2017 4th International Conference on Social Science (ICSS 2017)
ISBN: 978-1-60595-525-4

Stock Price Prediction Based on ARIMA-RNN Combined Model

Shui-Ling YU and Zhe Li[a,*]

School of Science, Changchun University of Science and Technology, Changchun 130022, China

[a]lizhe@amss.ac.cn

*Corresponding author

Keywords: Stock price forecasting, Moving-average filter, RNN model, Hybrid ARIMA–ANN model.

Abstract: In this paper, we proposed a new hybrid ARIMA–RNN model to forecast stock price, the model based on moving average filter. This model can not only overcome the volatility problem of a single model, but also avoid the overfitting problem of neural network. We forecast stock price using ARIMA, RNN and ARIMA-RNN respectively, and we compare the value of MAE, MSE and MAPE of each model. We conclude that the hybrid ARIMA–RNN model has the best forecasting result.

Introduction

The change in stock price is a measure of the stability of the stock market, at the same time it is also the most concerned issue by stock investors. The accurate prediction of trends in stock price can not only help investors avoid and control risks in time, but also have important guiding significance in reducing investment losses.

The traditional time series prediction methods mainly contain exponential smoothing, ARIMA family models, GARCH family models et al. [1]. The ARIMA family models and GARCH family models are most widely used in the prediction of stock price. Contreras uses the ARIMA model to make short-term forecast of electricity price [2]; Morley et al. use GARCH model to predict stock volatility [3]; While Sohn and Lim apply the AR-GARCH model to predict time series hierarchically [4]. However, the stock price sequence is a complex nonlinear dynamic system, thus the accuracy of prediction of traditional methods can’t meet people's needs. In recent years, with the development of artificial intelligence, neural network (ANN) is widely used in stock price forecasting owing to its excellent quality, such as strong non-linear mapping ability, self-learning, self-adaptation and fault tolerance. The most widely used neural networks are BP and RBF which belong to static feedforward neural network (FNN) [5] [6]. Researchers proposed kinds of combination of models by combining traditional methods with neural network. Zhang and Bijari et al., proposed ARIMA-ANN combination model based on different data decomposition techniques [7] [8]. On the basis above, Narendra and Eswara decomposed the time series into linear and nonlinear parts, and established the ARIMA-ANN combined model based on the moving average filter method [9].

In financial time series, the data often show a strong correlation, and historical information also has very high value in application. Due to the traditional neural network represented by FNN network can only use the information of the current time, Elman proposed a recurrent feedback neural network (RNN) [10]. RNN is directly applied mainly in speech recognition, text classification, machine translation, visualization and so on. At the same time, with the further development of RNN network, the artificial intelligence trained by RNN and LSTM are not only applied to financial sector, but also to the UAV by Google, IBM, Jingdong and other enterprises. In addition, Bijari, Zhang, Tan et al. applied a variety of variants of RNN to predict time series [11] [12] [13].

In this paper, the data is decomposed into linear and nonlinear parts by using moving-average filtering method. The linear part is predicted by ARIMA model and the nonlinear part is predicted by RNN network. Thus the ARIMA-RNN combination model is established to predict the stock price, and by which we can get better prediction effect.

The ARIMA-RNN combined model based on MA filter

MA Filter

The main idea of moving average filter is if a smooth time series obey normal distribution, then the series is strictly stable. Usually we use the value of kurtosis to test whether a series $\{y\}$ obeys normal distribution:

$$\text{kurtosis} = \frac{E\{(y-E\{y\})^4\}}{(E\{(y-E\{y\})^2\})^2} \tag{1}$$

If the value of kurtosis is 3, then the series obeys the normal distribution; if the value of kurtosis is too large or too small, then the series is ether the leptokurtic or platykurtic.

The working principle of MA filter is first of all we apply MA filter to separate the component of low-volatility from the original time series$\{y_t\}$, which is the linear part l_t.

$$l_t = \frac{1}{m}\sum_{i=t-m+1}^{t} y_i \tag{2}$$

Then, use g_t to represent the component of high-volatility, which is the non-linear part:

$$g_t = y_t - l_t \tag{3}$$

The process of decomposition by MA filtering is shown in figure 1 as below:

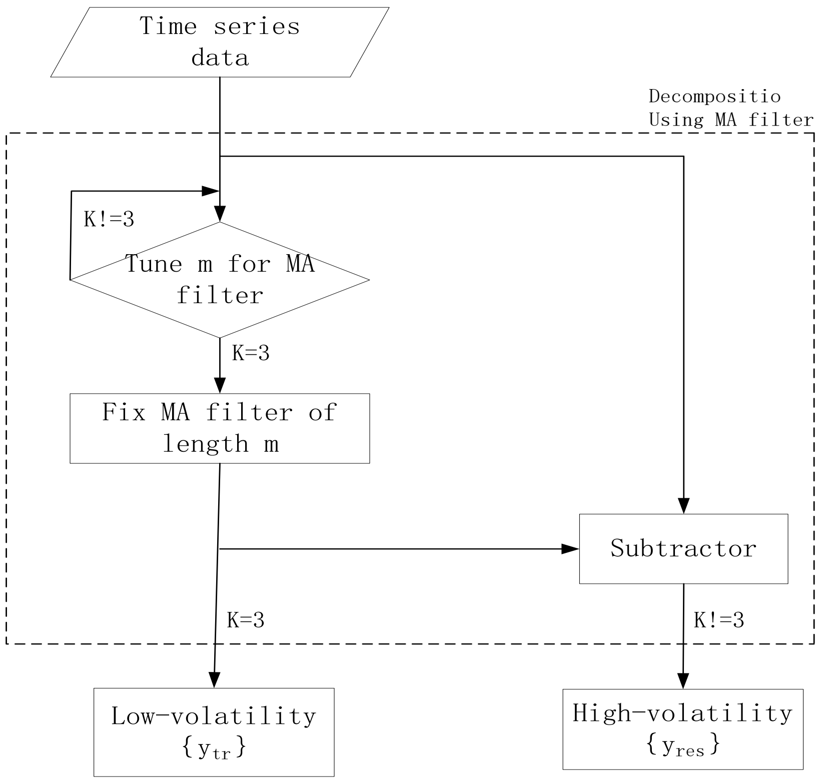
Time series data
Decompositio Using MA filter
K!=3
Tune m for MA filter
K=3
Fix MA filter of length m
Subtractor
K=3
K!=3
Low-volatility {y_{tr}}
High-volatility {y_{res}}

Figure 1. Data decomposition process using MA filter.

The ARIMA-RNN Combined Model

We can get two parts by decomposing time series y_t with the MA filtering method, the linear part l_t and the non-linear part g_t. The ARIMA-RNN combined model uses the ARIMA model to obtain the predicted value $\widehat{l_t}$, at the same time we can get the predicted value $\widehat{g_t}$ by applying RNN model. The final predicted value $\hat{y}_t$ is shown as follows:

$$\widehat{y_t} = \widehat{l_t} + \widehat{g_t} \tag{4}$$

The ARIMA model

The low-volatility component can be expressed as an ARIMA model of the series values $l_{t-1}, l_{t-2}, \dots, l_{t-p}$ and the random error series $n_t, n_{t-1}, n_{t-2}, \dots, n_{t-p}$.

$$l_t = f(l_{t-1}, l_{t-2}, \cdots, l_{t-p}, n_t, n_{t-1}, \cdots, n_{t-q}). \tag{5}$$

Where, f is a linear function of ARIMA, which means:

$$l_t = a_1 l_{t-1} + a_2 l_{t-2} + \cdots + a_p l_{t-p} + n_t + b_1 n_{t-1} + b_2 n_{t-2} + \cdots + b_q n_{t-q}. \tag{6}$$

The RNN Model

The high-volatility component g_t can be expressed as a non-linear function of $g_{t-1}, g_{t-2}, \dots, g_{t-N}$, which is as below:

$$g_t = h(g_{t-1}, g_{t-2}, \cdots, g_{t-N}) + \varepsilon_t . \tag{7}$$

Where, h is the model of RNN neural network.

The recurrent neural network (RNN) is a kind of node-oriented connection ring and time-related artificial neural network. It can remember the information of the previous time and apply it to the output calculation of the current time. The structure diagram is as follows:

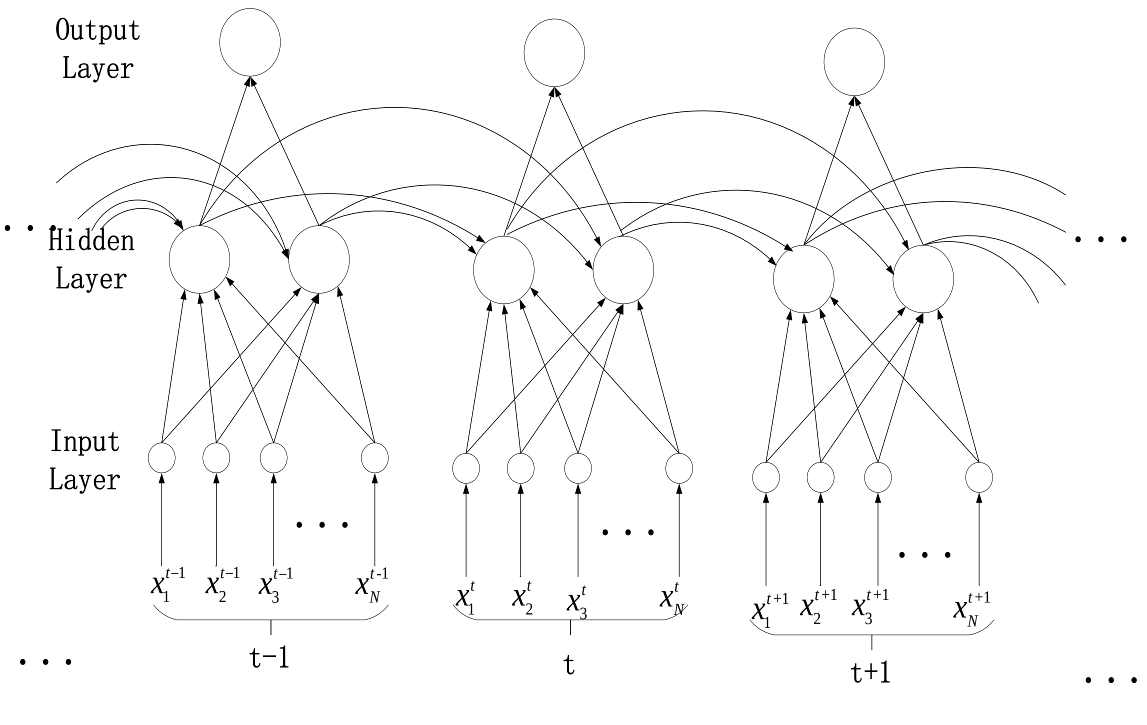
Output Layer
Hidden Layer
Input Layer
x_1^{t-1} x_2^{t-1} x_3^{t-1} x_N^{t-1}
x_1^{t} x_2^{t} x_3^{t} x_N^{t}
x_1^{t+1} x_2^{t+1} x_3^{t+1} x_N^{t+1}
t-1
t
t+1

Figure 2. RNN's structure.

The input layer is represented by v, and N is the numbers of neurons; the hidden layer is represented by h, and H is the numbers of neurons; the output layer is represented by o, and M is the numbers of neurons. Let the w_{hv} denotes the weight of the vth neurons in the input layer to the hth neurons in the hidden layer, and the $w_{h'h}$ denotes the weight of the hth neurons in the hidden layer to the h'th neurons in the hidden layer, the w_{oh} denotes the weight of the hth neurons to the oth neurons in the output layer. a is the summarized values. For example, a_h^t represents the weighted input value in the hidden layer at time t. b is the value calculated by the activation function. For example, b_h^t represents the output value calculated by the activation function in the hidden layer at time t. $\mathcal{L}$ denotes the loss function just as the loss function in the BP network. x_i^t represents the input value of the ith neurons in the input layer at time t.

1) Forward calculation

The input in the hidden layer at time t is as follows:

$$a_h^t = \sum_{i=1}^{N} \omega_{hv} x_i^t + \sum_{h'=1}^{H} w_{h'h} b_{h'}^{t-1} . \tag{8}$$

The output in the hidden layer at time t is as follows:

$$b_h^t = \theta_h(a_h^t). \tag{9}$$

The input in the output layer at time t is as follows:

$$a_o^t = \sum_{h=1}^{H} \omega_{0h} b_h^t . \tag{10}$$

The output in the output layer at time t is as follows:

$$b_o^t = \theta_h(a_o^t) . \tag{11}$$

2) The process of backward updating parameters based on the BPTT algorithm

$$\frac{\partial L}{\partial \omega_{hk}} = \sum_{t=1}^{T} \frac{\partial L}{\partial a_o^t} \frac{\partial a_o^t}{\partial \omega_{oh}} . \tag{12}$$

$$\delta_o^t \overset{\text{def}}{=} \frac{\partial L}{\partial a_o^t} . \tag{13}$$

The δ_o^t is the revised value of weight w_{oh}:

$$\delta_0^t = \frac{\partial L}{\partial \omega_{hk}} = \sum_{t=1}^{T} \frac{\partial L}{\partial a_o^t} \frac{\partial a_o^t}{\partial \omega_{oh}} = \sum_t^T \delta_j^t b_i^t . \tag{14}$$

The δ_h^t is the revised value of weight in the hidden layer:

$$\delta_h^t = \theta'(a_h^t)\left(\sum_{k=1}^{K} \delta_o^t \omega_{ho} + \sum_{h'=1}^{H} \delta_{h'}^{t+1} \omega_{h'h}\right) . \tag{15}$$

Experimental Results and Analysis

In this paper, the day closing prices of the Standard & Poor’s 500 index, the NASDAQ index, the Nikkei index, the Hong Kong Hang Seng Index and the Shanghai Composite Index are the empirical object. We use the analysis software python and R to make one step forward sliding forecast, and the original data are obtained from the Yahoo Finance website.

This article uses the python software to build the RNN network based on the karo library in theano. The closing price of the fifth day is predicted by the previous four days’ closing price. The numbers of neurons in the input layer, hidden layer and output layer of the RNN network are 4 4 and 1 respectively. The RNN network divides the data into training sets and test sets at a ratio of 3: 1.

The results of ARIMA-RNN combination models are as follows. The ARIMA(3,1,2)–RNN model was established while predicting the day closing price of the Standard & Poor’s 500 index. The ARIMA(7,1,2)–RNN model was established while predicting the day closing price of the NASDAQ index. The ARIMA(1,1,2)–RNN model was established while predicting the day closing price of the Nikkei index. The ARIMA(3,1,3) –RNN model was established while predicting the day closing price of the Hong Kong Hang Seng Index. The ARIMA(5,1,3)–RNN model was established while predicting the day closing price of the Shanghai Composite Index.

After a few steps tests, such as smoothness analysis, correlagram specification order and model test, the final results of single ARIMA models are as follows. The ARIMA(1,1,1) model was established while predicting the day closing price of the Standard & Poor's 500 index. The ARIMA(3,1,3) model was established while predicting the day closing price of the NASDAQ index. The ARIMA(2,1,2) model was established while predicting the day closing price of the Nikkei index. The ARIMA(4,1,2) model was established while predicting the day closing price of the Hong Kong Hang Seng Index. The ARIMA(6,1,6) model was established while predicting the day closing price of the Shanghai Composite Index.

Predicted Results

The predicted results are shown in figure 3 to figure 7. Through comparison we found that the ARIMA-RNN hybrid model established in this paper always has excellent prediction efficiency whether to forecast the stock price or the trend of stock price.

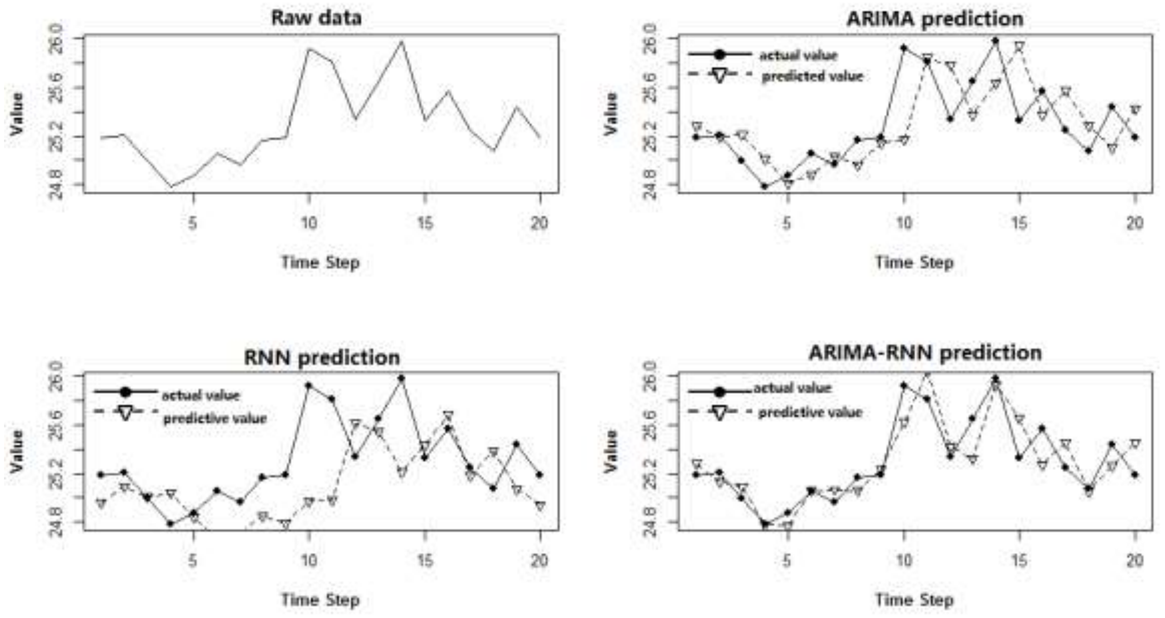

Figure 3. Prediction results of Standard & Poor's 500 index.

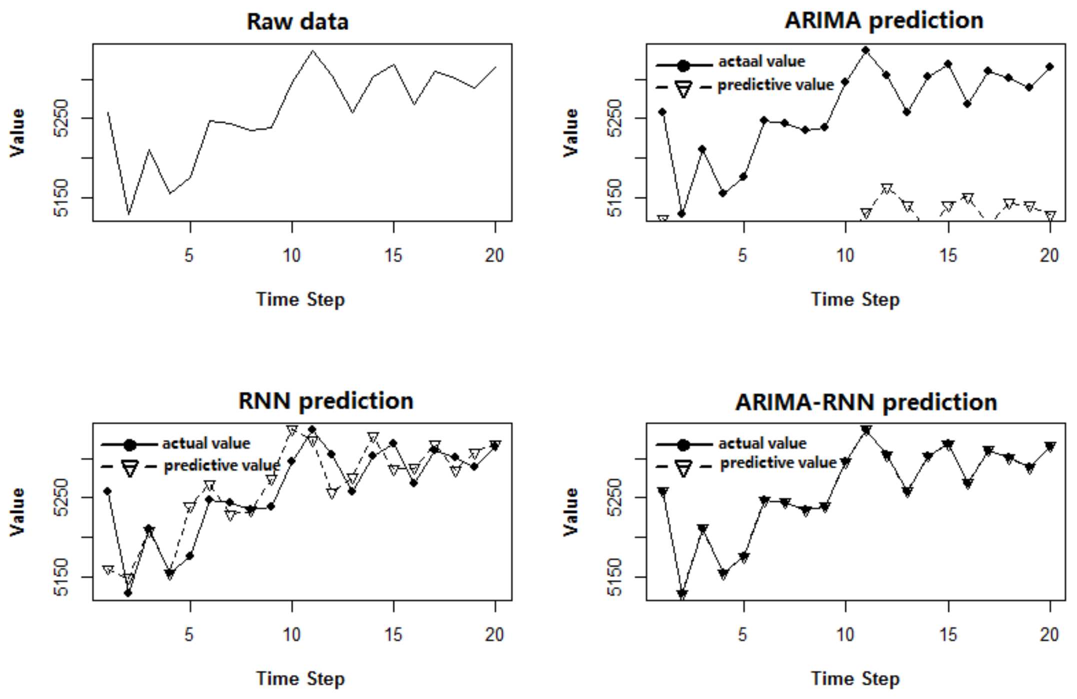

Figure 4. Prediction results of NASDAQ index.

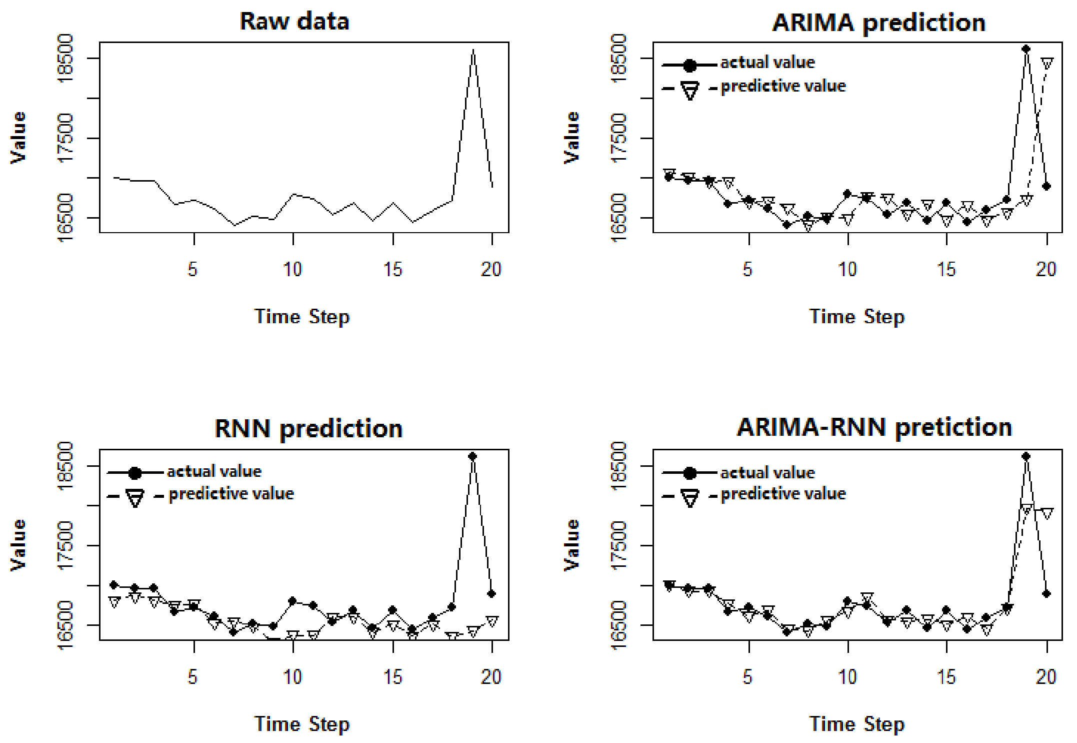

Figure 5. Prediction results of Nikkei index.

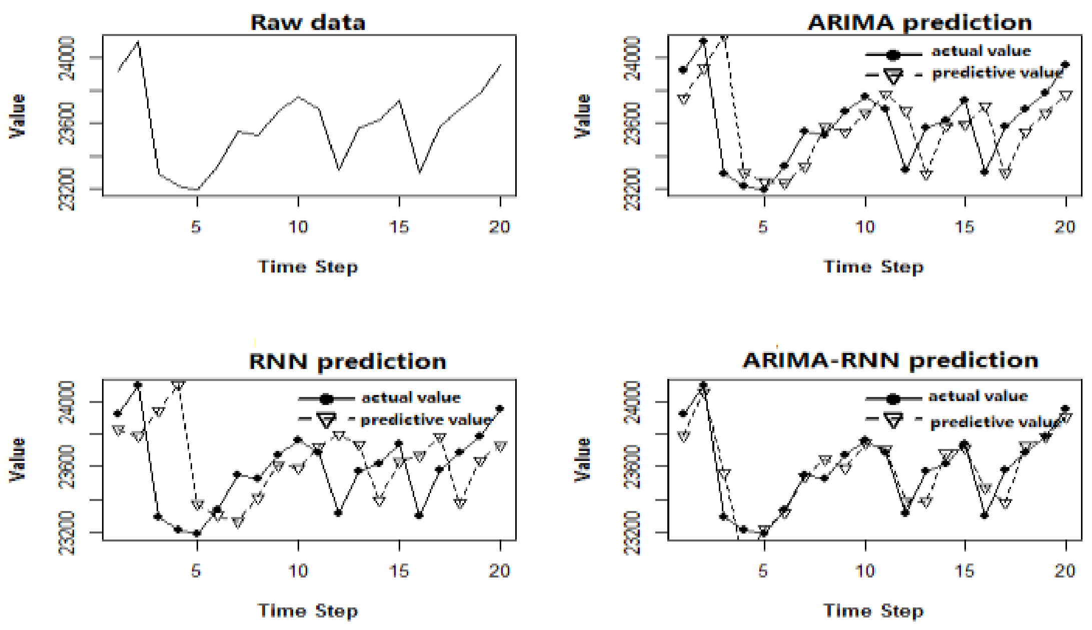

Figure 6. Prediction results of Hong Kong Hang Seng Index.

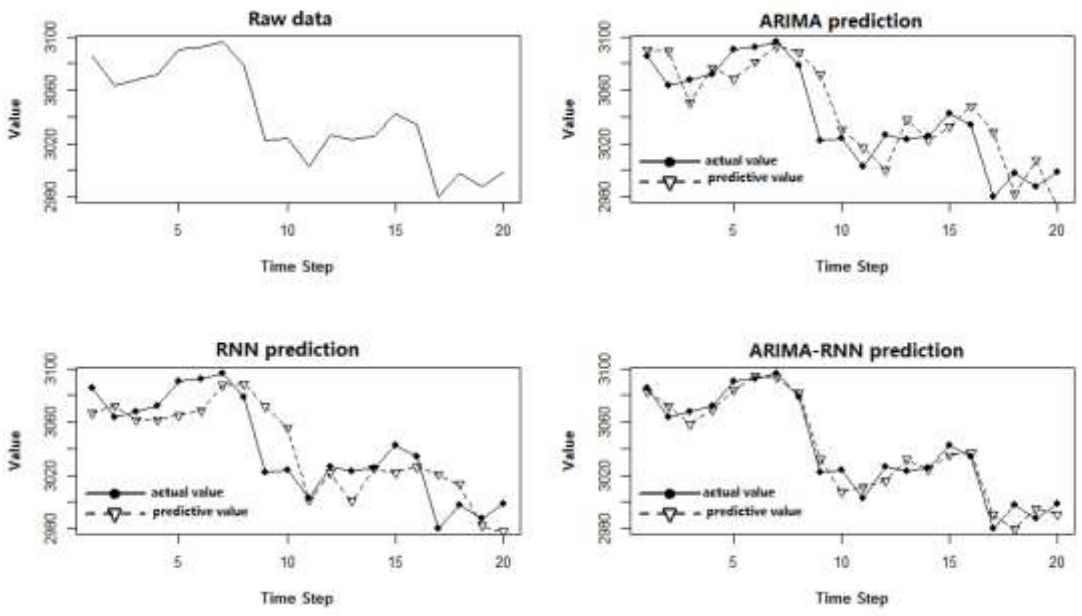

Figure 7. Prediction results of Shanghai Composite Index.

Prediction Error Comparison

Calculating the mean square error (MSE), mean absolute error (MAE), mean absolute percentage error (MAPE) of each model to compare their prediction accuracy.

Table 1. SP500 Prediction effect.

	ARIMA model	RNN model	ARIMA-RNN model
MAE	0.258574684	0.221050016	0.190790117
MSE	0.70138274	0.062692264	0.045997575
MAPE	0.008265211	0.007271456	0.002965063

Table 2. IXIC's Prediction effect.

	ARIMA model	RNN model	ARIMA-RNN model
MAE	156.1490757	38.53328218	18.51557172
MSE	27193.89002	2469.27135	616.324433
MAPE	0.029592174	0.007354554	0.003537766

Table 3. N225's Prediction effect.

	ARIMA model	RNN model	ARIMA-RNN model
MAE	298.9884572	174.4309752	160.7796026
MSE	324269.3434	47010.68172	82102.50201
MAPE	0.017303278	0.005588	0.009412003

Table 4. HSI's Prediction effect.

	ARIMA model	RNN model	ARIMA-RNN model
MAE	252.3557076	197.1381271	86.94371906
MSE	107337.5901	70516.02258	13650.66015
MAPE	0.01074269	0.008381815	0.00369975

Table 5. SSEC's Prediction effect.

	ARIMA model	RNN model	ARIMA-RNN model
MAE	24.74324407	21.98168158	8.87897304
MSE	771.5816093	733.0305489	111.1556798
MAPE	0.010217696	0.008744531	0.007521832

The results of Table 1 to Table 5 show that ARIMA-RNN hybrid model has the best prediction efficiency and accuracy because of the excellent characteristics of the mixed model. First the hybrid model avoids the volatility of the single model and the loss caused by improper operation in forecasting process. At the same time, hybrid model avoids the situation that predictive results are too large or too small. Most importantly hybrid model overcame the over-fitting problem of common neural networks, which made the overall predictive performance more stable.

Conclusion

In this paper we studied the stock daily closing price forecast and investigated the common model of forecasting stock price. At the same time the paper emphatically introduced fundamentals of RNN model and ARIMA-RNN hybrid model which is based on MA filter method. Through positivist research, major conclusions are as follows: combined with traditional forecasting model, RNN model will have a better prediction of stock price, for its memory of time. However, both ARIMA model and RNN model are more suitable for short-term predictions, so this paper only studied one-step forward prediction. And multi-step forward prediction will be further studied when we study the LSTM networks that have long-term memory.

References

[1] P.H. Franses, H. Ghijsels. Additive outliers, GARCH and forecasting volatility. International Journal of Forecasting, 1999, 15(1):1-9.

[2] J. Contreras. ARIMA models to predict next-day electricity prices Power Systems [J]. IEEE Transactions, 2003, (3).

[3] W. Liu, B. Morley, Volatility forecasting in the Hang Seng Index using the GARCH approach, Asia-Pacific Financial Markets 16 (1) (2009) 51-63.

[4] S.Y. Sohn, M. Lim, Hierarchical forecasting based on AR-GARCH model in a coherent structure, Eur. J. Operat. Res. 176 (2) (2007), 1033-1040.

[5] W. Lu, Sundararajan N., Saratchandran P. Performance evaluation of a sequential minimal radial basis function neural network learning algorithm [J]. IEEE Transactions on Networks, 1998, 9(2): 308-318.

[6] M. Bianchini, P. Frasconi, M. Gori. Learning Without local minima in radial basis function networks [J]. IEEE Transactions on Neural Networks, 1995, 6(3): 749-756.

[7] G. Zhang, Time series forecasting using a hybrid ARIMA and neural network model, Neuro computing, 2003, 50 (0):159-175.

[8] M. Khashei, M. Bijari, A novel hybridization of artificial neural networks and ARIMA models for time series forecasting, Appl. Soft Comput, 2011, 11 (2):2664-2675.

[9] C. Narendra Babu, B. Eswara Reddy, A moving-average filter based hybrid ARIMA–ANN model for forecasting time series data[J]. Applied Soft Computing, 2014, 27-38.

[10]J L. Elman. Finding structure in time [J]. Cognitive science, 1990, 14(2): 179-211.

[11] M. K. Bijari. "An artificial neural net-work (p,d,q) model for time series forecasting", Expert Systems with Applications, Volume 37, Issue 1, January 2010 , pp. 479-489.

[12] R. C. Zhang. "Cooperative convolution of Elman recurrent neural networks for chaotic time series prediction. Neuro-computing", Volume 86, June 2012, pp. 116-123.

[13] H. Tan. "The Application of Improved Elman Neural Network in the Exchange Rate Time Series". Artificial Intelligence and Computational Intelligence (AICI), 2010 International Conference on, vol.3, October 2010, pp. 440-443.