Expert Systems with Applications 40 (2013) 5501–5506

Contents lists available at SciVerse ScienceDirect

Expert Systems with Applications

journal homepage: www.elsevier.com/locate/eswa

A Bayesian regularized artificial neural network for stock market forecasting

Jonathan L. Ticknor *

Duke University, Pratt School of Engineering, 121 Hudson Hall, Durham, NC 27708, USA

ARTICLE INFO

Keywords:
Bayesian regularization
Neural network
Stock prediction
Overfitting

ABSTRACT

In this paper a Bayesian regularized artificial neural network is proposed as a novel method to forecast financial market behavior. Daily market prices and financial technical indicators are utilized as inputs to predict the one day future closing price of individual stocks. The prediction of stock price movement is generally considered to be a challenging and important task for financial time series analysis. The accurate prediction of stock price movements could play an important role in helping investors improve stock returns. The complexity in predicting these trends lies in the inherent noise and volatility in daily stock price movement. The Bayesian regularized network assigns a probabilistic nature to the network weights, allowing the network to automatically and optimally penalize excessively complex models. The proposed technique reduces the potential for overfitting and overtraining, improving the prediction quality and generalization of the network. Experiments were performed with Microsoft Corp. and Goldman Sachs Group Inc. stock to determine the effectiveness of the model. The results indicate that the proposed model performs as well as the more advanced models without the need for preprocessing of data, seasonality testing, or cycle analysis.

1. Introduction

Stock price prediction has recently garnered significant interest among investors and professional analysts. The prediction of stock market trends is very complex, owing to the inherent noisy environment and high volatility related to daily market trends. The complexity of the market and stock price movement is related to a number of factors including political events, market news, quarterly earnings reports, and conflicting trading behavior. Traders often rely on technical indicators based on stock data that can be collected on a daily basis. Despite the availability of these indicators, it is often difficult to predict daily to weekly trends in the market.

In the past two decades, a large body of research for predicting stock market returns has been developed. This body of knowledge contains many artificial intelligence (AI) approaches, namely artificial neural networks (ANN). A number of these networks have utilized feedforward neural networks to predict stock trends (Baba & Kozaki, 1992; Chenoweth & Obradovic, 1996; Fernandez-Rodriguez, Gonzalez-Martel, & Sosvilla-Rivebo, 2000; Ghiassi, Saidane, & Zimbra, 2005; Hamzacebi, Akay, & Kutay, 2009; Lendasse, De Bodt, Wertz, & Verleysen, 2000; Roh, 2007; Walezak, 1999). Leung, Daouk, and Chen (2000) analyzed a number of parametric and nonparametric models for predicting stock market trends and returns. The empirical results suggested that the probabilistic neural network (classification model) outperforms the standard feedforward neural network (level estimation model) in predicting stock trends.

One drawback of using standard back propagation networks is the potential for overfitting the training data set, which results in reduced accuracy on unknown test sets. To address the potential overfitting of neural network weights, some researchers have developed hybridized neural network techniques (Chu, Chen, Cheng, & Huang, 2009; Leigh, Purvis, & Ragusa, 2002; Oh & Kim, 2002). Hassan, Nath, and Kirley (2007) utilized a hybrid model that included Hidden Markov Model (HMM), Artificial Neural Networks (ANN), and genetic algorithm (GA) to predict the price of three major stocks. The results of their study indicated that the next day stock price could be predicted to within a 2% of the actual value, a significant improvement over other standard models.

In recent years, a number of techniques including fuzzy logic, genetic algorithms, and biologically based algorithms have been utilized to improve feedforward neural networks (Armano, Marchesi, & Murru, 2004; Atsalakis & Valvanis, 2009; Blanco, Delgado, & Pegalajar, 2001; Chang & Liu, 2008; Chang, Wang, & Zhou, 2012; Chen, Yanga, & Abraham, 2006; Ritanjali & Panda, 2007; Yudong & Lenan, 2009). Neural networks and support vector machines were compared by Kara, Boyacioglu, and Baykan (2011) to predict stock price index movement, the results of which showed the superiority of neural networks. These new techniques have im-

* Tel.: +1 919 660 5030.
E-mail address: jonathan.ticknor@duke.edu

0957-4174/$ - see front matter © 2013 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.eswa.2013.04.013

proved the capability of the original back propagation neural network with some success and reduced some of the drawbacks of ANNs related to noise fitting.

In this paper the feedforward artificial neural network coupled with Bayesian regularization is introduced as a novel approach to predicting stock market trends. The networks ability to predict stock trends is shown for two major United States stocks and is compared to another advanced hybrid technique. The simulations indicate that the weight optimization technique offers an enhanced level of performance compared to other advanced techniques. A brief discussion of Bayesian regularization is included to provide a brief background on the equations governing the technique and its potential applications in financial time series forecasting.

2. Bayesian regularization of neural networks

The back propagation neural network is a very popular technique in the field of ANN which relies on supervised learning, typically through the use of a gradient descent method to reduce a chosen error function (e.g. mean squared error). Fig. 1 provides a general architecture for a feedforward neural network, including the three main layers of the model: input, hidden, and output. The connection between neurons in Fig. 1 in each layer is termed a link. This link is stored as a weighted value which provides a measure of the connection between two nodes (Garson, 1991; Goh, 1995). The supervised learning step changes these weights in order to reduce the chosen error function, generally mean squared error, in order to optimize the network for use on unknown samples. A major issue for these techniques is the potential for overfitting and overtraining which leads to a fitting of the noise and a loss of generalization of the network.

To reduce the potential for overfitting, a mathematical technique known as Bayesian regularization was developed to convert nonlinear systems into "well posed" problems (Burden & Winkler, 2008; MacKay, 1992). In general, the training step is aimed at reducing the sum squared error of the model output and target value. Bayesian regularization adds an additional term to this equation:

$$F = \beta E_D + \alpha E_w \tag{1}$$

where F is the objective function, E_D is the sum of squared errors, E_w is sum of square of the network weights, and α and β are objective function parameters (MacKay, 1992). In the Bayesian network the weights are considered random variables and thus their density function is written according the Baye's rules (Forsee & Hagan, 1997):

$$P(w|D,\alpha,B,M) = \frac{P(D|w,\beta,M)P(w|\alpha,M)}{P(D|\alpha,\beta,M)} \tag{2}$$

where w is the vector of network weights, D represents the data vector, and M is the neural network model being used. Forsee and Hagan (1997) assumed that the noise in the data was Gaussian, and with this assumption were able to determine the probability density function for the weights. The optimization of the regularization parameters α and β require solving the Hessian matrix of $F(w)$ at the minimum point w^{MP}. Forsee and Hagan (1997) proposed a Gauss–Newton approximation to the Hessian matrix which is possible if the Levenburg–Marquardt training algorithm is used to locate the minimum. This technique reduces the potential for arriving at local minima, thus increasing the generalizability of the network.

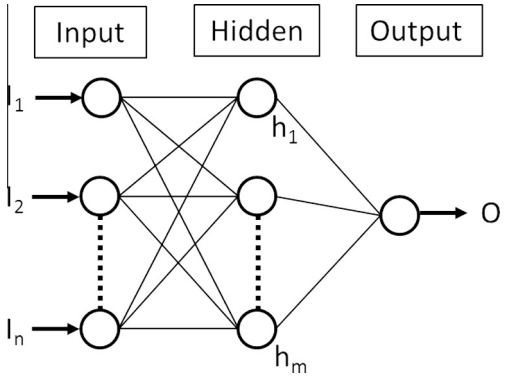

Fig. 1. General architecture of three-layer feedforward neural network.

The novelty of this technique is the probabilistic nature of the network weights in relation to the given data set and model framework. As a neural network grows in size through additional hidden layer neurons, the potential for overfitting increases dramatically and the need for a validation set to determine a stopping point is crucial. In Bayesian regularized networks, overly complex models are penalized as unnecessary linkage weights are effectively driven to zero. The network will calculate and train on the nontrivial weights, also known as the effective number of parameters, which will converge to a constant as the network grows (Burden & Winkler, 2008). The inherent noise and volatility of stock markets introduces a high probability of overfitting and overtraining for general back propagation networks. These more parsimonious networks reduce the chance of overfitting while eliminating the need for a validation step, thus increasing the available data for training.

3. Prediction model

In this paper, a three layered feedforward ANN was used to predict daily stock price movements. The model consisted of an input layer, hidden layer, and output layer depicted in Fig. 1. Inputs to the neural network included daily stock data (low price, high price, opening price) and six financial indicators, which were represented with nine neurons in the input layer. The output of the network was the next day closing price of the chosen stock. The number of neurons in the hidden layer was chosen empirically by adjusting the number of neurons until the effective number of parameters reached a constant value.

The layer weights of the network were optimized according the procedure of Forsee and Hagan (1997), using mean squared error as the performance evaluation function. A tangent sigmoid function was chosen for the hidden layer, meaning the return value will fall in the range $[-1,1]$, using the following function (Harrington, 1993):

$$\tanh(n) = \frac{2}{1+\exp(-2n)} - 1 \tag{3}$$

The architecture of the network and effective number of parameters are shown in Table 1, where a hidden layer with five neurons was deemed to be sufficiently large. The network was permitted to train for no more than 1000 epochs, at which point the optimization routine would cease.

Neural network training is best suited to data that falls within a small specified range, generally produced by some normalization procedure. In this study, data was normalized within the range $[-1,1]$ by the use of the "mapminmax" function in MATLAB software. This normalization procedure is utilized so that larger input values do not overwhelm smaller inputs, helping reduce network error (Kim, 2003). A reverse of this procedure is performed on

Table 1
ANN parameters for stocks in initial experiment.

Stock	Neurons	Layers	Effective number of parameters
Microsoft (MSFT)	5	3	20.0
Goldman Sachs (GS)	5	3	20.2

the output data to obtain the stock value in the appropriate units before comparison to actual data.

4. Simulation experiments

4.1. Test data

The research data used for stock market predictions in this study was collected for Goldman Sachs Group, Inc. (GS) and Microsoft Corp. (MSFT). The total number of samples for this study was 734 trading days, from 4 January 2010 to 31 December 2012. Each sample consisted of daily information including low price, high price, opening price, close price, and trading volume. The training data set was chosen as the first 80% of the samples, while the testing data comprised the remaining 20% of the samples. The neural network model was used to predict the price of the chosen stock one day in the future. All of the available data was utilized to estimate the appropriate size of the network, increasing the number of hidden layer neurons until the effective number of parameters converged to a constant value.

A comparison study was also performed to test the efficacy of the network in this study to an advanced hybrid model by Hassan et al. (2007). For this experiment, data was collected for Apple Inc. (AAPL) and International Business Machines Corp. (IBM). The total number of samples for this study was 492 trading days from 10 February 2003 to 21 January 2005. The training set for this experiment includes trading days from 10 February 2003 to 10 September 2004, while the testing data includes 91 trading days from 13 September 2004 to 21 January 2005. These trading and testing periods were chosen to ensure uniformity in experiment conditions for both models.

4.2. Technical indicators

Six technical indicators were used in this study as part of the input variables to the network. Technical indicators are often used by investors in the stock market as a potential mechanism for predicting market trends (Kim, 2003). A number of indicators have been developed, though only a subset of the major indicators were used in this study. A general search of the existing literature helped deduce the major indicators to be used in this study (Yudong & Lenan, 2009; Kara et al., 2011; Kim & Han, 2000). The indicators used in this study, including their formulae, are summarized in Table 2. All of these indicators are computed from the raw data collected for each stock.

5. Results and discussion

5.1. Network size determination

The first stage of the experiment involved determining the appropriate network size for the Bayesian regularized network. The data for GS and MSFT were analyzed separately, utilizing the full data set. The number of hidden layer neurons was adjusted at each iteration until a constant number of effective parameters were found. The total hidden layer neurons were adjusted between two and ten, in which a value of five was found to be sufficiently large as shown in Table 1.

5.2. Network output for test stocks

The results of the network prediction are shown in Fig. 2, which depicts the one day future actual stock value and predicted stock

Table 2
Selected technical indicators and their formulas.

Name of indicator	Indicator formula
Exponential moving average (5 and 10 day)	$EMA(h)_{t-1} + \alpha \times (C_t - EMA(h)_{t-1})$
Relative strength index (RSI)	$100 - \frac{100}{1+(\sum_{i=0}^{n-1} Up_{t-i}/n)/(\sum_{i=0}^{n-1} Dw_{t-i}/n)}$
Williams R%	$\frac{H_n - C_t}{H_n - L_n} \times 100$
Stochastic K%	$\frac{C_t - L_{t-n}}{HH_{t-n} - LL_{t-n}} \times 100$
Stochastic D%	$\frac{\sum_{i=0}^{n-1} K_{t-i}\%}{n}$

C_t is the closing price, L_t is the low price, H_t is the high price at time t.
Smoothing factor, $\alpha = 2/(1+h)$, h is the time period for h day moving average (i.e. 5 or 10).
HH_t and LL_t are highest high and lowest low in the last t days.
Up_t means upward price change; Dw_t means downward price change at time t.

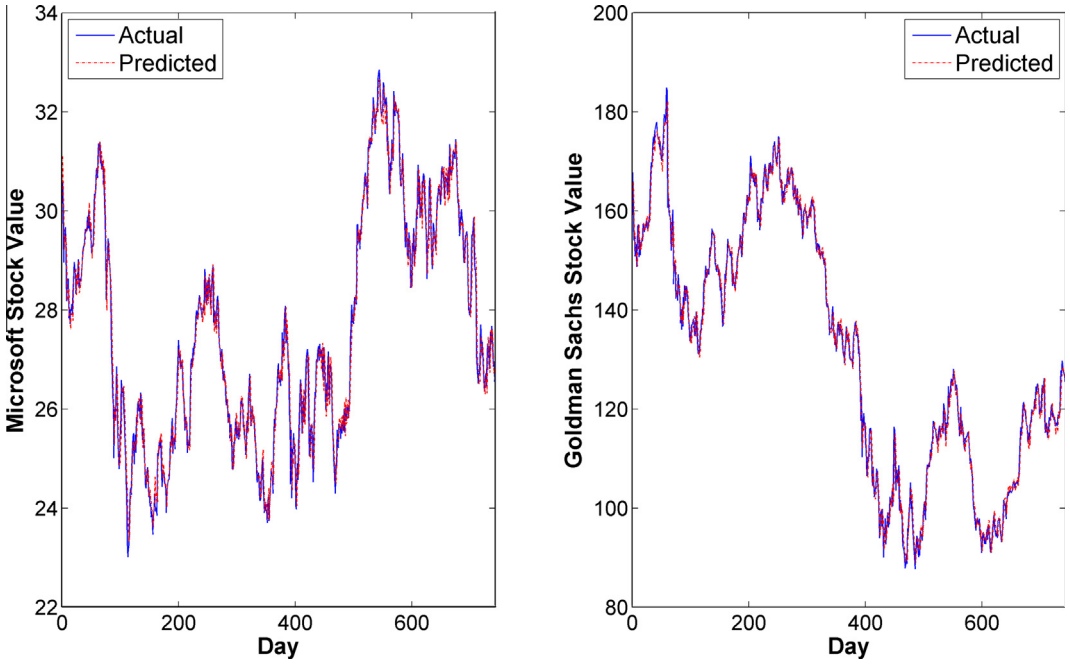

Fig. 2. One day future stock price versus trading day: model prediction and actual stock price.

value. The Bayesian regularized network provides on average a >98% fit to future stock prices for both stocks over the full trading period. The technology and banking stocks were chosen because of the differences in industry market behavior and volatility. Microsoft stock was highly volatile over the length of the trading period as well as in daily trading. The proposed model was able to handle this noise and volatility without overfitting the data which can be seen in the fit for testing data (points beyond day 600). This result provides evidence that the model can handle large datasets with significant noise and volatility while maintaining generalizability. Figs. 3 and 4 shows the results of this experiment plotted as target stock price versus predicted price. The dashed line indicates a line with a slope of one, which would indicate a perfect fit. The reduced error in the training data is expected, however, the limited spread in the testing data provides evidence that the model is providing appropriate generalization.

5.3. Performance metric

The performance of the Bayesian regularized artificial neural network was measured by computing the mean absolute percentage error (MAPE). This performance metric has been used in a number of studies and provides an effective means of determining the robustness of the model for predicting daily trends (Chang &

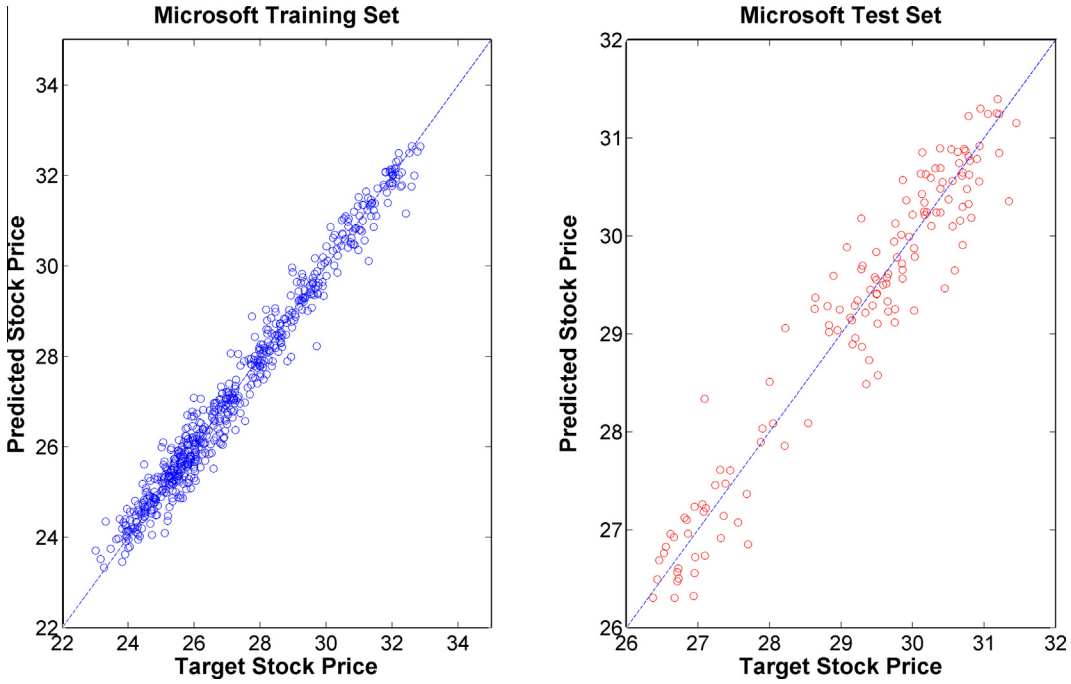

Fig. 3. Comparison of microsoft target and predicted stock price – one day future value.

Fig. 4. Comparison of Goldman Sachs target and predicted stock price – one day future value.

Table 3
Forecast accuracy of the Bayesian regularized ANN.

Stock	Training MAPE (%)	Test set MAPE (%)	Total MAPE (%)
Microsoft (MSFT)	1.0494	**1.0561**	1.0507
Goldman Sachs (GS)	1.5235	**1.3291**	1.4860

Table 4
Forecast accuracy comparison with Hassan et al. (2007).

Stock name	Mean absolute % error (MAPE) in forecast for 91 test dataset		
	Proposed Bayesian ANN	Fusion model with weighted average[a]	ARIMA model[a]
Apple Inc. (AAPL)	**1.9688**	1.9247	1.8009
IBM Corporation (IBM)	**0.7441**	0.8487	0.9723

[a] Models from Hassan et al. (2007)

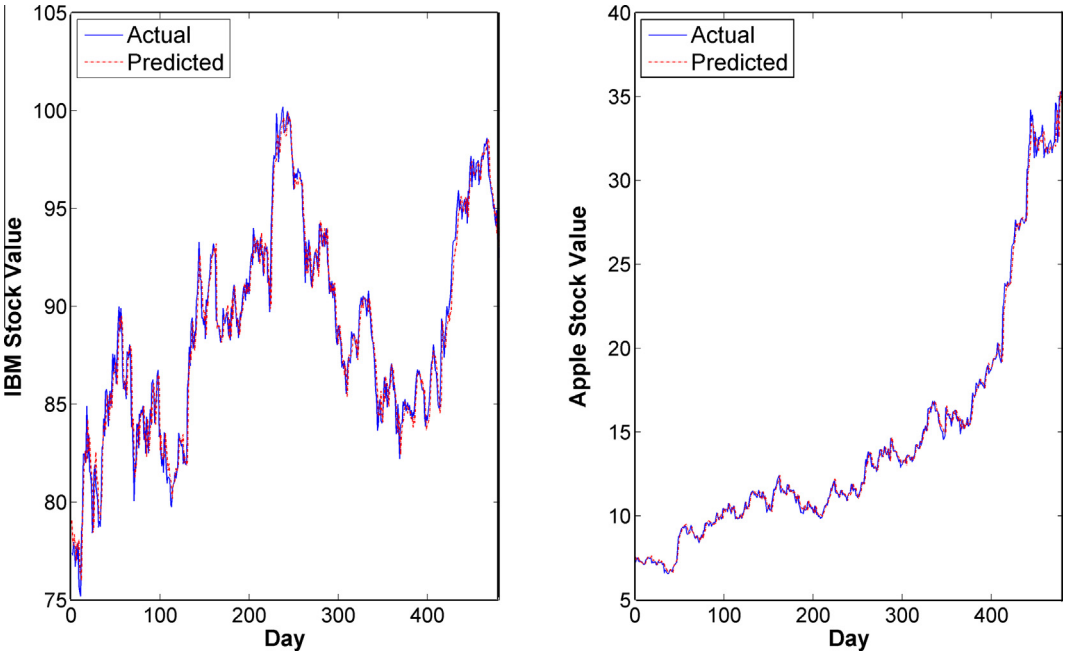

Fig. 5. One day future stock price prediction for comparison with Hassan et al. (2007).

Liu, 2008; Chang et al., 2012; Hassan et al., 2007). This metric is first calculated by finding the absolute value of the deviation between the actual stock price and the predicted stock price. This value is then divided by the actual stock price and multiplied by 100 to determine the percentage error of the individual data point. This procedure is performed over the entire trading space, summed, and then divided by total trading days to arrive at the MAPE value. The MAPE value is computed using the following equation:

$$MAPE = \frac{\sum_{i=1}^{r}(abs(y_i - p_i)/y_i)}{r} \times 100\% \quad (4)$$

where r is the total number of trading days, y_i is the actual stock price on day i, and p_i is the predicted stock price on day i.

5.4. Results

Table 3 presents the experimental results for the two stocks (GS and MSFT) chosen for this study. The MAPE value was calculated for the training, testing, and total data sets to monitor the effectiveness of the model at predicting unknown data. It is interesting to note the small MAPE value for each stock during the testing phase. These values indicate that on average, the Bayesian regularized network can predict the stock price to within 98% accuracy.

In order to test the efficacy of this approach, the network was tested against the same data set used in Hassan et al. (2007) to directly compare the two models. After a thorough analysis, the network architecture for this model included twenty hidden neurons, as opposed to the five neurons in the first experiment. The results of this experiment are shown in Table 4 and plotted in Fig. 5. The results show that Bayesian regularized network forecasts are as good as both the fusion model and the ARIMA model in Hassan et al. (2007). The data from Tables 3 and 4 showed that the proposed model is an effective technique for estimating next day closing prices and can provide a simplified financial time series prediction tool compared to other similar advanced neural network techniques.

6. Conclusions

In this paper a novel time series forecasting tool was introduced. The model combines Bayesian regularization with the Levenberg–Marquardt algorithm to forecast the movement of stock prices. The results of this model indicate that this tool reduces the potential for overfitting and local minima solutions that commonly plague neural network techniques. The probabilistic

nature of the network weights allows the investor to safely expand the network without increasing the risk of overfitting through the use of the effective number of parameters. The proposed model showed strong generalization with respect to recent stock trends for two companies in varying market domains. To evaluate the efficacy of this model, the network was compared to an advanced hybrid neural network model by Hassan et al. (2007). The results indicate that the forecasting ability of the model in this study is comparable, and potentially superior, to the fusion model and ARIMA model. Prediction of stock market trends is very important for the development of effective trading strategies. This model is a novel approach to solving this problem type and can provide traders a reliable technique for predicting future prices. It is important to note that addition or substitution of other technical indicators could improve the quality of this model in future applications.

Acknowledgements

This work was supported by the Department of Energy Office of Science Graduate Fellowship Program (DOE SCGF), made possible in part by the American Recovery and Reinvestment Act of 2009, administered by ORISE-ORAU under contract no. DE-AC05-06OR23100. The views expressed in this work do not represent those of the Department of Energy or the Office of Science and are solely those of the author.

References

Armano, G., Marchesi, M., & Murru, A. (2004). A hybrid genetic-neural architecture for stock indexes forecasting. *Information Sciences, 170*, 3–33.

Atsalakis, G. S., & Valvanis, K. P. (2009). Surveying stock market forecasting techniques – Part II: Soft computing methods. *Expert Systems with Applications, 36*, 5932–5941.

Baba, N., & Kozaki, M. (1992). An intelligent forecasting system of stock price using neural networks. In *Proceedings of the IEEE international joint conference on neural networks* (pp. 371–), http://dx.doi.org/10.1109/IJCNN.1992.287183..

Blanco, A., Delgado, M., & Pegalajar, M. C. (2001). A real-coded genetic algorithm for training recurrent neural networks. *Neural Networks, 14*, 93–105.

Burden, F., & Winkler, D. (2008). Bayesian regularization of neural networks. *Methods in Molecular Biology, 458*, 25–44.

Chang, P., & Liu, C. (2008). A TSK type fuzzy rule based system for stock price prediction. *Expert Systems with Applications, 34*, 135–144.

Chang, P., Wang, D., & Zhou, C. (2012). A novel model by evolving partially connected neural network for stock price trend forecasting. *Expert Systems with Applications, 39*, 611–620.

Chen, Y., Yanga, B., & Abraham, A. (2006). Flexible neural trees ensemble for stock index modeling. *Neurocomputing, 70*, 697–703.

Chenoweth, T., & Obradovic, Z. (1996). A multi-component nonlinear prediction system for the S&P 500 index. *Neurocomputing, 10*, 275–290.

Chu, H. H., Chen, T. L., Cheng, C. H., & Huang, C. C. (2009). Fuzzy dual-factor time-series for stock index forecasting. *Expert Systems with Applications, 36*, 165–171.

Fernandez-Rodriguez, F., Gonzalez-Martel, C., & Sosvilla-Rivebo, S. (2000). On the profitability of technical trading rules based on artificial neural networks: Evidence from the Madrid stock market. *Economics Letters, 69*, 89–94.

Forsee, F.D., & Hagan, M.T. (1997). Gauss-Newton approximation to Bayesian learning. In *1997 IEEE international conference on neural networks* (Vols. 1–4, pp. 1930–1935). Houston, TX, USA, http://dx.doi.org/10.1109/ICNN.1997.614194.

Garson, G. D. (1991). Interpreting neural-network connection weights. *AI Expert, 6*, 47–51.

Ghiassi, M., Saidane, H., & Zimbra, D. K. (2005). A dynamic artificial neural network model for forecasting time series events. *International Journal of Forecasting, 21*, 341–362.

Goh, A. T. C. (1995). Back-propagation neural networks for modeling complex systems. *Artificial Intelligence in Engineering, 9*, 143–151.

Hamzacebi, C., Akay, D., & Kutay, F. (2009). Comparison of direct and iterative artificial neural network forecast approaches in multi-periodic time series forecasting. *Expert Systems with Applications, 36*, 3839–3844.

Harrington, P. B. (1993). Sigmoid transfer functions in backpropagation neural networks. *Analytical Chemistry, 65*, 2167–2168.

Hassan, M. R., Nath, B., & Kirley, M. (2007). A fusion model of HMM, ANN, and GA for stock market forecasting. *Expert Systems with Applications, 33*, 171–180.

Kara, Y., Boyacioglu, M. A., & Baykan, O. K. (2011). 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, 38*, 5311–5319.

Kim, K. (2003). Financial time series forecasting using support vector machines. *Neurocomputing, 55*, 307–319.

Kim, K., & Han, I. (2000). Genetic algorithms approach to feature discretization in artificial neural networks for prediction of stock price index. *Expert Systems with Applications, 19*, 125–132.

Leigh, W., Purvis, R., & Ragusa, J. M. (2002). Forecasting the NYSE composite index with technical analysis, pattern recognizer, neural network, and genetic algorithm: A case study in romantic decision support. *Decision Support Systems, 32*, 361–377. http://dx.doi.org/10.1109/IJCNN.1992.287183.

Lendasse, A., De Bodt, E., Wertz, V., & Verleysen, M. (2000). Non-linear financial time series forecasting application to the Bel 20 stock market index. *European Journal of Economical and Social Systems, 14*, 81–91.

Leung, M. T., Daouk, H., & Chen, A. S. (2000). Forecasting stock indices: A comparison of classification and level estimation models. *International Journal of Forecasting, 16*, 173–190.

MacKay, D. J. C. (1992). A practical Bayesian framework for backpropagation networks. *Neural Computation, 4*, 448–472.

Oh, K. J., & Kim, K. J. (2002). Analyzing stock market tick data using piecewise nonlinear model. *Expert Systems with Applications, 2*, 249–255.

Ritanjali, M., & Panda, G. (2007). Stock market prediction of S&P 500 and DJIA using bacterial foraging optimization technique. In *2007 IEEE congress on evolutionary computation* (pp. 2569–2579), http://dx.doi.org/10.1109/CEC.2007.4424794.

Roh, T. H. (2007). Forecasting the volatility of stock price index. *Expert Systems with Applications, 33*, 916–922.

Walezak, S. (1999). Gaining competitive advantage for trading in emerging capital markets with neural networks. *Journal of Management Information Systems, 16*, 178–194.

Yudong, Z., & Lenan, W. (2009). Stock market prediction of S&P 500 via combination of improved BCO approach and BP neural network. *Expert Systems with Applications, 36*, 8849–8854.