2019

Time Series Analysis on Stock Market Data

Peiyuan Zhou (20803542)

SYDE 631 PROJECT REPORT

周培媛

MA.S.C. SYSTEM DESIGN ENGINEERING | University of Waterloo

Contents

Abstract

The problem of stock analysis is typically formulated as a problem of predicting stock movements based on daily closing prices collected over a period. To solve the forecasting/predicting problem, both time series modeling approach (e.g. ARIMA model) and machine learning approach (e.g. SVM, Decision Tree, ANN prediction method) could be considered. The aims of the study are to identify a model best fitting the time series of DJA stock price from 2014 to 2018 and to forecast the stock price in 2019. I also provide two discussion on the result by using traditional time series analysis approach, the one is whether the seasonal model with different periods would improve the forecast result, the other is whether different time intervals would improve the forecast result. In addition, I also apply the machine learning approaches to construct classifiers and discuss the differences between the result of time-series modeling and the result of machine learning.

Introduction

The stock market is well-known to be difficult to model and predict because it is chaotic in nature and highly random [1]. Analysts and investors have been relying on the use of different mathematical techniques to do so in the past [2]. The time series models, such as ARMA, ARIMA, are popular to be applied to solve the forecasting problem when treating the stock price as a univariate time series. For example, Gahirwal [3] decomposed a time series into multiple components, which included trend, seasonality and an irregular component to represent noise data. Then he performed forecasting based on each component separately. His study demonstrated that the prediction accuracy could be improved when more information was added.

More recently, as a powerful analytical tool, data mining has gained popularity over the years and has been used for solving financial problems or assisting related tasks. A number of traditional classification algorithms have been successfully used for predicting stock market movements, such as decision trees, artificial neural networks [4], support vector machines [5], and genetic algorithms [6] etc., to analyze stock data in order to predict the trends that can be predicted [7], and stock prices of future trading days [8], etc. For example, in [9], attempts have been made to use an SVM classifier to predict the upward or downward direction of stock prices and the Korean composite stock price index based on five parameters of stock. In [10], an artificial neural network-based approach is used to develop a classifier to predict the closing prices of selected stocks that are traded in the Bombay Stock Exchange (BSE). The network developed consists of an input layer (open, high, low, close and volume), one hidden layer and an output layer.

In this study, considering time series analysis is a basic concept within the field of statistical learning that allows the user to find meaningful information in data collected over time, I apply the strategy of model construction (i.e. identification, estimation and diagnostic test) to find the best model for the stock market data to predict future stock values. So, the main part of the report demonstrates the whole process of model construction considering different components of time series.

In addition, I also list more discussions about the dataset, for example in Discussion 1, I discussed whether the seasonal models with different periods can improve the forecasting result. Then in Discussion 2, I use the previous 8 differenced observations as the attributes to transform original time series into a relational table and apply some classification algorithms (e.g. SVM, C5.0, ANN) to discuss whether the machine learning approach could improve the forecasting result. In addition, in Discussion 3, besides the close price, I add more attributes, such as the highest price, lowest price and volume to transform a univariate time series into a multivariate time series and applying the classifiers on it to discuss the prediction result.

Data Description

In this case, for performance evaluation, I applied time series analysis methods on "Dow Jones Composite Average" of New York Stock Exchange which is collected from Yahoo Finance Website [11]. The dataset contains 5 years stock price data from Nov. 2014 to Oct. 2019. The Dow Jones Composite Average, it should be noted, is comprised of 65 stocks from eight standard industrial categories that make up the Dow Jones Industrial Average, the Dow Jones Transportation Average, and the Dow Jones Utility Average. For forecasting the stock price, I choose the first four-year data as training data (from 2014-11 to 2018-10), the last year as testing data (from 2018-11 to 2019-10). I use the monthly data as an illustration case to show the whole process of implement, and I will discuss seasonality, machine learning and multivariate time series in the Discussion section.

Model Construction and Forecasting

1. <u>Model Identification</u>

Firstly, the monthly changes of DJA stock price is plotted as Fig. 1(a). When decompose the time series, the original data shows obviously trend. The components are plotted in Fig. 1(b).

According to the shown in Fig. 1(a) and Fig. 1(b), I listed all the statistical properties of the time series as follows and summarized them as Table 1.

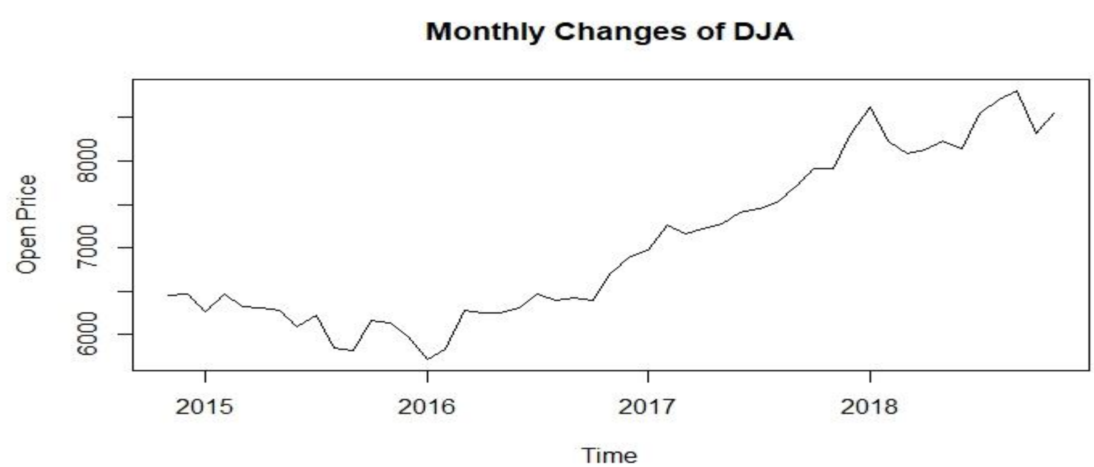


Fig. 1(a) Monthly Changes of DJA Stock Price

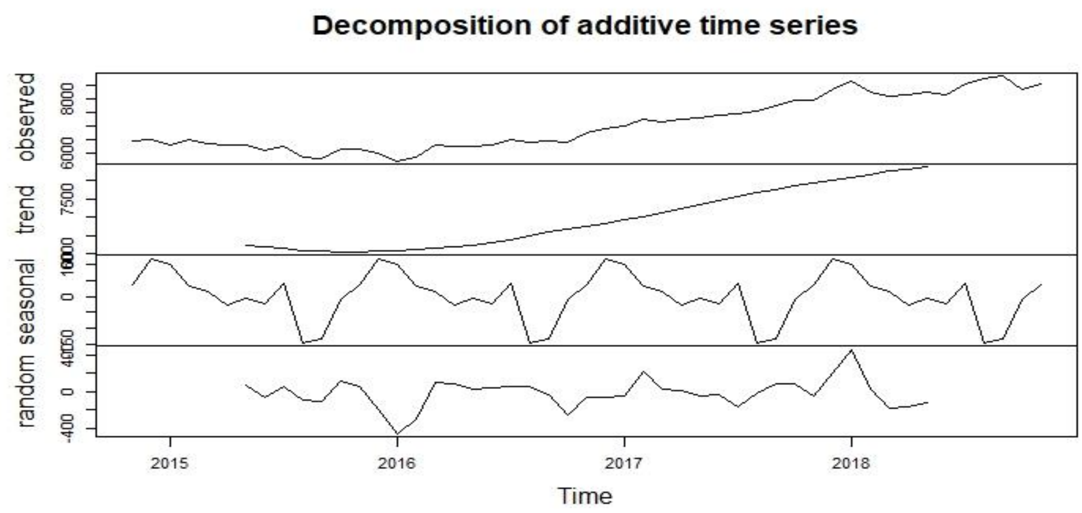


Fig. 1(d) Decomposition of Stock Prices

1. **Addictive Decomposition**: Since the variance looked to be about the same magnitude across time, so an additive decomposition is applied in this case.

2. **Non-Stationary with Trend**: There is obvious trend, and the augmented Dickey–Fuller test (ADF test) result shows the p-value=0.8 for the hypothesis stationary. Since 0.8>0.05(95% confidence), the time series should be non-stationary. In this case, the differencing should be considered, so ARIMA model would be a better choice.

3. **Seasonality**: It is difficult to determine the seasonality according to the plot, so I didn't consider seasonality firstly, and will discuss the forecasting result when different periods of seasonality is considered in Discussion 1.

4. **Transformation:** After differencing for removing trends, Fig 2(a) shows the plot of differenced time series. The plot looks like random time series. Then to test the normality of the time series, I use Q-Q plot in R as Fig.2(b) shows. In addition, in the Shapiro-Wilk normality test, p-value equals 0.6656 (larger than 0.05, 95% confidence). The data is normal if the p-value is above 0.05, so the data do not need transformation.

5. **Autocorrelation:** The ACF and PACF plot for differenced time series is shown as Fig. 2(c). For both ACF and PACF, only when lag=1, the value is above the 95% confidence. While the plot of ACF and PACD tails off not cut off. So, I consider ARMA(1,1) for the differenced time series.
6. **Extreme values:** I cleaned the series using *tsclean* command in R to remove the outliers. The result is same with the original time series, so the original observations doesn't contain extreme values.

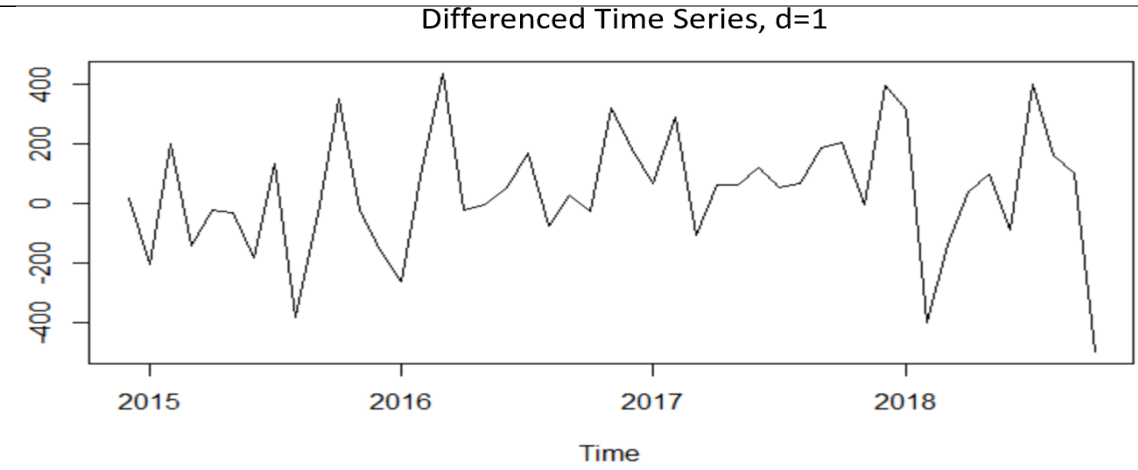


Fig. 2(a) Differenced Monthly Changes of DJA Stock Price

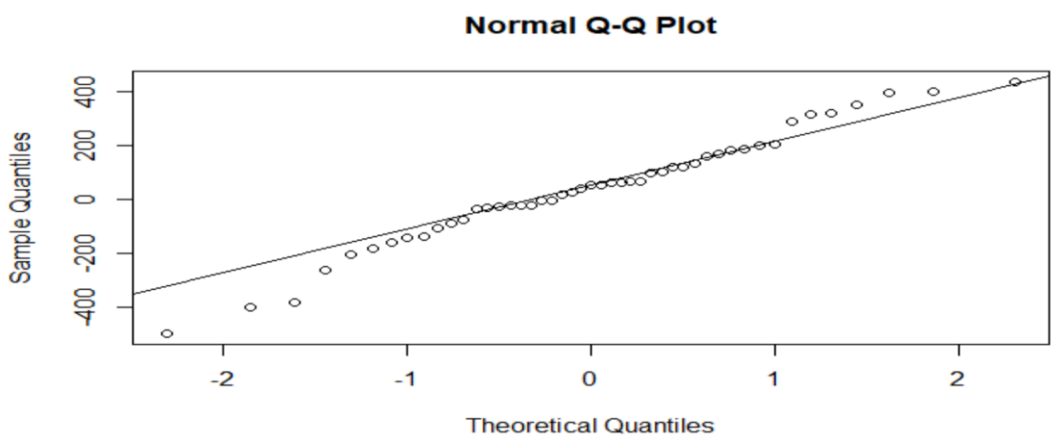


Fig. 2(b) QQ Plot - Normality

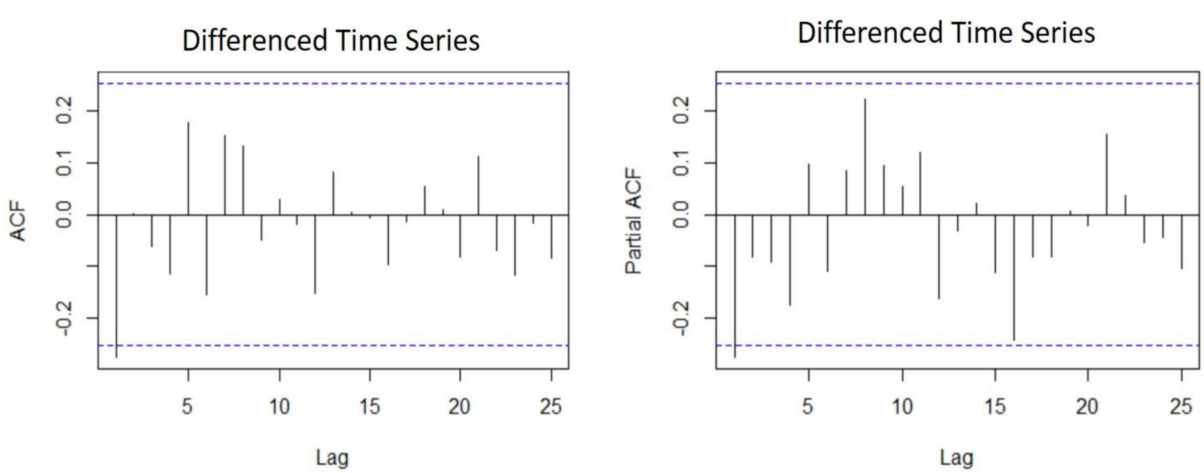


Fig.2(c) ACF and PACF

Statistical Properties	Detected or not
Non-Stationarity	Yes
Trends	Yes
Need for a transformation	No
Seasonality	No
Autocorrelation	Yes
Extreme Values	No
Long Term cycles	No
Known or unknown interventions	No

Table 1. General Statistical Properties of the Series Are Detected in The Graph

2. <u>Estimation</u>

In this stage, I used AIC, BIC and MLE to estimate the best overall ARIMA model. The definition of AIC is ***AIC=-2lnML + 2k***, where ML is maximum likelihood, so *lnML* is the value of maximized log likelihood function for a fitted model, k is the number of model parameters. In addition, BIC can be treated as another estimator, defined as ***BIC= -2lnML+ kln(n)***. It is like AIC but change ***2k*** into ***ln(n)k*** where n is the number of samples, so ***BIC = AIC-2k+kln(n).*** Table 2 summaries the value of AIC, BIC and MLE for model estimation.

As a result, the ARIMA(0,1,0) is selected as the final model. This result shows that the increase and decrease of the stock price is totally random.

	AIC	BIC	MLE
ARIMA(0,1,0)	**636.27**	**638.12**	**-317.14**
ARIMA(1,1,0)	638.22	641.93	-317.11
ARIMA(0,1,1)	638.2	641.91	-317.1
ARIMA(1,1,1)	639.65	645.2	-316.82
ARIMA(1,1,2)	641.46	648.86	-316.73
ARIMA(2,1,2)	639.38	648.63	-314.69
ARIMA(0,1,8)	636.34	652.99	-309.17

Table 2 AIC, BIC and MLE for different ARIMA model.

To test whether the ARIMA (0,1,0) is the ideal model for the original observations, I tested the overfitting model further. Since Fig. 2(d) shows the ACF plot for the residual when ARIMA (0,1,0) is applied for the original observations, when lag=8, the value of ACF is larger than significant statistical level. So ARIMA (0,1,8) is selected to test whether it is better than ARIMA (0,1,0). Firstly, the MLE value of ARIMA(0,1,8) is -309.17, larger than -317.14 for ARIMA(0,1,0), ARIMA(0,1,8) is more accurate than. The AIC of ARIMA(0,1,8) is like the model ARIMA(0,1,0). However, if ARIMA(0,1,0) is chosen, that is to say the differenced series is treated as random time series, it looks not helpful for the forecasting. So, I will plot both forecasting result for ARIMA(0,1,0) and ARIMA(0,1,8)

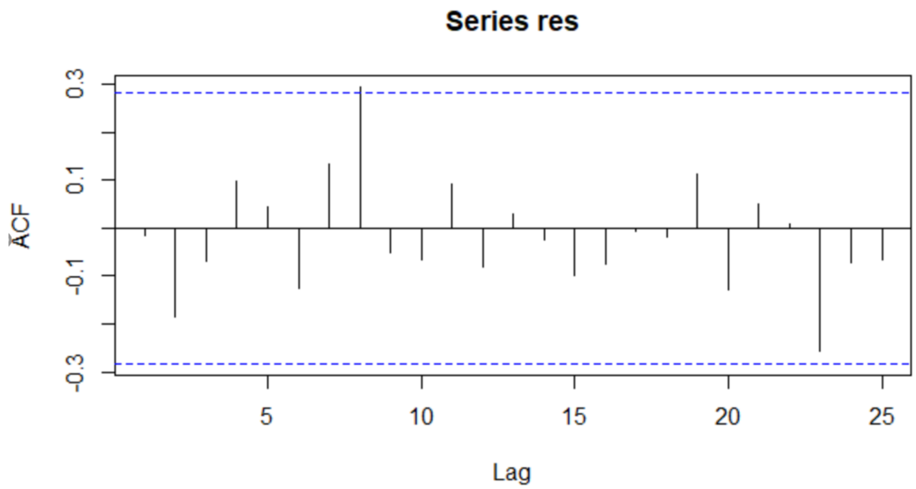


Fig. 2(d) The ACF of residual time series when ARIMA(0,1,0) is applied

3. **<u>Diagnostic Checking</u>**

   In the third stage, I finished diagnostic checking for the selected model to determine whether the assumptions (i.e. independent, homoscedasticity and normality) underlying the series are satisfied by the residuals of ARIMA(0,1,8) model.

   1. **<u>Independent:</u>** Fig 3(a) shows the ACF and PACF residuals and the plot illustrate the residual is white noise because all the value of ACF are lower than the significant level with 95% confidence.
   2. **<u>Heteroscedasticity:</u>**
      The heteroscedasticity is checked by the stationarity of the model by *adf.test()*. The result of is 0.04952, which is less than significant level (0.05), which means the residual is stationary. Thus, the residual has constant variance.
   3. **<u>Normality:</u>** To test the normality of the residual, I use Q-Q plot in R as Fig.3(b) shows. In addition, in the Shapiro-Wilk normality test, p-value equals 0.3545 (larger than 0.05, 95% confidence), so the residual is normality.

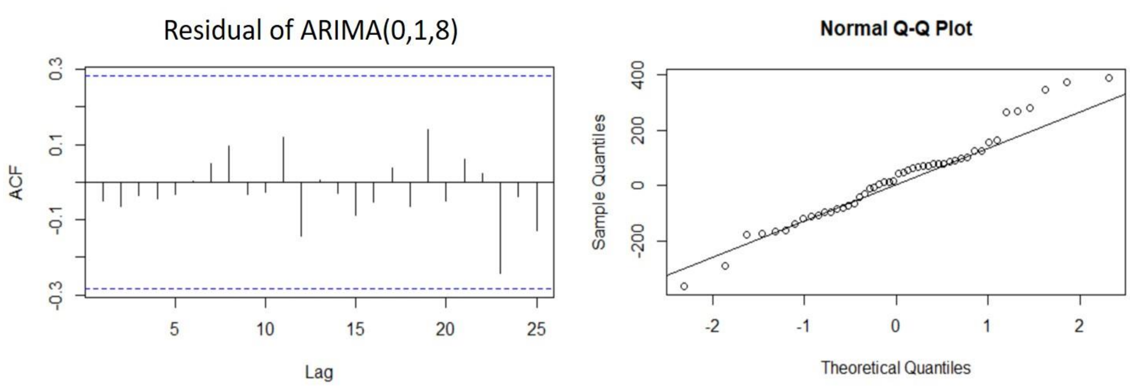


Fig. 3(a) The ACF (b) the QQ-plot of residual time series when ARIMA(0,1,8) is applied

4. Forecasting

Figure 4 (a) and (b) shows the different forecasting result when different ARIMA models are applied. In conclude, for ARIMA(0,1,0) the forecasting result is just a line with trend, while when ARIMA (0,1,8) is applied, the forecasting result follows the trends with fluctuation. To evaluate the forecasting result, I use Mean Absolute Error (MAE) and Root mean squared error (RMSE) as introduced in [12]. Table 3 compared the accuracy of two models for training dataset and testing dataset, and it shows ARIMA(0,1,8) can provide a slightly more accurate forecasting result. Therefore, the models that we can selected for this case as: $\boldsymbol{x_t = -0.13a_{t-1} - 0.15a_{t-2} - 0.18a_{t-3} + 0.13a_{t-4} + 0.4a_{t-5} - 0.23a_{t-6} + 0.37a_{t-7} + 0.33a_{t-8} + 34.52}$, $\boldsymbol{a_{t-1} \sim N(1, \sigma_a)}$

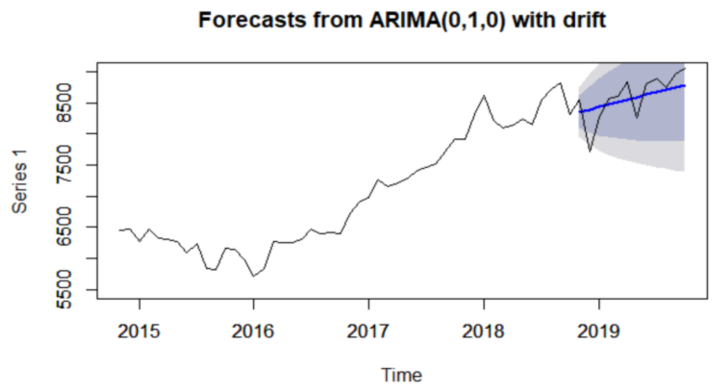


Fig. 4(a) The forecasting result when ARIMA(0,1,0) is applied

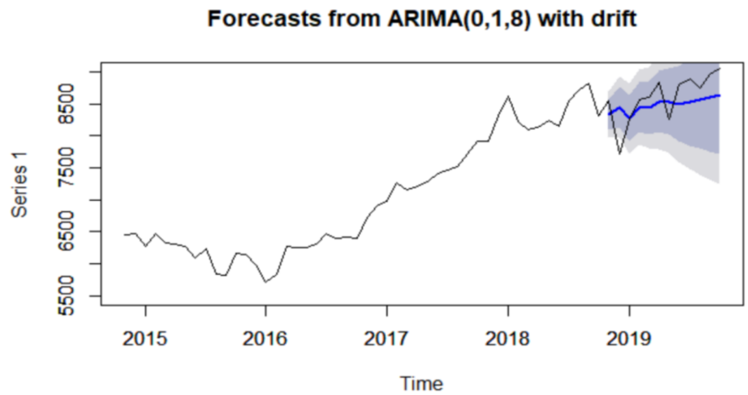


Fig. 4(b) The forecasting result when ARIMA(0,1,8) is applied

Training Set	**ME**	**RMSE**	**MAE**	**MPE**	**MAPE**	**MASE**	**ACF1**
ARIMA(0,1,8)	2.60	**157.23**	**122.56**	0.0009	1.73	1.15	-0.01
ARIMA(0,1,0)	0.13	200.2	150.6	-0.07	2.16	0.18	-0.017
Testing Set	**ME**	**RMSE**	**MAE**	**MPE**	**MAPE**		
ARIMA(0,1,8)	-2260.8	**2278.5**	**2260.8**	-36.49	36.49		
ARIMA(0,1,0)	-2342.7	2365.9	2342.7	-37.83	37.83		

Table 3 Comparison between two ARIMA model

Discussion 1: Seasonality or non-Seasonality?

In above result, I didn't consider seasonality because according to the plot of ACF and decomposition result, there is no obvious seasonality is discovered in the time series. In this section, I will discuss whether the seasonality with different periods will impact the forecasting result. Firstly, according to setting different periods for the original observations, I applied the different seasonal differencing (i.e. diff lag = 4, 6, 12, 18) for removing seasonality components from the observations. The ACF and PACF plots for the differencing result are shown in Fig. 5.

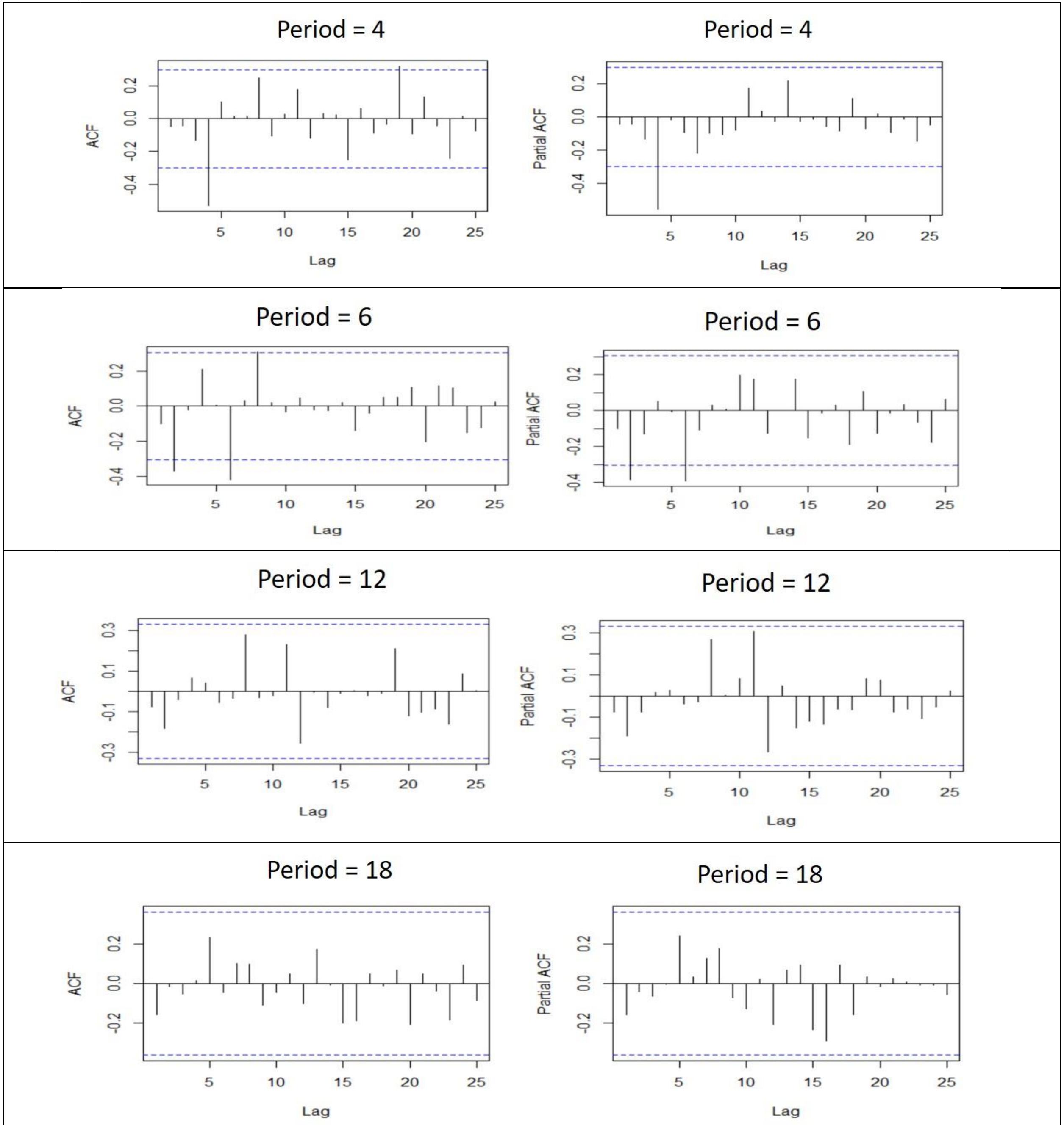


Fig. 5 ACF and PACF plots for Seasonal Differencing Series when Different Periods are Applied

According to the ACF and PACF result, I select the different models. For example, when period equals four, both ACF and PACF dies off but not cut off, and when lag=4, the values of ACF and PACF are larger than 95% confidence level, so ARIMA(0,1,0)(1,1,1)$_4$ is selected. And similarly, when period equals six, neither ACF and PACF shows dies off, and when lag=2 and lag=6, the values are over 95% confidence level, so ARIMA(2,1,2)(1,1,1)$_6$ is selected. When period equals 12 and 18, all the values of ACF and PACF are lower than 95% confidence level, looks like white noise, so ARIMA(0,1,0)(0,1,0)$_{12}$ and ARIMA(0,1,0)(0,1,0)$_{18}$ are selected. Table 4 shows the value of AIC, BIC and MLE for different models. In addition, the forecasting result are plotted in Fig. 6.

Models	AIC	BIC	MLE
ARIMA(0,1,0)(1,1,1)$_4$	588.85	594.13	-291.42
ARIMA(2,1,2)(1,1,1)$_6$	578.52	590.51	-282.26
ARIMA(0,1,0)(0,1,0)$_{12}$	496.72	498.28	-247.36
ARIMA(0,1,0)(0,1,0)$_{18}$	410.44	411.81	-204.22

Table 4 AIC, BIC and MLE for different ARIMA models with different periods of seasonality.

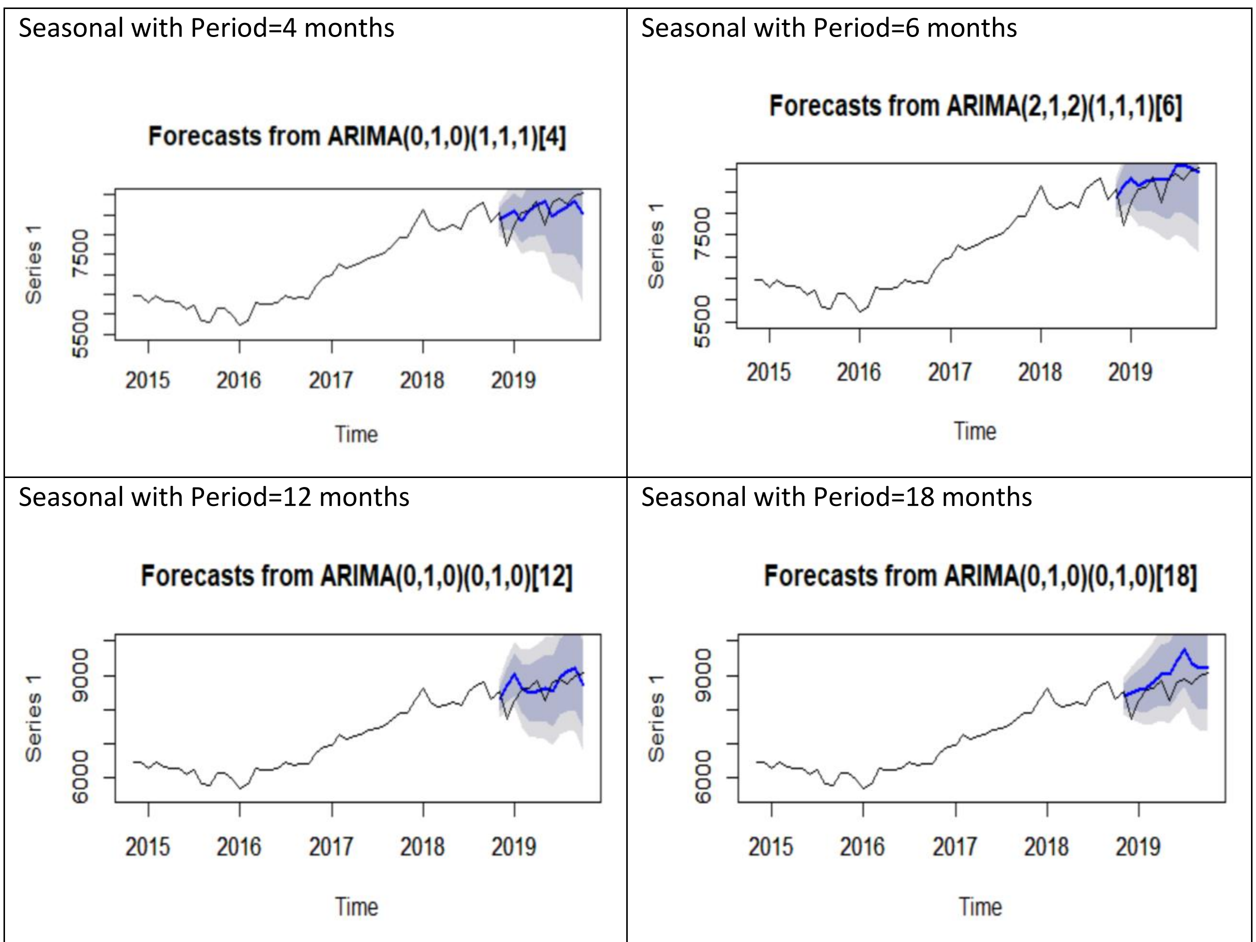


Fig. 6 The Plotting of Forecasting Result when Different Models are Applied.

To evaluate the forecasting result, I also use Mean Absolute Error (MAE) and Root mean squared error (RMSE) [12]. Table 5 shows the comparison of the forecasting accuracies on testing data when different models are used. Compared to the result of non-seasonality models, the values forecasting errors are even higher than the forecasting errors of non-seasonality models. Hence, we can conclude that the seasonal model cannot improve the forecasting accuracy in this case.

Testing Set	ME	RMSE	MAE	MPE	MAPE
ARIMA(0,1,0)(1,1,1)$_4$	-2356.686	**2379.318**	**2356.686**	-38.0509	38.0509
ARIMA(2,1,2)(1,1,1)$_6$	-2501.544	2537.455	2501.544	-40.42022	40.42022
ARIMA(0,1,0)(0,1,0)$_{12}$	-2509.703	2547.685	2509.703	-40.56518	40.56518
ARIMA(0,1,0)(0,1,0)$_{18}$	-2751.511	2807.694	2751.511	-44.49005	44.49005

Table 5 Forecasting Result for Testing Dataset when Different Seasonality Models are Applied.

Discussion 2: Machine Learning Strategies

In the application of stock prediction, one may ask if we are to predict the stock movement in future based on the previous price? So, in this section, I transform the problem into a classification problem by considering the previous stock price as several attributes' values and the stock movement (i.e. Up and Down) as the class label. I will combine the machine learning classifiers with some of above conclusions by time series analysis to see whether the machine learning strategies can help to improve the forecasting result.

2.1 Transforming

Firstly, I removed the trend components from the time series by differencing to avoid trend impact for the forecasting. And the change of stock price for previous 8 months stock price are used as eight attributes in the relational table because in our previous case, lag=8 could be a good choice for model construction. And then the stock movement (i.e. Up, Down) is used as class. As an example, after transforming original time series into a relational table, Table 6 shows the preview of the data. For example, the first records show the stock price changes from Dec-2014 to July-2015 as attribute values, then the stock movement (shown as Class) on Aug-2015 is "Up".

	Lag = 1	…	Lag = 8	Class (Two Levels)	Class (Three Levels)	Class (Five Levels)
Aug-2015	-181.08	…	181.77	Up	Up	Up
Sep-2015	130.1099	…	26.02002	Down	Down	LDown
…	…	…	…	…	…	…
Oct-2018	151.0098	…	263.0098	Up	Up	Up
Nov-2018	141.7402	…	-373.06	Down	Down	LDown
…	…	…	…	…	…	…
Oct-2019	-199.01	…	630.7505	Up	Up	LUp

Table 6. Preview of Dataset as Inputting for Machine Learning Classifiers.

In addition, I consider the different levels for evaluate the stock movement, as shown the last three columns in Table 6. When two levels are considered, I set *UP* as “the amount of increases of stock price is higher than 0%” and *DOWN* as “the amount of decreases of stock price is lower than 0%”; when three levels are considered I set *UP* as the stock price increases more than 1%, *DOWN* as the stock price decreases more than -1%, and *EQUAL* as the change level in the interval [-1%, 1%]; when five levels are considered, I set *LUP* and *LDOWN* in which the stock price changes are more than 3% or -3%, respectively, *UP* as the changes between 1% and 3%; *DOWN* as the stock price decreases more than [-3%, -1%] and *EQUAL* as the change level in the interval [-1%, 1%].

By using the transformed table, I also use the first four years stock price as training dataset, marked as green color in Table 6; and the stock price of 2019 as testing dataset, marked as yellow color in Table 6.

2.2 Experiment Process and Evaluation

After transforming time series into a relational table, each row represents a feature vector for the target. Then the classifiers can be applied to classify the feature vectors into different classes. The following are three classical classification algorithm, Support Vector Machine (SVM) with RBF [13], MLP Artificial Neural Networks (ANN) [8] and Decision Tree C5.0 [14].

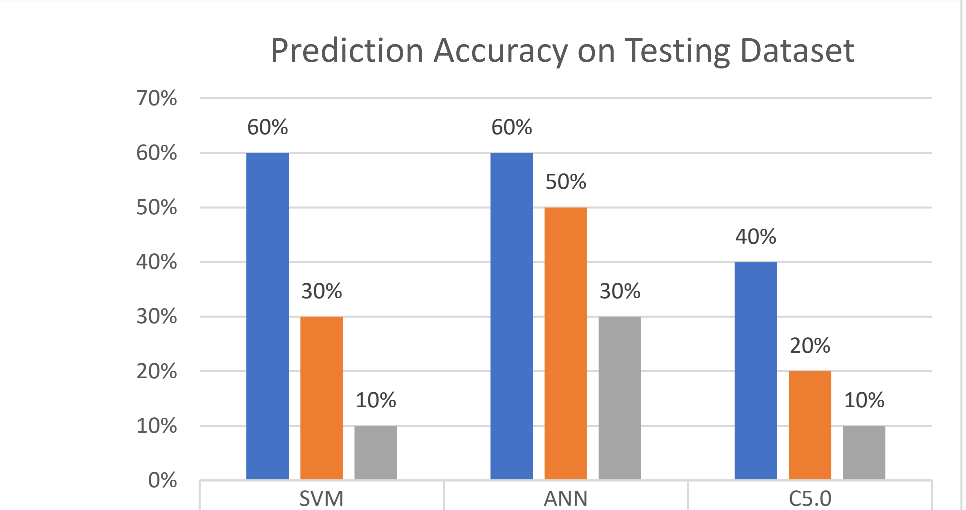
Support Vector Machine (SVM): The SVM transforms original input data into a higher dimensional space using a nonlinear mapping and then searches for a linear separating hyper-plane. In this case, if we treat the input as a m*n matrix, m represents the number of months records are used, and n represents the number of features (e.g. change of stock price in 8 previous months).

Artificial Neural Network (ANN): ANN is composed of interconnecting artificial neurons that can compute values from inputs. Multi-Layer Perceptron (MLP) and Radial Basis Function (RBF) are two of the most widely used neural network architecture in literature for classification or regression problems. RBF is a local type of learning which is responsive only to a limited section of input space, and MLP is a distributed approach. In this case we choose ANN with RBF function.

C5.0: C5.0 is one of the most well-known decision trees for classification. Compared to more advanced and sophisticated machine learning models (e.g. Neural Networks and Support Vector Machines), the decision trees under the C5.0 algorithm generally perform nearly as well but are much easier to understand and deploy.

When 10 months in 2019 are selected as testing dataset, Figure 7 summarized the prediction accuracy. When two levels of class are set, 50% is the random chance to estimate “Up” and “Down” of the stock price change. And the result of time series analysis model also shows MA model is more suitable for the change of stock price. So, when machine learning approach is applied, the result looks similar, for the two levels class, the prediction accuracies are 60%, 60% and 40% for SVM, ANN and C5.0 respective.

Similarly, when three levels of class are considered, 33% is the random change to estimate "Up", "DOWN" and "EQUAL". And the prediction accuracies are 30%, 30% and 20% for SVM, ANN and C5.0 respective. And when five levels are considered, 20% is the random change to estimate "LUP", "UP", "LDOWN", "DOWN" and "EQUAL". And the classifiers perform around 20%. In conclude, ANN could perform better than the other two classifiers. And this result is almost consistent with Time Series Modeling result to shows that the change of stock price looks random.



	SVM	ANN	C5.0
Two Levels	60%	60%	40%
Three Levels	30%	50%	20%
Five Levels	10%	30%	10%

Fig. 7 The Prediction Accuracies of Different Classifiers (Univariate Case)

Discussion 3: Multivariate Time Series

For stock analysis to be more complete, other than opening/closing prices, other data parameters, such as the daily maximum and minimum, the bid and ask spread, the trading volume, etc., which are usually also collected for each stock, should also be considered. In other words, there is a need for an effective computational technique to be developed to ensure that important associations between different parameter values observed at different time instants be taken into consideration during stock analysis, since a univariate time series analysis technique may not discover such relationships. For example, it is possible for the differences between the maximum and minimum stock prices on a time point to indicate a significant rise or drop in a stock's closing price the following time point. Compared to the univariate time series is plot in Fig. 1(a), Fig. 8 shows the plot of normalized multivariate time series.

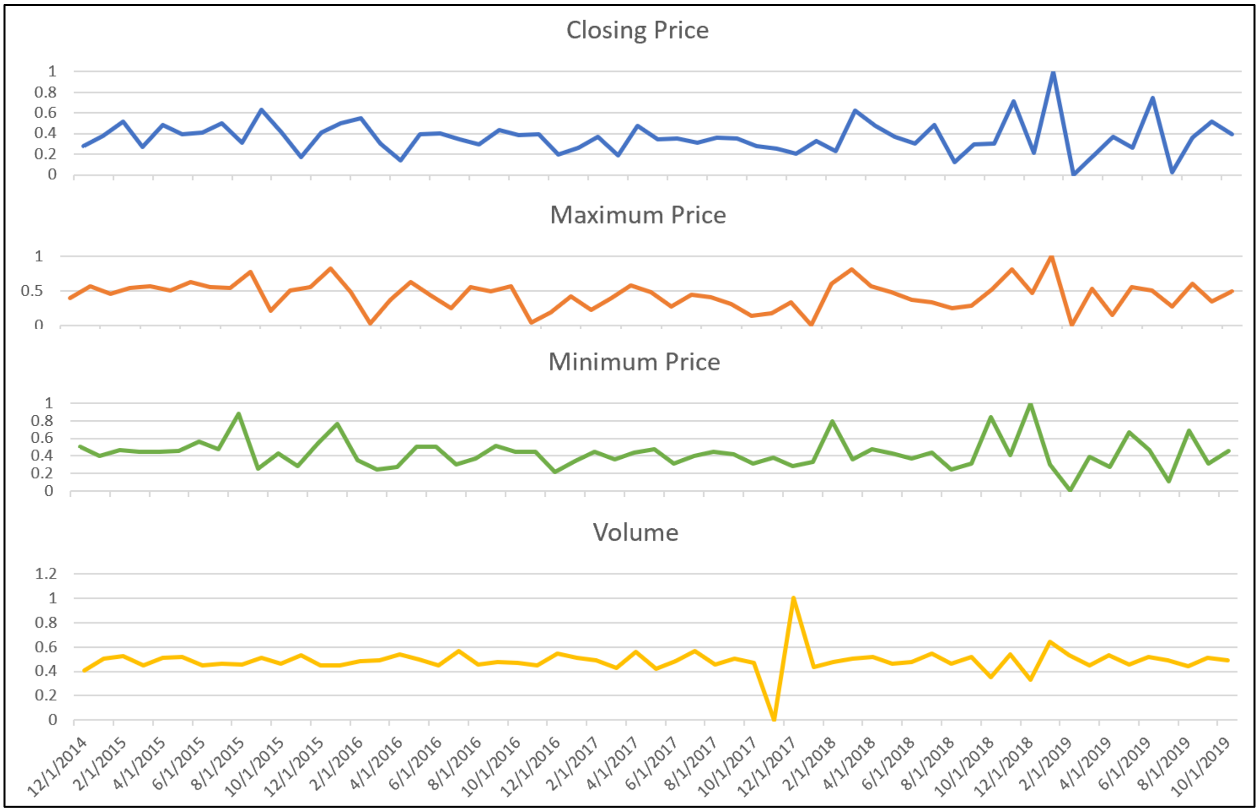
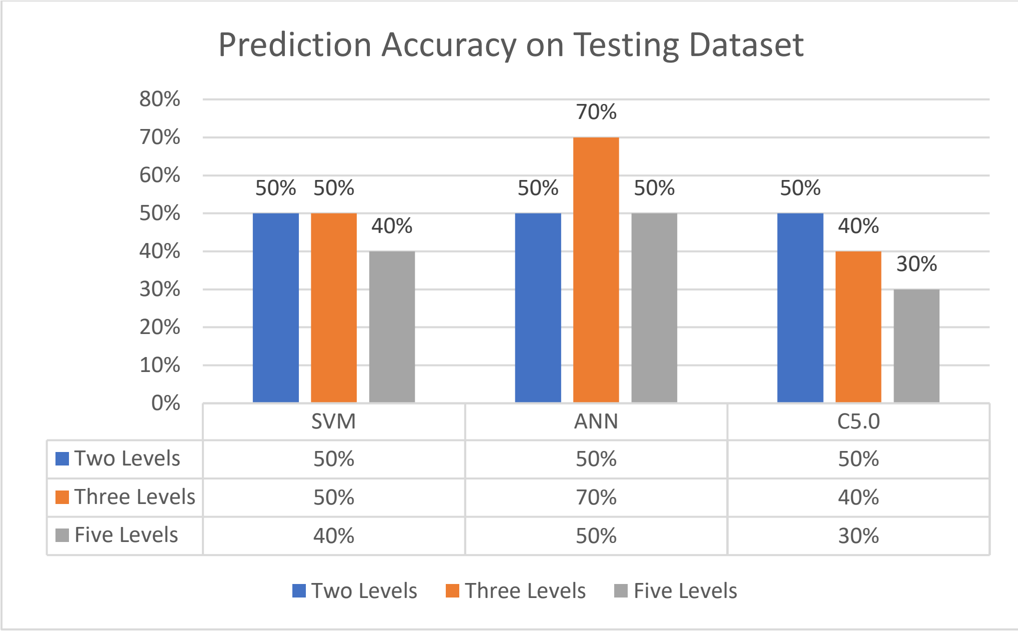


Fig. 8 Plot of Multivariate Time Series

In this section, I will add more attributes, such as the "differences between maximum and minimum stock price" and "volume". And adding all previous 8 months for each attribute. Then, in this case, the previous feature vector containing 8 features is extended into a larger feature vector containing 8*3=24 features. The class with different levels is also used to be target for prediction. I still use the first four years data as training data, and the last ten months in 2019 as testing data.

As shown in Fig. 9, the summarized prediction accuracy, when more attributes are used for classification, the prediction result is improved, especially for the case of three levels and five levels of class are used. When two levels of class are set, the result of classification using different classifiers almost same with the result in Discussion 2. While when three-level class and five-level class are used, the prediction accuracy is improved 20% for all classifiers.

Fig. 10 shows the comparison of prediction result when less or more features are used for classification. The solid line shows the result when multivariate time series is used for classification and dot line shows the result when univariate time series is used.



	SVM	ANN	C5.0
Two Levels	50%	50%	50%
Three Levels	50%	70%	40%
Five Levels	40%	50%	30%

Fig. 7.2 The Prediction Accuracies of Different Classifiers (Multivariate Case)

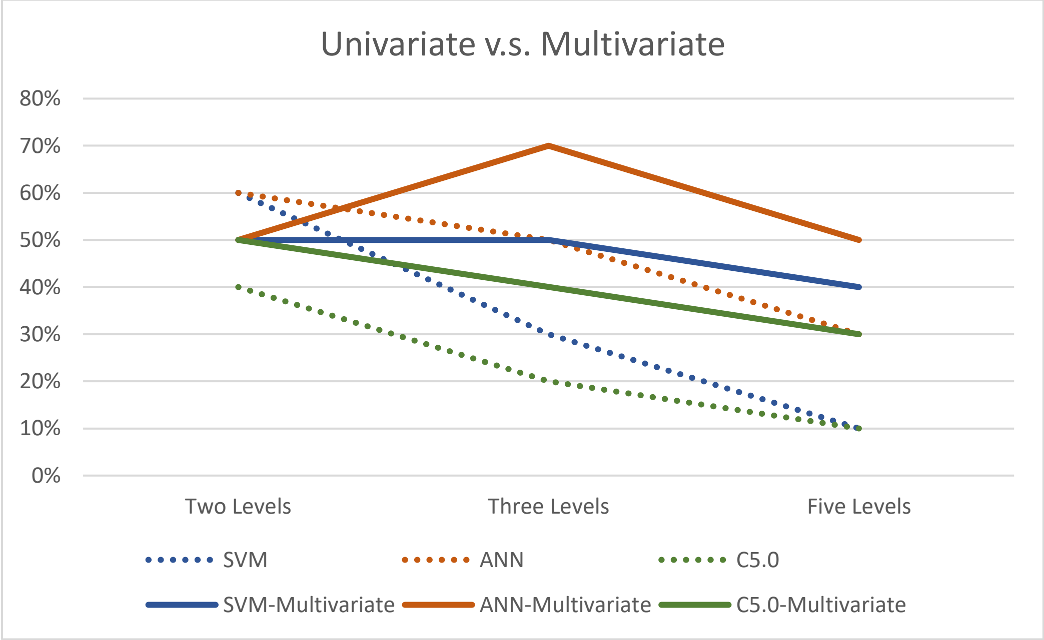


Fig. 7.3 Comparison of Prediction Result for Univariate Time Series and Multivariate Time Series

Conclusion

In this study, I discussed both time series modeling and machine learning strategies for analyzing the change of stock price monthly. Firstly, I constructed the model by identifying model, estimate model and diagnostic test and the final model is selected as ARIMA(0,1,8) and illustrate the forecasting result. Then I construct some seasonal and stationary models but considering seasonality didn't improve the forecasting result. In discussion 2 and 3, I also use some classification models (SVM, C5.0 and ANN) to predict the stock price in future. When more attributes, or called multivariate time series, are used, the prediction result is improved.

During processing the stock market data, I met some problems, for example, the change of stock price looks random, and it is difficult to find a fitted stationary model for forecasting; and the series doesn't show obviously seasonality and it is difficult to determine the period of seasonality.

Hence, in future work, on one hand, I can use some short-term stock price dataset, such as daily or weekly instead of monthly, since by using the monthly data, the cycle may be discovered in a long period which require we use the long-term dataset instead of 5 years data set. On the other hand, I can combine both machine learning and time series modeling furthermore to obtain the more accuracy prediction result.

References

[1] P. Roy, R. Kumar and S. Sharma, "A Fuzzy Logic Model to forecast stock market momentum," *Int. J. Adv. Engg. Res. Studies/IV/II/Jan.-March,* vol. 98, no. 101, 2015.

[2] S. Nicholis and D. Sumpter, "A dynamical approach to stock market fluctuation," *International Journal of Bifurcation and Chaos,* vol. 21, no. 12, pp. 3557 -564, 2011.

[3] M. Gahirwal and M. Vijayalakshmi, "Inter time series sales forecasting," *International Journal of Advanced Studies in Computers, Science and Engineering,* vol. 2, no. 1, pp. 55-66, 2013.

[4] Z. Theodorus and K. Dimitrus, "Short Term Prediction of Foreign Exchange Rates with a Neural-Network Based Ensemble of Financial Technical Indicators," *International Journal on Artificial Intelligence Tools,* vol. 22, no. 3, pp. 220-241, 2013.

[5] T. Sands, D. Tayal and M. Morris, "Robust stock value prediction using support vector machines with particle swarm optimization," in *Evolutionary Computation (CEC), IEEE Congress on. IEEE,*, 2015.

[6] D. Chandwani and M. Saluja, "Stock Direction Forecasting Techniques: An Empirical Study Combining Machine Learning System with Market Indicators in the Indian Context," *International Journal of Computer Applications,* vol. 92, no. 11, 2014.

[7] M. Gahirwal, "Inter Time Series Sales Forecasting," *International Journal of Advanced Studies in Computers Science and Engineering,* vol. 2, no. 1, pp. 55-66, 2013.

[8] B. Nair, M. Patturajan and V. Mohandas, "Predicting the BSE Sensex: Performance comparison of adaptive linear element, feed forward, and time delay neural networks," in *International Conference on Power, Signals, Controls and Computation*, 2012.

[9] Y. Wang and I. C. Choi , "Market index and stock price direction prediction using machine learning techniques: an empirical study on the KOSPI and HSI," *arXiv preprint arXiv:1309.7119.*, 2013.

[10] A. Devadoss, and T. Ligori, , " Stock prediction using artificial neural networks," *International Journal of Data Mining Techniques and Applications,* vol. 2, pp. 283-291, 2013.

[11] "Yahoo! Financial," [Online]. Available: http://finance.yahoo.com..

[12] R. J. Hyndman and G. Athanasopoulos, Forecasting: principles and practice., OTexts., 2018.

[13] P. Ou and H. Wang, "Prediction of stock market index movement by ten data mining techniques," *Modern Applied Science,* vol. 3, no. 12, p. 28, 2009.

[14] B. Nair, N. Dharini and V. P. Mohandas, "A stock market trend prediction system Using a hybrid decision treeneu-ro-fuzzy system," in *International Conference on Advances in Re-cent Technologies in Communication and Computing*, 2009.