

# Stock Price Forecasting using Hidden Markov Model

*Yulin Lei, Fu Wen, Xin Xu, Youyuan Zhang*

*MAY 6, 2016*

## Abstract

## Introduction

Stock prices are rather difficult to forecast. Proponents of efficient market hypotheses argue that stock prices cannot be predicted since the market prices have already reflected all known and expected fundamentals. However, precisely because the current market price reflects all relevant information, and therefore the current and historical prices are extremely useful for predicting the future prices since some relevant information is entering the stock markets continuously. It is well documented evidence that stock prices demonstrate strong momentum, short-term continuation and long-term reversal, and other identifiable patterns.

## Time series Analysis: ARIMA, SARIMA and Spectral Analysis

A time series is a sequence of vectors,  $x(t)$ ,  $t = 0, 1, \dots$ , where  $t$  represents elapsed time. For simplicity we will consider here only sequences of scalars, although the techniques considered generalise readily to vector series. Theoretically,  $x$  may be a value which varies continuously with  $t$ , such as a Closing price. In practice, for any given physical system,  $x$  will be sampled to give a series of discrete data points, equally spaced in time. The rate at which samples are taken dictates the maximum resolution of the model; however, it is not always the case that the model with the highest resolution has the best predictive power, so that superior results may be obtained by employing only every  $n$ th point in the series.

Time series are generally sequences of measurements of one or more visible variables of an underlying dynamic system, whose state changes with time as a function of its current state vector  $u(t+1) = F(u(t))$ . Such dynamic systems may evolve over time to an attracting set of points that is regular and of simple shape; any time series derived from such a system would also have a smooth and regular appearance. However another result is possible: the system may evolve to a chaotic attractor. Here, the path of the state vector through the attractor is non-periodic and because of this any time series derived from it will have a complex appearance and behaviour. In a real-world system such as a stock market, the nature and structure of the state space is obscure; so that the actual variables that contribute to the state vector are unknown or debatable. The task for a time series predictor can therefore be rephrased: given measurements of one component of the state vector of a dynamic system is it possible to reconstruct the (possibly) chaotic dynamics of the phase space and thereby predict the evolution of the measured variable?

Example: Amazon For our interested stocks, we have conducted the time series analysis with similar steps to construct a single variable ARIMA model. A detailed exploration process will be shown as below.

Firstly, we check the ACF and PACF plots for the first-order or second-order difference of the original data. By observing the number of bars outside of the blue lines, which give the values beyond which the autocorrelations are significantly different from zero.

With this routine step, though there are very much unlikely for the financial data to show a pattern of seasonality, we have still tried to use spectral analysis to detect periodicity of the data by transferring the data from time-domain to frequency domain with the help of FFT(Fast Fourier Transform) and visualize the Raw Periodogram.

In this specific example, there is an obvious peak when the frequency is about 0.11. Thus a possible seasonality worth testing is about 8 or 9. It is worth noting that the rest of the stocks do not show such pattern. However, taking this into consideration would be an essential step to dig potential information.

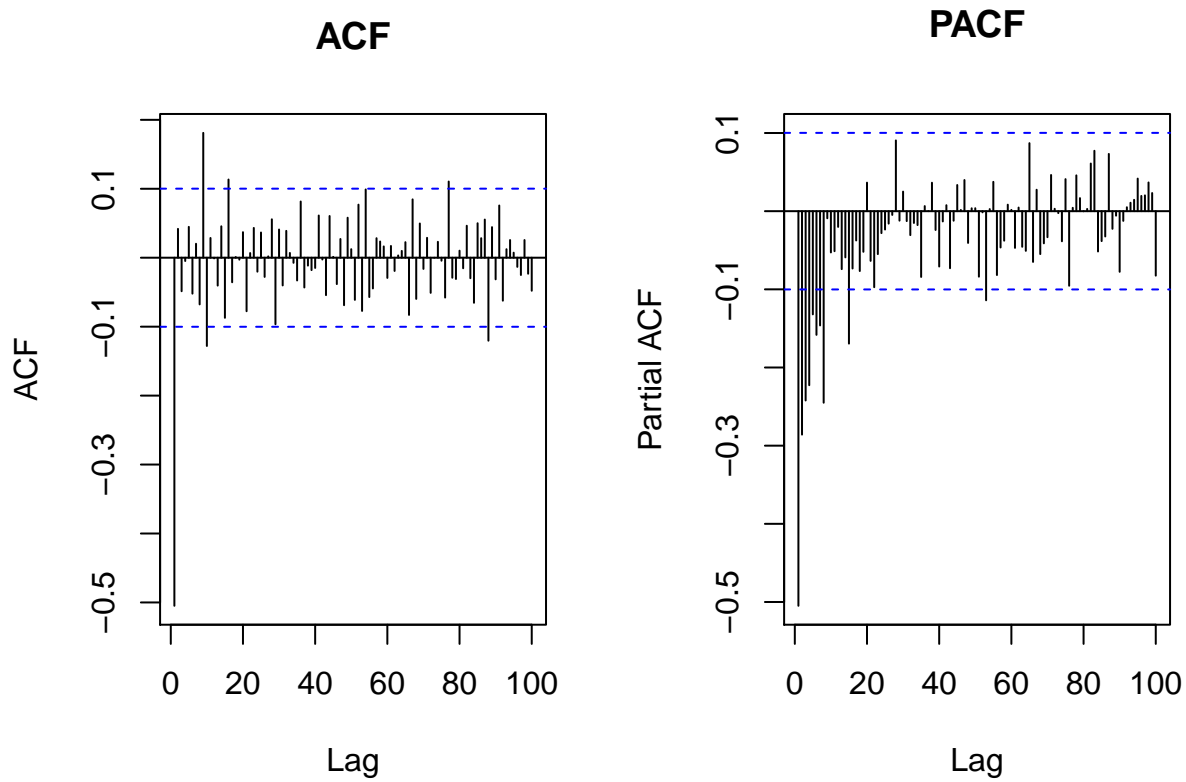


Figure 1:

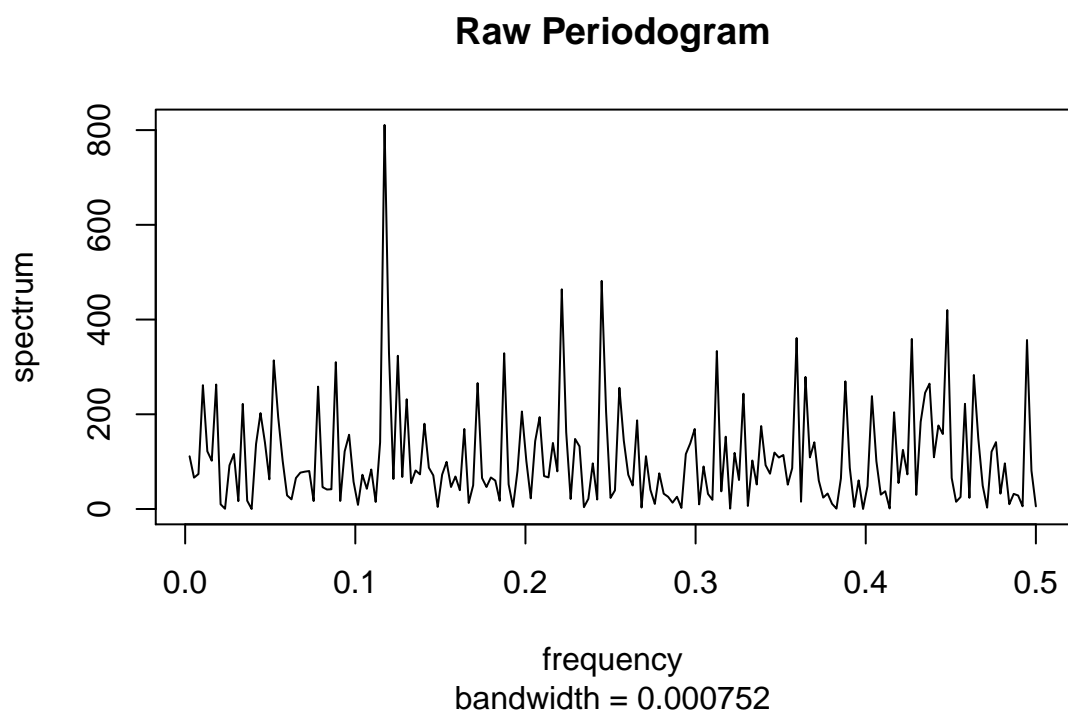


Figure 2: Raw Periodogram

After comparing several SARIMA(Seasonal Autoregressive Integrated Moving Average) models based on their AIC, a best model is selected for making forecast. Here the best for Amazon is :  $Arima(8, 0, 2) \times (0, 1, 0)_{(9)}$

The prediction plot can be seen as below with a shaded area of confidence interval of the forecast.

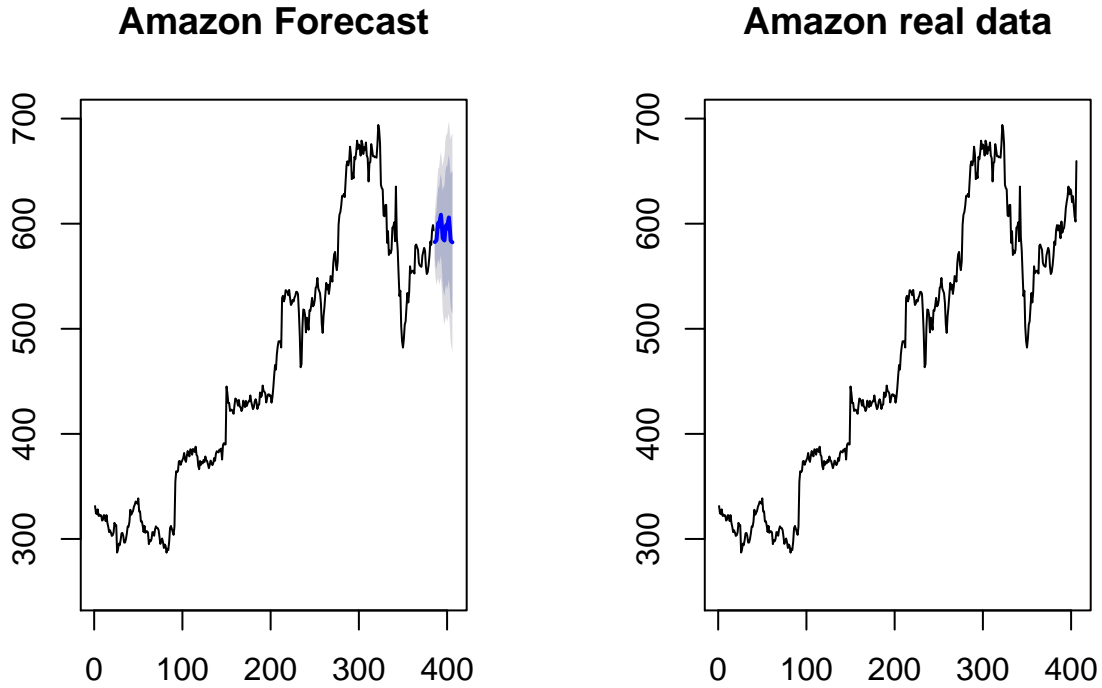


Figure 3: Amazon Time Series Forecast

From the graph we can see that the SARIMA model shows a similar trend of the data. However, the MSE is 408.088.

The similar analysis procedure is conducted for the rest of the five stocks, the forecasts plots are in the attachment and their MSEs are as below:

	MSE
Facebook	11.40
LinkedIn	8.94
Google	234.88
Apple	3.90
Microsoft	2.03
Alibaba	2.24

Table 1: MSE for Time Series Forecasting

## Gaussian Hidden Markov Model

A Hidden Markov Model deals with the Markov process with hidden(unobserved states). Markov chain demonstrates visible states directly. Transition probabilities are the only parameters. In a Hidden Markov Model, the states are not visible, but there are visible observations of output which depend on the hidden states. Each observation has a probability given a specific state, which is called the emission probability.

The Hidden Markov Models are broadly used in various areas like “temporal pattern recognition such as speech, handwriting, gesture recognition, part-of-speech tagging, musical score following, partial discharges and bioinformatics”.<sup>[2]</sup>

In this report, Hidden Markov Model is applied to make forecasts. Firstly, from past data, we fit Gaussian Hidden Markov Model to locate some patterns. Then, we forecast tomorrow's stock price of the variable of interest.

### Hidden Markov Model:

We build a hidden markov model as the following:

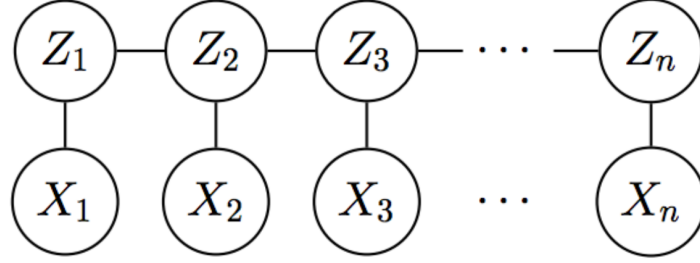


Figure 4: Hidden Markov Model

Denote

$T$  as the number of observations

$diff_t$  as the difference between closing prices of consecutive two days  $open_t$  as the the opening price

$close_t$  as the the closing price

$h_t$  as the highest price

$l_t$  as the highest price

$X_t, t = 1, \dots, T$  as the observation

$Z_t \in \{1, 2, 3, 4\}$  as the hidden states

### Algorithm

Firstly, we use the Bawm-Welch algorithm to estimate the parameters for HMM (the initial distribution  $\pi$ , the transition matrix  $T$ , and the emission distributions  $\epsilon_i$ ).

Since the observations are continuous series, we set the emission distributions as multinormal ones:

$$\epsilon_t = p(X_t | Z_t = j) = N(X_t | \mu_j, V_j), j = 1, 2, 3, 4$$

where  $\mu_j$  and  $V_j$  are the mean vector and covariance matrix for the observations at state  $j$ .

Secondly, plugging the estimated parameters into forward algorithm, we can get  $s_j(z_j) = p(x_{1:j}, z_j)$  and then we can calculate:

$$\begin{aligned}
p(x_{j+1}|x_{1:j}) &= \frac{p(x_{1:j}, x_{j+1})}{p(x_{1:j})} = \sum_{z_j, z_{j+1}} \frac{p(x_{1:j}, x_{j+1}, z_j, z_{j+1})}{p(x_{1:j})} \\
&= \sum_{z_j, z_{j+1}} p(z_j|x_{1:j})p(z_{j+1}|z_j)p(x_{j+1}|z_{j+1}) \\
&= \sum_{z_j, z_{j+1}} p(z_{j+1}, z_j|x_{1:j})p(x_{j+1}|z_{j+1}) \\
&= \sum_{z_{j+1}} p(z_{j+1}|x_{1:j})p(x_{j+1}|z_{j+1})
\end{aligned}$$

After obtaining the distribution of  $x_{j+1}|x_{1:j}$ , we sample 100 points and take the mean (posterior mean) as the prediction of  $\hat{x}_{j+1}$ .

### Implementation

**Data** We pick seven high-tech companies: Apple, Facebook, LinkedIn, Amazon, Google, Microsoft and Alibaba, including their price related statistics for nearly seven months.

**Forecast and Performance** Firstly, based on previous research[1], we take opening prices, closing prices, highest prices and lowest prices as the observations.  $X_t = (open_t, close_t, h_t, l_t)$ . We make one day forecast based on the previous 210 days' data.

From Viterbi Algorithm, we get the the most likely sequence of hidden states and plot them as below. x label is the date and y label is the real closing price.

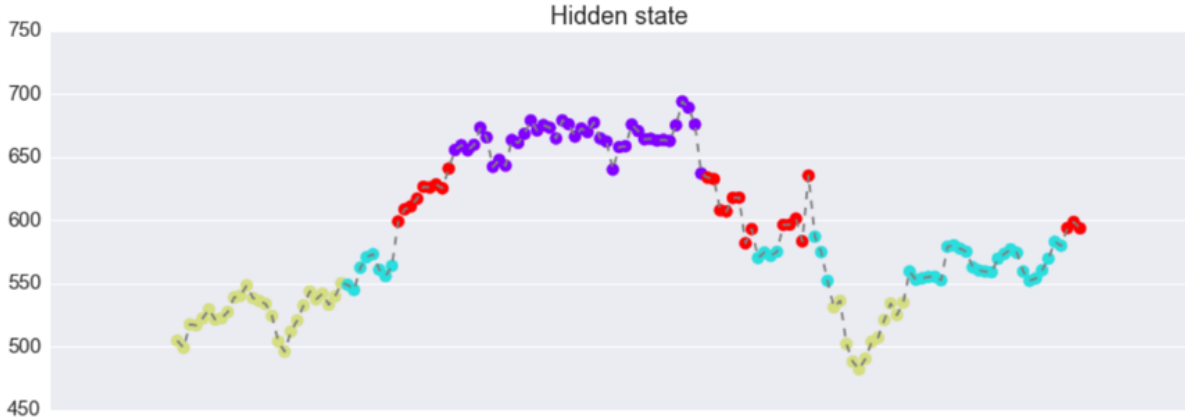


Figure 5: Hidden States of Amazon without Closing Price Difference

As we can see, the four price levels are caught well by the models. The forecasting is shown in the plot below. In terms of MSE(as the table shows), the performance is not that satisfying. From the plot, we can obviously found that the model failed to catch the trend of the stock. Since the model is fit on the previous data, the four hidden states correspond to the previous price level. For a given hidden state, the multinormal emission distribution  $p(x_i|z_i)$  has a fixed mean and variance. It's not surprising that the model can not catch the ups and downs of prices.

After notice that the prediction can't catch the trend, we add closing price difference  $diff_t = close_t - close_{t-1}$  to observations  $X_t = (diff_t, open_t, close_t, h_t, l_t)$ .

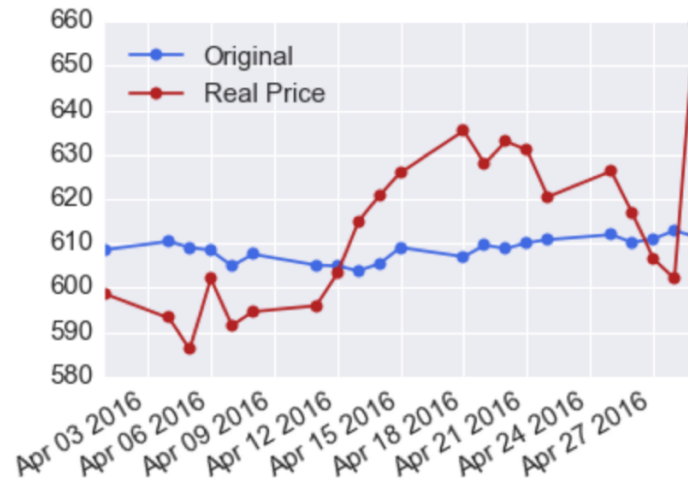


Figure 6: Forecasting of Amazon Stock Price without Closing Price Difference

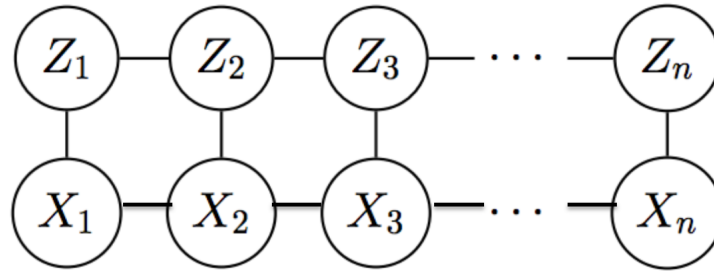


Figure 7: Hidden Markov Model with Correlated Observations

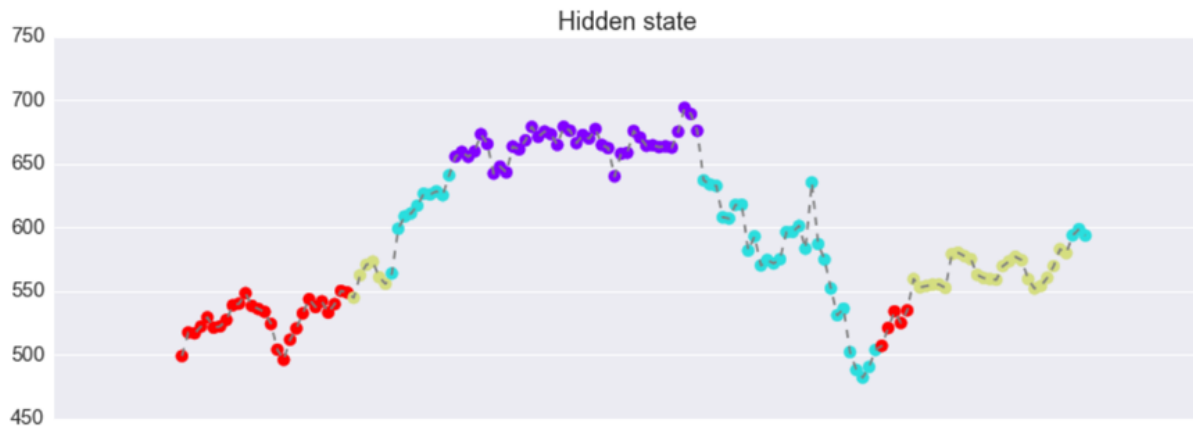


Figure 8: Hidden States of Amazon with Closing Price Difference

	MSE
Apple	17.91
Facebook	32.22
LinkedIn	81.72
Amazon	1066.49
Google	475.79
Microsoft	4.55
Alibaba	7.99

Table 2: MSE for HMM without Closing Price Difference

Under this model setting,  $X_t$  is correlated with  $X_{t-1}$  since the relationship:  $diff_t = clost_t - clost_{t-1}$

The MSE and forecasting plot is shown below. In terms of MSE, the performance is better and the trend can be caught to some degree.

	MSE
Apple	11.18
Facebook	17.37
LinkedIn	36.96
Amazon	671.68
Google	397.12
Microsoft	3.19
Alibaba	3.35

Table 3: MSE for HMM with Closing Price Difference

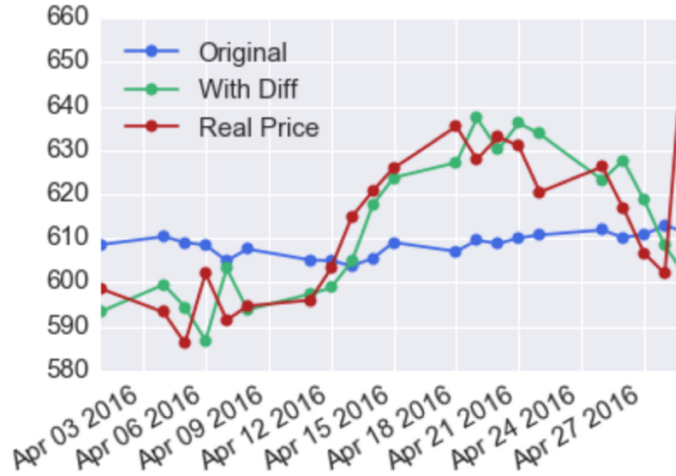


Figure 9: Forecasting of Amazon Stock Price without Closing Price Difference

## Investment Strategy

Using MSE as the evaluation of the forecasting performance might not be that reasonable. The most important goal is to transfer the results of forecasting to profits.

Assume a naive investment strategy used to evaluate whether the prediction is accurate and meaningful. Suppose the current day's closed price of one stock and the prediction of its tomorrow's closed price are

known. For example, if one stock's price is projected to increase 3% on the next day, 3 shares of this stock will be purchased. If it is projected to decrease 2%, 2 shares will be sold out. Calculate the daily return with the real closed price. Finally, calculate the annualized returns for these 20 days. The result is shown in the table.

	Revenue
Microsoft	-9.56
Amazon	39.11
Apple	17.64
Facebook	1.68
Google	-8.04
LinkedIn	-24948.39

Table 4: The Investment Revenue

For HMM model, the investment will turn into positive returns on 3 stocks and negative returns on 2 stocks. The profits from these 3 stocks are notably higher than the loss from those 2 stocks. Therefore, if a good portfolio is designed, profits can be ensured from the prediction, which indicates that the prediction is informative and useful.

For time series model, sometimes the investment will earn a large profit and sometimes may make a great loss. This prediction is not so satisfying because of great risk and the expected returns are less than that of HMM model.

The reason may be because HMM model takes more variables into account than time series, such as open price, high price, low price and price difference. It has been proved by Neural Network that closed price is indeed influenced by these 4 factors. Moreover, HMM model predicts one day in one prediction while time series predicts 20 days in one prediction. Long time prediction may lead to more discrepancy.

## Neural Network

Before training the models, the data is normalized before being input to the ANN. We have guaranteed the input vectors of the training data to have zero-mean and unit variance, while considering the activation function may be Unipolar sigmoid the target values are normalized between 0 and 1. We could also normalized them to between -1 and 1 and 0 and  $\frac{1}{\sqrt{2\pi\sigma}}$  when the activation function is Bipolar sigmoid, Tan hyperbolic or RBF, but these goes beyond the scope of this report and we stick to the Unipolar sigmoid activation function for "simplicity". Similarly, the test data vector is also scaled by the same factors with which the training data was normalized. The output values from the ANN for this test vector are also scaled back with the same factor as the target values for the training data and plotted together with the real data for comparison.

In spite of all the features mentioned for neural networks, building a neural network for prediction is somehow complicated. In order to have a satisfactory performance, we must consider some crucial factors in designing of such a prediction model. One of the main factors is the network structure including number of layers, neurons, and the connections. In the two proposed models, we followed the empirical rules when choosing the number of layer and neurons and test the model based on error.

In the first model, the inputs to the neural networks are the lowest, the highest and the open values in the d previous days. In the second model, the same variables are taken from the past few d days such that the data can be seen as longitudinal.

1. The first Neural Network model takes the data from a fixed period of time(January and February). The estimated network is shown as below.



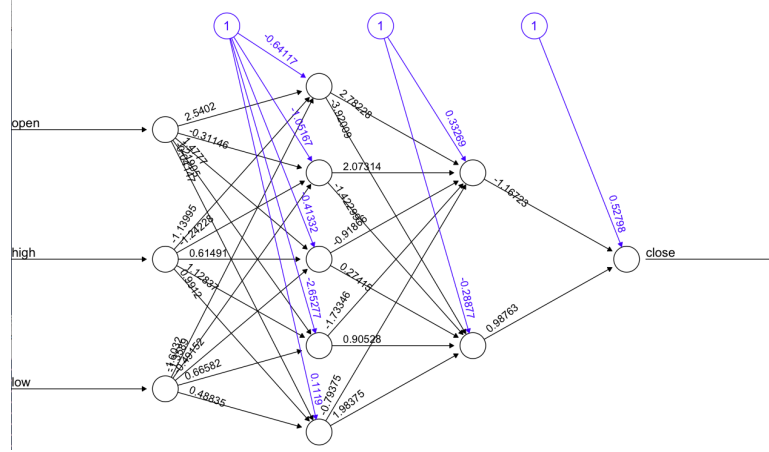


Figure 10: The First Neural Network Model

With this training model, we put in the test data with the same variables for prediction(March and April). The true data is labeled as the black line and the red line is the predicted values. The pattern can be caught very well, but there may be more improvement.

### Prediction based on Test data(fixed period)

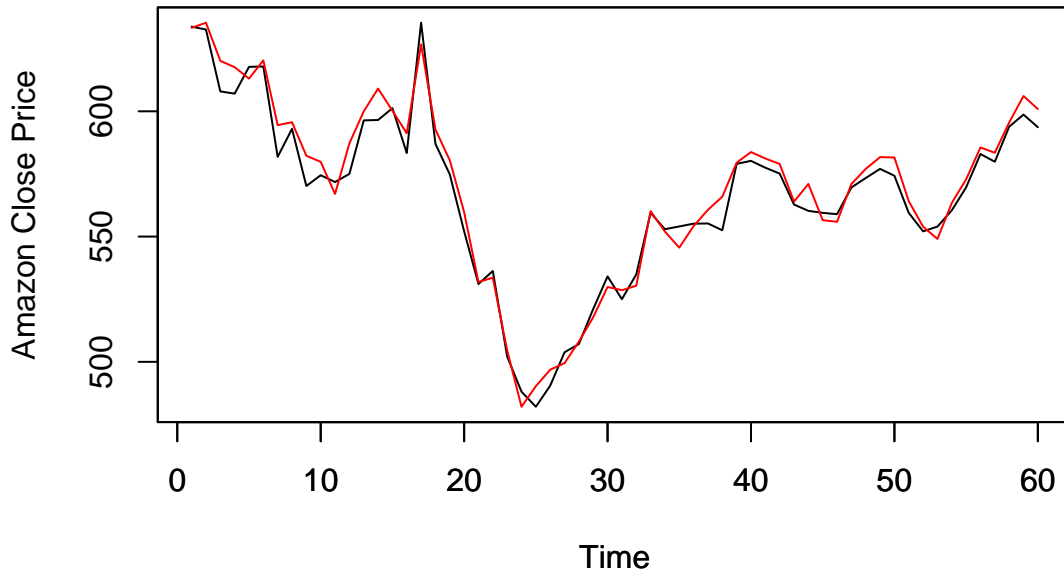


Figure 11: Prediction based on Test data(fixed period)

Next, we try to analyze in terms of longitudinal data to improve prediction. The input will be all the variables in the past 3 periods of time (60 days). The best trained network is show as below.

This time the red and black lines are almost overlapping. Thus by taking into account the data across a relatively long period of time, the prediction accuracy has been improved to a great extent.

Neural networks (NN) can assist in finding a relationship, or mapping, between inputs (Open, Low and High price) and an output (Closing price). Neural Networks have the advantage that can approximate any nonlinear functions without any apriori information about the properties of the data series. Comparing with traditional methods such as linear regression and logistic regression which are model based, Neural Networks

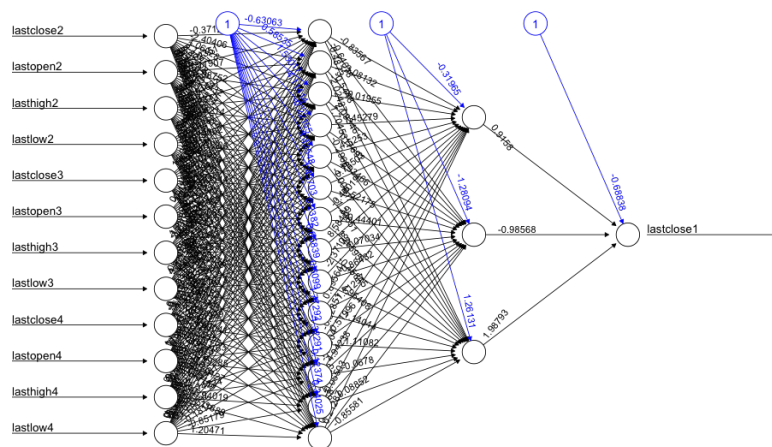


Figure 12: The Second Neural Network Model

### Prediction based on past data(longitudinal)

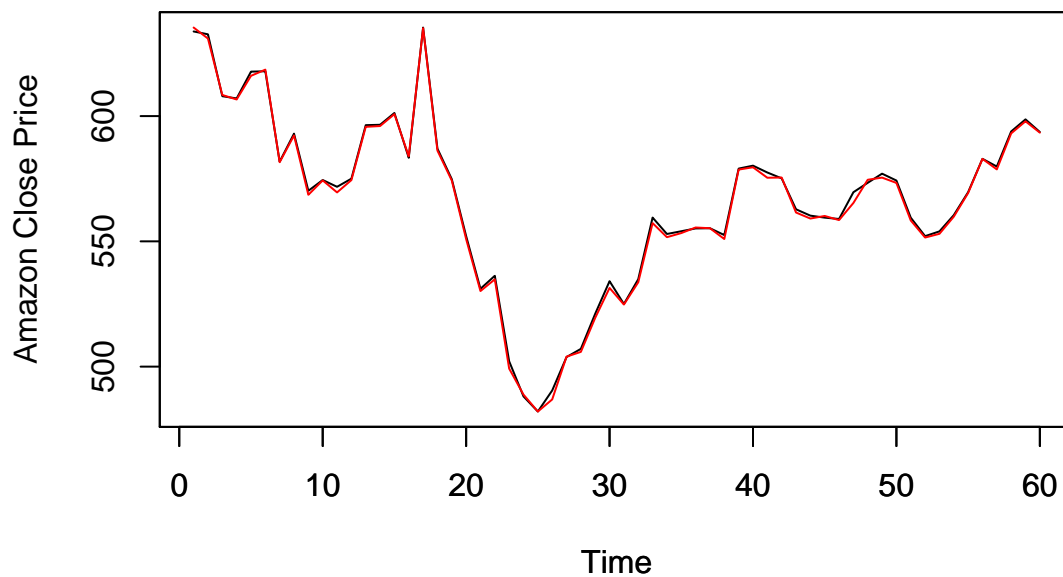


Figure 13: Prediction based on past data(longitudinal)

are self-adjusting methods based on training data, so they have the ability to solve the problem with a little knowledge about its model and without constraining the prediction model by adding any extra assumptions. Besides, neural networks can find the relationship between the input and output of the system even if this relationship might be very complicated because they are general function approximators. Consequently, neural networks are well applied to the problems in which extracting the relationships among data is really difficult but on the other hand there exists a large enough training data sets. It should be mentioned that, although sometimes the rules or patterns that we are looking for might not be easily found or the data could be corrupted due to the process or measurement noise of the system, it is still believed that the inductive learning or data driven methods are the best way to deal with real world prediction problems. Furthermore, Neural Networks have generalization ability. After training, they can recognize the new patterns even if they haven't been in training set. Since in most of the pattern recognition problems predicting future events (unseen data) is based on previous data (training set), the application of neural networks would be very beneficial.

The application of neural networks in prediction problems is very promising due to some of their special characteristics. But we still need to be clear that the model is mostly strong in explaining about the prediction rather than forecasting. By implementing neural networks model in this report, it is further proved that including more variables other than merely the Closing price can be helpful. Also, it also shows a strong potential of making forecasting if we are able to get the required variables. Though this may require a complete reconstruction of the data frames and complexity of computation, it can be a promising model worth optimizing.

## Conlusion

## Reference

- [1] Hassan, Md Rafiul, and Baikunth Nath. "Stock market forecasting using hidden Markov model: a new approach." Intelligent Systems Design and Applications, 2005. ISDA'05. Proceedings. 5th International Conference on. IEEE, 2005.
- [2] [https://en.wikipedia.org/wiki/Hidden\\_Markov\\_model](https://en.wikipedia.org/wiki/Hidden_Markov_model)

## Attachment

Time series forecast plots:

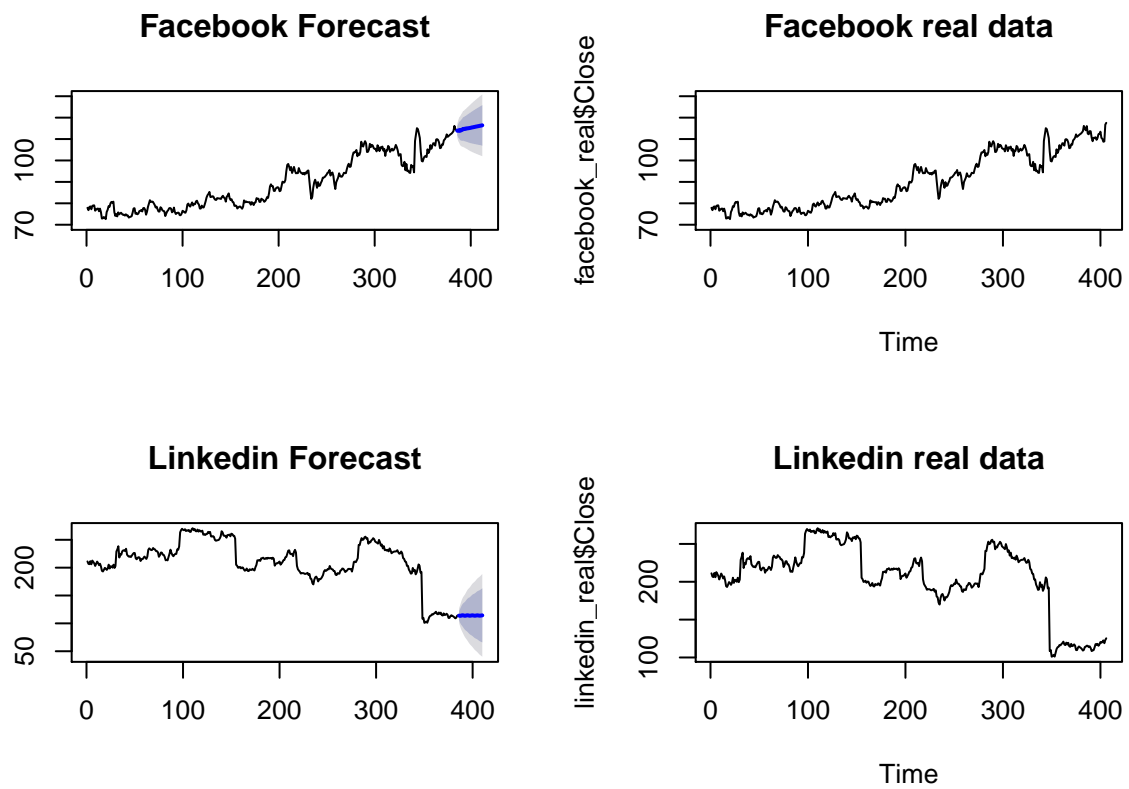


Figure 14:

Code

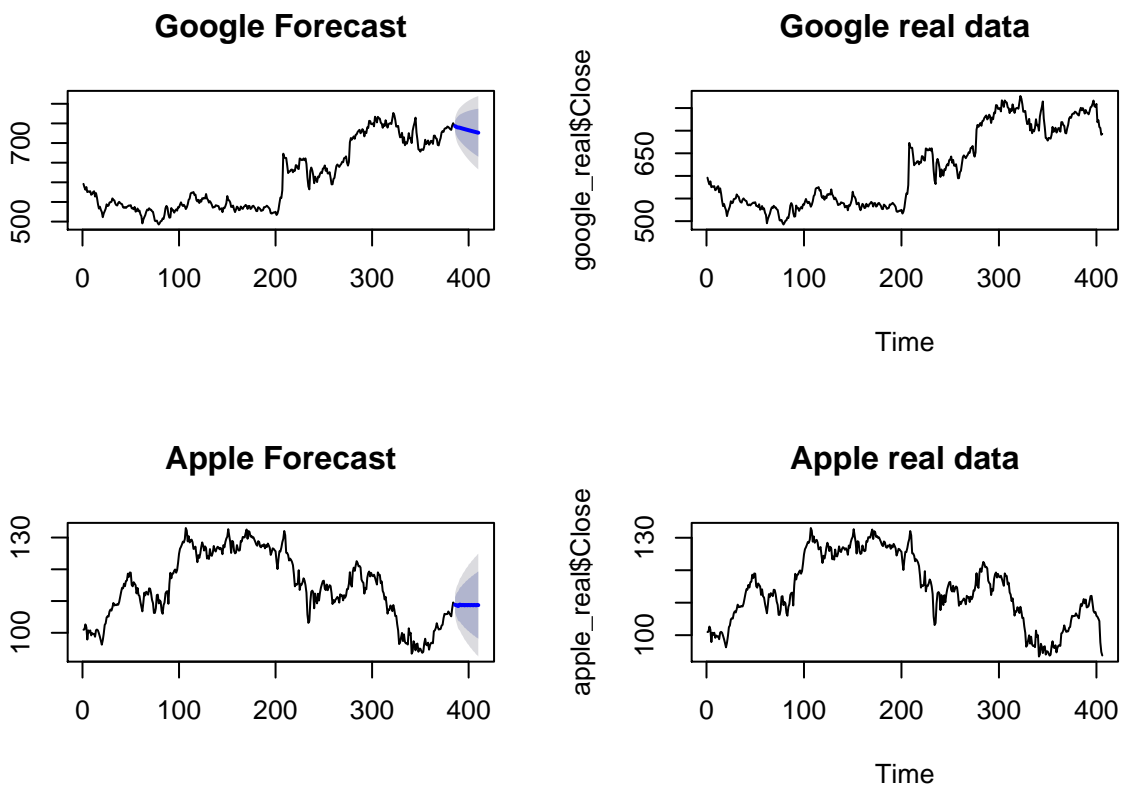


Figure 15:

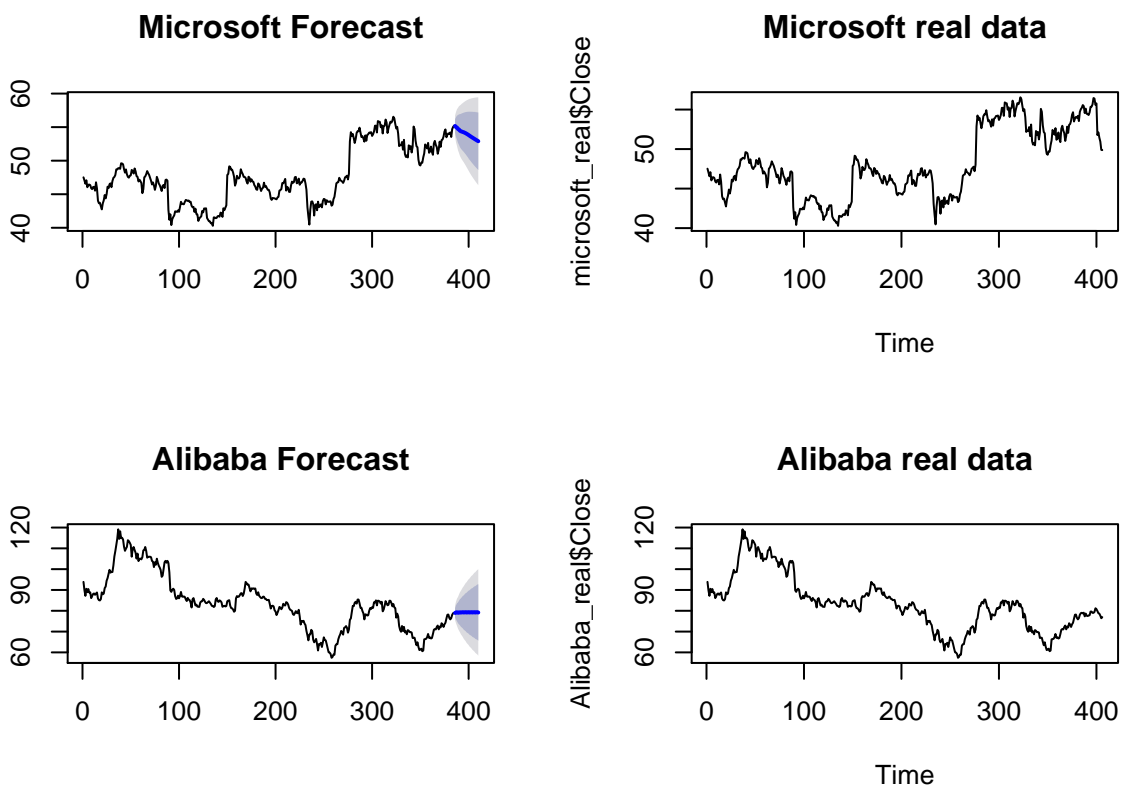


Figure 16: