On the development and performance evaluation of a multiobjective GA-based RBF adaptive model for the prediction of stock indices

Babita Majhi a,*, Minakhi Rout b, Vikas Baghel c

a Dept. of CSIT, Guru Ghasidas Vishwavidyalay, Central University, Bilaspur, India 
b Dept. of CSE, ITER, Siksha ‘O’ Anusandhan Deemed to be University, Bhubaneswar, India 
c School of Electrical Sciences, IIT Bhubaneswar, Bhubaneswar, India

Received 12 January 2013; revised 17 September 2013; accepted 16 December 2013
Available online 25 December 2013

KEYWORDS
Multi-objective optimization; Radial basis function (RBF); Fuzzy decision making

Abstract This paper develops and assesses the performance of a hybrid prediction model using a radial basis function neural network and non-dominated sorting multiobjective genetic algorithm-II (NSGA-II) for various stock market forecasts. The proposed technique simultaneously optimizes two mutually conflicting objectives: the structure (the number of centers in the hidden layer) and the output mean square error (MSE) of the model. The best compromised non-dominated solution-based model was determined from the optimal Pareto front using fuzzy set theory. The performances of this model were evaluated in terms of four different measures using Standard and Poor 500 (S&P500) and Dow Jones Industrial Average (DJIA) stock data. The results of the simulation of the new model demonstrate a prediction performance superior to that of the conventional radial basis function (RBF)-based forecasting model in terms of the mean average percentage error (MAPE), directional accuracy (DA), Thelis’ U and average relative variance (ARV) values.

© 2013 King Saud University. Production and hosting by Elsevier B.V. All rights reserved.

1. Introduction

The accurate prediction of stock price indices is of interest to both private and institutional investors. However, accurate forecasts of this type are challenging due to the inherently noisy and non-stationary nature of stock prices (Abu-Mostafa and Atiya, 1996; Li et al., 2003). Many macro-economical factors affect stock prices, such as political events, firms’ policies, general economic conditions, commodity price indices, interest and exchange rates, investors’ expectations and psychological factors. Many studies of the prediction of stock prices have been conducted over the past two decades. The forecasting
techniques used in the literature can be classified into two categories: statistical and soft computing models. The statistical models include exponential smoothing, the autoregressive moving average (ARMA), autoregressive integrated moving average (ARIMA) and generalized autoregressive conditional heteroskedasticity (GARCH) models (Franses and Gijsels, 1999). These models are based on the assumption that the data of various time series linearly correlate. These real-life stock market data are nonlinear and non-stationary in nature, and the linear forecasting models provide poor prediction performance. To overcome this limitation, in the recent past, soft and evolutionary computing methods have been suggested (Atsalakis and Valavanis, 2009a,b) to forecast these data.

Artificial neural networks (ANNs), which can efficiently model nonlinear systems, have been found to efficiently predict the stock market. Probabilistic neural networks (PNN) (Kim and Chun, 1998), functional link ANNs (Majhi et al., 2009), generalized regression neural networks (Mostafa, 2010) and cerebellar neural networks (Lu and Wu, 2011) have been proposed in the literature for forecasting purposes. In recent years, many researchers (Jilani and Burney, 2008; Chang and Liu, 2008; Dong and Pedrycz, 2008) have used fuzzy time series in forecasting problems. A rough set data analysis model for the discovery of decision rules from stock exchanges has also been reported (Yao and Herbert, 2009). However, a single technique cannot efficiently handle the entire spectrum of forecasting problems. Thus, researchers have introduced different hybrid forecasting models. A neuro-fuzzy system composed of an adaptive neuro-fuzzy inference system (ANFIS) has been used for the short term forecasting of stock market trends (Atsalakis and Valavanis, 2009a,b). A combination of a hidden Markov model (HMM) and fuzzy model has been presented in Hassan, 2009. A self-organizing feature map technique hybridized with support vector regression shows improvement in the prediction and training time (Huang and Tsai, 2009). A forecasting model that integrates the data clustering technique, fuzzy decision tree and genetic algorithm has been reported for stock price forecasting (Lai et al., 2009). Hadavandi et al. (2010) presented an integrated approach based on genetic fuzzy systems and neural networks to optimize the results using minimum required input data and the least complex stock market model. To develop a forecasting model that is more efficient than using ANNs, a hybrid model using ARIMA with ANN (Khashei and Bijari, 2010, 2011) has been reported. Recently, an adaptive pole-zero model with a differential evolution-based training scheme has been reported (Rout et al., 2014). This model has shown an improved prediction of various currency exchange rates. A regression based-data mining technique has also been proposed (Aljumah et al., 2013) for the predictive analysis of diabetic treatment. Esfahanpour and Aghamiri (2010) proposed a neuro-fuzzy inference system that employs Takagi–Sugeno–Kang-type fuzzy rules to predict Tehran stock exchange indices. Lu (2010) integrated independent component analysis with neural networks to build a new forecasting model. Boyacioglu and Avci (2010) predicted stock market returns with an ANFIS model. A mixture of multilayer perceptron (MLP) experts has been presented to predict the Tehran stock exchange (Ebrahimipour et al., 2011). A combination of wavelet transforms and a recurrent neural network based on an artificial bee colony algorithm was proposed to forecast several international stock indices (Hsieh et al., 2011). A three-stage stock market prediction system (Enke et al., 2011) using multiple regression analysis, differential evolution-based type-2 fuzzy clustering and a neural network was recently introduced. Huang forecast stock indices with wavelet analysis and kernel partial least-squares regressions (Huang, 2011). Another efficient hybrid model of ANN and decision trees was proposed to forecast ten different stocks indices (Chang, 2011). Different neural networks, such as the multilayer perceptron (MLP), dynamic artificial neural network and hybrid neural networks, have been proposed to predict the NASDAQ stock exchange (Guresen et al., 2011). A novel stock prediction system has been presented based on neuro-fuzzy architecture and Elliott wave theory (Atsalakis et al., 2011). A type-2 neuro-fuzzy model has been recently applied to predict stocks (Liu et al., 2012). An integrated functional link interval type-2 fuzzy neural system with particle swarm optimization (PSO)-based learning has been proposed to predict stock market indices (Chakravarty and Dash, 2012). A combination of nonlinear independent component analysis with neural networks (Dai et al., 2012) and support vector regression (SVR) (Kao et al., 2012) to predict stock market indices has been recently reported. A hybrid intelligent model that uses an ANN structure trained with a Levenberg–Marquardt algorithm was reported to predict the fluctuations in the stock market (Asadi et al., 2012). Another hybrid approach that combines an exponential smoothing model, the ARIMA and ANN (Wang et al., 2012) has been suggested to forecast stock indices. Most of the conventional derivative-based learning algorithms suffer from slow convergence and a long training time. Therefore, new models that overcome these limitations are necessary to facilitate online and accurate predictions. In recent years, evolutionary algorithms have been introduced to train the weights of neural network models (Hsieh et al., 2011; Chakravarty and Dash, 2012; Asadi et al., 2012; Wang et al., 2012; Shen et al., 2011). Several approaches to stock index forecasting using ANNs have been proposed in the last two decades, but an evolving general method to determine the optimum structure of neural networks is an interesting research idea. If the structure is complex, the generalization ability is low due to the high variance error. Conversely, if a structure is simple, it cannot accurately correlate the input and output data. Thus, an optimum design involves a compromise between the two competing objectives, namely, the performance and the architectural complexity. Thus, the performance constitutes an interesting multi-objective optimization problem to achieve a trade-off between the structure of the model and the prediction. Using a multi-objective approach, the solution can escape from a local minima problem, which can yield improvements from the learning model (Teixeira et al., 2000; Abbass, 2003). Multi-objective evolutionary algorithms have been suggested to determine the number of trade-off solutions between the number of fuzzy rules and the prediction accuracy of financial time series (Hassan et al., 2012). Recently, multi-objective evolutionary algorithms with fuzzy decision-making have been successfully applied to efficiently design cognitive radio parameters (Pradhan and Panda, 2012). Another interesting paper (Hassan et al., 2012) has been reported that employs a hybrid multi-objective evolutionary algorithm and fuzzy-hidden Markov (HM) model to predict time series. Examples of several more recent applications of multi-objective approaches include the following: (Qasem et al., 2012; Guillen et al., 2009; Qasem and Shamsuddin, 2012; Lou et al., 2012; see also references therein).
dis the maximum distance between these selected
m
and RBF are given in Section 2. Section 3 develops the hy-
erature review, the problem formulation and the motivation
ated and demonstrated to be superior to the conventional RBF
networks and accurately predict stock markets. A fuzzy-based
posed to obtain a set of trade-off structures of RBF
mization-based approach known as NSGA-II has been pro-
are given in Section 6.

is presented in Section 5. Finally, the conclusions of the paper
data, and a comprehensive discussion on the obtained results
of the proposed model was carried out using real life stock
performance metrics are presented in Section 4. A simulation study
section a fuzzy decision based methodology is outlined to
brid-forecasting model using the RBF and NSGA-II. In this
series, such as stock indices, using a multi-objective approach.
Conversely, many interesting and promising meta-heuristic
multi-objective optimization techniques, such as the non-domi-
ated sorting genetic algorithm version II (NSGA-II) (Deb et al., 2002) and multi-objective particle swarm optimization
(MOPSO) (Coello et al., 2004), have been reported and have
been applied to various fields. Hence, NSGA-II was chosen to
simultaneously optimize two objectives associated with the prediction problem in an attempt to improve the performance
of the optimal RBF structure.

In this paper, an efficient and popular multi-objective opti-
mization-based approach known as NSGA-II has been pro-
posed to obtain a set of trade-off structures of RBF
networks and accurately predict stock markets. A fuzzy-based
scheme was employed to generate the best compromised pre-
diction model. The performance of this new model was evalu-
ad and demonstrated to be superior to the conventional RBF
(Hatanaka et al., 2003)-based prediction model.

This paper is organized as follows: Section 1 contains a lit-
erature review, the problem formulation and the motivation
behind the problem selection. The details of the NSGA-II and RBF are given in Section 2. Section 3 develops the hy-
brid-forecasting model using the RBF and NSGA-II. In this
section a fuzzy decision based methodology is outlined to
determine the best compromised prediction model. The perfor-
ance metrics are presented in Section 4. A simulation study
of the proposed model was carried out using real life stock
data, and a comprehensive discussion on the obtained results
is presented in Section 5. Finally, the conclusions of the paper
are given in Section 6.

2. Methodology

2.1. Non-dominated sorting genetic algorithm (NSGA-II)

Multi-objective optimization problems yield multiple solu-
tions, each of which makes a tradeoff between objectives.
Hence, each solution is considered optimal. The NSGA-II is
a popular and efficient multi-objective genetic algorithm
(GA) (Deb et al., 2002). In NSGA-II, a parent population of
size N is created, which subsequently undergoes selection,
crossover and mutation processes to produce an offspring pop-
ulation of size N. The offspring population is combined with
the parent population to form a combined population of size
2N, which undergoes a non-dominated sorting process. This
process partitions the complete population into several non-
dominated fronts based on the values of the objective func-
tions. The members of the first front are completely non-dom-
inant. The members of the first front only dominate the
members in the second front. Similarly, the other fronts are
determined until each member of the population falls into
one front. A new population of size N is created by taking
the members of the non-dominated fronts starting from the
first level. Since the population size is predefined, the combined
population cannot be completely accommodated in the new
population. Thus, several non-dominated fronts are discarded.
If none of the members of a front can be accommodated, the
required number of members for the new population is selected
based on the crowding distance technique. An operator, such
as binary tournament selection, simulated binary crossover
or polynomial mutation, is introduced into NSGA-II to im-
prove the overall performance.

2.2. Radial basis function (RBF) neural network

A RBF network can be viewed as a special two-layer network
that contains linear parameters by fixing all RBF centers and
non-linearities in the hidden layer (Haykin, 1999). Fig. 1 de-
picts a schematic diagram of an RBF network to be used as
a stock market predictor with M inputs and one output. The
performance of an RBF network depends on many factors,
including the number of centers. Of the many basis functions,
the Gaussian function is more popular and used in the
proposed RBF network predictor.

The output, Y of the network is given by

\[ Y(t) = w_0 + \sum_{j=1}^{N} w_j \phi(||x - c_j||), \]

where \( w_j \), 0 ≤ j ≤ N are the weights of the output layer, and

\[ \phi(||x, c_j||) = \exp \left( -\frac{m}{d_{\text{max}}^2} ||x - c_j||^2 \right), \quad j = 1, 2, \ldots m, \]

where \( d_{\text{max}} \) is the maximum distance between these selected
centers. \( c_j \) denotes the Euclidean distance, and \( m \) is the
number of centers. The standard deviation or width of all the
Gaussian radial basis functions is fixed at \( \sigma = \frac{d_{\text{max}}}{\sqrt{2m}} \).

By providing a set of the inputs, \( x(t) \), and the corresponding
desired value \( d(t) \), t = 1, 2, ..., n, the weights, \( w_j \), are deter-
mined using the linear least squares (LS) method. The weight
vector is updated using the pseudo-inverse method (Broom-
head and Lowe, 1988) as follows:

\[ w = \phi^+ d, \]

where \( d \) is the desired response vector in the training set. The
matrix \( \phi^+ \) is the pseudo-inverse of matrix \( \phi \) and is defined as

\[ \phi^+ = (\phi^T \phi)^{-1} \phi^T. \]
 Appropriately choosing the center from the data set is a key point of the RBF network. The best RBF network is required to garner the optimum performance from the data. This condition is generated by appropriately selecting the centers using a multi-objective algorithm, such as NSGA-II.

3. Development of stock market forecasting model using NSGA-II and Fuzzy decision making

3.1. Problem formulation

In this paper, the multi-objective NSGA-II algorithm is used to select the centers for the RBF network. The binary chromosomes of NSGA are initialized first to select the proper network. The number of genes in each chromosome equals the length of the training data set. The chromosomes of the population indicate whether each data point is employed as a center of the basis functions. A “1” in the chromosome indicates that the center of the basis function is located at the corresponding training data point, and “0” represents a lack of center, as shown in Fig. 2.

In the chromosome, the position of gene value “1” indicates the center position of the basis function (selected center), and the number of “1” genes in the chromosome indicates the number of basis functions (number of centers).

Two objectives are considered in the multi-objective approach of designing an efficient predictor: the minimization of number of centers in the hidden layer \( f_1 \), which relates to the complexity of the RBF network, and the minimization of the MSE \( f_2 \), which relates to the prediction performance measure. The MSE of the stock index predictor is defined as

\[
MSE = \frac{1}{n} \sum_{i=1}^{n} (y_i - d_i)^2
= \frac{1}{n} \sum_{i=1}^{n} \left( y_i - \left( \sum_{k=1}^{K} w_k \phi(\|x_i - c_k\|) + w_0 \right) \right)^2,
\]

where \( d_i \) is the desired output and \( y_i \) is the output of the RBF network.

The algorithm of developing an RBF-based forecasting model using NSGA-II proceeds as follows:

1. \[Start\]: Generate a random population of \( N \) chromosomes (binary). Each chromosome contains a number of genes.
2. \[Fitness\]: Evaluate the multi-objective fitness \( f_1 = \text{no. of centers}, f_2 = \text{MSE} \) of each chromosome in the population.

   \[\text{Fig. 2} \quad \text{Chromosome-center representation.}\]

3. \[Non-dominated sorting\]: Rank the population according to the following steps:
   a. \[Dominance rank\]: Rank the population with Algorithm-1.
   b. \[Crowding distance\]: Calculate the crowding distance with Algorithm-2.
4. \[New population\]: Create a new population by repeating the following steps:
   a. \[Selection\]: Select two parent chromosomes from the population based on the crowding selection operator as given in Algorithm-3.
   b. \[Crossover\]: With a predefined crossover probability, crossover the parents to form the new offspring.
   c. \[Mutation\]: With a predefined mutation probability, mutate the new offspring.
   d. Combine the parent chromosomes, offspring and mutated offspring.
   e. Select \( N \) number of best chromosomes for the next generation and discard the others.
5. Repeat Steps 2-4 with the new population obtained from the previous generation.
6. If the end condition is satisfied, stop; the non-dominated chromosomes give the required solution.
7. Otherwise, go to Step-2.

3.1.1. Algorithm-1 (Non-dominated sorting)

Let there be \( n \) objective functions. A solution \( x \) dominates another solution \( y \) when the following conditions are satisfied. Otherwise, \( x \) and \( y \) are non-dominated solutions.

1. \( x \) is not worse than \( y \) for all \( n \) objective functions.
2. \( x \) is strictly better than \( y \) in at least one of the \( n \) objective functions.

The non-dominated solutions in population \( N \) can be obtained as follows:

1. Set rank counter \( r = 0 \).
2. Obtain \( r = r + 1 \).
3. Find the non-dominated chromosomes based on the definition given.
4. Assign rank \( r \) to these individuals.
5. Remove these individuals from the population \( N \).
6. If population \( N \) is empty, stop. Otherwise, go to step 2.

3.1.2. Algorithm-2 (crowding distance)

Consider a number of non-dominated solutions in \( N \) of size \( M \), and a number of objective functions \( f_k, k = 1, 2, \ldots, n \) are given. Let \( d_i \) or \( d_j \) be the value of the crowding distance of the solution \( i \) or \( j \). The crowding distance is calculated via the following steps:

1. Let \( d_i = 0, i = 1, 2, \ldots M \)
2. For each objective function \( f_k, k = 1, 2, \ldots n \), sort the set in ascending order.
3. Set \( d_1 = d_M = \infty \).
   For \( f = 2 \) to \( (M - 1) \)
\[
d_j = d_i + (f_{j+1} - f_{j-1}).
\]

End of loop.

3.1.3. Algorithm-3 (crowding selection operator)

A solution \( x \) is better than another solution \( y \) if one of the following conditions is satisfied:

1. The domination rank of solution \( x \) is smaller than that of solution \( y \).
2. If the domination ranks are equal, the crowding distance of \( x \) is larger than that of \( y \).

3.2. Fuzzy decision making

NSGA-II provides a set of solutions, each of which represents a particular performance trade-off between the multiple objectives. Because the decision is mostly imprecise in nature, each objective function associates fuzziness with its goal. The degree of fuzziness can be represented by a membership function that varies between 0 and 1. When the solutions in the non-dominated front are close to each other and distinguishing between the solutions that provide almost equal weight to each objective is difficult, the fuzzy-based approach enables a compromise solution. This approach examines the way the solutions are contributing to each objective and assigns a fuzzy variable. In this paper, a method similar to that proposed in (Pradhan and Panda, 2012) is employed to determine a compromised solution on the non-dominated front.

The membership value of the \( i \)th objective of \( j \)th solution in the non-dominated front is computed as follows:

\[
\mu'_i = \begin{cases} 
1; & \text{if } F_i \leq F_i^{\min} \\
\frac{F_i^{\max} - F_i}{F_i^{\max} - F_i^{\min}}; & \text{if } F_i^{\min} < F_i \leq F_i^{\max} \\
0; & \text{if } F_i > F_i^{\max}
\end{cases}
\]

(7)

\( \mu'_i \) indicates how the \( j \)th non-dominated solution can best satisfy the \( i \)th objective. The sum of membership values for all objectives of the \( j \)th non-dominated solution suggests how well it satisfies different objectives. The contribution of each non-dominated solution with respect to all the \( N \) non-dominated solutions can be obtained as follows:

\[
\mu_j = \frac{\sum_{i=1}^{M} \mu'_i}{\sum_{i=1}^{M} \sum_{j=1}^{N} \mu'_i},
\]

(8)

where \( M \) represents the total number of objectives. The solution that contains the maximum value of \( \mu_j \) is a compromised solution that is better accepted by the decision maker. However, this compromised solution is not binding for a decision maker to accept.

3.2.1. Algorithm-4 (steps of Fuzzy decision making)

1. Simulate the NSGA-II program for ten independent runs and obtain its Pareto fronts.
2. Apply the fuzzy rule and calculate the values of \( \mu_1 \) and \( \mu_2 \) for each objective function \( f_1 \) (number of centers) and \( f_2 \) (mean square error) on the Pareto front using (7).
3. Calculate the value of \( \mu \) for each \( \mu_1 \) and \( \mu_2 \) using (8).
4. Choose the corresponding chromosome as the optimized solution that contains the highest value of \( \mu \).

4. Performance metrics

The mean absolute percentage error (MAPE), directional accuracy (DA), Theil U and average relative variance (ARV) are used to gauge the performance of the proposed prediction model for the test data. These values are calculated as follows:

\[
\text{MAPE} = \frac{\sum_{i=1}^{n} \left| \frac{A_i - P_i}{A_i} \right|}{n} \times 100,
\]

(9)

where \( A_i \) is the actual and \( P_i \) is the predicted value for \( i \)th test pattern. \( n \) is the total number of test patterns.

\[
\text{DA} = \frac{100}{n} \sum_{i=1}^{n} d_i, \quad \text{where } d_i = \begin{cases} 
1, & (P_i - P_{i-1})(A_i - A_{i-1}) \geq 0 \\
0, & \text{otherwise}
\end{cases}
\]

(10)

This measure accounts for the number of correct decisions when predicting whether the value of the series will increase or decrease during the subsequent time steps. The values assigned by DA should fall between 0 and 100; the closer the values are to 100, the more accurate the prediction model is. This measure is more important when applied to the stock market because a correct prediction of the direction of the series of the stock quotation directly impacts the financial gains and losses of the investment (Ferreira et al., 2008; Chang and Tsai, 2007). Another important measure for performance comparison is defined as

\[
\text{Theil’s } U = \sqrt{\frac{1}{n} \sum_{i=1}^{n} (A_i - P_i)^2} + \sqrt{\frac{1}{n} \sum_{i=1}^{n} P_i^2}.
\]

(11)

This measure associates the model performance with a random walk (RW) model, as given in Table 1.

The predictor is usable if its Theil’s U statistics approach the perfect model, i.e., Theil’s U approaches zero (Ferreira et al., 2008). The third performance measure, the ARV is defined as

\[
\text{ARV} = \frac{\sum_{i=1}^{n} (P_i - A_i)^2}{\sum_{i=1}^{n} (P_i - \bar{A})^2}.
\]

(12)

The characteristics of this measure are given in Table 2.

The model is practical if the value of ARV is less than one; a value closer to zero indicates that the predictor tends to be perfect (De and Araújo, 2012).

5. Simulation study

5.1. Data collection and feature extraction

The data for the experiment on stock market prediction were collected from a website for two stock indices, namely the DJIA and S&P 500. The data were collected from January 2005 to December 2006, totaling 630 data patterns for both the DJIA and S&P 500 indices. The data obtained for the stock indices consisted of the closing price, opening price and lowest price in the day, highest value in the day and the total volume of stocks traded in each day. The technical indicator is a metric
whose value is derived from generic price activity in a stock or asset. Technical indicators look to predict the future price levels, or simply the general price direction, of a security examined asset. Technical indicators look to predict the future price level of a security examined.

<table>
<thead>
<tr>
<th>Name of the technical indicator</th>
<th>Formula</th>
</tr>
</thead>
<tbody>
<tr>
<td>Simple moving average (SMA)</td>
<td>[P = \frac{\sum_{i=1}^{N} x_i}{N}] (x_i = \text{today’s price})</td>
</tr>
<tr>
<td>Exponential moving average (EMA)</td>
<td>[A = \frac{2}{(N+1)}] (P = \text{current price, } A = \text{smoothing factor, } N = \text{time period})</td>
</tr>
<tr>
<td>Accumulation/distribution oscillator (ADO)</td>
<td>[% K = \left(\frac{\text{Highest high in K period} - \text{Lowest low in K period}}{\text{Today’s close} - \text{Lowest low in K period}}\right) \times 100]</td>
</tr>
<tr>
<td>Stochastic oscillator (STOC)</td>
<td>[% D = \text{SMA of } % K \text{ for the period}]</td>
</tr>
<tr>
<td>Relative strength index (RSI)</td>
<td>[\text{RSI} = 100 - \frac{100}{1 + \left(\frac{D}{n}\right)}] (U = \text{total gain, } D = \text{total loss, } n = \text{no. of RSI period})</td>
</tr>
<tr>
<td>Price rate of change (PROC)</td>
<td>[\text{(Today’s close} - \text{close X period ago})] (\times 100)</td>
</tr>
<tr>
<td>Closing price acceleration (CPACC)</td>
<td>[\text{(close price} - \text{close price N period ago})] (\times 100)</td>
</tr>
<tr>
<td>High price acceleration (HPACC)</td>
<td>[\text{(high price} - \text{high price N period ago})] (\times 100)</td>
</tr>
</tbody>
</table>

5.2. Training of the Pareto RBF model

Of the 630 patterns, 500 patterns were used to train the forecasting model, and 130 patterns were used for testing purposes. Each of the patterns consisted of ten technical indicators. Each pattern was sequentially applied as an input to the RBF network; the output was calculated and compared with the corresponding desired value to yield the error value. The desired value to be applied to the model depended on how many days ahead the prediction was to be made. After the application of all input patterns, the mean square error (MSE) was calculated.

The NSGA-II algorithm, as described in section III, was used to optimize the number of centers and the MSE of the RBF network. The different values of the parameters used in the simulation-based experiments are listed in Table 4.

A binary representation of the chromosome, binary tournament selection, single point binary crossover and bit reversal mutation was used in this study. The simulation study was carried out for a one-day, one-week and one-month forecast, and the Pareto-optimal solution was obtained in each case.

Figs. 3(a)–3(c) show the optimal Pareto fronts obtained for one day, one week and one month, respectively, for the S&P 500 stock index using NSGA-II. In each of these figures, a square box is indicated that corresponds to a fuzzy-based best compromised solution. Table 5 shows the best compromised solution obtained using the fuzzy decision stated in (8) for ten independent runs. This table lists the number of centers and the MSE for the S&P 500 for the one-day, one-week and one-month forecast, respectively. Similarly, the simulation results of the DJIA stock index for one day, one week and one month were obtained and are listed in Table 6 and Figs. 4(a)–4(c).
Fig. 3a  Fuzzy optimized Pareto front for S&P 500 stock index for the one-day forecast.

Fig. 3b  Fuzzy optimized Pareto front for S&P 500 stock index for the one-week forecast.

Fig. 3c  Fuzzy optimized Pareto front for S&P 500 stock index for the one-month forecast.
5.3. Testing of the model

After the completion of the training phase, each of the non-dominated solutions provides the weights and the centers corresponding to fitness values. One compromised structure has been obtained during the training phase using the fuzzy decision rule. This structure was then chosen for testing purposes. The number of centers of the RBF and the proposed multi-objective RBF (MORBF) were maintained the same to facilitate a comparison. For the conventional RBF, this choice allowed a comparison of the prediction performance with an equivalent multi-objective model. The comparison of the actual and predicted values obtained by the conventional RBF and the MORBF model for one day, one week and one month is given in Figs. 5(a)–5(c) for the S&P500 and in Figs. 6(a)–6(c) for the DJIA. The values of the MAPE, DA, Theli’s U and AVR were also calculated for different experiments for both the MORBF and RBF and are listed in Tables 7–9 for the S&P500 and in Tables 10–12 for the DJIA stock indices. These

![Fig. 4a](image1)

**Fig. 4a** Fuzzy optimized Pareto front for the DJIA stock index for the one-day forecast.

![Fig. 4b](image2)

**Fig. 4b** Fuzzy optimized Pareto front for the DJIA stock index for the one-week forecast.

<table>
<thead>
<tr>
<th>Prediction model</th>
<th>No. of centers</th>
<th>MSE ($10^{-5}$)</th>
</tr>
</thead>
<tbody>
<tr>
<td>One day ahead</td>
<td>42</td>
<td>0.7243</td>
</tr>
<tr>
<td>One week ahead</td>
<td>47</td>
<td>0.8525</td>
</tr>
<tr>
<td>One month ahead</td>
<td>51</td>
<td>1.6890</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Prediction model</th>
<th>No. of centers</th>
<th>MSE ($10^{-5}$)</th>
</tr>
</thead>
<tbody>
<tr>
<td>One day ahead</td>
<td>47</td>
<td>1.199</td>
</tr>
<tr>
<td>One week ahead</td>
<td>54</td>
<td>1.218</td>
</tr>
<tr>
<td>One month ahead</td>
<td>45</td>
<td>2.650</td>
</tr>
</tbody>
</table>

**Table 5** Best compromised values of two objectives obtained from fuzzy decision for S&P 500 stock index.

**Table 6** Best compromised values of two objectives obtained from fuzzy decision for DJIA stock index.
Fig. 4c  Fuzzy optimized Pareto front for the DJIA stock index for the one-month forecast.

Fig. 5a  Comparison of the actual and predicted values during the testing of the S&P500 stock index using MORBF with fuzzy decision-making and conventional RBF for the one-day forecast.

Fig. 5b  Comparison of the actual and predicted values during the testing of the S&P500 stock index using MORBF with fuzzy decision-making and conventional RBF for the one-week forecast.
Fig. 5c  Comparison of the actual and predicted values during the testing of the S&P500 stock index using MORBF with fuzzy decision-making and conventional RBF for the one-month forecast.

Fig. 6a  Comparison of the actual and predicted values during the testing of the DJIA stock index using MORBF with fuzzy decision-making and conventional RBF for a one-day forecast.

Fig. 6b  Comparison of the actual and predicted values during the testing of the DJIA stock index using MORBF with fuzzy decision-making and conventional RBF for the one-week forecast.
results demonstrate that the MORBF provides superior performance in all cases for both the stock indices in comparison to the RBF forecasting model under identical conditions.

Table 7 Comparison of the performance measures for the S&P 500 stock index for a one-day forecast (number of centers 42).

<table>
<thead>
<tr>
<th>Methods</th>
<th>MAPE</th>
<th>DA</th>
<th>Theli’s U</th>
<th>AVR</th>
</tr>
</thead>
<tbody>
<tr>
<td>NSGA-Fuzzy</td>
<td>1.14288</td>
<td>57</td>
<td>0.00749</td>
<td>0.16460</td>
</tr>
<tr>
<td>RBF</td>
<td>5.23668</td>
<td>59</td>
<td>0.03429</td>
<td>0.88028</td>
</tr>
</tbody>
</table>

Table 8 Comparison of the performance measures for the S&P 500 stock index for the one-week forecast (number of centers 47).

<table>
<thead>
<tr>
<th>Methods</th>
<th>MAPE</th>
<th>DA</th>
<th>Theli’s U</th>
<th>AVR</th>
</tr>
</thead>
<tbody>
<tr>
<td>NSGA-Fuzzy</td>
<td>2.19308</td>
<td>59</td>
<td>0.01472</td>
<td>0.57355</td>
</tr>
<tr>
<td>RBF</td>
<td>5.81531</td>
<td>50</td>
<td>0.03557</td>
<td>1.07750</td>
</tr>
</tbody>
</table>

Table 9 Comparison of the performance measures for S&P 500 stock index for the one-month forecast (number of centers 51).

<table>
<thead>
<tr>
<th>Methods</th>
<th>MAPE</th>
<th>DA</th>
<th>Theli’s U</th>
<th>AVR</th>
</tr>
</thead>
<tbody>
<tr>
<td>NSGA-Fuzzy</td>
<td>4.68460</td>
<td>51</td>
<td>0.02976</td>
<td>0.97141</td>
</tr>
<tr>
<td>RBF</td>
<td>10.3620</td>
<td>47</td>
<td>0.06942</td>
<td>1.20544</td>
</tr>
</tbody>
</table>

Table 10 Comparison of performance measures for DJIA stock index for one-day forecast (number of centers 47).

<table>
<thead>
<tr>
<th>Methods</th>
<th>MAPE</th>
<th>DA</th>
<th>Theli’s U</th>
<th>AVR</th>
</tr>
</thead>
<tbody>
<tr>
<td>NSGA-Fuzzy</td>
<td>1.35878</td>
<td>57</td>
<td>0.00896</td>
<td>0.26589</td>
</tr>
<tr>
<td>RBF</td>
<td>5.33802</td>
<td>42</td>
<td>0.03299</td>
<td>0.97148</td>
</tr>
</tbody>
</table>

Table 11 Comparison of performance measures for DJIA stock index for one-week forecast (number of centers 54).

<table>
<thead>
<tr>
<th>Methods</th>
<th>MAPE</th>
<th>DA</th>
<th>Theli’s U</th>
<th>AVR</th>
</tr>
</thead>
<tbody>
<tr>
<td>NSGA-Fuzzy</td>
<td>2.78601</td>
<td>53</td>
<td>0.01775</td>
<td>0.67152</td>
</tr>
<tr>
<td>RBF</td>
<td>5.97773</td>
<td>59</td>
<td>0.03623</td>
<td>0.90452</td>
</tr>
</tbody>
</table>

Table 12 Comparison of performance measures for DJIA stock index for the one-month forecast (number of centers 45).

<table>
<thead>
<tr>
<th>Methods</th>
<th>MAPE</th>
<th>DA</th>
<th>Theli’s U</th>
<th>AVR</th>
</tr>
</thead>
<tbody>
<tr>
<td>NSGA-Fuzzy</td>
<td>5.22487</td>
<td>53</td>
<td>0.03233</td>
<td>0.85846</td>
</tr>
<tr>
<td>RBF</td>
<td>18.0648</td>
<td>43</td>
<td>0.10965</td>
<td>1.07920</td>
</tr>
</tbody>
</table>

6. Conclusion

This paper developed an efficient set of RBF-based stock index prediction models by formulating the prediction problem as a multi-objective optimization problem. Two conflicting objectives, the number of centers and the MSE of the model, were chosen to be optimized using NSGA-II and a fuzzy decision-making scheme. The prediction performance in terms of four metrics was evaluated to predict different stock indices for various forecast periods. The results of various simulation-based experiments using real life data demonstrate that the MORBF models developed in this paper show superior prediction performance in terms of four performance measures compared to its single-objective counterpart. Further research work is being carried out to efficiently predict other time series using the proposed multi-objective-based approach.

References


