

# Forecasting KOSPI using Elman network

Hongchul Ahn, Hotak Hong<sup>a</sup>, Jongho Nang and Saejoon Kim

*Department of Computer Science and Engineering, Sogang University, Seoul 121-742, Korea*

**Abstract.** Due to the non-stationary nature of stock market index, making a prediction on its course is a truly challenging task. Research has been actively conducted to predict stock market indices by means of machine learning in recent years. In our research, we made a prediction of KOSPI for one week based on Elman Network. Based on the predictive result, we ran a simulation from which we obtained 3.16% yield over a period of one year. In this paper, we describe how we exploited Elman network to make predictions on stock markets, then we propose a method for using the predictive values for investment.

## 1. Introduction

Stock market indices are influenced by a great range of indicators hence making predictions on them is an extremely difficult task. As machine learning has been applied to many fields lately, many researchers are exploring to apply machine learning to predict stock prices. With so many indicators that affect the stock price, the choice of input parameters is critical to the prediction result [1, 2]. There are two major types of stock price analysis: fundamental and technical. While a firm's intrinsic values are studied in a fundamental analysis, past data such as stock price history and stock trade volume is reviewed to predict future stock price. There has been a widely accepted consensus among investors that investors cannot beat the stock market with technical analysis if the capital market operates efficiently. However, it has been demonstrated that information obtained with technical analysis is as powerful as data obtained with macroeconomic indicator analysis [3]. Technical analysis was also observed to be very effective in an uncertain market circumstances [4]. In experiments for this paper, we exploited technical analysis indices as input parameters. Features were selected with stepwise regression to prevent entering features repeatedly. Elman network, which is a type of RNN, was chosen as it shows solid performance on time series analysis and prediction was made on weekly KOSPI level, on which we base our investment decision.

The organization of this paper is as follows. In Section 2, related work is discussed. Section 3 explains our approach to input data selection and data processing.

Section 4 demonstrates our experiment design including experiment length and machine learning method followed by experimental results. Section 5 concludes the paper with suggestions for future work.

## 2. Related Work

Machine learning has been actively embraced by more and more academic fields in recent years. One notable area that has adopted machine learning is stock market prediction. In the fields of business administration and economics, fundamental analysis and technical analysis have been produced [5, 6]. In statistics, numerous time series approaches [7]. Despite the existing many attempts to predict the stock market, controversy has persisted if stock market prediction would be valid or effective in the actual stock market. With the introduction of various machine learning methods, stock market prediction now shows satisfying performances.[8, 9] In this paper, we used Elman network, a type of RNN, for stock market prediction.

## 3. DATA

Many different types of information can be used to predict the direction of the stock market. There are two types of stock price analysis methods: fundamental analysis and technical analysis. The purpose of our experiment was predicting the stock market index of one week in the future, hence we excluded fundamental data and only used technical analysis data for input parameters, which we obtained from Daewoo Securities HTS. In order to prevent repeated input of input parameters, stepwise regression was used to make indicator selection. In our experiment, each input parameter is introduced and discussed on manipulation method. We used a statistics program R for experiment.

---

<sup>a</sup> hotakhong@sogang.ac.kr

This work was supported by the ICT R&D program of MSIP/IITP. [Open ICT Convergence Support Program (R0215-15-1001) and ICT/SW Creative Research Program (R2215-15-1003)]

### 3.1 Stock Price Index

The following items are included when describing a stock: the opening and closing prices, the highest and lowest prices in a specific period of time. The price of a stock of a given period generally refers to the final price of the stock. Most technical indices used in this paper are calculated based on stock price and trading volume, hence a solid understanding of these indices is essential in successful configuration of input parameters.

### 3.2 Technical Indices for Experiment

The data frame we used consists of 14 features and yields one output variable. The features have been selected among meaningful technical indices of distinctive nature [10]. Input parameters were categorized as trend, oscillator, and momentum types, and representative technical indicators of each type were selected as our input parameters. Among the technical indices, moving average (2, 5, 20, 60) was chosen among trend-types, moving average convergence/divergence (MACD) and relative strength index (RSI) from oscillator types, and MO(3) and MO(9). We also added opening price, highest and lowest price, trading volume as well as sigma(20) in estimating stock prices. The opening and closing prices, highest and lowest prices at time point T are the most basic pieces of information in calculating the closing price of T+1. Trade volume and sigma were also added to our input parameter list as these add depth to estimation. From trend types, we chose moving average as an input parameter. MACD and RSI were chosen from oscillator type indices and MO from momentum types. As shown, we selected input parameters from trend, oscillator, and momentum types and weekly closing price at t+1 was estimated as output. Stepwise regression was adopted for final selection of indicators so as to suppress repeated impact in the experiment.

#### 3.2.1 MA

The moving average during the period of N days is calculated as follows:

$$MA(t) = \frac{1}{N} \sum_{i=t-N}^{t-1} price(t-i) \quad (1)$$

We used MA of 2, 5, 20, and 60 days

#### 3.2.2 Momentum

Momentum is one of the most basic technical indices, calculated by comparing the current stock price to the stock price at a specific time in the past. Momentum of 0 indicates there was no change in the stock price over the course of a specific period of time while a positive value means the stock price has increased and a negative momentum means otherwise. Momentum is calculated as follows:

**Table 1.** Number of states for the indicators.

Label	Index
x1	open price (t)
x2	Highest price (t)
x3	Lowest price (t)
x4	Weekly close price (t)
x5	MA(2)
x6	MA(5)
x7	MA(20)
x8	MA(60)
x9	volume
x10	Momentum(3)
x11	Momentum(9)
x12	RSI(14)
x13	MACD(12,26)
x14	Sigma(20)
Y	Weekly close price (t+1)

$$MO(t) = price(t) - price(t-n) \quad (2)$$

The term t is the closing price at day t. Comparing the current stock price and n days back shows whether the stock is going up or down in price. In our experiment, we used 3 day and 9 day momentum indices

#### 3.2.3 RSI

Relative strength index (RSI) shows the magnitude of price increase of a stock. The value of RSI lies between 0 and 100. RSI values below 25 is interpreted as depressed increase while the stock is thought to be overheated if above 75. RSI is calculated as follows:

$$RSI(t) = \frac{A}{A+B} \times 100 \quad (0 \leq RSI \leq 100) \quad (3)$$

The magnitude of each increase in the stock price over N days is added up and divided by the magnitude of each increase and decrease of stock price over N days to calculate RSI. In our calculation of RSI, 14-day period was used.

#### 3.2.4 MACD

Moving average convergence/divergence (MACD) was devised by Gerald Appel when he observed short-term and long-term moving averages repeat convergence and divergence and address the MA only reflects the past and does not directly predict the future. MACD is calculated as follows:

$$MACD(t) = SMA(t) - LMA(t) \quad (4)$$

where SMA stands for short-term moving average and LMA is long-term moving average. In our experiment, SMA(12) and LMA(26) were used.

## 4. Experiment

### 4.1 Data Processing

Our input parameters consist of a variety of technical indicators with different characteristics, and the indicators lie in different scales, hence each parameter needs to be normalized to be used as a feature. This is necessary because neural networks perform effectively when the values of input data are concentrated around zero. In our approach, all of the input parameters used as feature were normalized in  $[0, 1]$  as follows[11]:

$$\frac{X_t - \text{Min}(X)}{\text{Max}(X) - \text{Min}(X)} \quad (5)$$

The drawback of normalization as Equation (5) is that the minimum value becomes 0 hence data loss could incur. Hence we chose to add 0.00001 to all of the input parameter values to prevent this. The range of each feature after all is normalized within  $[0.00001, 1.00001]$ .

### 4.2 Stepwise regression

Stepwise regression is a method for selecting variables that meaningfully affect the result in regression analysis. In selecting variables, a few methods are available such as forward selection, backward elimination, and stepwise selection. In our experiment, backward elimination was used, which starts with all of the candidate variables and deletes variables that least affect the model one by one. With backward elimination, we selected 7 out of 14 parameters. Stepwise regression was performed in our experiment with the 7 input parameters.

**Table 2.** The 7 features selected with stepwise regression

Label	Index
x2	Highest price (t)
x3	Lowest price (t)
x6	MA(5)
x7	MA(20)
x9	volume
x12	RSI(14)
x13	MACD(12,26)
Y	Weekly close price (t+1)

### 4.3 Experimental period

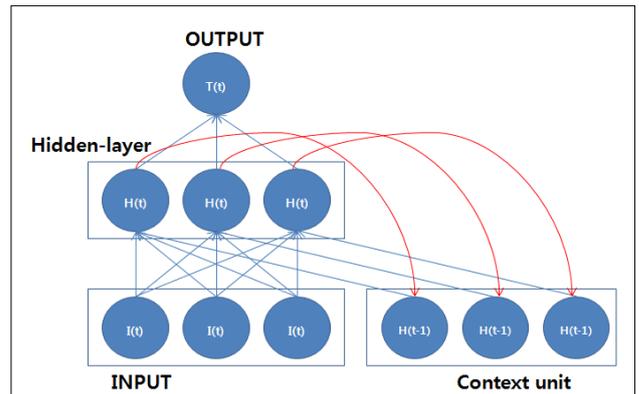
In our experiment, our method was applied to the weekly data collected between January 2, 2000 and October 26, 2015. The 722 weekly data points between January 2, 2000 and October 28, 2013 were used for training; data for validation comes from between November 4, 2013 and October 27, 2014. Test was performed for the period between November 3, 2014 and October 26, 2015. In our approach, the initial model was formed with the training data and validated to identify parameters that improve performance to construct the final model.

2000.01.02	2013.11.04	2014.11.03	2015.10.26
Training		Validation	Test
Final Training Period			

**Figure 1.** Data range for training, validation, and test.

### 4.4 Elman network

Elman network is known for satisfying performance for time-series data[12, 13], but it is a deep learning algorithm requiring a large volume of data. We resorted to Elman network, which is a simple recurrent network, for prediction since the amount of available data is not sufficiently large. There is a single hidden layer in a typical multi-layer perceptrons network, and a variety of data from the input parameters are extracted in the hidden layer before relayed to the output layer. A hidden layer consists of multiple nodes, and each node extracts different types of data from the input variables. An Elman network is similar to the multi-layer perceptrons model in many aspects such as the existence of a hidden layer consisting of multiple nodes as well as relay of input data from the input layer to the hidden layer and then to the output layer. Unlike the multi-layer perceptrons model however, in an Elman network, output variables are calculated at the same time when data is extracted in the hidden layer and fed as the input data for the next iteration. Figure 2 shows how a model is constructed in an Elman network by accumulating data over time. This characteristic was one major reason we opted for Elman network. To compare performance we also tested other machine learning algorithms such as SVM and multi-layer perceptrons with identical input data.



**Figure 2.** Elman network.

### 4.5 Forecast performance

Three different performance measuring methods were used in evaluating the performance of predicted values by each model. RMSE (root-mean-square error) (6) and MAE (mean absolute error) (7) evaluates the discrepancy between the estimated values by a model and the real data. Smaller values of RMSE and MAE indicate that the estimations are more accurate. MAPE (mean absolute percentage error) (8) expresses the difference between estimation and actual data in a ratio, and lower ratio indicates that the estimation is closer to the actual data.

$$RMSE = \sqrt{\frac{1}{N} \sum_{t=1}^N (X_t - F_t)^2} \quad (6)$$

$$MAE = \frac{1}{N} \sum_{t=1}^N |X_t - F_t| \quad (7)$$

$$MAPE = \frac{1}{N} \sum_{t=1}^N \frac{|X_t - F_t|}{X_t} \quad (8)$$

#### 4.6 Experimental Results

Testing with each machine learning method showed that Elman network produces the best results for all RMSE, MAE, and MAPE compared to linear SVM or multilayer perceptron network, hence we chose Elman network in our method.

#### 4.6 Profit simulation

With estimation data from the Elman network model, we simulated trading on the data.

**Table 3.** Experimental Performance.

Performance method	SVM(linear)	Neural Network (MLP)	Elman Network
RMSE	69.43	37.37	34.75
MAE	54.44	28.94	27.40
MAPE (%)	2.70%	1.45%	1.36%

Trading rules were defined based on the work[14] by Cheng et al.:

Rule 1 : Sell rule

$$IF |forecast(t) - actual(t)| < 0 \quad (9)$$

Rule 2 : Buy rule

$$IF |forecast(t) - actual(t)| > 0 \quad (10)$$

Our trading simulation was performed on an imaginary exchange-trade fund (ETF) that estimates stock prices with 100% accuracy and zero tracking error. For each week, a sell signal triggers buying an Inverse ETF and a buy signal triggers buying an Index ETF. If a sell signal is received for a stock currently owned, then all of the stock in question is all sold.

Definition of profit (11)

$$Profit = \sum_{t_s=1}^p \left( \frac{actual(t_s+1) - actual(t_s)}{actual(t_s)} \right) + \sum_{t_b=1}^q \left( \frac{actual(t_b) - actual(t_b+1)}{actual(t_b)} \right)$$

When we simulated stock trading based on these rules after making weekly predictions on KOSPI for one year starting in November 2014, we could obtain a profit of 3.16%. Our experiment demonstrates that investing in the stock market based on Elman network could yield profit.

## 5. Conclusion

Making weekly predictions on the stock market index based on machine learning was most accurate when the Elman network was used as evaluated with RMSE. Our trading simulation over one year of period starting in November 2014 produced profit at 3.16%. This is a meaningful result but it could have been affected by many other variables not considered in our experiment, and the result does not qualify as solid because the financial market is volatile. In the future work, we will work on making estimated values more accurate and applying the estimated results to stock trading.

## References

1. M. C. Chan, C. C. Wong, and C. C. Lam, Financial Time Series Forecasting by Neural Network Using Conjugate Gradient Learning Algorithm and Multiple Linear Regression Weight Initialization, Computing in Economics and Finance, (2000)
2. R. J. Frank, N. davey, and S. P. Hunt, Time Series Prediction and Neural Networks, Journal of Intelligent and Robotics System, Vol.31 Issue1-3, pp.91-103, (2001)
3. Zhou, G., and Y. Zhu. An Equilibrium Model of Moving-Average Predictability and Time-Series Momentum. Working Paper, Tsinghua University and Washington University in St. Louis (2013)
4. Zhu, Y., and G. Zhou. Technical Analysis: An Asset Allocation Perspective on the Use of Moving Averages. Journal of Financial Economics, **92**, 519-522. (2009)
5. R. D. Edwards and J. Magee, Technical Analysis of Stock Trends, John Magee, Inc., (1974)
6. S. Cottle, R. F. Murray, F. E. Block, and D. L. Dodd, Graham and Dodd's *Security Analysis*(5th edition), McGraw-Hill. (1988)
7. S. M. Kendall and K. Ord, *Time Series*, Oxford, New York. (1997)
8. Vatsal H. Shah, Machine Learning Techniques for Stock Prediction, Technical Report, New York University, (2007)
9. Li Deng. A Tutorial survey of architectures, algorithms, and applications for deep learning, APSIPA Transactions on Signal and Information Processing. Vol.3. (2014)
10. T. Y. Mieko., T. Seiji, Adaptive Use of technical indicators for the prediction of intra-day stock prices. (2007)
11. Adebisi, A. A. and Ayo, C. K. and Adebisi, M. and Otokiti, So, O. An Improved Stock Price Prediction

- using Hybrid Market Indicators, African Journal of Computing & ICT, Vol.5. No.5, pp.124-135. (2012)
12. H. Shi, X. Liu, Application on stock price prediction of Elman neural networks based on principal component analysis method, IEEE Transactions on Pattern Analysis and Machine Intelligence, Wavelet Active Media Technology and Information Processing (ICCWAMTIP), Vol.10, 411-414. (2014)
  13. J Wang, J Wang, W. Fang, H. Niu, Financial Time Series Prediction Using Elman Recurrent Random Neural Networks, Hindawi Publishing Corporation Computational Intelligence and Neuroscience, Article ID 613073. (2015)
  14. C. H. Cheng, L. Y. Wei, Y. S. Chen, Fusion ANFIS models based on multi-stock volatility causality for TAIEX forecasting, Neurocomputing, 3462-3468. (2009)