Stock Prediction Based on ARIMA Model and GRU Model

Zijie Zhong[1,*], Difei Wu[1], Wenxuan Mai[2]

[1]School of Data Science, Beijing Normal University - Hong Kong Baptist University United International College, Zhuhai, 519087, China
[2]School of Data Science, Beijing Normal University - Hong Kong Baptist University United International College, Zhuhai, 519087, China
[*]Corresponding author: q030026220@mail.uic.edu.cn

Abstract: *Although the market is getting back into order after COVID-19, it is still continuously affected by the epidemic virus. Researchers try to understand how a piece of stock changes over a period of time before and after COVID-19, in order to provide reliable marketing decisions to society. In this paper, using ARIMA (Auto-regressive Integrated Moving Average) model and GRU (Gate Recurrent Unit) model, we analyse close price of Apple stock from 2018 to 2023 (1258 data in total) and use 70% of data for model training, whereas the remaining 30% of data are for model prediction and evaluation. Results show that the ARIMA model is able to extract trend of the stock but has bad performance on fitting and predicting real data; the R-square for GRU on test data reaches to 0.936.*

Keywords: *Stock Prediction, ARIMA model, GRU model*

1. Introduction

Nowadays, time series data are important objects to study and stock analysis is necessary to dive into a market. People usually need to extract patterns of the data to make decisions on weather forecasting, disease prevention and investment. However, time series data, especially stock, are not easy to extract due to unstable volatility caused by national and social policies. In the past, people could only follow a rule called Buy Low and Sell High to do trade in stock market, which was random. It could either go high when a company is boosting, or go down when financial status is too bad to continue the dividend and support stock. Under unpredictable circumstance, a feasible way to avoid potential risks and gain benefits is that some models can be trained based on historical data of stock and make predictions in a specific time interval. As machine learning is swiftly developing, Neural Network derives many transformation models which are proved to have better performance on prediction problems as they can discover complex and hidden patterns in data more efficiently compared to physical and empirical models [1]. In this paper, we build the ARIMA model and GRU model with the best parameters; then we generate judge criterions, including diagnostic analysis, R-square, MAE (Mean Absolute Error) and so on.

2. Model Construction

2.1. Data Source and feature selection

Apple stock data used in this paper are downloaded from Yahoo finance database. In the dataset, there are open price data and close price data for Apple and they are collected from May 14th, 2018, to May 11th, 2023. We only study the close price data from May 14th, 2018, to May 11th, 2023, shown in Figure 1.

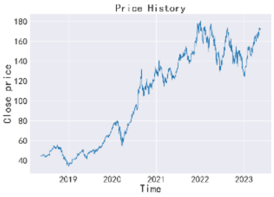

Figure 1: Close price data

2.2. ARIMA model

ARIMA model is widely applied in solving problems related to time series. There are three important parameters p, d and q, to define Auto-Regressive term, Integrated term and Moving Average term, respectively. Auto-Regressive term describes lags of original time series data, while Integrated term describes difference order and Moving Average term describes lags of prediction error in the model. ARIMA model can be further written in the following way:

$$Y_t = \alpha + \beta_1 Y_{t-1} + \beta_2 Y_{t-2} + \cdots + \beta_p Y_{t-p} + \phi_1 \epsilon_{t-1} + \phi_2 \epsilon_{t-2} + \cdots + \phi_p \epsilon_{t-p} \quad (1)$$

We need to follow process to analyze data and build the model successfully:

2.2.1. Data cleaning

Data cleaning aims at recognizing and removing outliers and null values, by which model performance could be harmed. In this paper, we use data statistics and box-plot method to achieve this goal shown by Table 1 and Figure 2:

Table 1: Data statistics

Column	Non-null Count
Date	1258 non-null
Close	1258 non-null

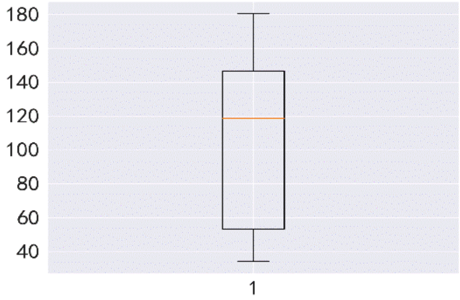

Figure 2: Box-plot for close price

2.2.2. Stationarity test

ARIMA model requires that data should be stational. From Figure 1 we have seen that the original data may be non-stable because they fluctuate and show upward trend overall. In this paper, we apply ADF (Augmented Dickey-Fuller) test to check whether time series is stable or not.

Mathematically, stationarity can be classified as weak stationary random process and strictly stationary random process. Mostly we discuss the weak stationary random process for the following conditions:

$$E(y_t) = E(y_{t+m}), for\ any\ t, m \quad (2)$$

$$Cov(y_t, y_{t+k}) = cov(y_{t+k}, y_{t+k+m}), for\ any\ t, k, m \quad (3)$$

According to ADF test, if there exits a unit root, time series would be unstable. In this case, there would be deceptive relationship between independent variables and dependent variables because any error in residual series would not decrease as sample size increases. That is to say, residual effect in a model is permanent, which can be explained by spurious regression. Table 2 give results of ADF test:

Table 2: ADF test (before smoothing)

T-statistics	-0.693
P-value	0.848
Lag used	1
Number of observations used	1256
T-statistics in 1% interval level	-3.436
T-statistics in 5% interval level	-2.864
T-statistics in 10% interval level	-2.568

From Table 2, it is clear that T-statistics of original data is larger than that in any of three confidence intervals; P-value is greater than 0.05. Thus, we cannot reject the null hypothesis, which means original data series is not stable.

2.2.3. White noise test

Before simulating the transformed time-series and making predictions, it is necessary to introduce the concept of ‘white noise’ for two reasons [2]. Firstly, a white noise series is a sequence of random numbers which cannot be predicted with the following conditions:

$$E(\epsilon_t) = \mu \tag{4}$$

$$Var(\epsilon_t) = \sigma^2 \tag{5}$$

$$Cov(\epsilon_t, \epsilon_s) = 0, t \neq s \tag{6}$$

In this paper, we use Ljung-Box test to determine whether time series is a white noise sequence. If correlations of some terms in the sequence are zero, statistics approaches to follow Chi-square distribution. Since stable time series is short-term dependent, we do not need to calculate all the correlations. By Ljung-Box test, we have:

$$Q(m) = T(T+2)\sum_{l=1}^{m}\frac{\hat{\rho}^2}{T-l} \tag{7}$$

where T is sample size, m is a chosen number and $\hat{\rho}_k$ is auto-correlation coefficient after k lags.

Table 3: Ljung-Box test results (before smoothing)

Lb_stat	Lb_pvalue
1254.045	1.097×10^{-274}
2502.535	0
3745.658	0
4983.548	0
6216.190	0
7443.312	0
8665.103	0
9881.231	0
11092.032	0
12297.710	0
13498.403	0
14693.732	0
15883.724	0
17068.397	0
18247.713	0
19422.208	0
20591.753	0
21756.459	0
22915.780	0
24069.539	0

From Table 3, it is clear that all P-values are smaller than 0.05. Because the null hypothesis assumes that the original data series is white noise, we reject it and assume that it is not a white noise sequence.

2.2.4. Data smoothing

To guarantee data to be stationary, in this paper we take log-first difference for all the data. Our sequence becomes:

$$A(n) = A(n) - A(n-1), n = 1,2, \ldots, k \tag{8}$$

where k is total number of values in the sequence.

Again, we do ADF test and Ljung-Box test for data transformation. Results are shown by the following Table 4 and Table 5, respectively.

Table 4: ADF test (after smoothing)

T-statistics	-10.820
P-value	1.816×10^{-19}
Lag used	8
Number of observations used	1248
T-statistics in 1% interval level	-3.436
T-statistics in 5% interval level	-2.864
T-statistics in 10% interval level	-2.568

Table 5: Ljung-Box test results (after smoothing)

Lb_stat	Lb_pvalue
17.928	2.294×10^{-5}
17.973	1.251×10^{-4}
18.646	3.235×10^{-4}
18.937	8.085×10^{-4}
22.693	3.863×10^{-4}
24.880	3.593×10^{-4}
37.934	3.119×10^{-6}
51.037	2.581×10^{-8}
67.663	4.365×10^{-11}
69.880	4.676×10^{-11}
70.597	9.408×10^{-11}
72.278	1.198×10^{-10}
74.307	1.281×10^{-10}
78.243	5.984×10^{-11}
83.684	1.472×10^{-11}
88.056	5.697×10^{-12}
88.555	1.117×10^{-11}
97.153	7.336×10^{-13}
97.790	1.343×10^{-12}
100.268	1.128×10^{-12}

Data now are stationary and not a white noise sequence.

2.2.5. Parameters modification

In this paper, we use two methods to decide which parameters we are going to apply. The first method is that we draw ACF (Auto-correlation Coefficient) and PACF (Partial Auto-correlation Function) plots. The second method is that we apply BIC (Bayesian Information Criterion) for evaluation.

The Auto-Regressive term "p" can be defined by PACF plot and the number of non-zero partial auto-correlations gives the order of "p" [3]. The ACF plot identifies Moving Average term "q" and it describes how well the present value of the time series is related to its previous values [4].

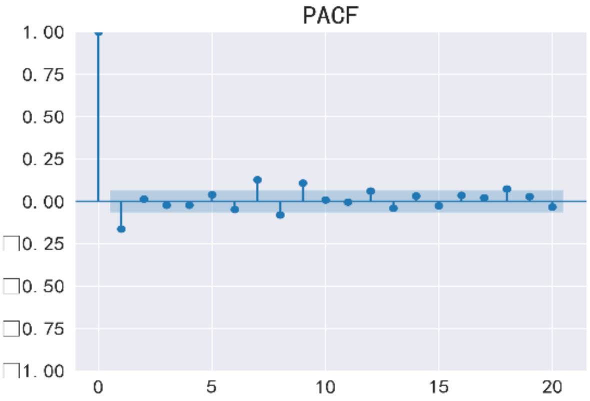

Figure 3: PACF plot

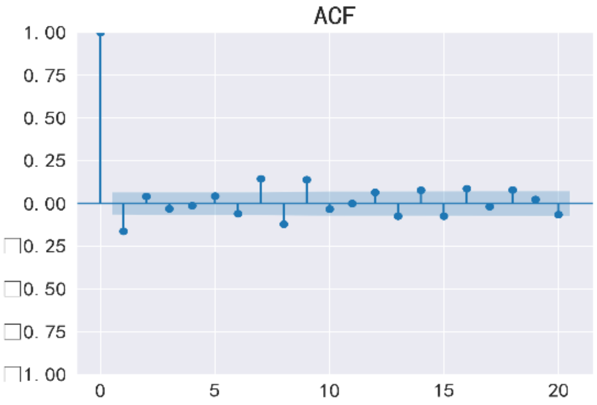

Figure 4: ACF plot

Figure 3 and Figure 4 show that after 1 lag, both auto-correlation coefficient and partial auto-correlation coefficient approach to 0. However, since two plots are censored, we need to further determine p and q by using BIC. The following formula gives:

$$BIC = -2\ln(L) + kln(n) \tag{9}$$

where L is maximum of likelihood function for a model, k is the number of free parameters to be estimated and n is sample size.

Setting a range of p and q, we loop all possibilities and calculate corresponding values of BIC shown in the following Table 6:

Table 6: BIC with different (p, q)

p	d	q	BIC
0	1	0	-3532.041
0	1	1	-4260.064
0	1	2	-4275.341
0	1	3	-4269.518
1	1	0	-3897.029
1	1	1	-4276.477
1	1	2	-4242.630
1	1	3	-4264.203
2	1	0	-4002.496
2	1	1	-4269.868
2	1	2	-4267.120
2	1	3	-4284.451
3	1	0	-4056.027
3	1	1	-4262.125
3	1	2	-4262.029
3	1	3	-4277.574

Since the lower BIC is, the better a model is, we select p=2 and q=3 for the ARIMA model. This selection gives the following information:

Table 7: ARIMA model results

	coef	P>|z|
ar.L1	-1.756	0.000
Ar.L2	-0.903	0.000
ma.L1	0.670	0.000
ma.L2	-0.862	0.000
ma.L3	-0.802	0.000
Sigma2	0.0004	0.000

Table 7 shows that all P values for terms in ARIMA model are smaller than 0.05, which means they are significant.

2.2.6. Model evaluation

To evaluate ARIMA model, in this paper we combine diagnostic plot with Durbin-Watson Test. A diagnostic plot includes residual plot, estimated density histogram, Q-Q (Quantile-Quantile) plot and correlogram, shown in following Figure 5, Figure 6, Figure 7 and Figure 8, respectively.

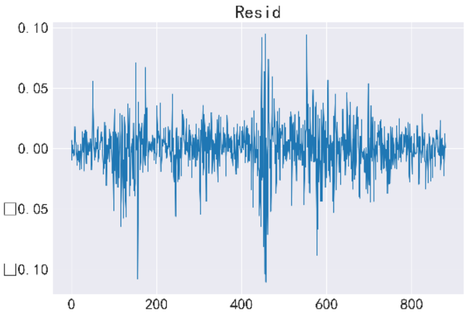

Figure 5: Residual

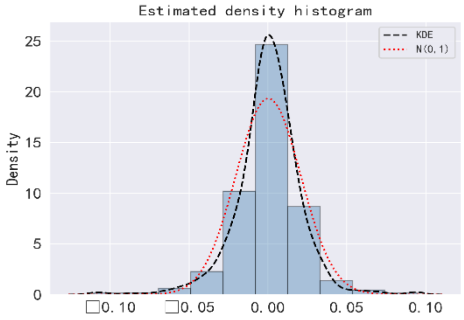

Figure 6: Estimated density histogram

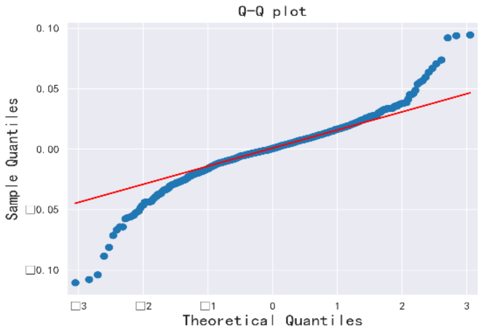

Figure 7: Q-Q plot

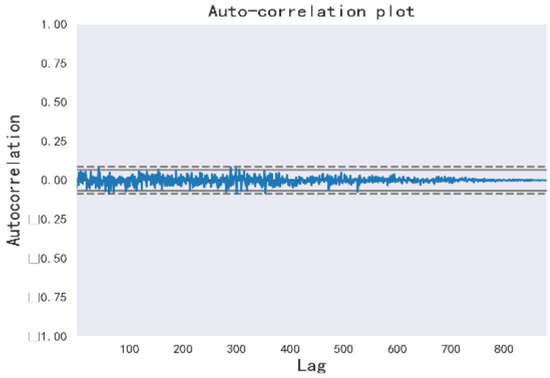

Figure 8: Auto-correlation plot

ARIMA model satisfies conditions that residuals are independent and that residuals generally follow normal distribution. More precisely, by Durbin-Watson test:

$$DW = \frac{\sum_{t=2}^{n}(e_t - e_{t-1})^2}{\sum_{t=2}^{n} {e_t}^2} \tag{10}$$

where e_t is the residual at t moment calculated by:

$$e_t = \rho e_{t-1} + v_t \tag{11}$$

where ρ is any real number and v_t is a constant term at t moment.

In our result, Durbin-Watson statistics is 2.075. This implies that ARIMA model passes Durbin-Watson test and residuals are independent.

2.2.7. Model forecasting

Using a model to predict futural data is always important. In this paper, we forecast the test data after first difference. The result is shown in Figure 9.

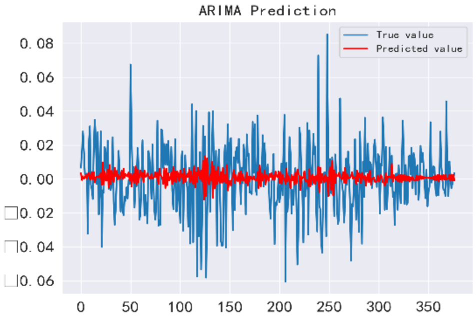

Figure 9: ARIMA forecasting visualization

2.3. GRU model

GRU networks fall into the category of RNNs, i.e., neural networks whose underlying topology of inter-neuronal connections contains at least one cycle [5]. Introduced in 2014, GRU is one kind of gated RNNs which are applied to solve gradient vanishing and explosion in traditional RNNs. A basic structure for GRU is illustrated in Figure 10:

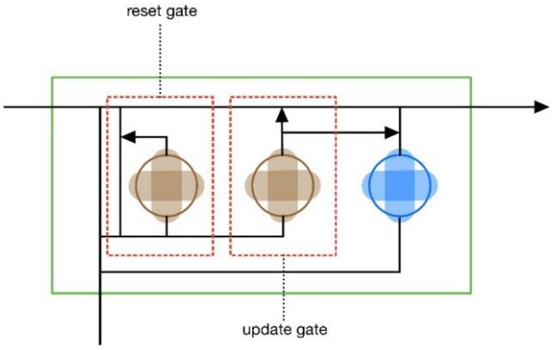

Figure 10: Basic structure of GRU

Reset gate and update gate are two important structures in GRU. Similar with LSTM (Long and Short-Term Memory), first it calculates gate values for update gate and reset gate, denoted by $z(t)$ and $r(t)$, respectively by the following formulas:

$$z(t) = \sigma(W_z \cdot [h_{t-1}, x_t]) \tag{12}$$

$$r(t) = \sigma(W_r \cdot [h_{t-1}, x_t]) \tag{13}$$

where σ is the linear transformation of x_t and h_{t-1}.

After that, with a sigmoid activation function, value for reset gate is used in h_{t-1}, which represents how much information passed down from last moment can be used. Based on this modified h_{t-1}, x_t

can do linear transformation with it. By $tanh$ activation function, we get new h_t. The following two formulas show how they are transformed.

Gate value for update gate will make effect on this new h_t while value for (1-gate value) will make effects on h_{t-1}. If we sum up these two results, finally we have output h_t in hidden condition. This process enables update gate to preserve previous results. When gate value reaches to 1, new h_t is output; instead, when it approaches to 0, h_{t-1} is output. The following two formulas show how they are transformed:

$$\widetilde{h_t} = \tanh(W \cdot [r_t \times h_{t-1}, x_t]) \tag{14}$$

$$h_t = (1 - z_t) \times h_{t-1} + z_t \times \widetilde{h_t} \tag{15}$$

2.3.1. Time series supervision transformation

Time series data must be transformed to a supervisory form before they are sent into any RNN. In this paper, we use every 7 data as inputs and 1 data after them as output. Precisely, our input timestamp is 7 while output timestamp is 1. Based on this transformation, we split data into training dataset (70%) and testing dataset (30%) for input vectors and output vectors.

2.3.2. Data normalization

Stock data are unstable and we need to normalize them to $[0,1]$ by the following formula:

$$X_{scaled} = \frac{(X - X_{min})}{(X_{max} - X_{min})} \cdot (max - \min) + min \tag{16}$$

where X is the data to be normalized, X_{min} is the minimum in scaled data sequence, X_{max} is the maximum in scaled data sequence, max is the maximum in original data sequence and min is the minimum in original data sequence.

2.3.3. Model training

We set cell size as 32, step of time as 7 and output size as 1 in the first layer of GRU. Then, three other GRU layers with the same cell size and a final dense layer are added. In this dense layer, it gives final calculation for output vectors:

$$output = activation(dot(input, kernel) + bias) \tag{17}$$

2.3.4. Model forecasting

The output of GRU model should be inversely transformed to the original predictive data because we conduct normalization when processing them. In this paper, first, we generate criterions including RMSE (Root Mean Square Error), MSE (Mean Square Error), MAE, explained variance regression score and R-square for test data of GRU model by formulas:

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(y_i - f(x_i)\right)^2} \tag{18}$$

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - f(x_i))^2 \tag{19}$$

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|f(x_i) - y_i| \tag{20}$$

$$Explained\ variance\ score = 1 - \frac{Var(y_i - f(x_i))}{Var(y_i)} \tag{21}$$

$$R^2 = 1 - \frac{\sum_{i=1}^{n}\left(y_i - f(x_i)\right)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2} \tag{22}$$

where n is the total number of data, y_i is the i^{th} real data and $f(x_i)$ is the i^{th} fitting data.

Table 8: GRU criterions

Criterions	Results
RMSE	3.323
MSE	11.040
MAE	2.586
Explained variance regression score	0.939
R-square	0.936

Table 8 illustrates these criterions for GRU model.

For intuitive understanding, we plot Figure 11 where truth value and prediction are illustrated.

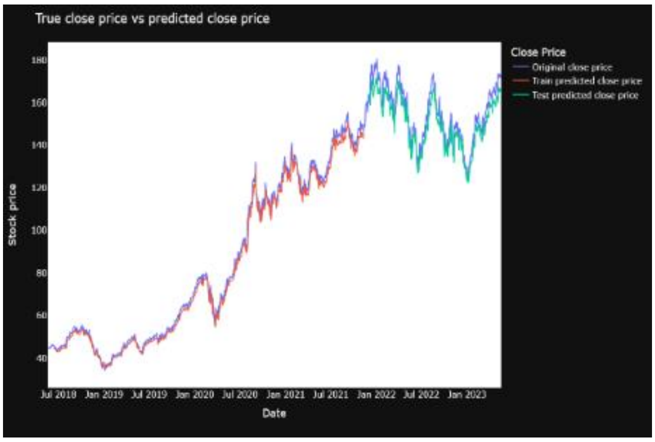

Figure 11: GRU fitting and prediction results

It is also possible for us to predict futural data which are not in testing dataset using GRU. In Figure 12, we use data in last 15 days to predict data in next one week, which are from May 12th to May 19th.

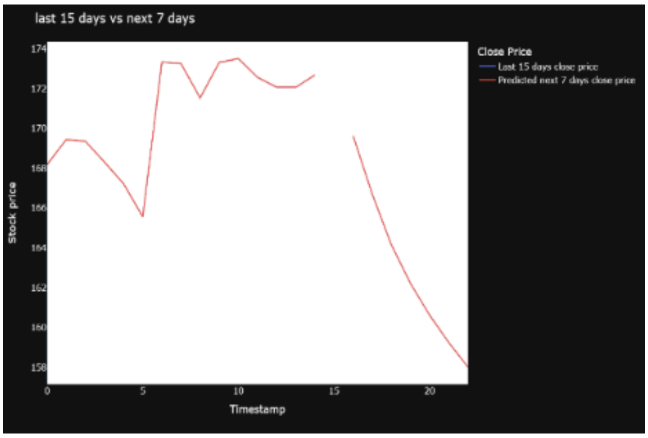

Figure 12: Next one week prediction

3. Conclusions

This study applies two different models to make stock prediction. We focus on modifying the best parameters in these models by statistical methods and trying to visualize prediction results. The following points are resulted from our study:

1) Prediction from ARIMA is more concentrated compared with GRU

2) GRU has better performance in long-term prediction compared with ARIMA

3) When data have greater volatility, GRU performs better than ARIMA

For hypothesis, we propose the following reasons for these conclusions:

1) ARIMA applies moving average and auto-regression, which cause results generated by ARIMA model to be close to mean value of a data sequence.

2) Historical data are stored in gates of GRU where there are activation functions. For data with greater volatility, predictions would be more radical.

For improvement, GRU cannot solves gradient vanishing and explosion problems utterly, although it is computationally cheaper. In future work, solving gradient vanishing and explosion problems completely would be a feasible direction.

References

[1] C.I. Noshi, A.I. Assem, J. J. Schubert. The role of big data analytics in exploration and production: a review of benefits and applications. SPE International Heavy Oil Conference and Exhibition, Society of Petroleum Engineers (2018).
[2] J. Brownlee. White Noise Time Series with Python (2017). Retrieved from White Noise Time Series with Python - MachineLearningMastery.com.
[3] PennState. Retrieved from https://online.stat.psu.edu/stat510/lesson/2/2.2 (2014).
[4] Yanrui Ning, Hossein Kazemi, Pejman Tahmasebi. A comparative machine learning study for time series oil production forecasting: ARIMA, LSTM, and Prophet. Computers & Geosciences, Volume 164, July 2022, 105126.
[5] L. Jain, L. Medsker. Recurrent Neural Networks: Design and Applications. International Series on Computational Intelligence. CRC-Press (1999).