Implementation of Artificial Neural Network to Predict S&P 500 Stock Closing Price

Ike Fitriyaningsih, Anthon R. Tampubolon, Harry L. Lumbanraja, Grace E. Pasaribu and Pita S.A. Sitorus
Institut Teknologi Del, Diploma III Teknik Informatika, Indonesia

Abstract. Artificial Neural Network (ANN) is a learning method that can be used for prediction and classification. In this study, ANN back-propagation is implemented to predict the closing price of the S&P 500 stock exchange using historical data. Historical data consisting of five variables, namely open, high, low, close, and volume. The historical data are taken from finance.yahoo.com which stores historical data of daily stock prices up to 65 years earlier. The data is designed based on daily prediction scenarios of the closing stock price of the S&P 500 stock exchange. Using the scenario an application prototype using R and Java software has been successfully built. The prototype is dynamic during select the data set (10, 50, 100, 500 past data) to get the best stock closing prediction for the next day. The selection of the best stock closing prediction uses the MAPE (Mean Average Percentage Error) criterion which is in contrast to the prediction accuracy level. The smaller the MAPE value the better the predicted result. Based on the implementation results, the average MAPE for daily forecast for 1 month in April 2017 was 0.2307 indicating that the average daily prediction accuracy rate for 1 month was 99.77%. Our prototype application can make prediction automatically every day, every time when connected to the internet.

1. Introduction
Artificial Neural Networks (ANN) is learning process like the human brain. The human nervous system is able to recognize different patterns and has been seen before. Back-propagation is one of the ANN algorithm that consisting of two stages; the forward and backward step. The forward step works from input layer, hidden layer, and output layer. The backward stage works by using a comparison value between the output and the target value returned from the output layer to the hidden layer and input layer. Learning process that has a target value like this is called supervised learning.

Supervised learning works repetitively until the output value approaches the target value. This learning process takes place in the backwards step where the error value is used to fix the weights on the hidden layer, so that the weights can be used to produce better output and closer to the actual value. ANN can be used to generate good predictive values using existing data.

The objective of this study was to produce the best data sets of experiments that have been designed so as to generate predictions on the closing price S&P 500 stock for daily predictions. The experiments were conducted for the selection of the best data sets which will then be used as models for input data on the ANN so as to generate stock price predictions. The input data used are open data (opening price), high (high), low (low price), close (closing price), and volume (number of shares traded). Implementation of the application will use R connected JAVA.
2. Basic Theory

ANN is one of the learning methods of artificial intelligence (AI) which is presented like human brain concept with millions of nerve cells to process information. Each nerve cell will interact with each other to create cooperation to train work skills during the learning process [1].

ANN architecture is as follows [2]:
1. Network with Single layer (Single-Layer Networks)
2. Network with multiple layers (Multi-Layer Networks)

The Multi-Layer Network consists of an input layer, an output layer and a hidden layer. Each node of the input layer is connected to each hidden layer node and each node of the hidden layer is connected to the output layer node. Usually there are some weights associated with each connection. The input layer is the information or the initial data entered into the network [3].

![Figure 1. Mathematical Illustration of ANN [4]](image)

The mathematical model of the Artificial Neural Network in Figure 1 describes as the input value to be forwarded to the input layer. The input value will be calculated by summation function as output. The output must then pass through the threshold function (threshold). If the resulting value passes the threshold value then the neuron is activated otherwise the neuron is not activated. The activation is done using the activation function to produce the final output at the output layer.

The activation function ANN back-propagation method, must continuous, differentiated easily, and the function not down [1]. The activation function Sigmoid Biner is one of the activation functions that satisfied the three requirements and this function is used on Artificial Neural Network with output at interval 0-1. The function of sigmoid activation is as follows:

\[ y = f(x) = \frac{n(n-1)x^2}{1+e^{-\sigma x}} \]

where \( f' = \sigma f(x)[1-f(x)] \).

Before the data is processed, first the data in normalization with min-max normalization method. The normalized data to be convert again after that [5]. The min-max normalization is:

\[ X_n = (X_0 - X_{\min})/(X_{\max} - X_{\min}) \]

where,

- \( X_n \) : value of data normal
- \( X_0 \) : value of actual data
- \( X_{\min} \) : value of minimum data
- \( X_{\max} \) : value of maximum data

ANN testing process is using the weight of the training result. The process that occurs in the testing process is similar to the training process to produce predictions. Then the predicted result in the testing process will be compared with the target as the error value. The sum of all errors (actual predictive
differences) is calculated using the MAPE (Mean Average Percentage Error) formula in the following equation:

$$\text{MAPE} = \frac{1}{n} \sum_{t=0}^{n} |\frac{P_t - A_t}{A_t}| \times 100$$

with,

$$PE_t = \frac{A_t - P_t}{A_t} \times 100$$

where:

- $PE_t$: percentage error of $t$ period
- $A_t$: actual closing stock price
- $P_t$: predicted closing stock price

3. Implementation

R is a system of statistical calculations and graphs. R provides programming languages, high-level graphics, interfaces to other languages and debugging facilities. Package from R allows flexible settings through custom-choice error and activation functions. JAVA programming language is ease of use, functionality, stable, secure and cross-platform [6].

The implementation of ANN in R is using neuralnet and Rserve package. It is used to perform the training and testing process so as to produce output in the form of predictions. JAVA is used as data processing. R will work on the training and testing process, JAVA works on updating the database using the API, retrieving data from the database for later processing, data normalization, and error counting. To connect R with JAVA is using the library can support R and JAVA to connect. The library used is as follows:

1. JRI.jar
2. JRIEngine.jar
3. REngine.jar
4. Rserve-0.6-1.jar
5. RserveEngine.jar

The Rserve package connects R and JAVA. One example is to update the data through the yahoo API and save it into the database. Data has been entered into the database and then processed again, such as normalizing the database. The result of normalization of the database is then saved into the CSV in the specified directory where R language accesses the data in the form of the CSV.

The ANN architecture used in the implementation of this paper can be seen in Figure 2

![Figure 2. Architecture Design of ANN](image)
GUI (Graphic User Interface) implemented using JavaFX. In accordance with the design that has been designed before, the GUI is already implemented. Figure 3 is the initial display.

![Figure 3. Initial Display GUI](image)

Figure 4 is a page that displays the results of the train process that displays the prediction and actual price one day before to show the movement of rising or falling stock prices.

![Figure 4. Daily Prediction GUI](image)

To view the graph, the “Grafik” button on Figure 4 and graph movement of seven last days are show like Figure 5.

![Figure 5. Graph of Actual and Prediction Data](image)
4. Experimental Result
Predicted and price movements can be seen in Figure 6. The graph shows the predicted results using a prototype not far from the actual price of the S&P500 closing stock price. S&P500 closing stock price are stable at start of April but there is decreasing trend. S&P500 closing stock price increase after 19th April.

![Graph of Prediction versus Actual Closing Price](image)

The predicted data is generated one day after. Detail data can be seen at Table 1. Predictive data is also compared with the actual data a day before to see whether joint stock price movements rise or fall. Implementation results do not produce predictions on all date because Saturday and Sunday stock market are close.

| Date | Prediction | Actual Price | Error | MAPE | Accuracy(%) |
|------|------------|--------------|-------|------|-------------|
| 03   | 2359.43    | 2360.13      | 0.70  | 0.0747 | 99.9253 |
| 04   | 2359.53    | 2352.95      | 6.58  | 0.2434 | 99.7566 |
| 05   | 2358.48    | 2357.48      | 1.00  | 0.2566 | 99.7434 |
| 06   | 2356.28    | 2355.54      | 0.74  | 0.2640 | 99.7360 |
| 07   | 2355.54    | 2357.16      | 1.62  | 0.2323 | 99.7677 |
| 10   | 2354.59    | 2353.78      | 0.81  | 0.1542 | 99.8458 |
| 11   | 2352.83    | 2344.93      | 7.90  | 0.1988 | 99.8012 |
| 12   | 2347.06    | 2328.95      | 18.11 | 0.1613 | 99.8387 |
| 13   | 2341.75    | 2349.01      | 7.26  | 0.2996 | 99.7004 |
| 17   | 2347.76    | 2342.19      | 5.57  | 0.2699 | 99.7301 |
| 18   | 2330.58    | 2338.17      | 7.59  | 0.0352 | 99.9648 |
| 19   | 2346.54    | 2355.84      | 9.30  | 0.1850 | 99.8150 |
| 20   | 2354.85    | 2348.68      | 6.17  | 0.2726 | 99.7274 |
| 21   | 2353.52    | 2374.15      | 20.63 | 0.2736 | 99.7264 |
| 24   | 2373.09    | 2388.61      | 15.52 | 0.3093 | 99.6907 |
| 25   | 2383.57    | 2387.45      | 3.88  | 0.3108 | 99.6892 |
| 26   | 2388.22    | 2388.77      | 0.55  | 0.3293 | 99.6707 |
| 27   | 2382.67    | 2384.19      | 1.52  | 0.3164 | 99.6836 |
| 28   | 2390.29    | 2388.33      | 1.96  | 0.1972 | 99.8028 |
| Average |          |              | 0.2307 |     | 99.7693 |
As a determinant indicator of either the bad predictive value can be seen from the MAPE value that has been available in the table column. The average MAPE forecast for the month was 0.2307 with an average accuracy of 99.77%.

5. Conclusion
In this chapter, we explain the conclusions and suggestions of the final work done. Here are the conclusions:

1. The prototype applied dynamic data set at 10, 50, 100, 500 past data selection so as to generate the best S&P500 stock predictions with MAPE criteria.
2. The daily prediction result of the S&P 500 stock price using implementation ANN by our prototype is good with the average MAPE prediction for a month is 0.2307 and the average accuracy is 99.77%.

Acknowledgments
The author also thanked to Institut Teknologi Del so we can realize and finish this research. The authors hope this journal can be useful for all those who need and to the reader.

References
[1] Trimulya. A.S. et.al. —Implementasi Jaringan Syaraf Tiruan Metode Backpropagation untuk Memprediksi Harga Saham.‖ Jurnal Coding, Sistem Komputer Unhan. vol. 03. pp. 66-75. 2015.
[2] Suyanto. Artificial Intelligence: Searching - Reasoning - Planing- Learning. Second Editor. Yogyakarta: Informatika Bandung. 2014.
[3] M. Cilimkovic. —Neural Networks and Back Propagation Algorithm‖. Institute of Technology Blanchardstown. Ireland
[4] G. Ferdika. —Dasar Komputasi Cerdas Neural Network (Jaringan Syaraf Tiruan).‖ 2011.
[5] N. Chamidah. W. dan U. Salamah. —Pengaruh Normalisasi Data pada Jaringan Syaraf Tiruan.‖ Jurnal ITSMART. pp. 28-29. Juni 2012.
[6] L. Afriyanti. —Rancang Bangun Tool untuk Jaringan Syaraf Tiruan (JST) Model Perceptron.‖ Seminar Nasional Aplikasi Teknologi Informasi 2010. ISBN: 1907-5022.