Hindawi Publishing Corporation
Advances in Artificial Neural Systems
Volume 2014, Article ID 246487, 7 pages
http://dx.doi.org/10.1155/2014/246487

Research Article

A Hybrid Intelligent Method of Predicting Stock Returns

Akhter Mohiuddin Rather

Woxsen School of Business, Sadasivpet, Kamkol, Hyderabad 502291, India

Correspondence should be addressed to Akhter Mohiuddin Rather; akhter.aki@gmail.com

Received 16 May 2014; Revised 26 August 2014; Accepted 26 August 2014; Published 7 September 2014

Academic Editor: Ozgur Kisi

This paper proposes a novel method for predicting stock returns by means of a hybrid intelligent model. Initially predictions are obtained by a linear model, and thereby prediction errors are collected and fed into a recurrent neural network which is actually an autoregressive moving reference neural network. Recurrent neural network results in minimized prediction errors because of nonlinear processing and also because of its configuration. These prediction errors are used to obtain final predictions by summation method as well as by multiplication method. The proposed model is thus hybrid of both a linear and a nonlinear model. The model has been tested on stock data obtained from National Stock Exchange of India. The results indicate that the proposed model can be a promising approach in predicting future stock movements.

1. Introduction

Prediction of stock returns has attracted many researchers in the past and at present it is still an emerging area both in academia and in industry. Mathematically, the techniques involved in obtaining prediction of stock returns can be broadly classified into two categories. The first category involves linear models such as autoregressive moving average models, exponential smoothing, linear trend prediction, random walk model, generalized autoregressive conditional heteroskedasticity, and stochastic volatility model [1]. The second category involves those models which are based on artificial intelligence such as artificial neural networks (ANNs) [2], support vector machines [3], genetic algorithms (GA), and particle swarm optimization (PSO) [4]. Linear models have a common limitation associated with them, that is, their linear feature which prevents them from detecting nonlinear patterns of data. Due to instability in stock market, the stock data is volatile in nature; thus, linear models are unable to detect nonlinear patterns of such data. Nonlinear models overcome the limitations of linear models, as ANNs embody useful nonlinear functions which are able to detect nonlinear patterns of data [5]. As a consequence, prediction performance improves by using nonlinear models [6, 7].

A lot of work has been done in this field; for instance, radial basis neural network was used for stock prediction of Shanghai Stock Exchange, wherein artificial fish swarm optimization was introduced so as to optimize radial basis function [8]. In time series prediction, ANNs have received overwhelming attention from researchers. For instance, Freitas et al. [9], Wang et al. [10], Khashei and Bijari [11], Chen et al. [12], and Jain and Kumar [13] report the use of ANN in time series stock prediction. A new approach called the wavelet denoising based backpropagation neural network was proposed for predicting stock prices [14]. In another work, researchers explored the use of activation functions in ANN, and to improve the performance of ANNs, they suggest the use of three new simple functions; financial time series data was used in the experiments [15]. For seasonal time series prediction, an ANN was proposed, which considers the seasonal time periods in time series. The purpose of such consideration is to determine the number of input and output neurons [16]. Multilayer perceptron and generalized regression neural networks were used to predict the Kuwait Stock Exchange [17]. The results showed that the models were useful in predicting stock exchange movements in emerging markets. Different ANN models were proposed that are able to capture temporal aspects of inputs [5]. For time series predictions, two types of ANN models are proved to be successful: time-lagged feedforward networks and dynamically driven recurrent (feedback) networks [18].

ANNs are not always guaranteed to yield desired results. In order to solve such problems, the researchers attempted to find global optimization approach of ANN to predict the stock price index [19]. With the goal to further improve the performance of predictors, researchers have also attempted to develop hybrid models for prediction of stock returns. The hybridization may include integration of linear and nonlinear models [10, 20, 21]. Researchers developed a hybrid forecasting model by integrating recurrent neural network based on artificial bee colony and wavelet transforms [22]. In another work, an adaptive network-based fuzzy inference system was used so as to develop a hybrid prediction model [23]. A hybrid system was proposed by integrating linear autoregressive integrated moving average model and ANN [11]. To overcome the prediction problem with the use of technical indicators, researchers proposed a hybrid model by merging ANNs and genetic programming [24]. Feature selection was also used to improve the performance of the system. In another work, random walk prediction model was merged with ANN and a hybrid model was thus developed [25]. Researchers also introduced a hybrid ANN architecture of particle swarm optimization and adaptive radial basis function [26]. Various regressive ANN models, such as self organising maps and support vector regressions, were used to form a hybrid model so as to predict foreign exchange currency rate [27]. For business failure prediction, a hybrid model was proposed in which particle swarm optimization and network-based fuzzy inference system were used [28]. In a recent work, researchers created a hybrid prediction model by integrating autoregressive moving average model and differential evolution based training of its feedforward and feedback parameters [29]. This hybrid model has been compared with other similar hybrid models and the results confirm that it outperforms other models.

The rest of the paper is arranged as follows. Section 2 discusses various prediction based models, including those models which are used in this work. The proposed hybrid model is described in detail in Section 3. Section 4 discusses two commonly used error metrics. In Section 5, experiments and results are presented. Finally, Section 6 presents conclusions.

2. Prediction Based Models

Stock return or simply return is used to refer to a profit on an investment. Return, R_t at time, is calculated using (1), where p_t and p_{t-1} are prices of stock at times t and $t-1$ respectively:

$$R_t = \frac{p_t - p_{t-1}}{p_{t-1}}. \tag{1}$$

Few prediction-based models available in literature including those used in this work are described below.

2.1. Exponential Smoothing Model. Exponential smoothing model is used to obtain predictions on time series data Brown [30]. It computes one-step-ahead prediction by means of computing geometric sum of past observations as shown in the following equation:

$$\hat{r}_{t+1} = \hat{r}_t + \alpha\left(r_t - \hat{r}_t\right), \tag{2}$$

where $\hat{r}_t$ is prediction of r_t, $\hat{r}_{t+1}$ is prediction for future value, α is a smoothing parameter in the range of $(0, 1)$, and $r_t - \hat{r}_t$ is prediction error. Exponential smoothing assigns exponentially decreasing weights over time. The observations are weighted, with more weight given to the most recent observations.

2.2. Autoregressive Moving Average Model. Autoregressive moving average model was introduced by Box and Jenkins [31], for times series prediction, which was actually inspired by the early work of Yule [32] and Wold [33]. The model consists of an autoregressive part AR(p) and moving average part MA(q). This is the reason why the model is referred to as ARMA(p, q) model. The autoregressive moving average model is thus defined as AR(p) process and it can be generally expressed as shown in the following equation:

$$r_t = \phi_1 r_{t-1} + \phi_2 r_{t-2} + \cdots + \phi_p r_{t-p} + \varepsilon_t. \tag{3}$$

Similarly, MA(q) process can be generally expressed as shown in the following equation:

$$r_t = \varepsilon_t - \theta_1 \varepsilon_{t-1} - \theta_2 \varepsilon_{t-2} - \cdots - \theta_q \varepsilon_{t-q}. \tag{4}$$

Hence, the ARMA(p, q) model can be expressed as

$$\begin{aligned} r_t = {} & \phi_1 r_{t-1} + \phi_2 r_{t-2} + \cdots + \phi_p r_{t-p} + \varepsilon_t - \theta_1 \varepsilon_{t-1} \\ & - \theta_2 \varepsilon_{t-2} - \cdots - \theta_q \varepsilon_{t-q}, \end{aligned} \tag{5}$$

where p and q represent the order of the autoregressive model and of the moving average model, respectively. ϕ and θ are coefficients that satisfy stationarity of series. ε_t are random errors or white noise at time t with zero mean and variance σ_ε^2.

2.3. Autoregressive Moving Reference Regression Model. This is a new regression model proposed by Freitas et al. [9] and is adopted in this work also. The same model has been used by [34, 35] along with different ANN models.

Let us consider time series data with T past returns as shown in the following equation:

$$r' = \left(r_{t-(T-1)}, \ldots, r_{t-1}, r_t\right). \tag{6}$$

Based on the available historical data, the future return of an action can be defined as the process in which the elements of the past returns are used to obtain an estimate of future return r_{t+l}, where $l \geq 1$. The value of l directly affects the choice of the adopted prediction method used.

Input-output pairs are generated by this model which are given to ANN in a supervised manner; thus, it can be called autoregressive moving reference neural network, AR-MRNN(p, k), where p is the order of regression and k is delay from the point of reference.

Using an autoregressive predictor, say $\mathfrak{R}$, AR-MRNN (p,k) implements the prediction system as shown in the following equation:

$$\widehat{r_{t+1}-z} = \mathfrak{R}\left(r_{t-(p-1)}-z,\ldots,r_{t-1}-z,r_t-z\right), \tag{7}$$

where $\widehat{r_{t+1}-z}$ is prediction for $r_{t+1}-z$ obtained at time t from the information available from the historical series $r' = (r_{t-(T-1)},\ldots,r_{t-1},r_t)$. p is the order of regression and z is moving reference given in the following equation:

$$z = r_{t-(p-1)-k}. \tag{8}$$

After training the neural network, prediction $\widehat{r}_{t+1}$ is obtained as shown in the following equation:

$$\widehat{r}_{t+1} = \widehat{r_{t+1}-z} + z. \tag{9}$$

According to (7), inputs to the neural network are given as differences rather than original values, and the network thus requires smaller values of weights which improves its ability to generalize. The output obtained from neural network is not the final predictions, rather than final predictions which are calculated by adding z value to the output.

3. The Proposed Hybrid Intelligent Prediction Model

This section discusses the proposed hybrid intelligent prediction model (HIPM) in detail. Consider the actual returns of time series given by r_t $(t = 1,\ldots,T)$. Let the predictions obtained by any linear model be denoted by $\widehat{r}_t$ $(t = 1,\ldots,T)$. The difference between actual time series (r_t) and predicted series $(\widehat{r}_t)$ is known as prediction error or simply error, which is calculated here in a similar fashion, that is, $\varepsilon_t = r_t - \widehat{r}_t$. In HIPM, the predictions are obtained via two methods, that is, summation method and multiplication method; these methods are defined below.

3.1. Summation Method. According to summation method, actual data is equal to predicted linear data summed up with error terms, as shown in the following equation:

$$r_t = \widehat{r}_t + \varepsilon_t. \tag{10}$$

Error terms are thus calculated as shown in the following equation:

$$\varepsilon_t = r_t - \widehat{r}_t. \tag{11}$$

3.2. Multiplicative Method. For multiplicative method, actual data is equal to predicted linear data multiplied with error terms, as shown in the following equation:

$$r_t = \widehat{r}_t \times \varepsilon_t. \tag{12}$$

The error terms are multiplied back to linear predictions because these error terms were calculated as shown in the following equation:

$$\varepsilon_t = \frac{r_t}{\widehat{r}_t}. \tag{13}$$

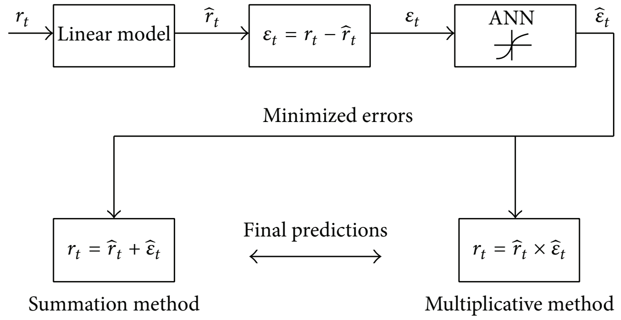

r_t
Linear model
$\widehat{r}_t$
$\varepsilon_t = r_t - \widehat{r}_t$
ε_t
ANN
$\widehat{\varepsilon}_t$
Minimized errors
$r_t = \widehat{r}_t + \widehat{\varepsilon}_t$
Final predictions
$r_t = \widehat{r}_t \times \widehat{\varepsilon}_t$
Summation method
Multiplicative method

Figure 1: Work flow of HIPM.

The series of errors obtained by the above two methods are given to ANN by means of AR-MRNN(p,k), where $k = 1$. Thus, in terms of AR-MRNN, (7) to (9) are modified as given below:

$$\widehat{\varepsilon_{t+1}-z} = \mathfrak{R}\left(\varepsilon_{t-(p-1)}-z,\ldots,\varepsilon_{t-1}-z,\varepsilon_t-z\right), \tag{14}$$

where $\widehat{\varepsilon_{t+1}-z}$ is estimate for $\varepsilon_{t+1}-z$ obtained at time t from the information available from the previously obtained error series $\varepsilon' = (\varepsilon_{t-(T-1)},\ldots,\varepsilon_{t-1},\varepsilon_t)$. p is the order of regression and z is moving reference given in the following equation:

$$z = \varepsilon_{t-(p-1)-k}. \tag{15}$$

After training, the neural network $\widehat{\varepsilon}_{t+1}$ is obtained as shown in (16); thus, minimized errors are obtained:

$$\widehat{\varepsilon}_{t+1} = \widehat{\varepsilon_{t+1}-z} + z. \tag{16}$$

These minimized errors are added to (10) and (12) after replacing original errors. Hence, final predictions from HIPM are obtained by summation method and multiplicative method as shown in the following two equations, respectively:

$$y_t = \widehat{r}_t + \widehat{\varepsilon}_t, \tag{17}$$

$$y_t = \widehat{r}_t \times \widehat{\varepsilon}_t. \tag{18}$$

Figure 1 shows the complete work flow of HIPM. Initially actual returns, r_t, are given as an input to linear prediction through which predictions $\widehat{r}_t$ are obtained. In the next step, errors ε_t are calculated. These errors are fed into ANN which does nonlinear processing. The minimized errors, $\widehat{\varepsilon}_t$, obtained from ANN are used to calculate final predictions via two methods, that is, summation method and multiplicative method.

3.3. Recurrent Neural Network with Autoregressive Moving Reference. The importance of a recurrent neural network (RNN) is that it responds to the same input pattern differently at different times. Figure 2 shows the recurrent neural network with one hidden layer which is used in this work. In this type of network model, input layer is not only fed to hidden layer but also fed back into input layer. The

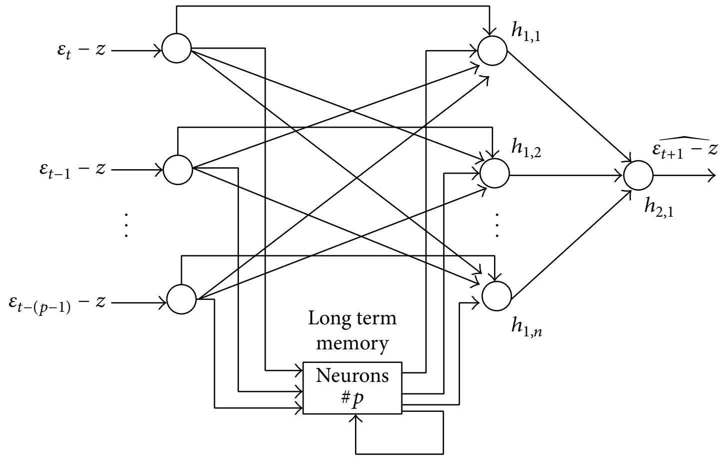


FIGURE 2: Recurrent neural network used in HIPM.

network is a supervised AR-MRNN with inputs ($\varepsilon_t - z, \varepsilon_{t-1} - z, \ldots, \varepsilon_{t-(p-1)} - z$). The desired output is $\widehat{\varepsilon_{t+1} - z}$. As shown, the receiving end for input layer is a long term memory for the network. The function of this memory is to hold the data and pass it to hidden layer immediately after each pattern is sent from input layer. In this way, the network is able to see previous knowledge it had about previous inputs. Thus, long term memory remembers the new input data and uses it when the next pattern is processed. The disadvantage of this network is that it takes longer to train.

The input layer has input neurons equal to the regression order chosen, that is, p, possessing linear activation function. The n are number of neurons $h_{1,1}, \ldots, h_{1,n}$ in hidden layer possessing some activation function. The output neuron, that is, $h_{2,1}$, also possesses some activation function. Thus, the network is able to learn how to predict complex nonlinear patterns of the data. This network is also known as Jordan Elman neural network and for training itself, the network uses backpropagation algorithm [36, 37].

4. Error Metrics

The performance of HIPM is checked here by using two error metrics, mean square error and mean absolute error. A brief discussion about these error metrics is given in the following subsections.

4.1. Mean Square Error (MSE). MSE is the arithmetic mean of the sum of the squares of the forecasted errors. MSE is a standard metric for comparing the differences between two time series and is defined as shown in the following equation:

$$\text{MSE} = \frac{1}{T}\sum_{t=1}^{T}\left(r_t - \widehat{r}_t\right)^2, \tag{19}$$

where r_t and $\widehat{r}_t$ are actual returns and predicted returns, respectively, and T is the length of the series.

4.2. Mean Absolute Error (MAE). MAE is a quantity used to measure how close forecasts are to the eventual outcomes. MAE measures the average magnitude of the errors in a set of forecasts, without considering their direction. The mean absolute error is given in the following equation:

TABLE 1: Company list.

1	2	3
Tech Mahindra	HCL	Mindtree

$$\text{MAE} = \frac{1}{T}\sum_{i=1}^{T}\left|r_t - \widehat{r}_t\right|, \tag{20}$$

where r_t and $\widehat{r}_t$ are actual returns and predicted returns, respectively, and T is the length of the series.

5. Application of HIPM on Stock Market Data

In order to verify the performance of HIPM predictor, real world stock data has been used for experiments. The stock data of three different information technology companies have been used here. These companies are given in Table 1.

Stock data of above three companies has been obtained from Bombay Stock Exchange, India (http://www.bseindia.com/). Daily adjusted closing prices of three stocks have been taken since 14-05-2013 to 30-12-2013 and returns of 164 days (15-05-2013 to 30-12-2013) are calculated using (1). Predictions were obtained using ESM and RNN.

5.1. Predictions Using ESM. Initially predictions are obtained using linear prediction model; here, ESM has been chosen for the purpose. Value for smoothing factor α in ESM has been obtained using the following optimization model:

$$\begin{aligned}&\text{Min} \quad \frac{\sum_{t=1}^{T}\left(r_t - \widehat{r}_{\text{ESM}(t)}\right)^2}{T}\\ &\text{s.t.} \quad 0 \le \alpha \le 1,\end{aligned} \tag{21}$$

where r_t is actual return, $\widehat{r}_{\text{ESM}(t)}$ is prediction obtained from ESM, α is smoothing factor, and T is length of historical

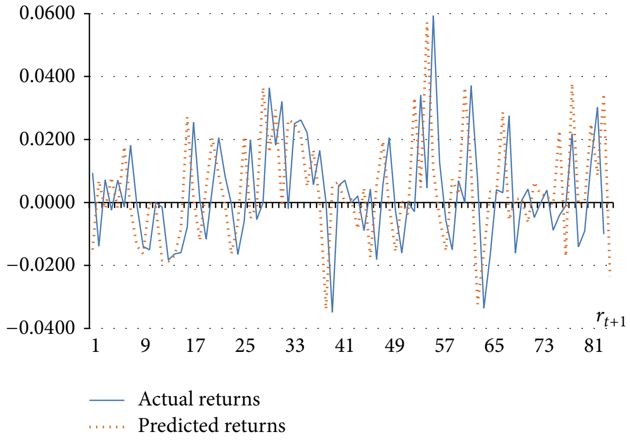


Figure 3: Multiplicative method: output of HIPM for stock 2.

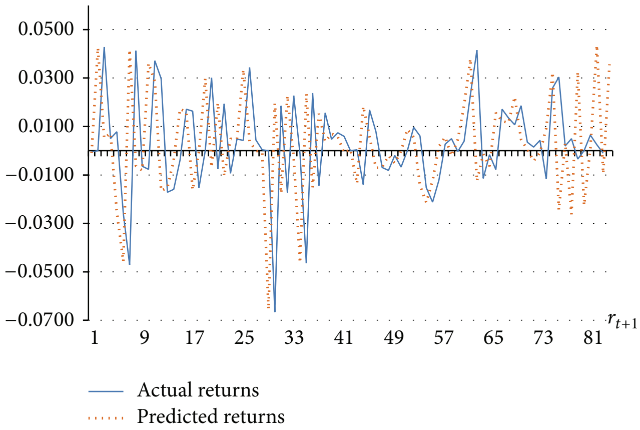


Figure 4: Summation method: output of HIPM for stock 3.

series. The smoothing factor α is associated with the term $\widehat{r}_{\mathrm{ESM}(t)}$ as shown in (2). Thus, (21) is an objective function which minimizes MSE of the predictions obtained from exponential smoothing technique. Its constraint guarantees that the value of smoothing factor α ranges between 0 and 1. Since ESM is a linear prediction model, it obviously did not produce satisfactory predictions, thus resulting in high prediction error.

5.2. Predictions Using RNN. After obtaining predictions using ESM and series of errors calculated, these errors were given to RNN by means of AR-MRNN(p,k) ((14) to (16)). AR-MRNN$(6,1)$ (RNN) was used to obtain stock predictions. Regression order $p = 6$ was chosen after trial and error as it was observed that, by using this particular regression order, RNN produced less prediction error. In each stock, data was divided into two equal parts (50:50). Out of 164 returns, 50% data or 82 returns between15-05-2013 and 05-09-2013 were kept for training RNN and the remaining 50% data, that is, 82 returns between 06-09-2013 and 30-12-2013, for testing. Sliding windows each of 82 returns were created. For each window, input-output pairs were calculated using AR-MRNN$(6,1)$ method which result in 76 input-output pairs in each window. 83 sliding windows were formed each giving prediction for future period; thus, 83 windows give 83 future predictions, while initial window gives prediction for the period r_{t+1}. By combing 83 sliding windows, 6308 input-output pairs were obtained. These sliding windows were given to RNN and trained in a supervised manner; the procedure was repeated for all stocks.

In the chosen RNN model, there are 16 neurons in hidden layer possessing sigmoid activation function and an output neuron in output layer also possessing sigmoid activation function. An error threshold of MSE $= 0.0002$ was preset for RNN; it means RNN converged only after average error reached below the threshold. For RNN, it took over 10,000 epochs for each stock to reach below preset error.

Figures 3 and 4 show the prediction output of HIPM (between 06-09-2013 and 31-12-2013) via multiplicative method and summation method for stock 2 and stock 3, respectively. Actual returns are shown by blue solid line, whereas predictions are shown by orange dotted line. HIPM predictor is able to capture nonlinear patterns of data very well. As shown, actual and predicted returns are very close to each other which imply that predictions are satisfactory. There are total six similar figures obtained, out of which only two are displayed above.

Table 2: Values of error metrics for HIPM.

Stock	1	2	3
		Summation method	
MSE	0.0005	0.0008	0.0025
MAE	0.0176	0.0205	0.0362
		Multiplicative method	
MSE	0.0004	0.0005	0.0007
MAE	0.0172	0.0175	0.0200

The performance of HIPM predictor can be better judged by Table 2, which displays values of error metrics obtained. As seen, Multiplicative method outperforms summation method in terms of less prediction error.

6. Conclusions

A new and promising approach for prediction of stock returns is presented in this paper. A hybrid prediction intelligent model is developed by combining predictions obtained from a linear prediction model and a nonlinear model. The linear model chosen is exponential smoothing model while autoregressive moving reference neural network is chosen as nonlinear model. This is a new approach wherein errors are fed into neural network so as to obtain minimized errors. Initially prediction of stock returns is obtained using exponential smoothing model and prediction errors calculated. Autoregressive moving reference method is used to calculate input-output pairs for the errors just obtained. These errors are fed into recurrent neural network, and the network learns using backpropagation algorithm in supervised manner. Finally, the prediction of stocks is calculated via two methods, summation method and multiplicative method. Based on results, it is observed that the proposed model is able to detect the nonlinear patterns of data very well and results are satisfactory. Input to neural network is given as differences rather than original values. The network thus needs to find smaller weights, thus increasing its prediction performance. The performance of proposed hybrid model can be further improved and applied in other areas too; this is certainly an important avenue for future research.

Conflict of Interests

The author declares that there is no conflict of interests regarding the publication of this paper.

References

[1] G. B. Durham, "Sv mixture models with application to s & p 500 index returns," *Journal of Financial Economics*, vol. 85, no. 3, pp. 822–856, 2007.

[2] C. H. Chen, "Neural networks for financial market prediction," in *Proceedings of the 1994 IEEE International Conference on Neural Networks*, pp. 1199–1202, Orlando, Fla, USA, June 1994.

[3] C. Burges, "A tutorial on support vector machines for pattern recognition," *Data Mining and Knowledge Discovery*, vol. 2, no. 2, pp. 121–167, 1998.

[4] R. Majhi, G. Panda, G. Sahoo, A. Panda, and A. Choubey, "Prediction of S&P 500 and DJIA stock indices using particle swarm optimization technique," in *Proceeding of the IEEE Congress on Evolutionary Computation (CEC '08)*, pp. 1276–1282, Hong Kong, China, June 2008.

[5] S. Haykin, *Neural Networks: A Comprehensive Foundation*, MacMillan College, New York, NY, USA, 1994.

[6] G. Armano, M. Marchesi, and A. Murru, "A hybrid genetic-neural architecture for stock indexes forecasting," *Information Sciences*, vol. 170, no. 1, pp. 3–33, 2005.

[7] K.-J. Kim and I. Han, "Genetic algorithms approach to feature discretization in artificial neural networks for the prediction of stock price index," *Expert Systems with Applications*, vol. 19, no. 2, pp. 125–132, 2000.

[8] W. Shen, X. Guo, C. Wu, and D. Wu, "Forecasting stock indices using radial basis function neural networks optimized by artificial fish swarm algorithm," *Knowledge-Based Systems*, vol. 24, no. 3, pp. 378–385, 2011.

[9] F. D. Freitas, A. F. de Souza, and A. R. de Almeida, "Prediction-based portfolio optimization model using neural networks," *Neurocomputing*, vol. 72, no. 10–12, pp. 2155–2170, 2009.

[10] J.-J. Wang, J.-Z. Wang, Z.-G. Zhang, and S.-P. Guo, "Stock index forecasting based on a hybrid model," *Omega*, vol. 40, no. 6, pp. 758–766, 2012.

[11] M. Khashei and M. Bijari, "An artificial neural network (p, d, q) model for timeseries forecasting," *Expert Systems with Applications*, vol. 37, no. 1, pp. 479–489, 2010.

[12] A.-S. Chen, M. T. Leung, and H. Daouk, "Application of neural networks to an emerging financial market: forecasting and trading the Taiwan Stock Index," *Computers & Operations Research*, vol. 30, no. 6, pp. 901–923, 2003.

[13] A. Jain and A. M. Kumar, "Hybrid neural network models for hydrologic time series forecasting," *Applied Soft Computing Journal*, vol. 7, no. 2, pp. 585–592, 2007.

[14] J.-Z. Wang, J.-J. Wang, Z.-G. Zhang, and S.-P. Guo, "Forecasting stock indices with back propagation neural network," *Expert Systems with Applications*, vol. 38, no. 11, pp. 14346–14355, 2011.

[15] G. S. da S.Gomes, T. B. Ludermir, and L. M. M. R. Lima, "Comparison of new activation functions in neural network for forecasting financial time series," *Neural Computing and Applications*, vol. 20, no. 3, pp. 417–439, 2011.

[16] C. Hamzaçebi, "Improving artificial neural networks' performance in seasonal time series forecasting," *Information Sciences*, vol. 178, no. 23, pp. 4550–4559, 2008.

[17] M. M. Mostafa, "Forecasting stock exchange movements using neural networks: empirical evidence from Kuwait," *Expert Systems with Applications*, vol. 37, no. 9, pp. 6302–6309, 2010.

[18] S. Samarasinghe, *Neural Networks for Applied Sciences and Engineering*, Auerbach Publications, Taylor & Francis, New York, NY, USA, 2007.

[19] K.-J. Kim and H. Ahn, "Simultaneous optimization of artificial neural networks for financial forecasting," *Applied Intelligence*, vol. 36, no. 4, pp. 887–898, 2012.

[20] Y.-K. Kwon and B.-R. Moon, "A hybrid neurogenetic approach for stock forecasting," *IEEE Transactions on Neural Networks*, vol. 18, no. 3, pp. 851–864, 2007.

[21] P. G. Zhang, "Time series forecasting using a hybrid ARIMA and neural network model," *Neurocomputing*, vol. 50, pp. 159–175, 2003.

[22] T.-J. Hsieh, H.-F. Hsiao, and W.-C. Yeh, "Forecasting stock markets using wavelet transforms and recurrent neural networks: an integrated system based on artificial bee colony algorithm," *Applied Soft Computing Journal*, vol. 11, no. 2, pp. 2510–2525, 2011.

[23] M.-Y. Chen, D.-R. Chen, M.-H. Fan, and T.-Y. Huang, "International transmission of stock market movements: an adaptive neuro-fuzzy inference system for analysis of TAIEX forecasting," *Neural Computing and Applications*, vol. 23, pp. 369–378, 2013.

[24] C.-M. Hsu, "A hybrid procedure with feature selection for resolving stock/futures price forecasting problems," *Neural Computing and Applications*, vol. 22, no. 3-4, pp. 651–671, 2013.

[25] R. Adhikari and R. K. Agrawal, "A combination of artificial neural network and random walk models for financial time series forecasting," *Neural Computing and Applications*, vol. 24, no. 6, pp. 1441–1449, 2014.

[26] G. Sermpinis, K. Theofilatos, A. Karathanasopoulos, E. F. Georgopoulos, and C. Dunis, "Forecasting foreign exchange rates with adaptive neural networks using radial-basis functions and particle swarm optimization," *European Journal of Operational Research*, vol. 225, no. 3, pp. 528–540, 2013.

[27] H. Ni and H. Yin, "Exchange rate prediction using hybrid neural networks and trading indicators," *Neurocomputing*, vol. 72, no. 13–15, pp. 2815–2823, 2009.

[28] M.-Y. Chen, "A hybrid ANFIS model for business failure prediction utilizing particle swarm optimization and subtractive clustering," *Information Sciences*, vol. 220, pp. 180–195, 2013.

[29] M. Rout, B. Majhi, R. Majhi, and G. Panda, "Forecas ting of currency exchange rates using an adaptive arma model with differential evolution based training," *Journal of King Saud University—Computer and Information Sciences*, vol. 26, pp. 7–18, 2014.

[30] R. Brown, *Smoothing, Forecasting and Prediction of Discrete Time Series*, Courier Dover Publications, 2004.

[31] G. E. P. Box and G. M. Jenkins, *Time Series Analysis, Forecasting and Control*, Holden-day, San Francisco, Calif, USA, 1970.

[32] G. U. Yule, "Why do we sometimes get nonsense correlations between time series? A study in sampling and the nature of time series," *Journal of the Royal Statistical Society*, vol. 89, pp. 30–41, 1926.

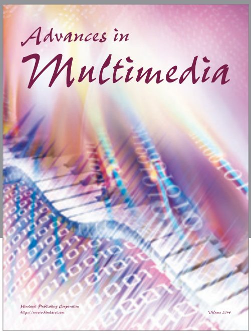
[33] H. O. Wold, *A Study in the Analysis of Stationary Time Series*, Almgrist & Wiksell, Stockholm, Sweden, 1938.

[34] A. M. Rather, "A prediction based approach for stock returns using autoregressive neural networks," in *Proceedings of the World Congress on Information and Communication Technologies (WICT '11)*, pp. 1271–1275, Mumbai, India, December 2011.

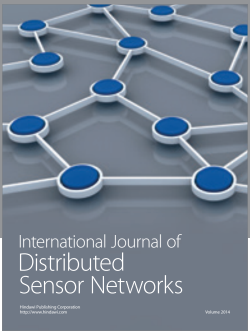
[35] A. M. Rather, "Optimization of predicted portfolio using various autoregressive neural networks," in *Proceedings of the International Conference on Communication Systems and Network Technologies (CSNT '12)*, pp. 265–269, IEEE, May 2012.

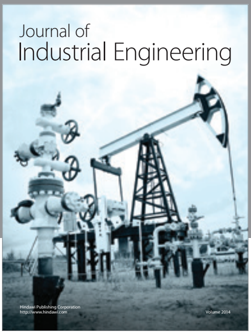
[36] D. Rumelhart and J. McClelland, *Parallel Distributed Processing*, MIT Press, Cambridge, Mass, USA, 1986.

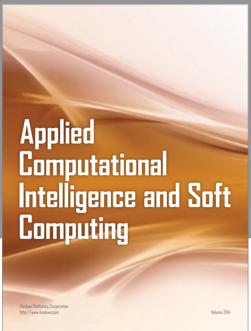
[37] S. Şeker, E. Ayaz, and E. Türkcan, "Elman's recurrent neural network applications to condition monitoring in nuclear power plant and rotating machinery," *Engineering Applications of Artificial Intelligence*, vol. 16, no. 7-8, pp. 647–656, 2003.

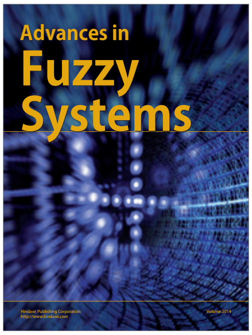

Advances in
Multimedia

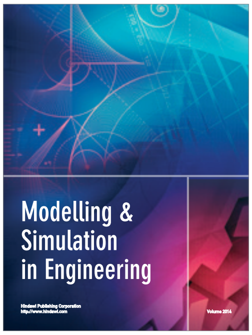
The Scientific
World Journal

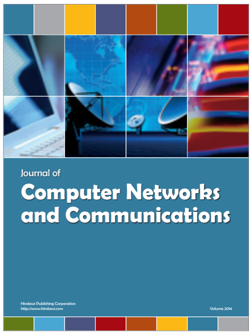

International Journal of
Distributed
Sensor Networks

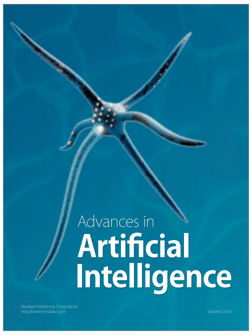

Journal of
Industrial Engineering

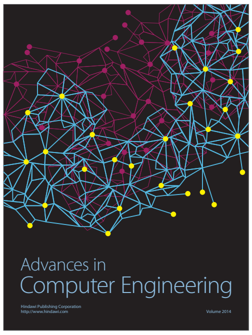

Applied
Computational
Intelligence and Soft
Computing


Advances in
Fuzzy
Systems

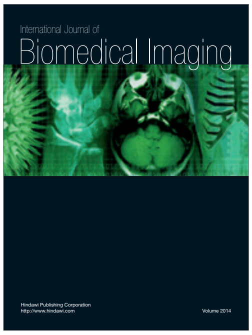

Modelling &
Simulation
in Engineering

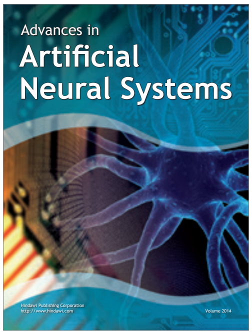

Journal of
Computer Networks
and Communications

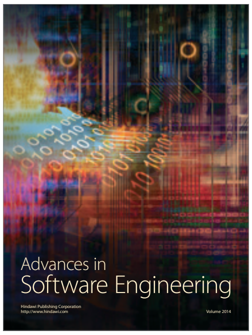

Advances in
Artificial
Intelligence


Advances in
Computer Engineering

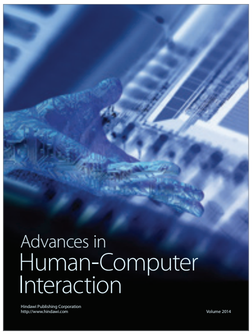
International Journal of
Computer Games
Technology

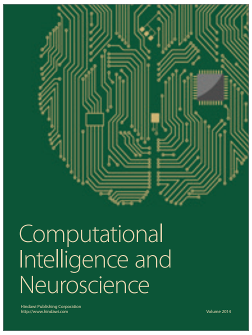

International Journal of
Biomedical Imaging


Advances in
Artificial
Neural Systems


Advances in
Software Engineering

Journal of
Robotics


Advances in
Human-Computer
Interaction


Computational
Intelligence and
Neuroscience

International Journal of
Reconfigurable
Computing

Journal of
Electrical and Computer
Engineering