Stock prediction model based on LSTM and ARIMA

Group 3

Zheng Wugeng Qu Weiting Chen Ziqi

p930026173@mail.uic.edu.cn p930031140@mail.uic.edu.cn p930026017@mail.uic.edu.cn

***Abstract* —** The prediction of stock prices has always been a hot research topic. However, the commonly used autoregressive integrated moving average (ARIMA) model still has its own advantages and disadvantages. The use of the long short-term memory (LSTM) network model for prediction also shows interesting possibilities. This article compares two models specifically through the analysis of the principles of the two models and the prediction results. The combination of time series and external factors may be a worthy research direction.

Keywords: Stock prediction; time series; deep learning, neural networks; auto regressive integrated moving average (ARIMA); long short-term memory (LSTM)

Introduction and Background

In the financial domain, stock prediction is a very crucial task. Based on this prediction, future transactions influence a lot. With the continuous application and development of artificial intelligence technology and big data technology, along with the further improvement of the financial market and the strong demand of the financial service industry, the stock market prediction has attracted extensive attention from the industry and academia.

This paper firstly establishes the ARIMA model, which is used to study the trend of the stock, and obtain the estimated value of the stock forecast, and check the fit and adaptability of the model. Combining the stock forecast model needs to deal with nonlinear problems and the stock has the characteristics of time series, so the work also uses the recurrent neural network to forecast the stock. While Recurrent Neural Networks (RNNs) allow persistence of information. However, the general RNN model has a weak ability to describe time series data with long memory. When the time series is too long, there are gradient dissipation and gradient explosion phenomena, which make RNN training very difficult. The Long Short-Term Memory (LSTM) model proposed is modified on the basis of the RNN structure, thus solving the problem that the RNN model cannot describe the long memory of time series.

To sum up, the ARIMA and LSTM model in deep learning can well describe the long memory of time series and get results for stock price prediction.

Related Works

The prediction of financial stocks has always been a research hotspot in the financial field. In terms of methods, it can be roughly divided into linear prediction models and nonlinear prediction models. Among them, linear prediction models mainly include the Autoregressive Integrated

Moving Average model (ARIMA), GARCH, EGARCH and IGARCH. As an early stock forecasting model, the above-mentioned linear forecasting model has played a pivotal role in promoting the forecasting development of the entire financial stock.

However, given the high noise and nonlinear characteristics of financial time series, it is still very difficult to accurately predict financial stock prices through linear prediction models. With the rapid development of computer technology and the advancement of deep learning research, neural networks in the field of machine learning are increasingly widely used in stock forecasting and have achieved more efficient and accurate forecasting results than linear forecasting models. The predicted accuracy using BP neural network and grey GARCH-BP model is significantly better than GARCH model.

Since the neural network prediction model has significant nonlinearity, we classify the neural network model as a nonlinear prediction model. Neural networks are divided into two categories: the first category is artificial neural network (ANN). However, due to the single structure of the ANN model, the generalization ability of the model is greatly weakened due to over-fitting, and the problem of local extreme values leads to a greatly weakened model prediction ability. And in the optimization process, it is easy to cause the gradient disappearance or gradient explosion problem due to too many neuron weights, and finally make the neural network model prediction invalid.

Deep neural network models (DNN), such as convolutional neural network (CNN), recurrent neural network (RNN) and long short-term memory neural network (LSTM), are the most efficient and cutting-edge forecasting models in the field of financial forecasting. Advantages: There is no restriction on the form of input variables, and the information that may be relevant to the forecasting problem can be used as the model input, considering the characteristics of the stock market being easily affected by various kinds of information. And it can effectively fit the nonlinear complex relationship between input variables, improve the degree of sample fitting, and at the same time, through the principle of neuron weight cycle, the number of neuron weights is greatly reduced, and the phenomenon of over-fitting is effectively prevented. Through the tanh activation function in DNN, the problems of gradient explosion and gradient disappearance in ANN can be significantly solved.

Methodology

ARIMA Model Architecture:

Autoregressive integrated moving average (ARIMA) models can be used to predict time series data based on the history data.

An ARIMA model is characterized by 3 terms: p, d, q where

p: the order of the Auto Regressive (AR) term

q: the order of the Moving Average (MA) term

d: the number of nonseasonal differences needed for stationarity

The ARIMA process is defined as:

$$X_t = c + \epsilon_t + \sum_{i=1}^{p} \phi_i X_{t-i} + \sum_{j=1}^{q} \theta_j \epsilon_{t-j}$$

$$\phi_i \; is \; AR \; parameters;$$

$$\theta_j \; is \; MA \; parameters$$

ARIMA model process:

a) Based on the scatter plot, autocorrelation function and partial autocorrelation function plots of the time series, the variance and trend of the series are identified with an ADF unit root test. Generally speaking, the time series of stock price data is not a stationary series.

b) Smoothing of non-stationary series. If the data series is non-stationary and has a certain increasing or decreasing trend, the data needs to be differenced.

c) According to the identification rules of the time series model, build the corresponding model If the bias correlation function and autocorrelation function of the smooth series are both trailing, the series is suitable for the ARIMA model.

d) Perform parameter estimation and test for statistical significance.

e) Perform a hypothesis test to diagnose whether the residual series is white noise.

f) Perform predictive analysis using the tested model.

LSTM Model Architecture:

A long short-term memory (LSTM) is a type of Recurrent Neural Network (RNN) specially designed to prevent the neural network output for a given input from either decaying or exploding as it cycles through the feedback loops. Memory of past input is critical for solving sequence learning tasks and Long short-term memory networks provide better performance than other Neural Networks.

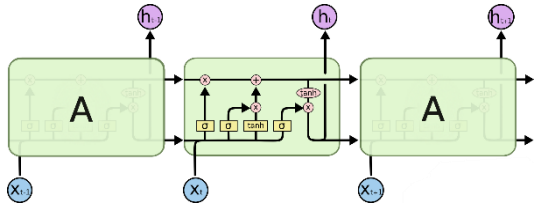


Figure 1 Each layer in LSTM

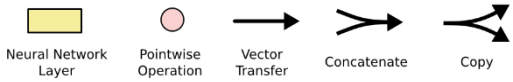
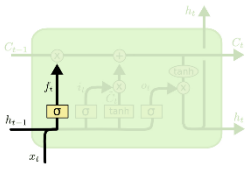


Figure 2 Notations in LSTM

LSTM Architecture consists of linear units and its neural network layers interact in a special way. In each layer, 4 different neural network layers make up on cell and in each cell, it has three inputs: C_{t-1}, h_{t-1} and x_t those are the states of last cell and the input x respectively. Then each cell output 2 values, C_t and h_t. They are both the state of this cell. When receiving C_{t-1}, h_{t-1} and x_t, it passes through three gates (forget gate, input gate and output gate).

a) For F gate, which is the forget gate. In this gate, we decide what information we're going to throw away from the cell state.

Determine how much the unit status of the previous moment C_{t-1} is retained to the current moment C_t.



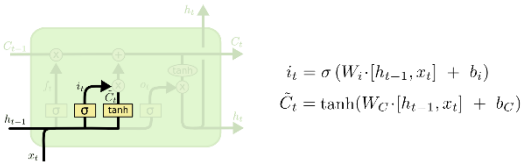
$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] + b_f\right)$$

Figure 3 Structure of Forget gate

When input x_t and h_{t-1}, the model merges the two values and process through the sigmoid function. The calculated value is then combined with another input C_{t-1}. Through this process, the cell can choose to forget unimportant information in the state. Selectively forgetting the information from the previous cell.

b) For input gate, which decides which values are updated. Next, a tanh layer and sigmoid layer create a vectors of new candidate values. In the next step,

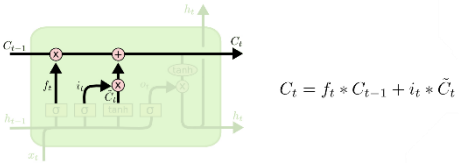
combine these two to create an update to the state.



$$i_t = \sigma\left(W_i \cdot [h_{t-1}, x_t] + b_i\right)$$
$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C)$$

Figure 4 Structure of Input gate

Then, drop the information which need to forget and add the new information. By this way, update the state of this cell

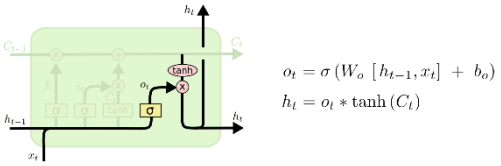


$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t$$

Figure 5 After update the cell

Inputs are C_{t-1}, h_{t-1} and x_t and layers are sigmoid, tanh. For now, update the state of this cell.

c) For output gate, need to decide what we're going to output and it depend on the state of this cell. First, we process a sigmoid function, then we put the state through tanh and combine together. Finally, we output the value.



$$o_t = \sigma\left(W_o\ [h_{t-1}, x_t] + b_o\right)$$
$$h_t = o_t * \tanh\left(C_t\right)$$

Figure 6 Structure of Input gate

Our LSTM model structure:

For this model, which contains 5 LSTM layers, 5 dropout layers and 2 dense layers. For LSTM layers, the first layer is to deal with input, the rest is in order to extract the information. For dropout layers, to prevent overfitting, we add dropout layers between each 2 of other layers. At last, add dense layers to output values.

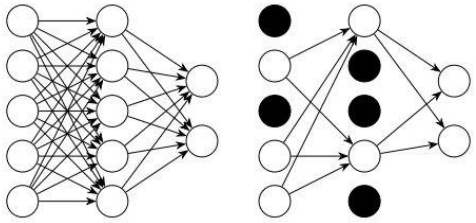


Figure 7 Dropout Layers

Parameters:

In the model, set parameters and hyper parameters.

Parameter	Explain
start_time	The start date of the training set
window	Window time for each row of data
timestamp	Training set Test Set Division Time (last n days)
epochs	Training set Number of training sessions
batch_size	the number of data samples captured in a training

Data Description and Preprocessing

Data Description:

When searching for stock data and the news from the National Association of Securities Dealers Automated Quotations (NASDAQ), we noticed that the current value of NVIDIA's stock is similar to the value of the stock around 2017-2018, when the stock went up due to the Cryptocurrency mining by GPUs. And in recent years, cryptocurrency becomes popular as well, which makes us pay more focused to NVIDIA's stock price.

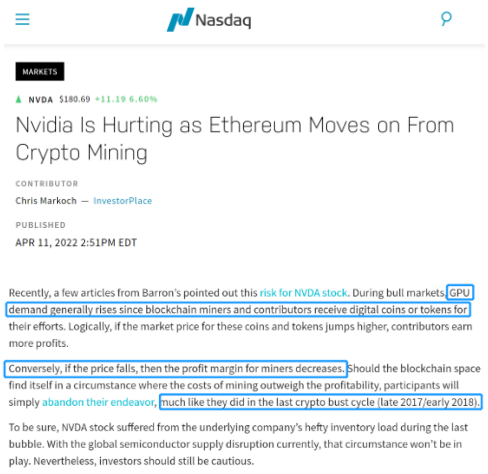


Figure 8 NVIDIA's news on Nasdaq

Hence, we select Nvidia stock data from (https://www.nasdaq.com/) NASDAQ official website, which is the world's largest stock market.

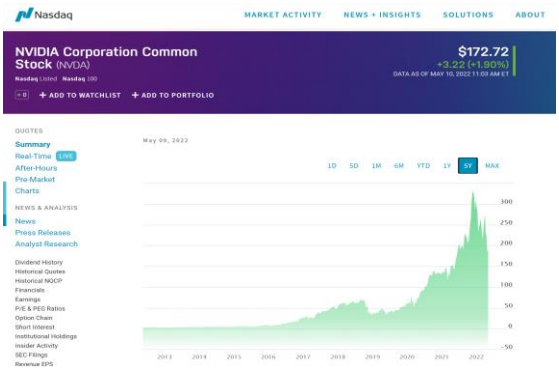


Figure 9 NVIDIA's stock data on Nasdaq

Select Nvidia's historical quotes. Download and process stock data from 1999-01-22 to 2021-11-12.

Data preprocessing:

After imported dataset, we got data that was started in 1999-01-22.

Date	Open	High	Low	Close	Volume
1999-01-22	0.401941	0.448595	0.356484	0.376820	271468800.0
1999-01-25	0.406726	0.421081	0.376820	0.416296	51048000.0
1999-01-26	0.421081	0.429455	0.378016	0.383998	34320000.0
1999-01-27	0.385194	0.394764	0.363661	0.382801	24436800.0
1999-01-28	0.382801	0.385194	0.379212	0.381605	22752000.0
...	...	...	...	...	...
2021-11-08	301.489990	311.000000	299.070007	308.040009	50310100.0
2021-11-09	322.820007	323.100006	299.640015	306.570007	64674600.0
2021-11-10	293.559998	308.500000	287.779999	294.589996	63620600.0
2021-11-11	304.679993	305.899994	297.769989	303.899994	33217200.0
2021-11-12	300.100006	306.799988	296.299988	303.899994	41215100.0

Figure 10 original dataset

Hence, select the close price to do the data training since the close price can represent a more truly price of a company.

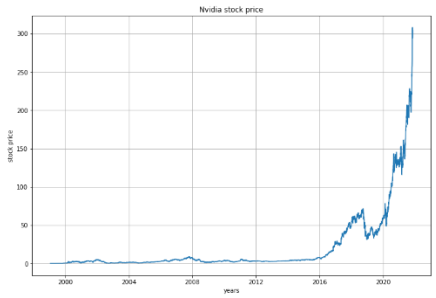


Figure 11 Plot for NVIDIA price

Then, we do the min-max normalization for data. The function of min-max normalization is

$$x' = \frac{x - min}{max - min}$$

So that got regularized data

```
Date
1999-01-22    0.000206
1999-01-25    0.000334
1999-01-26    0.000229
1999-01-27    0.000225
1999-01-28    0.000222
                ...
2021-11-08    1.000000
2021-11-09    0.995223
2021-11-10    0.956292
2021-11-11    0.986546
2021-11-12    0.986546
Name: value, Length: 5743, dtype: float64
```

Figure 12 after min-max normalization

Found that stock price of NVIDIA before 2015 can extract little information. So that we selected data that is from 2015-01-01.

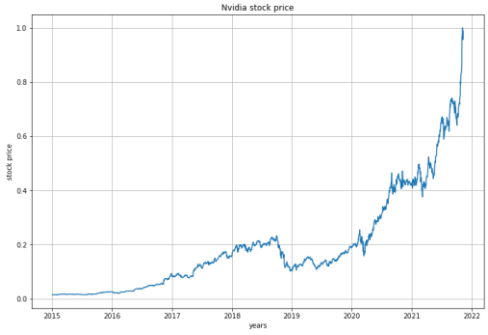


Figure 13 NVIDIA price since 2015-01-01

We set a window of 5 days, which means that for each piece of data, we train based on the stock price of the previous 5 days.

	0	1	2	3	4
0	[0.014709955…	[0.01444429…	[0.013975492…	[0.01393642…	[0.01449…
1	[0.01444429…	[0.01397549…	[0.013936425…	[0.01449899…	[0.01456…
2	[0.013975492…	[0.01393642…	[0.014498995…	[0.01456149…	[0.01436…
3	[0.013936425…	[0.01449899…	[0.014561499…	[0.01436616…	[0.01434…
4	[0.01449899…	[0.01456149…	[0.014366163…	[0.01434272…	[0.01440…
5	[0.01456149…	[0.01436616…	[0.014342723…	[0.01440522…	[0.01429…
6	[0.01436616…	[0.01434272…	[0.014405225…	[0.01429584…	[0.01457…
7	[0.01434272…	[0.01440522…	[0.014295843…	[0.01457712…	[0.01462…
8	[0.01440522…	[0.01429584…	[0.014577126…	[0.01462400…	[0.01484…
9	[0.01429584…	[0.01457712…	[0.014624006…	[0.01484278…	[0.01511…
10	[0.01457712…	[0.01462400…	[0.014842784…	[0.01511625…	[0.01516…
11	[0.01462400…	[0.01484278…	[0.015116251…	[0.01516313…	[0.01509…
12	[0.01484278…	[0.01511625…	[0.015163136…	[0.01509281…	[0.01431…
13	[0.01511625…	[0.01516313…	[0.015092816…	[0.01431928…	[0.01406…
14	[0.01516313…	[0.01509281…	[0.014319283…	[0.01406925…	[0.01443…
15	[0.01509281…	[0.01431928…	[0.014069257…	[0.01443648…	[0.01398…

Figure 14 Training set

After data preprocessing, we got the 1615 training data and 100 test data.

```
train_x (1624, 5, 1)
train_y (1624,)
test_x  (100, 5, 1)
test_y  (100,)
```

Figure 15 Quantity of data

Model Evaluation and Prediction

For ARIMA:

Model construction

Autoregressive process assumes that the observation at previous several time steps are useful to predict the next step. It depends on autocorrelation. We plotted the observation at the previous time step (t) with the observation at the next time step (t+1) as a scatter plot in the following figure, which clearly shows a relationship or some correlation.

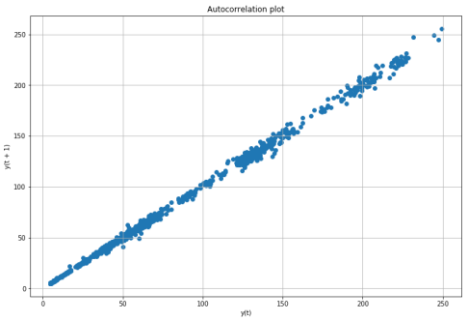


Figure 16 autocorrelation plot

From the data plot, the price has the trend of increasing, so this time series data is not stationary. We take a series of one order difference to make sure the data become stationary.

Do Dickey-Fuller Test on the differenced data and it showed that in 90% confidence level. The data after first order difference is stationary, and average change, show in figure below, is not large, so the data can be considered stable.

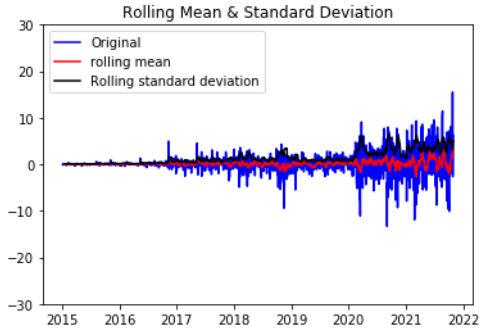


Figure 17 rolling mean & standard deviation

In ARIMA process, autocorrelation function and partial autocorrelation function could be used to determine parameter p and q of the model.

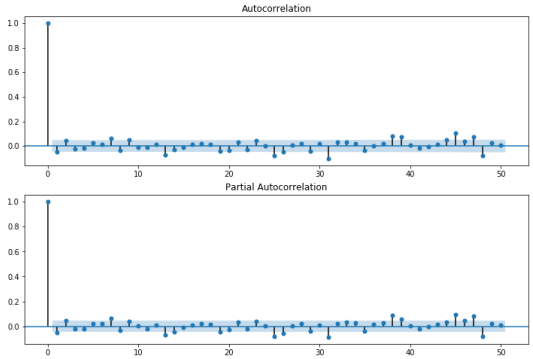


Figure 18 autocorrelation & partial autocorrelation

Based on Bayesian Information Criterion (BIC), use the tool in statsmodels package, BIC min order of (p, q) is (4, 2). We choose to use the Mean Square Error (MSE), Root Mean Square Error (RMSE), Mean Absolute Error (MAE) to evaluate our models.

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

$$RMSE(X,h) = \sqrt{\frac{1}{m}\sum_{i=1}^{m}(h(x_i) - y_i)^2}$$

$$\text{MAE}\ (X,h) = \frac{1}{m}\sum_{i=1}^{m}|h(x_i) - y_i|$$

The model performance on test set shown as follow:

ARIMA（4, 1 ,2 ）	
MSE	22.388
MAE	3.647
RMSE	4.731

Table 1 model evaluation

Residual white noise test:

A white noise test is performed on the residual series of the ARIMA model. If the residuals are white noise series, it means that there is no more information that can be mined. The results of visualizing the residuals of the model are as follows.

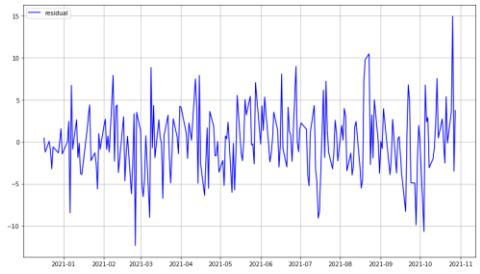


Figure 19 model residual

Here we use the ljung-box test. all p-values are greater than the significance level (e.g. 0.05) and the original hypothesis (that the series is a white noise series) cannot be rejected and the series can be considered as a white noise series. Therefore, there is no need to continue adding models.

```
[0.93754653 0.99677313 0.99954248 0.99996496 0.99999801 0.90740554
 0.93463142 0.96591488 0.98272858 0.9895207  0.99388265 0.98942624
 0.98311467 0.97617098 0.97419876 0.95521277 0.96300452 0.97545711
 0.83655811 0.81633406 0.84914263 0.88443476 0.80500401 0.837157
 0.53784668 0.45231719 0.4135847  0.44910952 0.48283134 0.50080892
 0.23252594 0.26557014 0.29764269 0.30195056 0.33892936 0.378934
 0.42368698 0.38006644 0.39592104 0.43699012]
```

Table 2 ljung-box test, p-value

ARIMA Evaluation and Prediction:

Finally, we use ARIMA (4, 1, 2) model to predict the stock price in last 90 days. The predicted price result as follow:

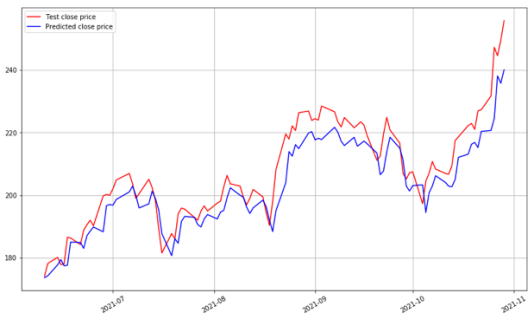


Figure 20 ARIMA (4,1, 2) price prediction

ARIMA（4, 1 ,2 ）	
MSE	27.832
MAE	6.654
RMSE	9.471

Table 3 ARIMA model evaluation on test set

For LSTM:

We train the model with epoch is 100, batch size is 512, optimizer is “Adam” and loss function is MSE.

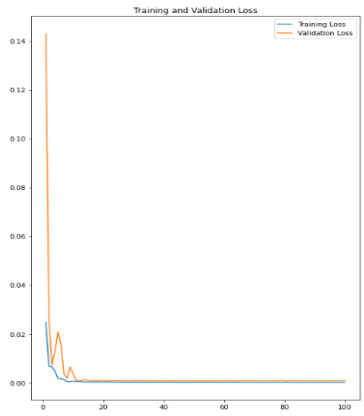


Figure 21 Loss during training

After training, we restore the values to that before min-max normalization. Then, compared with training set and test set and got the RMSE function.

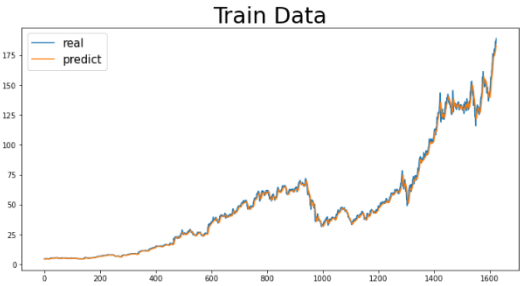


Figure 22 Compared with training data

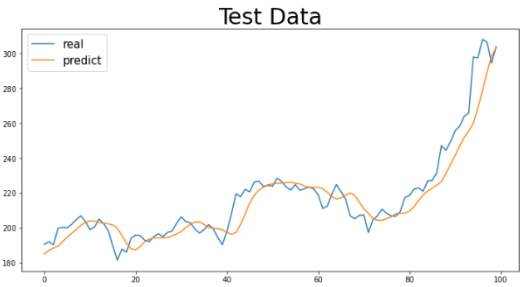


Figure 23 Loss during predicting

We choose to use the MSE, RMSE, MAE to evaluate our models. Here is our LSTM model results for different windows and eopch.

Window	Epoch	MSE	RMSE	MAE
5	50	92.791	9.633	7.48
5	100	84.992	9.219	5.519
10	50	114.404	10.696	7.663
10	100	133.182	11.540	8.206
20	50	115.178	10.732	7.717
20	100	117.007	10.817	7.96

Table 4 LSTM model evaluation on test set

Conclusion

Through the analysis of the establishment process and results of these two models, conclusions can be made. The stock prediction of the LSTM model is better than that of the ARIMA model. And ARIMA can further improve the accuracy of the ARIMA model if the residual series of the white noise sequence exists.

The disadvantage is that, as we all know, the fluctuation of stock prices is not only related to changes in time, but also related to economic factors, socio-political factors, and the listing of other stocks. These two models are essentially deduced by using possible relationships in the time series without considering other external factors. This is also the direction in which future research can be further in-depth. Of course, the further development and use of LSTM model in stock price prediction is also a subject of research value.

Group Work on GitHub Website:

https://github.com/ZhengWugeng/Financial_Computing_Group

References

[1] Long Short-Term Memory (LSTM). (2020, February 21). NVIDIA Developer. https://developer.nvidia.com/discover/lstm#:%7E:text=A%20Long%20shortterm%20memory%20%28LSTM%29%20is%20a%20type,better%20at%20pattern%20recognition%20than%20other%20neural%20networks.

[2] Olah, C. (27–08-15). U*nderstanding LSTM Networks* -- colah's blog. Understanding LSTM Networks. https://colah.github.io/posts/2015-08-Understanding-LSTMs/

[3] Team, K. (2022, February 3). Keras documentation: LSTM layer. LSTM-Keras. https://keras.io/api/layers/recurrent_layers/lstm/

[4] Mehtab, S., Sen, J., & Dutta, A. (2020, October). Stock price prediction using machine learning and LSTM-based deep learning models. In *Symposium on Machine Learning and Metaheuristics Algorithms, and Applications* (pp. 88-106). Springer, Singapore.

[5] Ariyo, A. A., Adewumi, A. O., & Ayo, C. K. (2014, March). Stock price prediction using the ARIMA model. In *2014 UKSim-AMSS 16th International Conference on Computer Modelling and Simulation* (pp. 106-112). IEEE.

[6] Munish Kumar, Surbhi Gupta, Krishan Kumar, and Monika Sachdeva. 2020. *SPREADING OF COVID-19 IN INDIA, ITALY, JAPAN, SPAIN, UK, US: A Prediction Using ARIMA and LSTM Model.* Digit. Gov.: Res. Pract. 1, 4, Article 24 (October 2020), 9 pages. https://doi.org/10.1145/3411760

[7] Newbold, P. (1983). *ARIMA Model Building and the Time Series Analysis Approach to Forecasting. Journal of Forecasting,* 2, 23-35. https://link.gale.com/apps/doc/A2582223/AONE?u=uicbnu&sid=bookmark-AONE&xid=12750cc3