \tanh(C_t) \quad (23)$$

W_o is the weight matrix of the output gate, while b_o is the bias term.

3. Empirical analysis

This paper's models employ Ping An of China's data from January 2016 to November 2023 is used as the training set, and the remaining data from December 2023 to December 2024 is used as the test set.

3.1. ARIMA model analysis

3.1.1. Stationarity test

The original data is examined for stationarity. This paper separates it into two rounds: ADF test and Ljung-Box test. Only when both sets of tests pass can the sequence be called stationary. The test

statistics are all below the critical thresholds of 1%, 5%, and 10%, and the p-value is less than the significance level of 0.05, which means that the test has passed. The ADF test results in Table 1. show that the original sequence fails, while the first-order difference sequence and the second-order difference sequence both pass. For the Ljung-Box test, the p-value less than the significance level of 0.05 is considered to have passed the test. The results are shown in Table 2.

Table 1. ADF test results

Sequence type	ADF statistics	Crucial value(1%)	Crucial value(5%)	Crucial value(10%)	p-value
Original	-1.789	-3.493	-2.889	-2.581	0.386
First-order difference	-10.196	-3.494	-2.889	-2.582	0
Second-order difference	-6.145	-3.494	-2.889	-2.582	0

Table 2. Ljung-Box test results

Sequence type	p-value	Test result
First-order difference series	0.064	The null hypothesis cannot be rejected, it is white noise
Second-order difference series	8.35×10^{-8}	The null hypothesis can be rejected, it is not white noise

Through the above two rounds of tests, it can be considered that the second-order difference sequence is a stationary sequence.

3.1.2. Parameter determination

The result of the stationarity test shows that the difference order is $d=2$. To determine p and q , this paper uses four methods: autocorrelation figure (ACF) and partial autocorrelation figure (PACF), AIC minimum criterion, BIC minimum criterion and cross-validation technique are used to find the best parameters by comparing their MSE, RMSE, and MAE values. Figures 3 and 4 illustrate the ACF and the PACF. It can be seen from the two figures that the ACF is shortened at order=1, whereas the PACF is truncated at order=2, so $p=2$, $q=1$. The final parameter results are shown in Table 3.

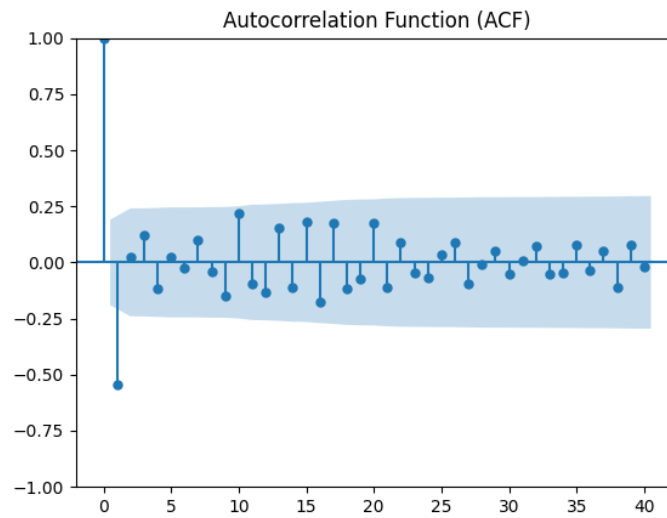


Figure 3. ACF plot of the second-order difference series (photo credit: original)

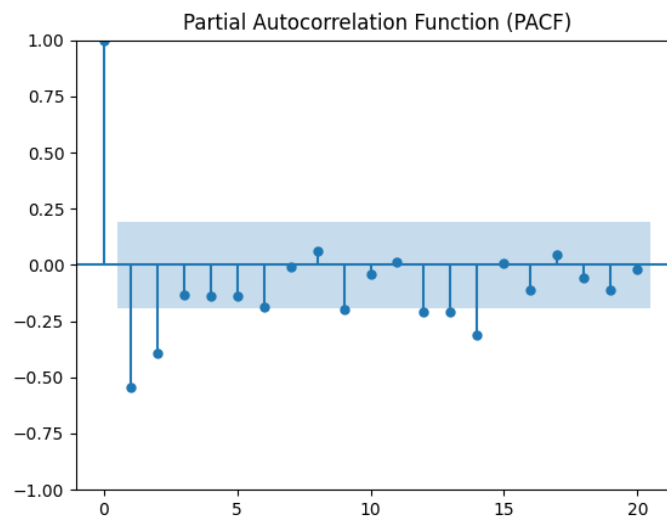


Figure 4. PACF plot of the second-order difference series.(photo credit: original)

Table 3. Parameter comparison table

Method	ARIMA Model	MSE	RMSE	MAE
ACF and PACF	(2,2,1)	56.745	7.533	5.375
AIC minimum criterion	(2,2,4)	31.673	5.628	4.406
BIC minimum criterion	(0,2,1)	60.298	7.765	5.545
cross-validation method	(4,2,4)	32.427	5.695	4.357

After comprehensive comparison, ARIMA (2,2,4) was finally selected as the optimal model.

3.1.3. Model checking

This paper uses QQ graph to test the model, as shown in Figure 5. After observing the QQ graph, it is found that most of the data points are near the straight line. It can be considered that the residuals conform to the normal distribution characteristics and the model can make effective predictions.

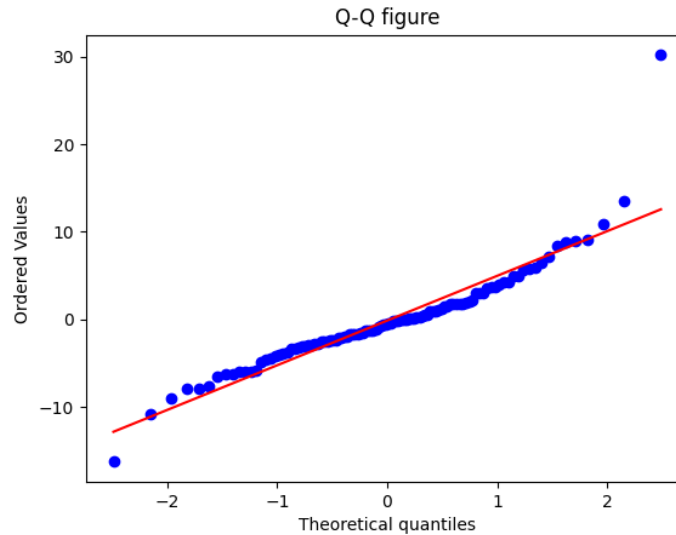


Figure 5. Q-Q figure (photo credit: original)

3.1.4. Prediction results

The prediction results of the ARIMA (2,2,4) model are shown in Figure 6.

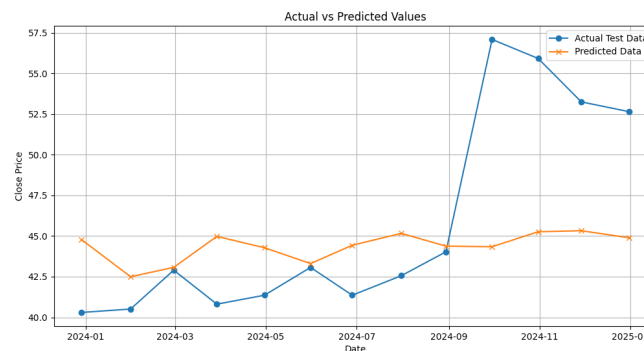


Figure 6. ARIMA model prediction graph (photo credit: original)

3.2. GM (2,1) model analysis

3.2.1. Prediction results

According to the formula mentioned above, the number of time points for moving average is selected as $h=3$, Using Python to calculate the modified GM (2,1) model parameters, and solve for $a_1 = -0.025$, $a_2 = 0.006$, $b = 36.189$. Substituting the parameters into the second-order

differential equation, continue to use Python to solve and predict, and obtain the results shown in Figure 7.

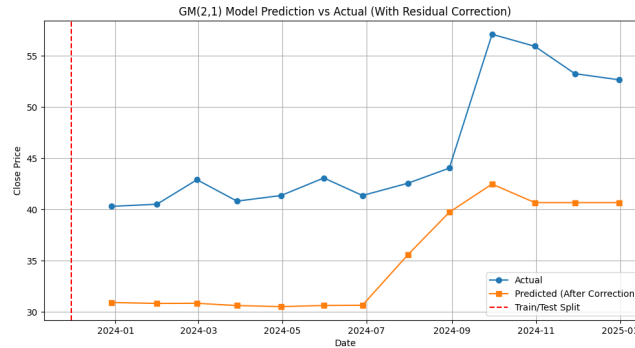


Figure 7. GM (2,1) model prediction graph (photo credit: original)

3.2.2. Model evaluation

The model was subjected to posterior difference test and association test. The posterior difference ratio $C = 0.214 < 0.35$, the small error probability $p = 1 > 0.95$, and the correlation is 0.67, close to 1. The above shows that the model has good prediction accuracy.

3.3. ARIMA (2,2,4)-GM (2,1) hybrid model analysis

This paper proposes an ARIMA-GM hybrid model based on the ARIMA model and the GM model, and the hybrid method is weighted average hybrid. The optimal weight is selected according to the above formula (18), and the results are shown in Table 4.

Table 4. Hybrid model weight comparison table

Method	Optimal Weight(ARIMA)	MSE	RMSE	MAE
Grid Search	0.1	7.578	2.753	1.989
Optimization Algorithm	0.11	7.575	2.752	1.988
Cross Validation	0	8	2.829	2.014
Bayesian Optimization	0.11	7.575	2.752	1.987

The above table shows that 0.11 is the optimal weight that may be obtained by adopting the minimum MSE strategy:

$$\hat{y}_t = 0.11\hat{y}_t^{ARIMA} + 0.89\hat{y}_t^{GM(2,1)} \quad (24)$$

Results of the forecast are displayed in Figure 8.

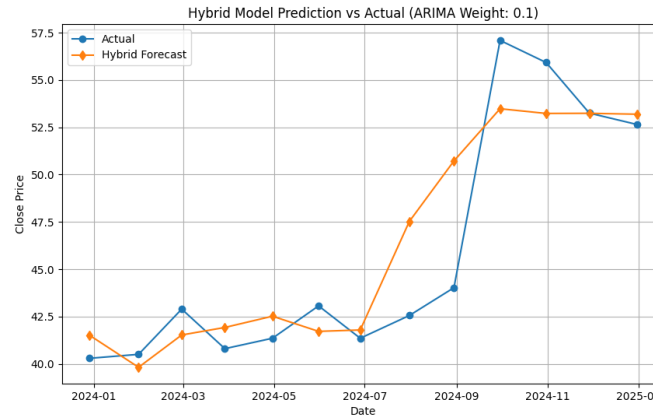


Figure 8. ARIMA-GM hybrid model prediction graph (photo credit: original)

3.4. LSTM model analysis

3.4.1. Data processing

In order to speed up the training process and improve model performance, the original data is generally not directly input into the LSTM model, but is normalized first. This paper uses the Min-Max normalization method to scale the data to the interval $[0,1]$. The specific formula is as follows:

$$X_{norm} = \frac{X - X_{min}}{X_{max} - X_{min}} \quad (25)$$

Where X is the original data, X_{max} is the maximum value, and X_{min} is the minimum value.

3.4.2. Prediction results

The LSTM model in this paper contains 2 hidden layers and 1 fully connected layer. Each hidden layer has 50 neurons, whereas the fully connected layer has 25 neurons. MSE is the loss function, whereas Adam is the optimization function. Each training uses 12 samples and is trained 100 times. The final prediction results are shown in Figure 9.

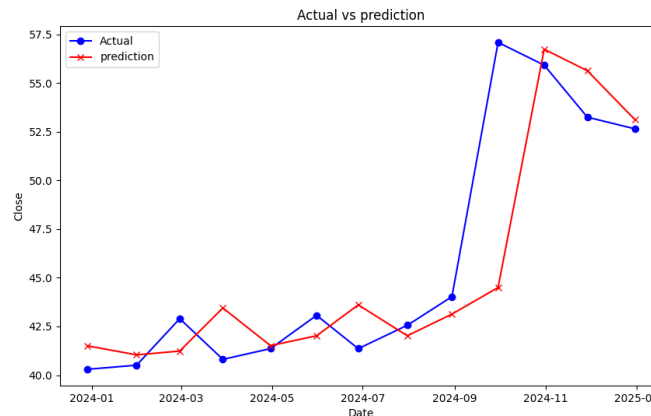


Figure 9. LSTM model prediction graph (photo credit: original)

3.5. Model comparison

The model performance comparison of ARIMA (2,2,4), GM (2,1), ARIMA-GM hybrid model and LSTM model is shown in Table 5.

Table 5. Comparison of four models' predictive abilities

Model	MSE	RMSE	MAE
ARIMA(2,2,4)	35.790	5.980	4.540
GM(2,1)	125.630	11.208	10.847
ARIMA-GM	7.575	2.752	1.988
LSTM	14.186	3.766	2.060

Table 5 demonstrates that the ARIMA and GM models' prediction results are inferior to the LSTM model, while the ARIMA-GM model after weighted mixing outperforms the LSTM model.

4. Conclusion

This paper selects the historical closing price of Ping An of China as a time series, and constructs ARIMA, GM, LSTM and ARIMA-GM hybrid model to predict future stock prices. The empirical results reveal that the hybrid model's prediction impact is superior to both the LSTM model and the individual ARIMA and GM models. The hybrid model's ARIMA model can capture linear data features, while the GM model can catch nonlinear ones, which explains this. The accuracy of the predictions can be increased by combining the two. Future stock price predictions can be made more accurate by increasing the model's prediction accuracy., better help investors avoid risks and increase returns, and also help managers balance the market and promote market stability.

At the same time, the hybrid model described in this paper also has shortcomings. This paper's data set is divided in a predetermined way. This fixed division method may not adapt to the dynamic changes of the data, especially when the data has obvious trends or seasonality. In subsequent research, we can try to use rolling forecast or cross-validation methods to further optimize the performance of the model.

References

- [1] Yupeng Cui. Application of Time Series Analysis in Stock Returns——Based on ARIMA and ARCH Models. FUJIAN ZHILIANG GUANLI, 2020(2): 184, 183.
- [2] Mengting Huang. An Empirical Study on Stock Price Prediction Based on ARIMA Model. NEI JIANG Science & Technology, 2023, 44(03): 61-62.
- [3] Fu Wenzhi. Research on Stock Price Prediction based on ARIMA-RF Combination Model. Scientific and Technological Innovation, 2023, (08): 40-43.
- [4] GUAN Xueying. Stock price prediction based on ARIMA - RNN hybrid mode. Journal of Harbin University of Commerce (Natural Sciences Edition).2024, 40(02): 250-256.
- [5] Jie Xiao. Research on Stock Price Prediction Based on ARIMA and LSTM Neural Network—Taking Citic Securities (600030) as an Example. E-Commerce Letters, 2024, 13(4): 11.
- [6] Zhenhong Gu. Stock Price Prediction Based on ARIMA-LSTM—A Case Study of Kweichow Moutai. E-Commerce Letters, 2024, 13(4): 10.
- [7] Tian Hongli, Li Chengqun, Yan Huiqiang. Research and application of prediction method based on entanglement theory and similar grey model in prediction of stock price inflection point. Application Research of Computers, 2020, 37(06): 1666-1669+1678.
- [8] GUO Gaiwen, WANG Shihan. Stock Price Prediction Based on Grey Theory and ARIMA Mode. Journal of Henan Institute of Education (Natural Science Edition), 2023, 32(02): 22-27.
- [9] Mohd Faizan Rizvi1, Shivang Sahu2, Dr. Sadhana Rana. ARIMA Model Time Series Forecasting. International Journal for Research in Applied Science and Engineering Technology. 2024, 12(5), 3782-3785.
- [10] Wang Fei. Research progress and application of LSTM recurrent neural network. Heilongjiang University, 2021.