Stock Trend Analysis and Prediction Algorithm

Ivan B Wong, MSEE University of Southern California

Abstract

This paper outlines a stock trend analysis algorithm that predicts the trend price trends. The basic algorithm consists of clustering, analysis and prediction. A clustering algorithm is used to partition time series data from historical data of a given stock. Trends from each partition group are then analyzed and classified. The results of the partition groups and classifications are then used for trend prediction of the windowed time series data. Experimental results show that this approach is effective and efficient at predicting the forward trend of stock prices with preliminary results reported. Moreover, algorithm improvements and modifications are proposed for further enhancements to the stock price prediction.

Keywords: Stock Trend Analysis, Data Mining, Clustering, Time Series, Stock Prediction

Introduction

Trend analysis and prediction models play a vital role in the stock market. Experienced traders and portfolio managers use various tools which assist in determining the future trend of a given stock. Predictions often are based on observations from past performance of the stock. An early sign of a familiar pattern may alert a trader to what is likely to happen in the near future. Trading strategies can then be formulated and adjusted accordingly. Although there are no guarantees to a given method, the addition of trading tools may assist a trader in calculating risk for an increased return.

The old saying “history repeats itself” is a nomenclature in stock trading. Patterns in the market repeat themselves due to seasonal, cyclical or other unknown variables. To this effect, historical patterns and the analysis of the movement of a stock can provide valuable insight for forecasting future stock performance and trading strategies.

Algorithm

The algorithm consists of four main components: windowing, trend analysis & pre-clustering, clustering and classification. Figure 1 shows a block diagram of the algorithm which will be explained in more detail.

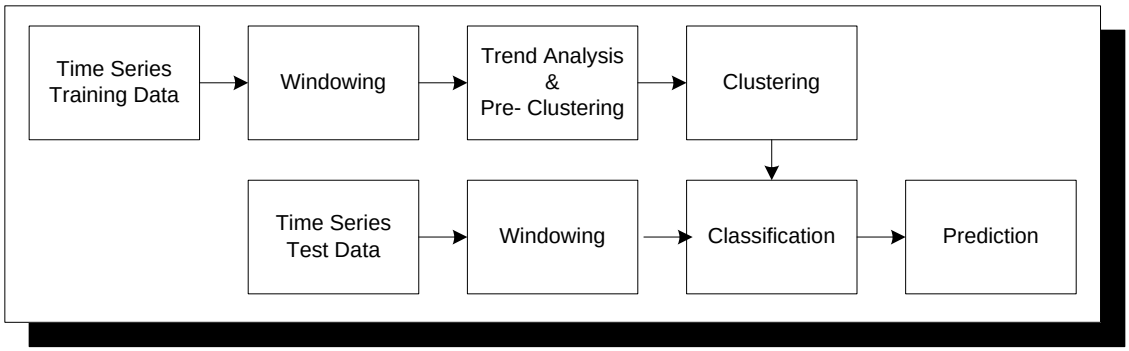

Figure 1. Basic Algorithm Block Diagram

Windowing / Time Series Preparation

Time series data or stock prices from a given period are windowed and used for test and training data.

Training data is acquired by sliding a fixed-length time window from time t_b to t_e . The total number of time series created is $N = t_e - t_b$ and the window length is given by w_{tr}.

$s_1 : p_1, p_2,\ldots,p_{wtr}$
$s_2 : p_2, p_3,\ldots,p_{wtr+1}$
…
$s_N : p_N, p_{N+1},\ldots,p_{wtr+N-1}$

where p_i (i= 1,2, … , w_{tr}+N-1) are stock prices at time i. A N by w_{tr} matrix or data set with N data records and w_{tr} attributes is created for the training data set. For the case of the testing data, a window of length $w_{te} < w_{tr}$ is used.

Training data is broken down into two parts. The first part is the same length as the testing data window, w_{te}, and the second part of length $w_{lm} = w_{tr} - w_{te}$ is used for trend analysis.

All windowed time series are then properly normalized using a basic linear scale equation given by:

$$G = H(F) = G_{\min} + (\frac{G_{\max} - G_{\min}}{F_{\max} - F_{\min}})(F - F_{\min})$$

where $G_{min} = 0$, $G_{max} = 1$, F_{min} and F_{max} denote the min and max values of the time series data, respectively. Figure 2 gives a schematic view of a windowed time series.

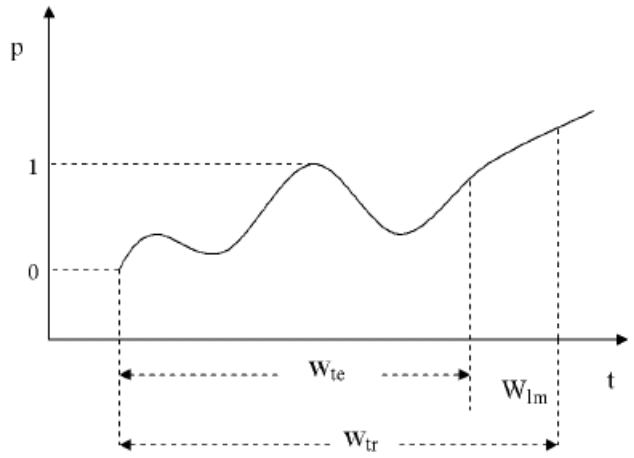

Figure 2: Schematic view of windowed time series and normalization

Trend Analysis & Pre-Clustering

The trend analysis and pre-clustering step sorts the training data according to trend movements and provides initial centroids for k-means clustering. Trend analysis is performed on the second half of the training time series of length w_{lm}. A one dimensional numerical gradient is calculated across the values resulting in a vector gradient. The vector gradient value is then summed together to give an overall gradient direction for the training time series. On average, values less than zero typically indicate a downward trend and a value greater than zero typically indicate an upward trend. After calculating the gradient directions, the training time series is then ordered in ascending rank based on the gradient direction of the w_{lm} window.

The ordered training data is then uniformly partitioned into K groups, where K > 1. To provide better k-means clustering in the latter step, initial centroids are computed from each of the K groups resulting in K initial vectors of length w_{te}.

Figure 3 provides a block diagram overview of the trend analysis and pre-clustering steps discussed.

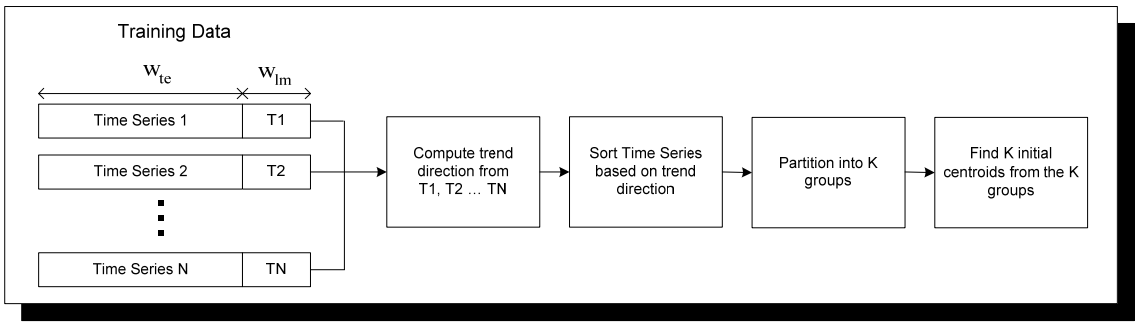

Figure 3. Trend Analysis & Pre-Clustering Block Diagram

Clustering

K-means clustering is performed to partition the training time series data. Initial centroids from the pre-clustering are used as inputs to the k-means clustering to provide the best centroid locations. As a result of the k-means clustering, the time series data is arranged in K groups of K centroid points with vector length w_{te}. The K^{th} centroid points denote the vector point with the closest Euclidian distance from the K^{th} group of time series data.

The K centroid points can then be labeled based on the original trend directions, where centroid C1 denotes a downward trend and centroid CK denotes an upward trend. The created respective centroids are then used to classify the testing time series data.

Figure 4 provides a block diagram overview of the clustering step discussed.

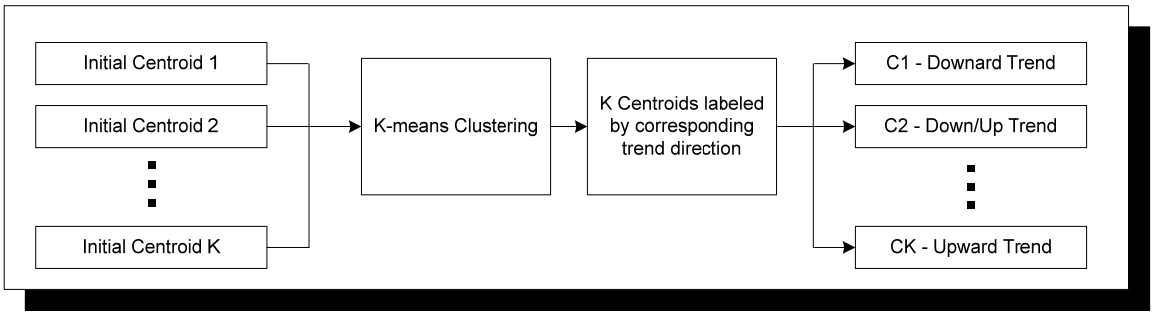

Figure 4. Clustering

Classification and Prediction

With the K centroid locations corresponding to a trend direction, classification can then be made on the testing time series data. Note the centroid vector length and windowed testing time series data length is w_{te}. The windowed testing data series is then classified to one of the K centroids based on the shortest Euclidian distance method. As a result, each of the window time series can then be predicted as a downtrend, uptrend, or a combination of both.

A prediction to each of the windowed time series can then be made based on its labeled trend direction.

Figure 5 shows a block diagram overview of the classification and prediction step discussed.

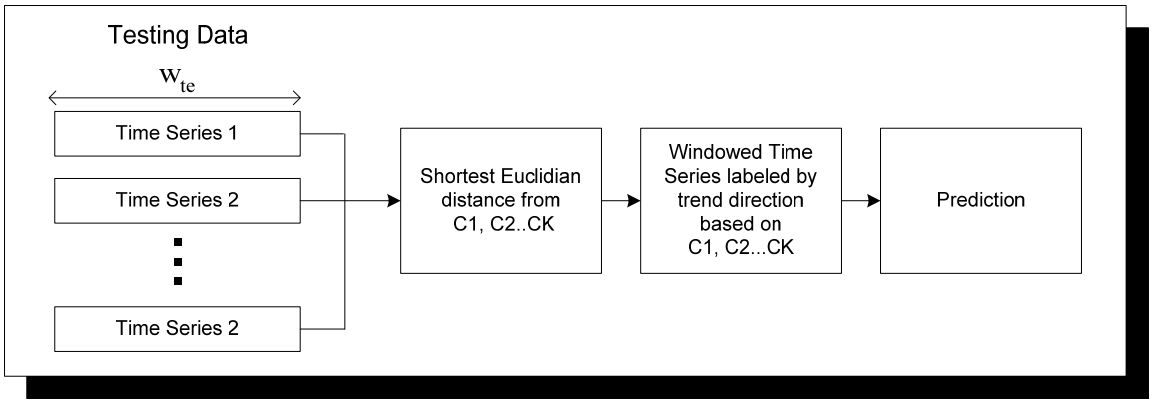

Figure 5. Classification and Prediction

Results

The algorithm was constructed into a Matlab program. Historical stock prices from the following three companies were used to demonstrate the stock trend analysis algorithm. As a note, the indicators at the lower portion of the stock chart are prediction indicators that forecast the future movement.

The following three tickers were used to demonstrate the algorithm: Apple (AAPL), Cisco (CSCO) and Ebay (EBAY). Data from 2004-2006 were used and retrieved from Yahoo Finance.

After some preliminary testing, the following values produced the best overall prediction results: training window = 15 days, testing window = 10 days, K = 3 clusters. The results are shown below in Figures 6, 7, 8.

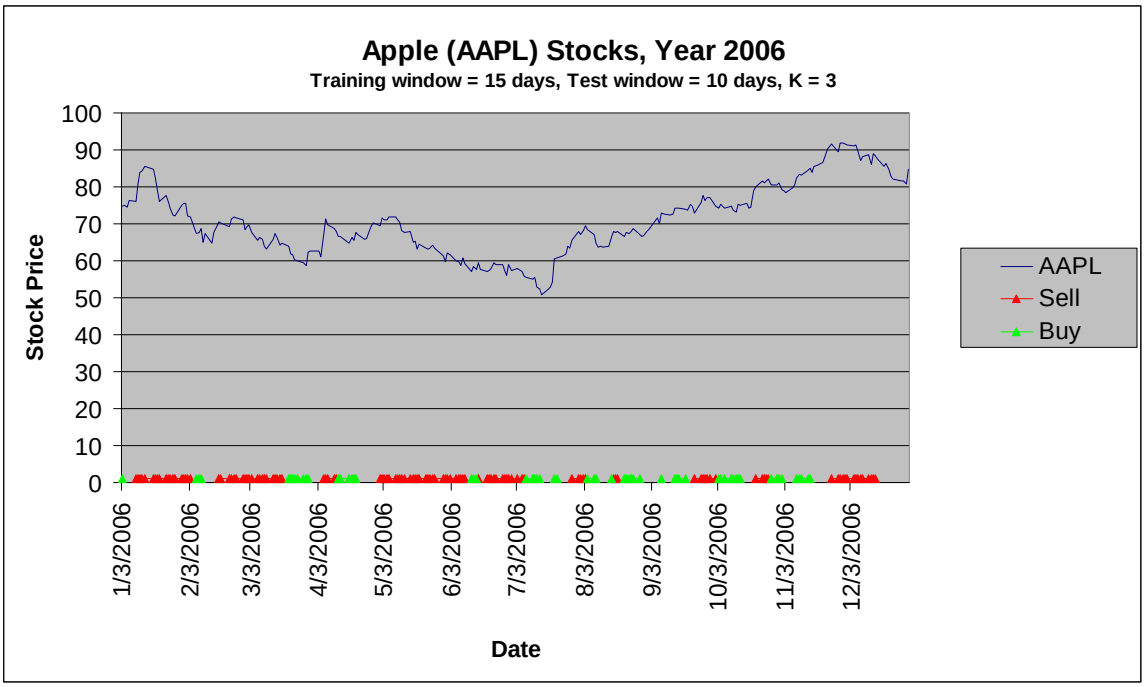

Figure 6: 2006 Stock Year, Apple, parameters: w_{tr} = 15 days, w_{te} = 10 days, K = 3

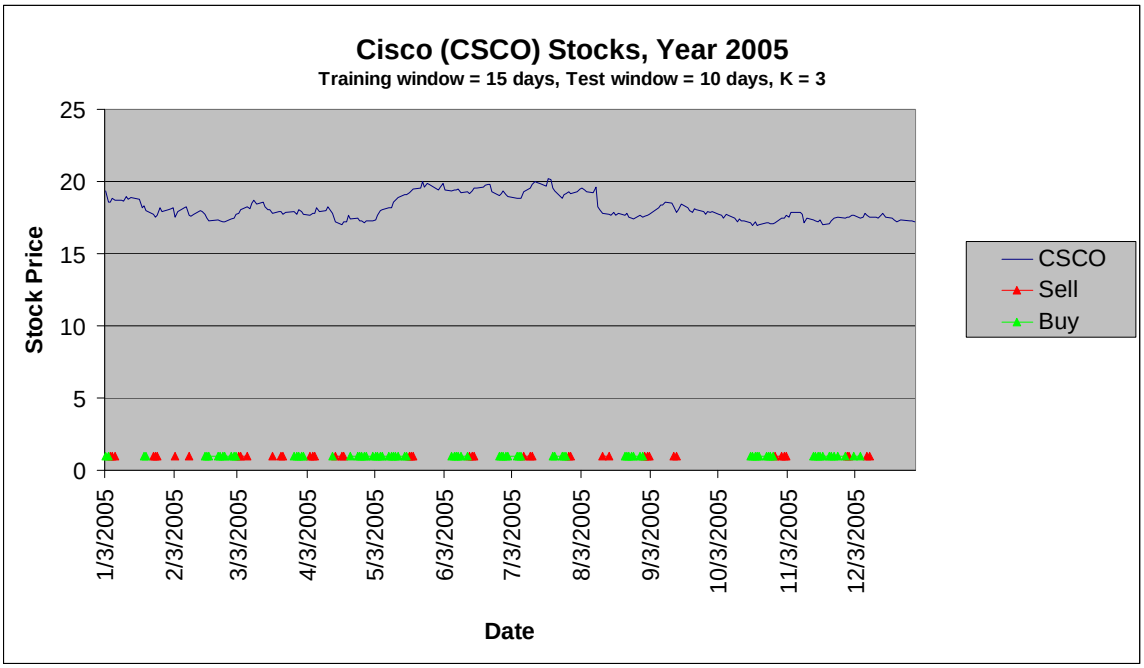

Figure 7: 2005 Stock Year, Cisco, parameters: w_{tr} = 15 days, w_{te} = 10 days, K = 3

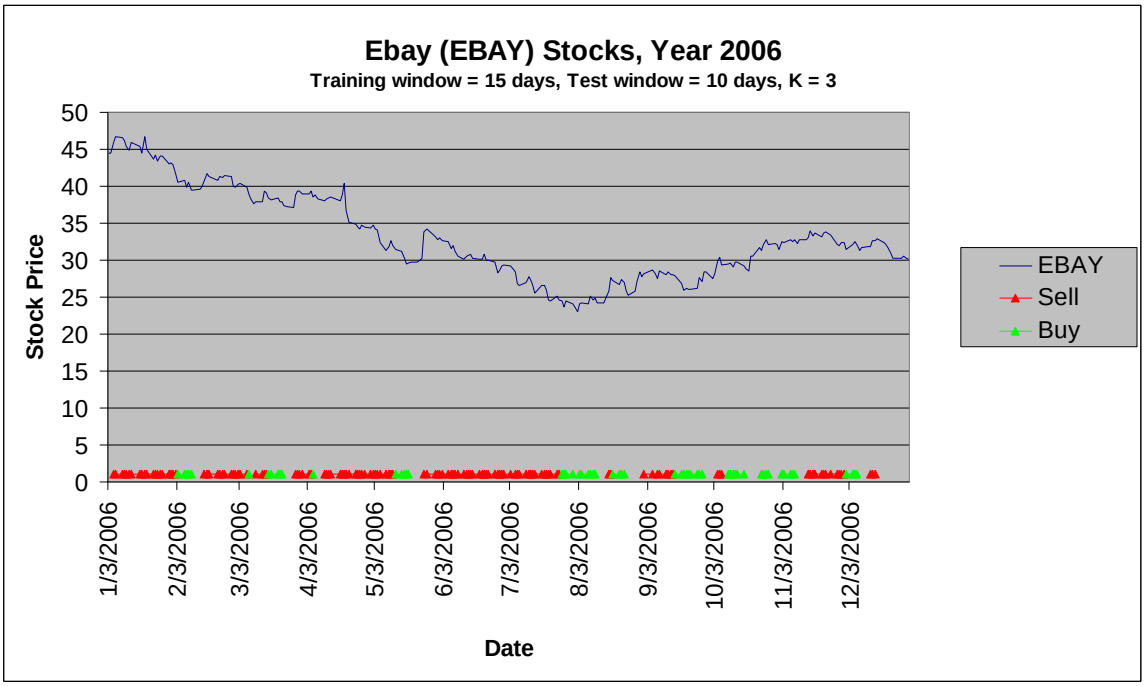

Figure 8: 2006 Stock Year, Ebay, parameters: w_{tr} = 15 days, w_{te} = 10 days, K = 3

The overall prediction results are accurate and track the respective stock movements quite closely. For the case of Apple and Ebay, the algorithm was able to closely predict the overall trend movement for a larger period movement.

For smaller spikes and trend changes such as the case of Cisco, the algorithm was less effective in determining the future movement.

There was not enough relevant data gathered from the selected window size and K-Clusters to predict the test data properly. Trend movements and the required granularity were insufficient and failed to provide accurate predictions with sporadic variances of the stock market. Moreover, the past training data provided insufficient data points to track the latter stock price movements.

To see the effects of both varying the window size and cluster value K, figure 9 plots the prediction signal for K = 3,4,6 for comparison

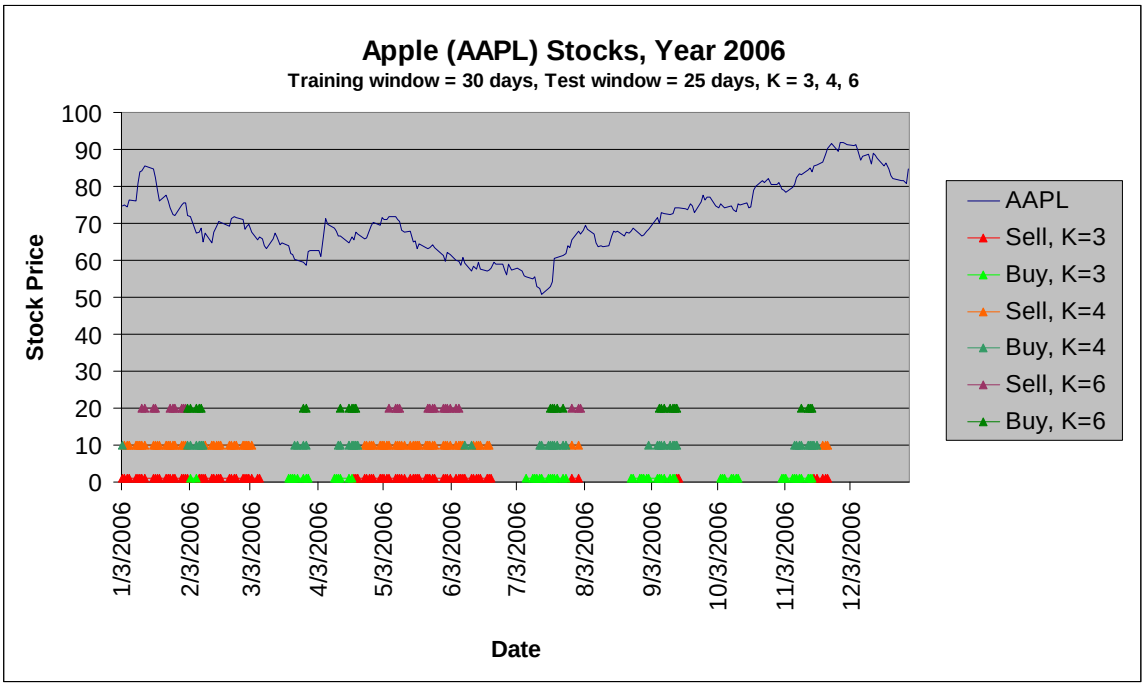

Figure 9: 2006 Stock Year, Apple, parameters: w_{tr} = 30 days, w_{te} = 25 days, K = 3, 4, 6

The larger window size of 30 days smoothed out the prediction signal. Comparing Figure 6 and Figure 9 for case K=3, the larger window removed some of the more granular trend predictions and provided a better trend prediction for larger trend movement periods. As the number of clusters K increased, the number of prediction signals decreased as a result of the smaller data time series sets per cluster group. The increase in clusters also had the effect of removing some of the granular trend predictions. In either case, the predominant trend prediction was not influenced greatly by changes in window size and number of clusters.

Improvements

Although the results are limited in predicting the overall trend of a stock, improvements are required to enhance the capabilities and prediction effectiveness.

First, k-means clustering can be improved by using a fuzzy clustering algorithm which takes into account a membership function and the degree to which a time series belongs to a cluster. Increased performance is dependent upon factors such as scalability and efficient clustering to improve the calculation speed for larger time series data and accurate real-time analysis.

Second, the data mining and pattern matching methods can be isolated to discover patterns in the data to remove unwanted outliers that are caused by unpredictable factors from the stock market. Also, introducing a scale change to pattern matching can discover similar patterns with different time scales.

Third, in the present work a simple gradient method was used to determine the trend. Further improvements can be realized by using a linear regression or probabilistic decision method.

Finally, the current algorithm can be modified to encompass various data that are correlated with market movement for example business cycle, inflation rates, interest rates, group movement, etc. The current algorithm can be combined with an existing prediction method that already takes these factors into account or integrated into the current algorithm. Figure 10 shows a block diagram of the improved stock trend analysis algorithm which takes into account other useful stock trend data that can help produce a better prediction.

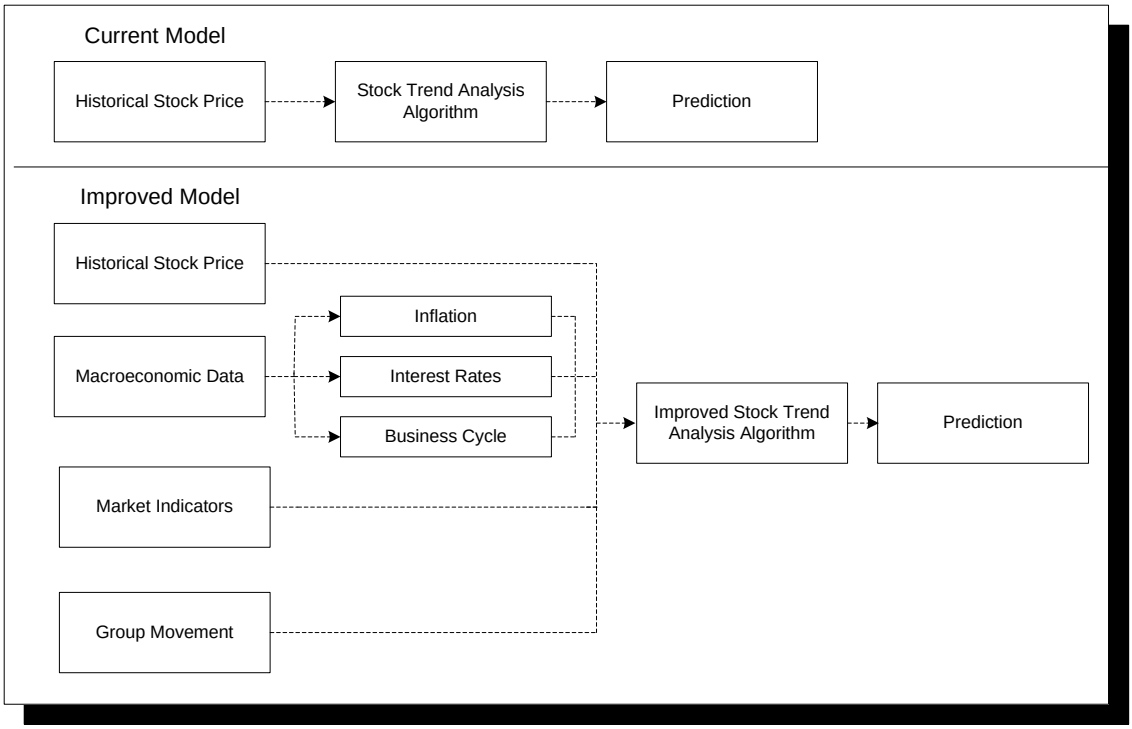

Figure 10: Modified model with Improved Stock Trend Analysis Algorithm

Conclusion

A stock trend analysis algorithm to analyze and predict the trend of a stock has been proposed. Results have shown that the algorithm provides prediction points which closely follow the overall movement of a stock. As a result, the algorithm can be used to create a trading strategy in itself or be a supplement to a trader's current trading strategy. In either case, the goal is to provide accurate prediction data points which lead to optimal stock trades and a higher return. While the algorithm correctly predicted the trend of stock prices for the given examples, it is unable to consistently predict accurately for differing variables. In conclusion, the old saying "history repeats itself" lives up to its meaning and historical data can be processed to forecast future movement accurately, in a volatile stock market.

References

D.C. Newton, Jr., J. Cunha, S. Da Silva, Stock Selecting Based on Cluster Analysis. Economics Bulletin, Vol. 13, No. 1 Page(s) 1-9, Oct. 2005.

H. He, J. Chen, H. Jin, et al. Stock Trend Analysis and Trading Strategy. JCIS-2006 Proceedings, Oct. 2006.

V. Niederhoffer. Clustering of Stock Prices. Operations Research, Vol. 13, No. 2 Page(s) 258-265, Mar.-Apr 1965.

W. K. Pratt. Digital Image Processing, Third Edition. John Wiley & Sons, Inc. 2001.

Y. Hiemstra. A Stock Market Forecasting Support System Based on Fuzzy Logic. System Sciences, 1994. Vol.III: Information Systems: Decision Support and Knowledge-Based Systems, Proceedings of the Twenty-Seventh Hawaii International Conference on Volume 3, Page(s):281 – 287, Issue 4-7 Jan 1994.