Improving Artificial Neural Network based Stock Forecasting using Fourier De-Noising and Hodrick-Prescott Filter

Aditya Mitra
School of Electrical and Electronic Engineering
Nanyang Technological University
Singapore 639798
Email: aditya3@e.ntu.edu.sg

Lipo Wang
School of Electrical and Electronic Engineering
Nanyang Technological University
Singapore 639798
Email: elpwang@ntu.edu.sg

***Abstract*—Accuracy in financial forecasting is a key determinant of profits in the financial markets. This paper proposes improvements to existing Artificial Neural Network based forecasting approaches using de-noising in frequency domain and the Hodrick-Prescott Filter. Traditionally used technical indicators are replaced with open, close, high, and low prices only. Forecasts achieved via these improvements are seen to outperform existing results. 8 stocks from the Dow Jones Industrial Average, were considered over a period of 6 years,between 2000 and 2005. The best and worst directional accuracy achieved were 90% and 79% respectively.**

Keywords*—*ANN, Back Propagation, Neural, Stock, Hodrick-Prescott Filter, Fourier, Denoise.

I. Introduction

Forecasting future prices of stocks has been the fundamental question in the minds of fund managers, investment banks and retail market participants. Despite the Efficient Market Hypothesis (EMH) [1] which claims that stock prices follow a random walk [2], there has been little consensus on the absolute validity of the hypothesis in real-world markets [3]. Empirical evidence from USA [4], Japan [5] and Europe [6] also indicate that markets only conform to the weak form of EMH. The argument against EMH has been that most investment decisions and financial trading models in the real world are built on historical data. The above results have offered encouragement for the research and development of trading systems which can accurately predict future stock prices in the short run. In the past decade, there has been a manifold increase in the use of computational intelligence methods for forecasting of stocks prices, indices and FX exchange rates. Azoff [7], Gately [8] and, Refenez and Bentz [9] discuss the methodology and efficacy of using Artificial Neural Networks(ANN) for forecasting stock prices. Slightly advanced methods including Fuzzy Methods [10] and Genetic Algorithms [11] have also been used in conjunction with neural networks (NN) for this purpose. Research has also been conducted on the use of feature selection methods [12] for selecting explanatory variables prior to neural network training. This research aims to implement and improve existing methods for forecasting stock prices using back propagation ANN. We select 8 stocks from the Dow Jones Industrial Average and test a system to predict closing price at t+1 using historical closing prices and a combination of 9 technical indicators [13]. Improvements are then suggested in three parts: first discarding technical indicators as inputs features and replacin them with price related features, second using a noise filter after implementing Fast Fourier Transform on the time series, and third using the Hodrick-Prescott filter to forecast the trend and cycle components of the time series separately. Finally, this paper is concluded by listing important findings, practical implications and suggesting areas for future research.

A. Back Propagation Neural Networks and the Levenberg Marquardt Algorithm

Neural networks are bio-inspired algorithms which are modelled after the human nervous system. An ANN system associates a set of input neurons with a set of output neurons making use of one or more hidden layers [14]. The most commonly used neural networks consist of one input layer, one hidden layer and one output layer. Such artificial neural networks have been shown to be powerful in approximating arbitrary functions with nonlinear characteristics [15].

The neural network used in this paper uses a back-propagation (BP) learning algorithm. BP based neural networks with a single hidden layer have been shown to be capable of providing accurate approximations of any continuous function [16]. Mathematical descriptions for the back-propagation algorithm can be found in [17]. Each hidden and output neuron processes its inputs by multiplying each input by its weight, summing the product and then passing the sum through a transfer function to produce a result [18]. The first phase of the NN learning process is the feed-forward stage, which calculates the network outputs in response to the corresponding inputs. In this phase, the connection weights are fixed. The second, Back Propagation phase modifies the connection weights. In this phase, an error signal will be obtained by comparing the network outputs with the target values. Then, the error signal spreads backward in the network layers and the weights are adjusted. [19]. This process is repeated for a specified number of iterations,or untill a specified maximum error is reached. The Levenberg Marquardt Algorithm (LM) is used to approach second order training speed without the computation of a Hessian Matrix [20]. The objective of training is to alter weights and biases in order to reduce global error E defined as : [21]

$$\frac{1}{p} \quad \sum_{p=1]}^{p} Ep, \tag{1}$$

where N is total number of training instances, and Ep is the error for training pattern p. Ep is given by the following formula:

$$Ep \quad = \quad \frac{1}{2} \quad \sum_{i=1}^{N} (0_i - t_i)^2 \, , \tag{2}$$

where N is total number of output neurons, o_i is the neuron predicted value at node i and t_i is target output at node i. When the performance function is given by a sum of squares function, the Hessian can be approximated using the Jacobian matrix. The LM Algorithm uses this approximation to implement a variation of Newton's Method. The LM Optimization technique is found to be more powerful than traditional gradient descent approaches [22] and is consequently used as our algorithm of choice.

B. Fast Fourier Transform and Low Pass Filters

A Fast Fourier transform (FFT) is an algorithm to compute the Discrete Fourier Transform (DFT) and it's inverse. Fourier analysis converts time (or space) to frequency (or wavenumber) and vice versa; an FFT rapidly computes such transformations by factorizing the DFT matrix into a product of sparse (mostly zero) factors [23]. [24]. A low-pass filter is a filter that passes signals with a frequency lower than a certain cutoff frequency and attenuates signals with frequencies higher than the cut-off frequency [25]. A first order filter reduces amplitude by half, and consequently reduces power by 6db every time the frequency doubles.

C. Hodrick Prescott Filter

The Hodrick–Prescott (HP) filter is a mathematical tool used to remove the cyclical component of a time series from raw data. It is used to obtain a smoothed-curve representation of a time series, one that is more sensitive to long-term than to short-term fluctuations. The adjustment of the sensitivity of the trend to short-term fluctuations is achieved by modifying a multiplier λ . Any series φ_t is composed of a trend component τ and a cyclical component γ such that:

$$\varphi_t \quad = \quad \tau_t + c_t + \epsilon_t. \tag{3}$$

Given an adequately chosen, positive value of λ , there is a trend component that will solve [26]

$$min\left(\sum_{t=1}^{T}(\varphi_t - \tau_t)^2 + \lambda\sum_{t=2}^{T-1}\left[(\tau_{t+1} - \tau_t) - (\tau_t - \tau_{t-1})\right]^2\right). \tag{4}$$

The value for the smoothing factor λ is varied according to frequency of data collection, and the appropriate values used in this paper are obtained from [27].

II. Data Selection and pre-processing

A. Sourcing and Normalizing

We selected 8 stocks from the Dow Jones Index to represent a diversity of industries and sizes. The stock along with their tickers are shown in Table I. Open, high, low and close price time series with a daily frequency, were obtained for each stock from Yahoo Finance [28]. We used 6 years worth of daily data starting on Jan 1st 2000 and ending on 31 Dec 2005, giving us 1502 data points.

Table I. Stocks selected for experiments

STOCK	TICKER
Microsoft	MSFT
Coke	KO
Exxon Mobil	XOM
International Business Machine	IBM
Proctor and Gamble	PG
E I Du Pont De Nemours And Co	DD
JP Morgan Chase	JPM
Johnson and Johnson	JNJ

The data was normalized before being used to train the neural network. If such a transformation is not employed, the value of the variable could be too large for the network to process, and might give skewed results [29]. We normalize each data point between (0, 1) using the following formula:

$$\chi_{norm} \quad = \quad \frac{\chi - \chi_{min}}{\chi_{max} - \chi_{min}}, \tag{5}$$

where, χ_{norm} is the normalized value,
χ is the actual data value,
χ_{min} is the smallest value within dataset
χ_{max} is the largest value within dataset

B. Calculating Technical Indicators

The methodology adopted in [13] used 9 technical indicators along with the closing price as input features. The indicators are listed in Table II along with their formulas.

Table II. Formulas for Technical Indicators

INDICATOR	FORMULA
10 Day Simple Moving Average (SMA)	$\frac{C_t+C_{t-1}+\ldots+C_{t-10}}{10}$
10 Day Weighted Moving Average	$\frac{nC_t+(n-1)C_{t-1}+\ldots+C_{t-10}}{(n+(n-1)+\ldots+1)}$
Momentum	$C_t - C_{t-n}$
Stochastic K%	$\frac{C_t-LL_{t-n}}{HH_{t-n}-LL_{t-n}} X \ 100$
Sochastic D%	$\frac{\sum_{i=0}^{n-1} K_{t-1}\%}{n}$
Relative Strength Index (RSI)	RS= $\frac{EMA(U,n)}{EMA(D,n)}$
Moving Average Convergence Divergence (MACD)	MACD:12 Day EMA - 26 Day EMA Signal Line: 9 day EMA of MACD Divergence: MACD - Signal
Larry William's R%	$\frac{H_n-C_t}{H_n-L_n} X \ 100$
Accumulation/ Distribution Oscillator	$\frac{H_t-C_{t-1}}{H_t-L_t}$
Commodity Channel Index	$\frac{M_t-SM_t}{0.015D_t}$

C_t is the closing price, L_t is the low price, H_t is the high price at time t. EMA is the exponential moving average shown in [30]. HH_t and LL_t are the highest and lowest price in the last t days respectively.

$M_t = H_t + L_t + C_t/3; SM_t = \left(\sum_{i-1}^{n} M_{t=i+1}\right)/n; D_t = \left(\sum_{i-1}^{n} |M_{t=i+1} - SM_t|\right)$. U and D represent upward and downward periods. Up periods represent days when close is higher than the previous close with $U = C_t - C_{t-1}$ and D=0. Down periods represent days when close is lower than the previous close with $D = C_{t-1} - C_t$ and U = 0. [13]

III. Neural Network Specifications

A. Formation of the ANN

The ANN used in this paper had one input layer, one hidden layer and one output layer. The choice of one hidden layer was motivated by literature which suggests that one layer with adequate neurons is able to forecast any arbitrary function [15]. The training of the neural network was carried on for 1000 epochs. The number of hidden layer neurons, feedback delays in input layer, momentum, and learning rate were determined experimentally using different combinations of values for each of the parameters. The transfer function used in the hidden layer was the tan sigmoidal function, and a pure linear function was used in the output layer. As described previously, we used the Levenberg-Marquardt Algorithm for training the network.

B. Segmentation of data

The data was segmented into a training and testing set. 60% of the data is reserved for training and 40% for testing the trained network. The division of data into these sets was performed randomly to avoid recency/cyclical biases.

C. Performance Measures

The neural network used Mean Squared Error (MSE) as a performance measure and stopping criterion. If $\hat{Y}_t$ is a vector of n predictions, and Y the vector of true values, then MSE is given by:

$$MSE = \frac{1}{n} \sum_{i=1}^{n} \left(\hat{Y}_t - Y_i\right)^2. \quad (6)$$

The other accuracy error measures used in the testing phase were Directional Accuracy, Root Mean Square Error, and Mean Average Percentage Error. The directional accuracy simply represents the percentage of the time the trained network is able to accurately predict the direction of the change in closing price. The formula for the other measures are as follows:

$$RMSE = \sqrt{MSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n} \left(\hat{Y}_t - Y_i\right)^2}, \quad (7)$$

$$MAPE = \frac{1}{n} \sum_{i=1}^{n} \left|\frac{\hat{Y}_t - Y_i}{Y_t}\right|. \quad (8)$$

IV. Implementation and Results

A. Using Closing Price and 9 Technical Indicators

We first implemented the methodology in [13] as our base case. We also used this experiment to determine the optimum values for the variable parameters (hidden neurons, feedback delays, momentum and learning rate) for the network. These values were held constant in following experiments to observe the ceterus paribus impact of our attempted improvements. The data was prepared as described in section II, and the network was configured as described in section III. The results for the first case are summarized in Table III. The experimentally determined values for network parameters which gave the best results are summarized in Table IV.

Table III. Results using closing price and 9 Technical Indicators

Stock	Accuracy	RMSE	MAPE	MSE
MSFT	71.88%	4.26	3.87	18.14
Coke	57.20%	2.43	3.89	5.94
Exxon	75.41%	1.41	1.29	1.99
IBM	72.76%	3.39	1.95	11.47
PG	62.36%	2.7	2.02	7.27
DD	55.05%	3.45	5.77	11.93
JPM	64.81%	1.92	3.15	3.69
JNJ	66.50%	3.29	2.83	10.84

Table IV. Optimum values for NN parameters

Parameter	Value
LR	0.6
Momentum	0.4
Hidden Neurons	10
Feedback Delays	10

This NN configuration and combination of inputs was found to predict the closing price with accuracy varying between 55% and 75% with relatively low MSE and MAPE,indicating a reasonable potential for profit. A feedback delay of 10 days in the input layer was found to give the best results for forecasting price at $t + 1$.

B. Using only price indicators

The first modification considered was to exclusively use price information and discard the technical indicators. This was motivated by Dow Theory, which suggests that price information already reflects all possible factors which affect stock prices. The open, close, high and low prices were used as input features into the network, with all other specifications for the network carried forward from IV-A. Results from the experiment are summarized in Table V.

Table V. Results using only price inputs

Stock	Accuracy	RMSE	MAPE	MSE
MSFT	78.75%	1.29	1.07	1.66
Coke	79.10%	0.61	0.60	0.37
Exxon	82.23%	1.56	0.87	2.45
IBM	79.90%	3.56	3.56	12.69
PG	82.30%	2.65	0.69	7.06
DD	87.53%	3.40	0.86	11.53
JPM	80.70%	6.50	1.89	42.25
JNJ	80.22%	2.43	0.76	5.88

The results from IV-B demonstrated a significant increase in accuracy for all 8 stocks tested. Directional accuracy varied between 78% and 87% indicating a more significant profit potential from this strategy. MAPE also compressed for 7 out of 8 stocks, signalling the superior and more accurate performance of this method over the one described in IV-A.

C. *Using a low pass filter for de-noising*

The third experiment involved the use of a low pass filter to remove high frequency noise components from the price time series. As a result of turbulent market events, stock prices may move haphazardly on days with abnormal activity. Including these days in the training process may distort the ability of the NN to accurately predict future prices. This experiment used the same configuration of inputs as described in IV-A, but added an extra feature which represented the de-noised time series of the closing price. Wavelet de-noising has previously been implemented in [31], but this paper opts for a FFT based de-noising approach instead. The de-noised signal was obtained by first implementing the Fast Fourier Transform on the time series, and then removing high frequency noise elements using a First Order Filter with an experimentally determined cut-off frequency. The results for this experiment are provided in Table VI.

Table VI. RESULTS WITH ADDITION OF DE-NOISED SIGNAL

Stock	Accuracy	RMSE	MAPE	MSE
MSFT	82.37%	1.07	0.89	1.16
Coke	81.97%	0.52	0.6	0.28
Exxon	87%	1.27	0.69	1.63
IBM	80.50%	1.8	0.89	3.27
PG	82.30%	2.47	0.66	6.12
DD	86.12%	2.35	0.91	5.55
JPM	79.36%	2.032	1.23	4.13
JNJ	80.97%	1.89	0.72	3.59

The results from IV-C indicated improvements in accuracy for all stocks over the method used in IV-A. Directional accuracy was found to increase in 6 out of the 8 stocks as compared to method used in IV-B . MSE and MAPE compressed for all 8 stocks in comparison to both previous methods, indicating more accurate price predictions.

D. *Using a Hodrick-Prescott Filter*

The final experiment involved the use of a Hodrick Prescott Filter. Stock prices are thought to be composed of a long-term slow moving trend and a faster cycle component. The HP filter was used to split all four input time series i.e. open, close, high and low into their trend and cycle components. The trend and cycle component of the forecast were modelled separately using two neural networks, both using same specifications as employed previously. The first NN was trained with the trend components of the four features and was used to forecast the trend component of the price at time $t + 1$. The second NN repeated the process for the cycle component. The linearity of the HP filter allowed us to simply sum the two component forecasts to obtain our forecast close price at time $t + 1$. The testing phase remained unchanged after the forecast was obtained. The results for this experiment are summarized in Table VII.

The results indicated an absolute increase in directional accuracy over methods in IV-A and IV-B and an increase/no change for 6 out of the 8 stocks over method in IV-C. The RMSE, MAPE and MSE were found to significantly compress in comparison to all previous methods.

Table VII. RESULTS WITH HODRICK-PRESCOTT FILTER

Stock	Accuracy	RMSE	MAPE	MSE
MSFT	80.89%	0.95	0.97	0.91
Coke	84.64%	0.35	0.49	0.13
Exxon	80.97%	1.00	0.79	1.01
IBM	83.97%	1.13	0.74	1.28
PG	82.30%	1.73	0.56	3.001
DD	90.40%	0.31	0.48	0.09
JPM	80.03%	1.39	1.04	1.94
JNJ	81.77%	0.91	0.61	0.82

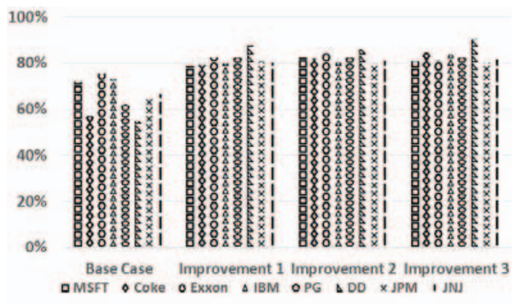

Figure 1. Comparison of Directional Accuracy

E. *Comparison of the four methods*

The experiment in section IV-D was found to produce the best combination of high directional accuracy and very tight MSE and MAPE measures, and was consequently considered the most profitable prediction model. All experimental improvements (sections IV-B to IV-D) were found to outperform our base case in section IV-A. Graphical comparison of the error measures for each of the methods are summarized in Figures 1, 2 and 3.

V. CONCLUSION

Prediction of stock prices is essential for the development and implementation of profitable trading strategies. The ability of machine learning techniques to model arbitrary non-linear functions has made them an extremely effective financial forecasting tool.This study attempted to improve existing Neural Network based forecasting models by altering choice of inputs features, employing a frequency domain noise filter and using the Hodrick Prescott Filter. A base case to forecast daily closing prices using ten days worth of previous closing prices and technical indicator data was first implemented on data from 2000 to 2005 for 8 stocks from The Dow Jones Industrial Average. Three distinct improvements were suggested and

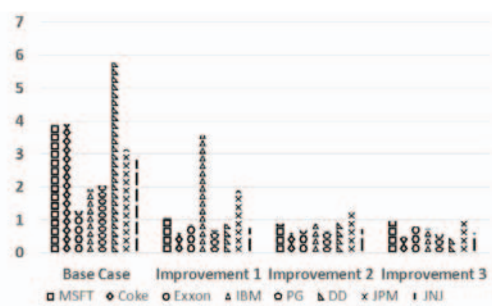

Figure 2. Comparison of MAPE

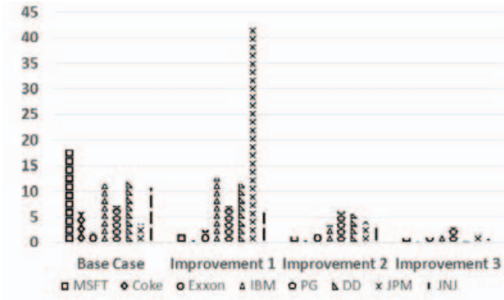

Figure 3. Comparison of MSE

implemented to improve the accuracy of the prediction. Based on the experimental results obtained, some important conclusions can be drawn. Firstly, it must be noted that the method found in literature was useful in forecasting both direction and actual price of the stocks considered, with reasonable accuracy. Artificial Neural Networks were hence found to be useful prediction tools in this field. The first approach for improvement involved using only price based indicators as inputs to the neural network. This approach was found to improve both the directional and value accuracy of the forecast. An average accuracy of 81.34% and average MSE of 10.48 was found to be significantly higher than the base case. The second and third improvement involved the use of a First Order Filter for de-noising the stock data, and the use of the Hodrick Prescott Filter to forecast the trend and cycle components of price separately. To the author's best knowledge, the average accuracy of 82.22% for the second improvement and 83.11% for the third, outperformed similar studies in literature. An average MSE of 1.48 for the third improvement was the lowest error achieved amongst all experiments conducted. However, our results could be improved by optimizing our network architecture through the use of Genetic Algorithms. An improvement can also be made by finding the best possible combination of input features, which may include macro and other fundamental data. Feature selection methods employed before training the neural network could help increase the accuracy of our results. Use of multi-step ahead prediction systems [32] in conjunction with our improvements could also help apply our work to a real world trading environment.

REFERENCES

[1] B. G. Malkiel, *"A Random Walk Down Wall Street: Including a Life-cycle Guide to Personal Investing"*. WW Norton & Company, 1999.

[2] E. F. Fama, "Efficient capital markets: A review of theory and empirical work," *The Journal of Finance*, vol. 25, no. 2, pp. 383–417, 1970.

[3] G. Tsibouris and M. Zeidenberg, "Testing the efficient markets hypothesis with gradient descent algorithms," in *Neural Networks in the Capital Markets*, vol. 8, pp. 127–136, Wiley: Chichester, 1995.

[4] J. Jarrett and E. Kyper, "Evidence on the seasonality of stock market prices of firms traded on organized markets," *Applied Economics Letters*, vol. 12, no. 9, pp. 537–543, 2005.

[5] J. S. Ang and R. A. Pohlman, "A note on the price behavior of far eastern stocks," *Journal of International Business Studies*, pp. 103–107, 1978.

[6] B. H. Solnik, "Note on the validity of the random walk for european stock prices," *The Journal of Finance*, vol. 28, no. 5, pp. 1151–1159, 1973.

[7] N. Meade, "Neural network time series forecasting of financial markets," Elsevier, 1995.

[8] Gately, Edward, *Neural Networks for Financial Forecasting*. Wiley, 1995.

[9] A.-P. Refenes, A. N. Burgess, and Y. Bentz, "Neural networks in financial engineering: a study in methodology," *IEEE Transactions on Neural Networks*, vol. 8, no. 6, pp. 1222–1267, 1997.

[10] D. Chorafas *et al.*, "Chaos theory in the financial markets: Applying fractals, fuzzy logic, genetic algorithms, swarm simulation & the monte carlo method to manage markets," *Probus*, 1994.

[11] W. Shen and M. Xing, "Stock index forecast with back propagation neural network optimized by genetic algorithm," in *Second International Conference on Information and Computing Science*, vol. 2, pp. 376–379, IEEE, 2009.

[12] Yuqinq He, Kamaladdin Fataliyev, and Lipo Wang, "Feature selection for stock market analysis," in *The 20th International Conference on Neural Information Processing (ICONIP2013)*, Daegu, Korea, 3-10 November 2013, Invited Paper, Part II, LNCS 8227, pp. 737–744, 2013.

[13] Y. Kara, M. A. Boyacioglu, and O. K. Baykan, "Predicting direction of stock price index movement using artificial neural networks and support vector machines: The sample of the istanbul stock exchange," *Expert Systems with Applications*, vol. 38, no. 5, pp. 5311–5319, 2011.

[14] R. Hecht-Nielsen, "Theory of the backpropagation neural network," in *International Joint Conference on Neural Networks*, pp. 593–605, IEEE, 1989.

[15] J. C. Principe, N. R. Euliano, and W. C. Lefebvre, *Neural and Adaptive Systems: Fundamentals Through Simulations*. Wiley, 2000.

[16] K. Hornik, "Approximation capabilities of multilayer feedforward networks," *Neural Networks*, vol. 4, no. 2, pp. 251–257, 1991.

[17] R. C. Eberhart, *Neural Network PC Tools: A Practical Guide*. Academic Press, 2014.

[18] A. Goh, "Back-propagation neural networks for modeling complex systems," *Artificial Intelligence in Engineering*, vol. 9, no. 3, pp. 143–151, 1995.

[19] Farideh Fazayeli, Lipo Wang, and Wen Liu, "Back-propagation with chaos," in *Proc. 2008 IEEE International Conference on Neural Networks and Signal Processing (ICNNSP2008)*, Zhenjiang, China, June 7-11, pp.5-8, 2008.

[20] J. J. More, "The levenberg-marquardt algorithm: Implementation and theory," in *Numerical Analysis*, pp. 105–116, Springer, 1978.

[21] O. KISI and E. Uncuoglu, "Comparison of three back-propagation training algorithms for two case studies," *Indian Journal of Engineering & Materials Sciences*, vol. 12, no. 5, pp. 434–442, 2005.

[22] M. T. Hagan and M. B. Menhaj, "Training feedforward networks with the marquardt algorithm," *, IEEE Transactions on Neural Networks*, vol. 5, no. 6, pp. 989–993, 1994.

[23] C. Van Loan, *"Computational Frameworks for the Fast Fourier Transform"*, vol. 10. Society for Industrial and Applied Mathematics, 1992.

[24] http://en.wikipedia.org/wiki/Fourier_transform.

[25] http://en.wikipedia.org/wiki/Low-pass_filter.

[26] R. J. Hodrick and E. C. Prescott, "Postwar us business cycles: an empirical investigation," *Journal of Money, Credit, and Banking*, pp. 1–16, 1997.

[27] M. Ravn and H. Uhlig, "On adjusting the hodrick-prescott filter for the frequency of observations," *Review of Economics and Statistics*, vol. 84, no. 2, pp. 37l–76, 2002.

[28] http://finance.yahoo.com.

[29] JingTao Yao and Chew Lim Tan, "Guidelines for financial forecasting with neural networks," in *International Conference of Neural Information Processing*, 2001.

[30] http://www.itl.nist.gov/div898/handbook/pmc/section4/pmc431.htm.

[31] Lipo Wang and Shekhar Gupta, "Neural networks and wavelet de-noising for stock trading and prediction," in *Time Series Analysis* (W. Pedrycz and S. M. Chen, eds.), Springer, 2012.

[32] Guanqun Dong, Kamaladdin Fataliyev, Lipo Wang, "One-Step and Multi-Step Ahead Stock Prediction Using Backpropagation Neural Networks," in *9th International Conference on Information, Communications and Signal Processing (ICICS 2013)*, 10 - 13 December 2013, Tainan.