One-Step and Multi-Step Ahead Stock Prediction Using Backpropagation Neural Networks

Guanqun Dong, Kamaladdin Fataliyev, Lipo Wang
School of Electrical and Electronic Engineering
Nanyang Technological University
Singapore

***Abstract*— Forecasting stock price with traditional time series methods has proven to be difficult. An artificial neural network is probably more suitable for this task, since no assumption of a mathematical model has to be made prior to the forecasting process. Furthermore, a neural network has the ability to extract the main influential factors from large sets of data, which is often required for a successful stock prediction task. In this paper, we explore one-step ahead and multi-step ahead predictions and compare with previous work.**

Keywords- Stock predicting; Neural Networks; Levenberg-Marquardt backpropagation; One-step and Multi-step ahead prediction

I. Introduction

Forecasting the stock market is not a simple task, mainly because of the random-walk behavior of the stock price movement in short time period. Several techniques are being used for stock forecasting. In last two decades, neural networks (NNs) have become one of these techniques. The behavior of stock price movement is highly complex. It is nearly impossible to model with a pure mathematical function. Moreover, a large set of related factors is required to explain a specific stock. These two factors are among the most important motivations for the neural network approach in stock prediction.

This paper begins with a brief review of neural networks and the back propagation (BP) learning algorithm in Chapter 2. Subsequently in Chapter 3.1, an improved one-step ahead forecasting system is proposed and compared to the original forecasting strategy. Then, in Chapter 3.2, a simple recursive algorithm for multi-step ahead prediction is discussed and an improved multi-step forecasting system is proposed. Their prediction results are compared. Lastly, in Chapter 4, we draw our conclusions and discuss future work.

II. A BRIEF INTRODUCTION TO NEURAL NETWORKS

Neural networks can perform tasks such as pattern recognition and classification. There are some important features which make NNs capable for a forecasting problem. The first feature is its data-driven trait which only requires few priori assumptions. NNs can learn from training data and capture the logic or relationship even unknown to us. Secondly, NNs can be used to generalize. After the training process, the neural network has updated its parameters (weights and bias) and can be used to make predictions on unknown data. Thirdly, NNs are universal functional approximators. It has been proved that a two-layer neural network with enough hidden neurons can approximate any continuous function with a satisfactory accuracy [3]. Lastly, NNs are nonlinear. Many real systems and forecasting models are nonlinear. All these features make NNs very useful in various problems, especially in forecasting.

A feed-forward neural network is a network which does not include a directed cycle between its neurons. It is quite different from a recurrent neural network which typically consists of a directed cycle structure.

The feed-forward neural network is the simplest neural network where the information transmits in only one direction, i.e., from the input neurons to the output neurons. The information may pass through some hidden layers, but there are no loops in this type of neural networks.

The Backpropagation (BP) learning algorithm is one of the most important neural network learning methods. The BP algorithm has been widely used in training feedforward multilayer neural networks. Levenberg-Marquardt algorithm is one such BP algorithms. In the Levenberg-Marquardt algorithm, the step size has to be kept small to ensure the validity of the approximation.

III. One-step ahead and multi-step ahead prediction

A. *One-Step Ahead Prediction*

In this section, a BP feed-forward neural network is applied to predict the weekly closing value of Shanghai Stock Exchange Composite Index (SHCOMP) from 2nd Nov 2007 to 11th July 2008. The prediction results are compared with the results obtained by Phua et al. [4].

1) Data Normalization: This process helps to normalize the minimum and maximum values of each data series to a boundary. In our case, the boundary is set to be [-1, 1]. The mathematical definition is given in eq. 1:

$$y = (y_{max} - y_{min}) * (x - x_{min}) / (x_{max} - x_{min}) + y_{min} \qquad (1)$$

2) Input Processing: For inputs, we apply the log-return. The definition is given eq. 2. The rationale is that we care more about the movement of the index value, whether it is going up and going down and the how much the movement is in percentage, rather than the absolute value of the movement [6].

$$\text{INPUT} = \log(Y(t) / Y(t - 1)) \quad (2)$$

3) Neural Network Topology: There are infinite ways to construct a neural network. Neuro-dynamics and architecture are two terminologies used to describe the way how a neural network is constructed. The combinations of these two define the topology of a neural network. Neuro-dynamics describe the properties of the basic cell, an individual neuron, such as its transfer function and how the inputs are connected [7]. The architecture of a neural network defines its geometrical structure, for example, the number of neurons in every layer. We have to consider the following factors in designing a proper neural network:

- The number of inputs: In our case, there are 5 inputs and some external factors (i.e., monthly social retail goods, price index, added value of industry, etc.), which we combine into one input. All in all, the inputs are Closing value and Opening value of Shanghai Stock Exchange Composite Index, Highest and Lowest value of Shanghai Stock Exchange Composite Index, Volume, and Other Factors.
- The number of hidden layers: Theoretically, a neural network that has one hidden layer and sufficient number of hidden neurons is capable of generalizing any continuous function [8]. If we increase the number of hidden layers, the computation time will increase as well as the danger of over fitting [9]. Over fitting, in simple words, is to memorize every point rather than being able to learn the general patterns. In our paper, we choose the number of hidden layers to be one.
- The number of neurons in each layer: In our paper, the number of hidden neurons is between 3 and 18.
- The number of outputs: In our case, the number is one, i.e., the output is the predicted closing value of the Shanghai composite index.

4) Training Procedure: The process of training a neural network is to let the neural network learn patterns in the training data, by iteratively presenting the NN with examples of correct known samples. The goal of this process is to find the set of weights and bias that result in a good minimum of the error function. There are four important factors in this process, i.e., training data preparation, learning algorithm, transfer function, and stopping criteria. In our case, Levenberg-Marquardt backpropagation has been applied.

- Training Data Preparation: We can divide the training data set into three categories, training, validation and testing. We use 70% of the data for training, 15% for validation, and 15% for testing. In our case, we use Shanghai composite index from 2002/11/04 to 2007/11/01 as our training data set.
- The Transfer Function: Transfer functions are mathematical functions that calculate the output of processing neuron based on the input. Klimasauskas [10] suggests that if the neural network is designed to learn average behavior, a sigmoid transfer function should be used. But if the learning process is about deviations from the average, the hyperbolic tangent function is the best. So in our case, we use hyperbolic tangent function as the transfer function:

$$\text{tansig}(n) = 2 / (1 + \exp(-2 * n)) - 1 \quad (3)$$

- The Stopping Criteria: There are mainly two types of stopping criteria, late stopping and early stopping [11]. Late stopping means that the neural network is trained until a certain error condition is reached. The problem for late stopping is over fitting. In order to obtain good generalization, pruning process should be involved in the training. The other stopping criterion is early stopping. It is a way to avoid over fitting but it is hard to know when to stop. In our case, we apply the late stopping.

5) Output error correction due to input outliers: For stock prediction, the outliers typically happen when the economic or political climate is changing rapidly. For example, big breaking news can cause a rapid drop to a particular stock. By selecting an appropriate error function, the impact of outliers can be reduced. We propose this process in our paper as one of the modifications compared to Phua et al. [4]. The error correction function is given below:

$$(1/3)\text{In} (\cosh (3 (o_\iota - \tau_\iota))) \quad (4)$$

In (4), o_i denotes the output from output neuron i and t_i is the corresponding target. This function is a smooth approximation of abs(o_i - t_i). It has such a property: when the difference between output predicting value and the real value is large, this function will produce a limited and reasonable error.

6) Performance Measurement and Comparison: Figure 1 shows the prediction results compared to actual results. Occasionally, our neural network successfully predicts some turning point of the trend, not just following the trend. But we have to admit that this successful prediction is not stable, since every time the neural network will be trained differently. Besides this, our predicting result for 2008.06.08 is around 3000, which is very close to the real value.

It is important to know how well the prediction is. There are several widely used performance measures. In this paper, “hit-rate” is used as a performance measure. The hit rate indicates how often the neural network gives a correct prediction in terms of the direction of price movement. For example, if the actual stock price movements for next five days are up-down-up-up-up and our prediction shows that the movements are up-down-down-down-up. Then in total, 3 out of 5 times the prediction is correct, so the hit rate in this case is 3/5 = 60%.

The following table shows a hit-rate comparison between Phua et al. and our one-step ahead prediction system.

TABLE I. COMPARISON IN HIT-RATE

	Phua et al	***Our System***
Correct prediction time	19 out of 34	22 out of 34
Hit Rate	55.8%	64.7%

The following table summarizes all the modifications and improvements made by us:

TABLE II. SUMMARY OF MODIFICATIONS AND IMPROVEMENTS

	Phua et al	***Our System***
Data Preparation	Input normalization	Input normalization and input data process
Number of hidden layers	2	1
Number of hidden neurons	20	5
Number of time-delays	1	5
Output error correction	No	Yes

B. *Multi-step Ahead Prediction for 000024.SZ*

According to our discussion, the one-step prediction result is relatively stable, the error is small, overall it is quite acceptable by achieving a hit rate more than 60%, compared to previous publications. But the one-step prediction does not help a lot in a real investment situation since it tends to follow the trend rather than foresee what is going to happen.

In a real investment or speculating situation, if we know the stock price will go up first then go down or vice versa, it would be a huge advantage in investment decision making. As a personal or institutional investor, this information will give you a niche to beat the market. The multi-step prediction may help us to realize this goal in some special cases.

Unlike one-step ahead prediction, multi-step prediction is more difficult [12], since it has to deal with more uncertainties, accumulation of errors and the accuracy can be reduced [13].

1) Data Preparation: In this case, we choose 000024.SZ as our target prediction stock. We have Closing price and Opening price of 000024.SZ, Highest and Lowest prices of 000024.SZ and Volume. Other factors are not considered in this case, since the prediction period is quite short and macro-economy only has a very limited impact on the fluctuation of stock price in a short period [14]. In addition, in order to reduce the impact of some big news or black swan, the 100-day period that we selected is a relatively stable period in terms of news and the change from fundamental perspective. The same process, inputs normalization and data processing, is done for this data set just as we did for one-step ahead prediction system's data preparation part.

2) Neural Network Topology: In this section, we summarize all the parameters being used in this multi-step neural network topology. For the first five factors, the selection criteria for a multi-step neural network is exactly same as a one-step ahead neural network, the only difference is that for a multi-step neural network, we have to use a close-loop structure to replace the open-loop structure after the training process has been done. The reason is simple. For a multi-step neural network, the prediction result of day T is deduced from the prediction results of day (T-1), day (T-2)…, day (T-n), rather than the actual data of day (T-1), day (T-2), day (T-n). So a close-loop structure must be adopted to realize this process.

TABLE III. MULTI-STEP NEURAL NETWORK TOPOLOGY PARAMETERS

Number of Inputs	5
Number of Time-delays	1:2:30 (it means we use day 1, day 3, . . . , day 30 to predict day 31. This parameter is determined by experiment.)
Number of hidden layers	1
Number of neurons in each layer	5
Number of outputs	1
Transfer function	Tansig
Learning algorithm	Levenberg-Marquardt backpropagation
Connection structure	Close-loop

3) Recursive Strategy for Multi-step Forecasting: A multi-step ahead time series prediction is to forecast the next H values [yN+1,…, yN+H] of a historical time series [y1,…,yN] which consists of N observations, where H > 1 denotes the forecasting horizon. The recursive strategy is one of the oldest and most intuitive forecasting methods for multi-step prediction problem [15]. In this method, a single model F is trained to perform a one-step ahead prediction first, such as

$$y_{t+1} = F(y_t, . . . ,y_{t-d+1}) + w. \tag{5}$$

In the equation above, t belongs to {d,…,N-1}, y_t is the actual closing price of day t, y_{t+1} is the predicted closing price.

When predicting H steps ahead, we first predict the first step by applying this model F. Secondly we use the value that we just produced as part of the input variables for predicting the next step by applying the same model F. We continue this process until we have predicted the entire horizon.

The recursive strategy may produce a low performance in multi-step forecasting task due to the noise present and the length of forecasting horizon. In fact, it is quite true when the forecasting horizon h exceeds the embedded dimension d, as at one point, all the inputs are predicted value rather than actual price. The reason for this possible inaccuracy is that this strategy is sensitive to accumulation of errors with a long forecasting horizon. The effect is known as Bufferfly Effect. The intermediate forecasting errors will propagate forward as these predicted results are used to determine the subsequent predictions.

4) Training Procedure: For this multi-step prediction case, we use the first 80 days out of 100 days as our training data set. The rest 20 days is our prediction target. We divide the training data set into three categories, training, validation and testing. We apply 70% for training, 15% for validation and 15% for testing.

5) Modifications to the Simple Recursive Method: According to our discussion, we know that simple recursive method’s potential poor performance is due to accumulation error. Hence we propose an error constraint strategy to reduce the accumulation error. Since we use the first 80 days as our training set, the rest 20 days as our prediction target, in these 20 days, we want to use the first 15 days to set up an error constraint, and the rest 5 days is for true forecasting purpose. All these 20 days are predicted results. The difference is that the first 15 days are error-reduced by an error constraint while the last 5 days are pure forecasting result. In our case, since the 000024.SZ stock price fluctuate between 11 and 35, we use 1 as our error constraint. For example, if the actual closing price at day 81 is 20, and our predicting closing price at day 81 is within the range 19 and 21, then the prediction process will continue and this predicted value will propagate as one of the inputs for the subsequent forecasting. But if the predicting closing price is 22, which falls outside the error constraints range, then the prediction result will not be accepted by the system. The system will predict again until the predicted result satisfies the error constraint condition. In our case, this constraint is set from day 81 to day 95. The logic behind this algorithm is that since the system has successfully (the error is controlled within certain range) predicted the first 10 days, the chance of being correct for the next five days will be possibly larger compared to a wrong prediction for the first 10 days. This is due to the fact that the neural network may select more suitable weights and bias for this specific situation. In addition, since the first predicted 10 days’ results are closer to the actual price, the accumulated error for these 10 days will possibly be reduced compared to the simple recursive strategy.

6) Multi-step Prediction Results: If we use this system in a real case, at day 95, the system predict that for the next five days, the stock price will go up first and then probably go down. According to this prediction, we can enter the market for a long position. Once we earn some money, we can choose either exit from the market, save the money, or set up a short position if we are quite sure that the price is going to drop. Although the prediction results are not very stable, with many times of running, the results tend to show that the price movement will behave like an “N”.

The following two figures show the results of the predictions. The figure on the left shows simple recursive strategy multi-step prediction results, while other one shows multi-step prediction results using improved recursive strategy. The black line is the actual closing price. The blue line is the predicted price under error constraint. The red line is the true predicted price without any correction.

By purely observation, we can tell that our improved recursive strategy outperforms the simple recursive strategy in terms of the movement of stock price. Above all, the improved strategy gives us a hint to tell whether it is a highly possible correct prediction or not.

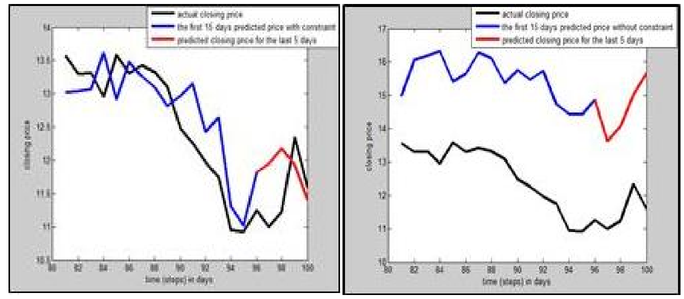

Figure 2.Prediction results of the two systems

If we run the system many times, in terms of probability, one or two patterns will appear more than other patterns. While the simple recursive strategy cannot tell us how big the chance is being possibly right. This is a very critical point for prediction especially for a real case investment. If we do not know the chance of winning the game, we also cannot decide how much risk we would like to take.

IV. CONCLUSION

In this paper, we first reproduced the one-step ahead prediction system from Phua et al. Secondly, we made some modifications and successfully outperform the original prediction system in terms of MSE value, hit-rate and absolute error.

Moreover, we explored a difficult multi-step prediction problem. Firstly, we reproduced a multi-step prediction system using simple recursive algorithm. Then, we proposed an error constraint algorithm in order to obtain better weights and bias, as well as smaller accumulated errors. The results outperform the simple recursive algorithm by observation. What is more important, the improved algorithm gives us a way to tell how big the chance of being right is. It is critical for a real investment decision.

In future work, we realize that the quality of training data is the most important thing for a successful prediction [16-19]. In stock prediction it is very critical to determine which factor is more important. With the power of neural network, we can try different factors and different combinations of those factors. This may help us to decide which factor(s) is(are) the main influential for a specific period of time. This will require adequate applications of feature selection techniques [20-30]. In addition, stock movements can be regarded time series, for which wavelets and wavelet packets have been shown to be powerful processing tools [31][32].

REFERENCES

[1] C. Aldrich, “Exploratory analysis of metallurgical process Data with neural networks and related methods”, Elsevier Science, British Library Cataloguing in Publication Data, vol. 1, pp. 50-67, 2002

[2] K. L. Priddy, and P. E. Keller, “Artificial Neural Networks: An Introduction”, SPIE – The International Society of Optical Engineering, Washington, 2005

[3] H. G. Zimmermann, and A. S. Weigend, “Representing Dynamical Systems in Feed-forward Networks: A Six Layer Architecture” (1997). In: *Decision Technologies for Financial Engineering*, edited by A. S. Weigend, Y. S. Abu-Mostafa, and A.-P. N. Refenes, p. 289-306. (Proceedings of the Fourth International Conference on Neural Networks

in the Capital Markets, NNCM-96, Pasadena, November 1996). Singapore: World Scientific.

[4] P. K. H. Phua, D. Ming, W. Lin, “Neural network with genetic algorithms for stock prediction”, Fifth Conference of the Association of Asian-Pacific Operations Research Societies, July 2000

[5] M. J. A. Berry and G. S. Linoff, “Data Mining Techniques: For Marketing, Sales, and Customer Relationship Management”, 2nd Edition", pp. 232-250, 2004.

[6] B. Yildiz, “Use of artificial neural networks in prediction of financial failures”, ISE Review, vol. 5, pp. 47-62, 2001.

[7] S. I. Gallant, “Neural Network Learning and Expert Systems”, MIT Press, Cambridge, 2003.

[8] A-S. Chen, M. T. Leung, H. Daouk, “Application of neural networks to and emerging financial market: forecasting and trading the Taiwan Stock Index”, Computers and Operations Resarch, vol. 30, 2003.

[9] G. Armano, M. Marchesi, A. Murru, “A hybrid genetic-neural architecture for stock indexes forecasting”, Information Series, vol. 170, pp. 3-33, 2005

[10] G. Zhang, E.Patuwo, M. Hu, “Forecasting with artificial neural networks: the state of the art”, International Journal of Forecasting, vol. 14, pp. 35-62, 1998.

[11] K. Levenberg, “A method for the solution of certain problems in least squares”, Quart. Appl. Math., no. 2, pp.164-168, 1944.

[12] T. Kimoto, K. Asakawa, M. Yoda, M. Takeoka, “Stock market prediction system with modular neural network”, Proceedings of the International Joint Conference on Neural Networks, pp. 1-6, 1990.

[13] D. Marquardt, “An algorithm for least squares in estimation of nonlinear parameters”, SIAM Journal of Applied Mathematics, no. 11, pp. 431-441, 1963.

[14] H. Tong, K. S. Lim, “Threshold autoregressive, limit cycles and cyclical data”, Journal of the Royal Statistics Society Series, vol. B-42, no. 3, pp. 245-292, 1980.

[15] A.S. Weigend, N.A. Gershenfeld, “Time Series Prediction: Forecasting the Future and Undestanding the Past”, AddisonWesley, Reading, MA, 1993

[16] L.P. Wang, “Wavelet neural networks for stock trading and prediction,” Invited Oral presentation, SPIE Defense, Security, and Sensing, 29 April - 3 May 2013, Baltimore, USA.

[17] Shekhar Gupta and L.P. Wang, “Stock Forecasting with Feedforward Neural Networks and Gradual Data Sub-Sampling,” Australian Journal of Intelligent Information Processing Systems, vol.11, pp.14-17, 2010.

[18] Ming Zhu and L.P. Wang, "Intelligent trading using support vector regression and multilayer perceptrons optimized with genetic algorithms," 2010 International Joint Conference on Neural Networks (IJCNN 2010), 2010.

[19] Shekhar Gupta and L.P. Wang, “Neural networks and wavelet de-noising for stock trading and prediction”, in Time Series Analysis, Witold Pedrycz and Shyi Ming Chen (eds.), 2012.

[20] L.P. Wang (Ed.): Support Vector Machines: Theory and Application. Springer, Berlin, 2005.

[21] L.P. Wang and X.J. Fu, Data Mining with Computational Intelligence, Springer, Berlin, 2005.

[22] L.P. Wang, Nina Zhou, and Feng Chu, “A general wrapper approach to selection of class-dependent features,” IEEE Transactions on Neural Networks, vol.19, no.7, pp.1267-1278, 2008.

[23] Xiuju Fu and L.P. Wang, "Data dimensionality reduction with application to improving classification performance and explaining concepts of data sets", International Journal of Business Intelligence and Data Mining (invited paper), vol.1, no.1, pp.65-87, 2005.

[24] L.P. Wang, “Feature selection for bioinformatics,” SPIE Defense Security Sensing, April 23~28, 2012 Baltimore.

[25] Nina Zhou and L.P. Wang, “Effective selection of informative SNPs and classification on the HapMap genotype data,” BMC Bioinformatics, 8:484, 2007.

[26] F. Fazayeli, L.P. Wang, and J. Mandziuk, “Feature Selection Based on the Rough Set Theory and Expectation-Maximization Clustering Algorithm”, Rough Sets and Current Trends in Computing, vol. 5306, pp. 272-282, 2008

[27] X.J. Fu and L.P. Wang, "Rule Extraction from an RBF Classifier Based on Class-Dependent Features", Cec'02: Proceedings of the 2002 Congress on Evolutionary Computation, Vols 1 and 2, pp. 1916-1921, 2002.

[28] X.J. Fu and L.P. Wang, "A GA-based novel RBF classifier with class-dependent features", Cec'02: Proceedings of the 2002 Congress on Evolutionary Computation, Vols 1 and 2, pp. 1890-1894, 2002.

[29] Xiuju Fu and L.P. Wang, "Rule extraction using a novel gradient-based method and data dimensionality reduction", Proceedings of the 2002 IEEE International Joint Conference on Neural Networks (IJCNN 2002), Honolulu, vol.2, pp. 1275 – 1280, May 12-17, 2002.

[30] Xiuju Fu and L.P. Wang, "Data dimensionality reduction and RBF neural networks for concise rule extraction", Proceedings of the 8rd International Conference on Neural Information Processing (ICONIP 2001), Shanghai, China, November 14-18, pp.149-153, 2001.

[31] L.P. Wang, K.K. Teo, and Z.P. Lin, "Predicting time series with wavelet packet neural networks", 2001 IEEE International Joint Conference on Neural Networks (IJCNN 2001), Vols 1-4, pp.1593-1597, 2001.

[32] K.K. Teo, L.P. Wang, Z.P. Lin, "Wavelet packet multi-layer perceptron for chaotic time series prediction: effects of weight initialization," Computational Science -- ICCS 2001, Proceedings Pt 2, Volume: 2074, pp. 310-317, 2001.

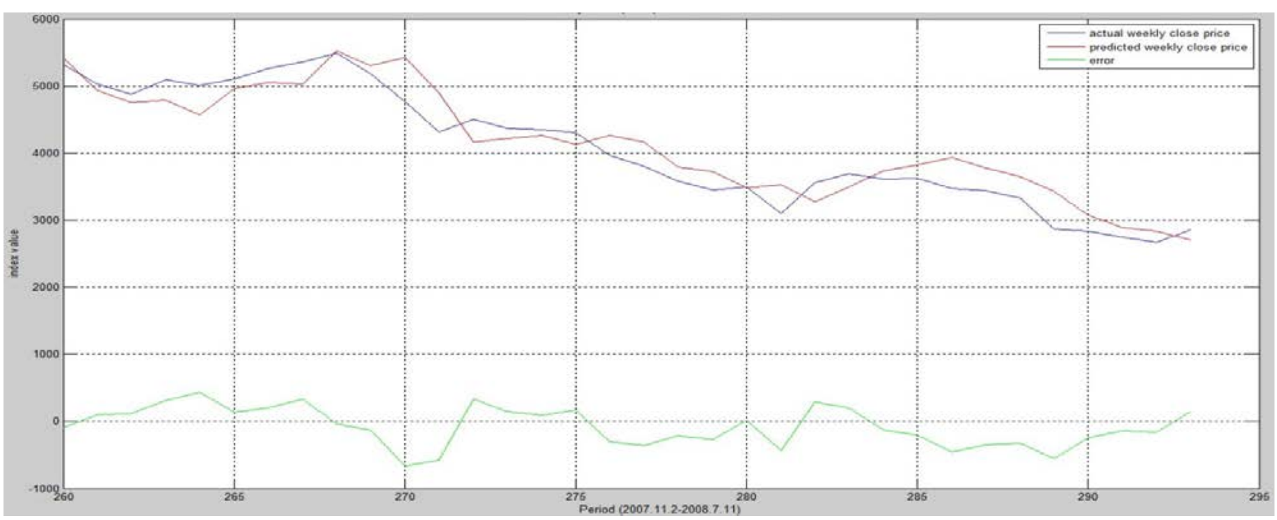

Figure 1. Our prediction result.