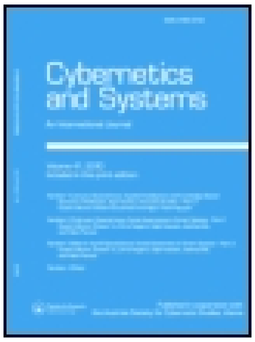

Taylor & Francis
Taylor & Francis Group

Cybernetics and Systems
An International Journal

ISSN: 0196-9722 (Print) 1087-6553 (Online) Journal homepage: http://www.tandfonline.com/loi/ucbs20

A Hybrid Artificial Neural Network with Metaheuristic Algorithms for Predicting Stock Price

Rahim Ghasemieh, Reza Moghdani & Shib Sankar Sana

CYBERNETICS AND SYSTEMS: AN INTERNATIONAL JOURNAL
http://dx.doi.org/10.1080/01969722.2017.1285162

A Hybrid Artificial Neural Network with Metaheuristic Algorithms for Predicting Stock Price

Rahim Ghasemieh[a], Reza Moghdani[b], and Shib Sankar Sana[c]

[a]Department of Management, Shahid Chamran University of Ahvaz, Ahvaz, Iran; [b]Industrial Management, Persian Gulf University, Bushehr, Iran; [c]Department of Mathematics, Bhangar Mahavidyalaya, Bhangar, India

ABSTRACT

Most investors change stock prices in long-term businesses because of global turbulence in the markets. Consequently, prediction of stock price is a difficult task because of unknown effective factors in this area although previous researches have shown that neural networks are more effective and accurate in many areas than traditional statistical models. The proposed study aims to predict prices on stock exchange via the hybrid artificial neural network models and metaheuristic algorithms which consist of cuckoo search, improved cuckoo search, improved cuckoo search genetic algorithm, genetic algorithm, and particle swarm optimization. The important 28 variables of value-added knowledge related to stock indices are identified as input parameters in this network, and then real values are obtained (http://www.tsetmc.com). The results of the proposed model suggest that particle swarm optimization is a dominant metaheuristic approach to predict stock price according to statistical performances of the above approaches.

KEYWORDS
Artificial neural network; metaheuristic algorithms; stock price

Introduction

Many management bodies of investors as well as researchers in stock price literature are being emerged to research for predicting future trend of stock price of the market. The investors are anxious enough thinking about the risk of investment in financial markets because of unknown future trends of the prices in the markets. As a result, it is a challenging task in an unstable market to determine the appropriate time of buying, holding, or selling the inventories. However, stock market prediction is surprisingly a sophisticated task because of its dynamic, nonlinear, nonparametric, and chaotic features. The researchers associated with stock price evaluation have observed that the analysis of historical trend of changes of stock price does not provide complete information to predict the future trend of stock price (Zahedi and Rounaghi 2015). Although outstanding forecasting models have been studied in the early decades, there is still lack of evidence as far as of having proper

CONTACT Shib Sankar Sana shib_sankar@yahoo.com Department of Mathematics, Bhangar Mahavidyalaya, South 24 Parganas, Bhangar 743502, India.
Color versions of one or more of the figures in the article can be found online at www.tandfonline.com/ucbs.

model in forcasting of stock price. In this direction of research works, we have suggested a proper model to fill up the gap of this research by developing and combining soft computing models. According to Kamruzzaman, Begg, and Sarker (2006), neural network (NN) has an important role in stock price analysis. Most international investment bankers and brokerage firms have major stakes in overseas markets. Merchandisable financial assets are a critical part of decision-making process of financial managers (Metghalchi 2011).

There are two main models in the field of artificial neural network (ANN) to forecast stock price, which are the statistical model and the soft computing model (Majhi et al. 2009). In the models of soft computing, pure models of ANN and combinations of other models are considered. Here, we can mention three familiar ANN tools for the said task. These are the radial basis function (RBF) (Han and Kamber 2001), the recurrent neural network (RNN) (Saad, Prokhorov, and Wunsch 1998), and the multilayer perceptron (MLP) (Guresen, Kayakutlu, and Daim 2011). Guresen, Kayakutlu, and Daim (2011)have proposed a model to evaluate the effectiveness of NN models. This model analyzes the multilayer perceptron (MLP) and the dynamic artificial neural network (DAN2) and uses the generalized autoregressive conditional heteroscedasticity (GARCH) method to determine new inputs. In the neuro-fuzzy model, adaptive neuro-fuzzy inference system (ANFIS) shows clearly that it is quite proper for stock market prediction and can be a functional tool for economists and practitioners dealing with forecasting of stock price index return (Boyacioglu and Avci 2010).

In the branch of combinations models, emerging new trend of soft computing methods in analyzing mathematical model has attracted researchers to apply hybrid artificial neural network (HANN) models along with other well known approaches such as fuzzy set and metaheuristic algorithms. These advanced approaches can deal with complex engineering problems which are difficult to solve by classical methods. Therefore, metaheuristic approaches, recently developed by many scholars, have been rigorously implemented in many research areas so as to deal with the complex real-world problems. In this case, genetic algorithm (GA) proposed by Holland (1975) is another popular approach. Kuo, Chen, and Hwang (2001) have presented a genetic algorithm-based fuzzy neural network (GFNN) to formulate the knowledge base of fuzzy inference rules which can measure the qualitative effect on the stock market. Yu and Zhang (2005) have suggested a novel hybrid evolutionary learning algorithm based on NN and GA. Aboueldahab and Fakhreldin (2011) have proposed the new hybrid genetic algorithm (HGA) as well as particle swarm optimization (PSO) with perturbation term based on the biological mechanism to solve the problem of local search restriction. Recently, Göçken et al. (2016) have used HANN models to select the most relevant technical indicators for stock market forecasting by performing Harmony Search (HS) and GA. In the work of Majhi et al. (2009), bacterial foraging optimization (BFO) and adaptive bacterial foraging optimization (ABFO) techniques have been discussed to develop an efficient forecasting model for the prediction of stock indices. Zhang and Wu (2009) have presented an improved bacterial chemo taxis optimization (IBCO) which is integrated into BPN to develop an efficient forecasting model for the prediction of various stock indices. Though the aforementioned studies are focused on hybrid metaheuristic approaches with ANN, there are some new approaches available in the literature that present a new perspective in forecasting of stock price. Wang (2002) has proposed a hybrid model that uses a data mart to reduce the size of stock data and combines fuzzy techniques with the grey theory to develop a fuzzy grey prediction of stock price in the Taiwan stock market. He concludes that the proposed model can effectively help stock dealers deal with day trading. Nayak, Misra, and Behera (2012, 2015) also have studied some neuro-genetic models. The above models have been applied to the Indian stock market, and the results of the model show that the evolutionary optimization techniques increase the adaptability of hybrid forecasting models.

The current study has two main goals. Firstly, it aims to develop a new general framework for the prediction of stock indices, combining metaheuristic approaches with ANN. It is rarely seen in the literature. The second objective is to present a comparative study of the performances of different metaheuristics in order to forecast stock prices based on various well-known technical indicators. Finally, the performances of prediction of the proposed approaches are evaluated according to various metrics.

As far as the knowledge of the authors goes, none of the existing literature has investigated and made a comparative study in stock price forecasting by implementing different kinds of approaches. On the other hand, several researchers have used metaheuristic algorithm with ANN to predict stock price, but none of them have proposed a general framework for considering those in the ANN structure. As a result, our proposed general framework of stock price forecasting and its comparative study of knowledge based on soft computing techniques will enrich the stock pricing literature.

The rest of this paper is organized as follows. In “Background of Artificial Neural Networks (ANNs),” a brief literature review of ANN is presented. The research methodology comes in “Research Methodology.” In “Metaheuristic Approaches,” metaheuristic approaches are proposed to solve and analyze the problem. Using raw data, numerical analysis is investigated in “Data Analysis” and conclusions are drawn in “Conclusion.”

Background of Artificial Neural Networks (ANNs)

The ANN is an information system that replicates the behavior of the human brain by emulating the operation and connectivity of the brain to generate a general solution of a problem (Adebayo, Saiang, and Nordlund 2015). It has strong fundamental concepts based on the performance of biological neural network system. The three most popular tools of ANN are the radial basis function (RBF) (Han and Kamber 2001), the recurrent neural network (RNN) (Saad, Prokhorov, and Wunsch 1998), and the multilayer perceptron (MLP). The models based on multibranch neural networks (MBNNs) (Yamashita, Hirasawa, and Hu 2005) and local linear wavelet neural networks (LLWNNs) (Chen, Dong, and Zhao 2005) are noteworthy, among others. Some forecasting applications of the MLP are found in financial time series forecasting (Yu, Wang, and Lai 2009), CMOL technology (Rezaia, Keshavarzia, and Mahdiye 2014), energy lost (Takia et al. 2016), road headers' performance market (Ebrahimabadi, Azimipour, and Bahreini 2015), automatic speech recognition systems (Park et al. 2011), performance evaluation (Kahrizia and Hashemib 2014), wind speed (Liu et al. 2013), automatic communication signal recognition (Shrme 2011), segmentation (Bae et al. 1998), wheat soaking (Kashaninejad, Dehghani, and Kashiri 2009), chemical plants (Lightbody et al. 1997), ozone level (Kashaninejad, Dehghani, and Kashiri 2009), macroeconomic data forecasting (Aminian et al. 2006), stock exchange movement (Mostafa 2010), maritime traffic forecasting (Mostafa 2014), electric load forecasting (Darbellay and Slama 2000), air pollution forecasting (Videnova et al. 2006), visual classification (Güler et al. 1998), ATM network (Ng and Tham 2000), design of production scheduling system (Feng et al. 2003), and many other contexts. This network also comprises of three layers: input, hidden, and output layers (Yasin et al. 2014). Hence, it is a multilayer network. Since each node is connected to all nodes of the other layers, it is a fully connected network. Figure 1 represents the structure of the network system as follows.

It is well known that the robustness of the fundamentals of ANN are studied based on human neural network and it executes computations similar to that of the natural neural networks (Srinivas et al. 2012). An ANN model is known as a computing system which is highly interconnected and transfers information processing elements like a neuron in the human body. The neuron collects inputs from both single and multiple sources and produces output in accordance with a predetermined nonlinear function. The primary elements of a NN are the distributed representation of knowledge-based information, local operations, and nonlinear processing (Sarkar and Pandey 2015). Generally, ANN consists of simple interconnected processing elements called neurons under a prespecified topology (layers) (Benardos and Kaliampakos 2004). Each neuron is connected to its neighbors with varying coefficients called weights (Hamed, Khalafallah, and Hassanien 2004). The knowledge of ANN is stored in its weights (Holubar et al. 2002; Hamed, Khalafallah, and Hassanien 2004). Due to the remarkable ability of the neural networks for deriving a general solution of complex systems, it can be used as

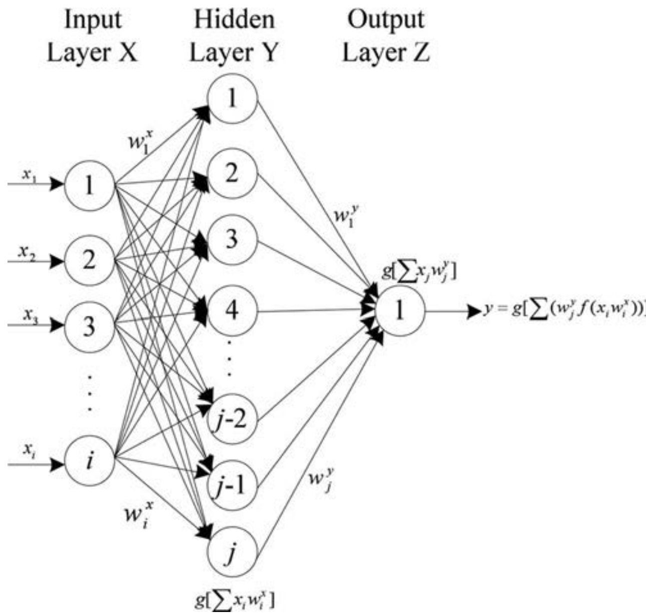

Figure 1. The structure of MLP network.

patterns of extraction and trend detection (Yilmaz and Kaynar 2011; Ebrahimabadi, Azimipour, and Bahreini 2015). With the development of artificial intelligence (AI), ANNs are widely applied in forecasting modeling (Chen et al. 2015). In comparison with the traditional statistical methods, ANNs can solve all nonlinear multivariate functions while the traditional statistical methods can only model the quadratic functions (Gemperline, Long, and Gregoriou 1991; Walczak and Massart 1996). The MLP is quite popular and used more than the other neural network tools associated with complex relationships between input and output variables (Kilic, Ekici, and Hartomacioglu 2015). Thus, practitioners use MLP based on feed-forward in prediction model as it can approximate any arbitrary functions to expect level of accuracy. In this experiment, MLP is trained with a gradient descent-based back-propagation algorithm. The back-propagation rule propagates the errors through the network and allows adaptation of the hidden neurons. Nonlinear activation function allows the neural network to deal with nontrivial problems using a small number of nodes. The NN supports a wide range of activation functions such as step function, linear function, sign function, and sigmoid function (see Figure 2). The sigmoid function is considered as the most popular activation function for two reasons: firstly, it is differentiable that helps to derive a gradient search learning algorithm for networks with multiple layers and; secondly, it generates continuous-valued output rather than binary output produced by the hard-limiter (Alarifi, Alarifi, and Al-Humidan 2012).

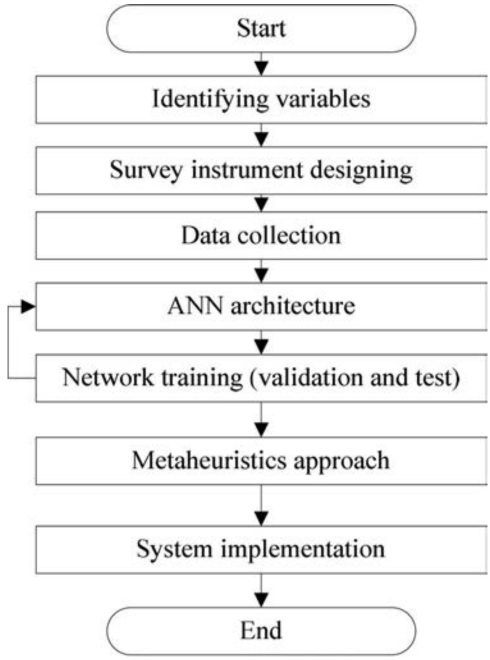

Figure 2. Research methodology.

Research Methodology

The main step of our research methodology following the model of Rouhani and Zare Ravasan (2013) is shown in Figure 2 as follow.

Here, the identification of input and output variables is performed in order to determine the framework of the model. According to Ince and Trafalis (2007), there are more than 100 technical indicators that can be used to analyze the market behavior. Most of them try to predict decisions of purchasing and selling of inventories. It is quite difficult to determine which indicators are proper in determining market fluctuations. We have considered the following input variables according to the literature and purpose of our research work. The 44 variables of which 13 variables from Kim and Han (2000), 6 variables from Chang and Liu (2008), 10 variables from Ince and Trafalis (2007), 15 variables from Göçken et al. (2016) are identified and considered as input variables. It is quite a difficult task to incorporate all available indicators in a model. To mitigate this difficulty, we have designed questionnaires which request opinions of experts about the perceived importance of technical indicators in predicting stock price and their responses are evaluated through the 5-point "Likert scale." We make a comprehensive list of 290 experts and the questionnaire is sent to all experts of stock exchange. The number of returned questionnaires is 225, a response rate of 77.58%. Finally, we select indicators for which the mean of responses are above 60%. Then, we utilize input variables as technical indicators, as shown in Table 1.

Table 1. Technical indicators used to build variables set.

Technical indicators	Notation
Today's close price	TCP
Previous close price	PCP
Previous highest price	PHP
Previous lowest price	PLP
Previous open price	POP
20 day simple moving average of close price	20SMACP
20 day exponential moving average of close price	20EMACP
20 day triangular moving average of close price	20TMACP
Close price moving average convergence/divergence	CPMACD
9-period exponential moving average of MACD	9EMACD
Acceleration opening price	AOP
Acceleration highest price	AHP
Acceleration lowest price	ALP
Acceleration close price	ACP
Momentum open price	MOP
Momentum highest price	MHP
Momentum lowest price	MLP
Momentum close price	MCP
Fast stochastic %K	FS
Slow stochastic %K	SS
Relative strength index	RSI
Bollinger middle band	BMB
Bollinger higher band	BHB
Bollinger lower band	BLB
Median price	MP
Price rate of change	PRC
Exponential moving average	EMA
Accumulation/distribution oscillator	ADO

In order to understand these indicators, some terms which are extensively used by experts are explained as follows:

- Close price: Final price at which a security is traded on a given trading day.
- Open price: The price at which a first bid is proposed on a given trading day.
- Bollinger Bands: Bollinger bands are made by three lines which help investment analyzers to find the trend of stock price.
- Momentum: It indicates the amount by which stock prices are altered over a period of time.
- Oscillator: This is a technical analysis tool that is made from a trend indicator for discovering short-term overbought or oversold conditions.
- Acceleration: Acceleration bands are plotted around a simple moving average as the midpoint and the upper and lower bands are at an equal distance from this midpoint.
- Stochastic: It is used to determine the signals of over-purchasing, overselling, or deviation.
- Moving Average Convergence/Divergence (MACD): This is a well-known indicator which shows the correlation between two price moving averages.
- Relative strength index: It compares the magnitude of recent gains with recent losses. It is an attempt to determine overbought and oversold conditions of an asset.

One of the most important aspects in designing an ANN structure is data collection and preparation. Thus, the cases or examples for training are representative of all the possibilities concerning the application. The data for study of stock price is collected from (http://www.tsetmc.com) February 2009 to April 2016. Generally, there are two main normalizing methods: linear and stochastic (Nguyen and Chan 2004) which are as follows.

1. Linear normalization in range [a, b] is computed by the following formula:

$$x_{norm} = (b-a)\left(\frac{(x-x_{min})}{(x_{max}-x_{min})}\right) + a \tag{1}$$

2. Stochastic normalization is obtained using the following formula, where μ and σ are the mean and the standard deviation, respectively:

$$x_{norm} = \left(\frac{x-\mu}{\sigma}\right) \tag{2}$$

In this work, data are scaled in the range [-1, 1] with respect to the minimum and maximum values of all the data. All inputs and outputs are normalized within a uniform range of [-1, 1] according to the following equation:

$$x_{norm} = 2 \times \left(\frac{(x-x_{min})}{(x_{max}-x_{min})}\right) - 1 \tag{3}$$

Where x is a variable, x_{max} is its maximum value, and x_{min} is its minimum value. One of the most important problems in ANN is called over fitting. Therefore, the data in neural network is divided into test, train, and valid data and it is guaranteed that the network is generalized and would not be over-fitted. To avoid random correlation, these subsets are randomly selected from all the data (Yasin et al. 2014). These data points are divided into train data (70%), validation data (15%), and test data (15%). The numbers of hidden neurons are varied from 1 to 20; and the hyperbolic tangent is utilized for hidden layer while the linear function is utilized for the output layer. The connection between inputs, hidden and output layers comprises of weights (w) and biases (b) which are the main parameters of the NN. Let the input $x = (x_1, x_2, x_3, \ldots, x_n)^T$ be the n-dimensional input vector, where additional B_j is the bias unit and x_j denotes the jth component of x. The nk-dimensional weight vectors are $w_{ij} = (w_{ij1}, w_{ij2}, w_{ij3}, \ldots, w_{ijn})^T$ where $i = 1, 2, \ldots, k$ and k is the analogous order of the network. The output at the hidden layer is computed by Eq. (4) as follows:

$$h_j = B_j + \sum_{i=1}^{n} x_i w_{ij}, \tag{4}$$

where h_j is the output at the hidden layer and w_{ij} represents the weight from the input, and x_i is the input parameter. The weighted output is then passed through an activation function. In Eq. (4), the sum of the weights in the hidden layer to the output layer is fixed to 1, and the output O is computed by the following equation.

$$O = f\left(\prod_{j=1}^{k} h_j\right), \tag{5}$$

where $f(\cdot)$ becomes a suitable activation function. The hyperbolic tangent-sigmoid (tansig), logsigmoid (logsig), and linear (purelin) functions are generally used as activation functions to solve nonlinear and linear regression problems. In this study, tansig is used as an activation function between the input and hidden layers, while purelin is used as activation function between the hidden and output layers. Figure 3 shows commonly used activation functions in ANNs.

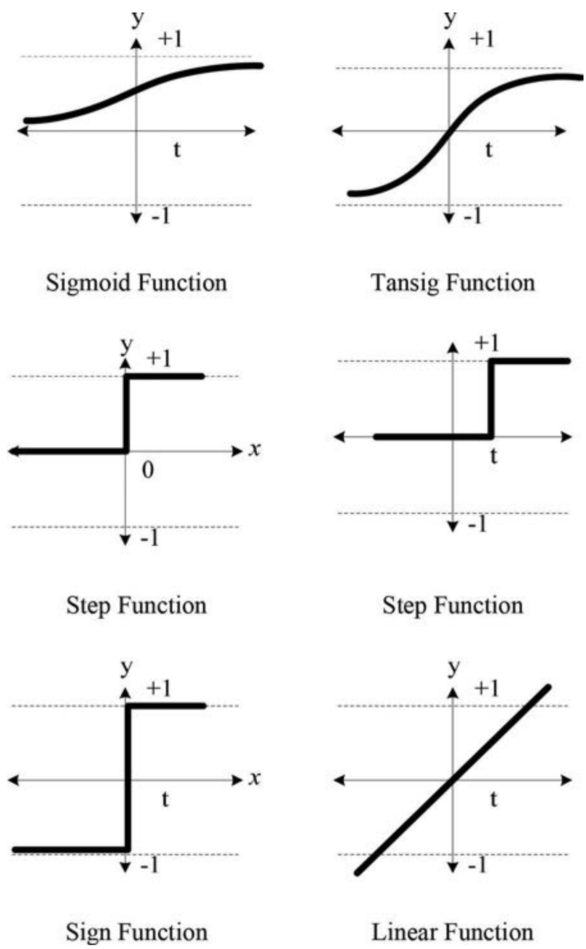

Figure 3. Common activation functions in ANN.

As far as network training is concerned, there are two different approaches that are considered in this literature. The first is the classical approach ANNs those are fundamentally nonlinear models used to distinguish patterns and classify variables. In this context, several researchers have developed some supervised and unsupervised training methods. The classical approaches related to the back-propagation (BP) model and Levenberg–Marquardt algorithm have better performance than other models in this classification. The second is metaheuristic approach to train ANNs. But, we have used a new metaheuristic approach to train ANNs. The selected activation function is tansig for the hidden layer and output layer. To validate our proposed model, the mean squared error (MSE), root mean squared error (RMSE), mean absolute error (MAE), and regression coefficient (R) are computed as follows:

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(fo_i - fe_i)^2 \tag{6}$$

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(fo_i - fe_i)^2} \tag{7}$$

$$MAE = \frac{1}{N}\sum_{i=1}^{N}|fo_i - fe_i| \tag{8}$$

$$R = \frac{\sum_{i=1}^{N}\left(fo_i - \overline{fo_i}\right)\left(fe_i - \overline{fe_i}\right)}{\sqrt{\sum_{i=1}^{N}\left(fo_i - \overline{fo_i}\right)^2 \sum_{i=1}^{N}\left(fe_i - \overline{fe_i}\right)^2}} \tag{9}$$

Here, fe_i and fo_i denote the experimental and network outputs, respectively. The terms $\overline{fe_i}$ and $\overline{fo_i}$ are the average of the above-mentioned data, respectively, and N is the total number of data. In such classical models, there are some limitations such as more training time and inexact prediction capability, especially for high range of data. To overcome these limitations, we can use metaheuristic techniques in order to train the network. The basic approach for stock market prediction is assumed to be a linear metaheuristic approach based on the forecasting model with parallel inputs as shown in Figure 4. The weights of the model are considered as the evolutionary operators and initialized with random numbers. In any metaheuristic concept, any solution of the model is represented as a population, and the mean square error (MSE) is considered as a fitness function, and values of the solutions are updated during iterations.

To estimate any complex relation between the input and output data, ANN uses transfer functions for hidden layers in short time with high accuracy. As each network has its own specification, a common method is not applied to determine how many hidden layers on how many neurons in ANN must have. As a result, the consensus of opinion to perform the best stricter in

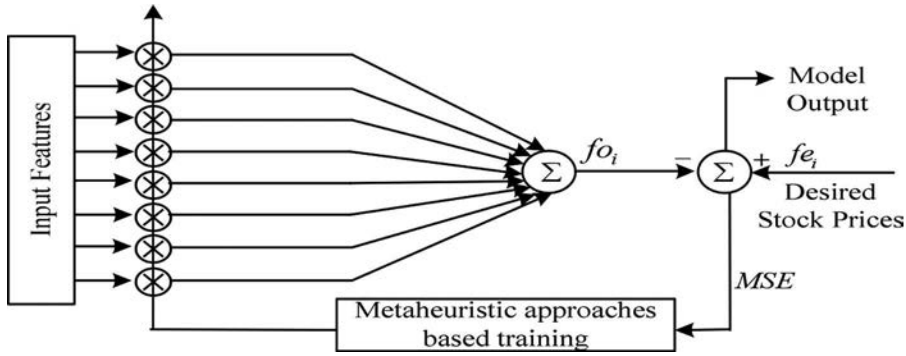

Figure 4. Linear metaheuristic approach-based forecasting model.

Table 2. Structure of neural network used for proposed approach.

Type of network	Explanation
Number of layers	4
Number of hidden layers	2
Number of neurons	5
Transfer function for hidden layers	tansig
Transfer function for input and outer layer	purelin
Preprocessing	Transfer data into the range [−1 1]
Percentage of training, test, and valid set	70, 15, 15
Number of input	28

ANN is evaluated by trial and error (Lin and Tseng 2000; Fernandez et al. 2013) method. Similar to the models of Yasin et al. (2014) and Göçken et al. (2016), the best configuration is achieved in this work using the linear transfer function (purelin) in the output layer and the tansig transfer function in the hidden layer. Other proper specifications for the structure of the employed network are shown in Table 2. It is noted that most specifications of this network are derived by trial and error.

Metaheuristic Approaches

In this section, the relevant approaches are described so as to analyze our problem. Cuckoo Search (CS) algorithm is presented in the first subsection, then how CS algorithm performance improves via successful manipulation is detailed and GA and PSO are described shortly. Finally, improved cuckoo search genetic algorithm (ICSGA) is studied at the end of this section.

Cuckoo Search Algorithm

In recent years, numerous works on this topic have been presented based on swarm algorithms. These algorithms include the biological evolutionary processes such as genetic algorithm (GA), particle swarm optimization algorithm (PSO) (Kennedy and Eberhard 1995), harmony search (HS) (Geem, Kim, and Loganatha 2001), bacterial foraging optimization algorithm (BFOA) (Passino 2002), artificial bee colony algorithm (ABC), central force optimization algorithm (CFO) (Formato 2007), group search optimizer (GSO) (He, Wu, and Saunders 2009), krill herd algorithm (KH) (Gandomi and Alavi 2012), optics-inspired optimization (OIO) (Husseinzadeh 2015), biogeography-based optimization (BBO) (Simon 2008), ant colony optimization (ACO), and backtracking search Optimization (BSO) (Civiciolu 2013). The CS algorithm was introduced by Yang and Deb based on the LÅvy flight behavior and brood parasitic behavior (Yang and Deb 2009). In many scientific literature, the CS algorithm is elegantly demonstrated as providing excellent performance in optimization such as power flow (Rao and Naresh Babu 2013), symmetric linear antenna array (Abdulrani, Abdmalek, and Siew-Chin 2012), neural network training (Valian, Mohanna, and Tavakoli 2011; Bhandari et al. 2014), image segmentation and other optimization (Arulanand, Subramanian, and Premalatha 2012). The main idea of the algorithm is based on the breeding behavior, such as brood parasitism, of some species.

Like other evolutionary methods, CS also starts with an initial population. The new solution $x^{(t+1)}$ for a cuckoo i is generated with the help of *Lévy* flight distribution as follows:

$$x_i^{(t+1)} = x_i^{(t)} + \alpha \otimes L\acute{e}vy(\lambda), \tag{10}$$

where $\alpha(\alpha > 0)$ represents a step size. To determine the step size, we should pay attention to the scales of problem. *Lévy* flight is a random walk with the random step size following a *Lévy* distribution, as follows:

$$L\acute{e}vy(\lambda) \sim u = t^{-\lambda}, \ (1 \leq \lambda \leq 3) \tag{11}$$

There are a few ways for the generation of steps of the Lévy flights. One of the most efficient and yet straightforward ways is the so-called Mantegna (1994) algorithm for a symmetric Lévy stable distribution. Here, "symmetric" means that the steps can be positive or negative. In Mantegna's algorithm, the step length can be calculated as follows:

$$S = \frac{u}{|v|^{1/\beta}}, \tag{12}$$

where $0 < \beta \leq 2$ is an index, and u and v are stochastic variables drawn from normal distributions as follows:

$$u \sim N\left(0, \sigma_u^2\right), \ v \sim N\left(0, \sigma_v^2\right), \tag{13}$$

$$\sigma_u = \left[\frac{\Gamma(1+\beta) sin \frac{\pi\beta}{2}}{\Gamma\left(\frac{1+\beta}{2}\right) \beta . 2^{(\beta-1)/2}} \right]^{1/\beta}, \ \sigma_v = 1 \tag{14}$$

Finally, Gamma function $\Gamma(x)$ is calculated by the following formulae:

$$\Gamma(x) = \int_0^\infty t^{z-1} e^{-t} dt \tag{15}$$

In a nutshell, CS is a population-based algorithm and the initial population is generated randomly within the limits of the control parameter. Then, the levy flight operator is performed on all individuals until a stopping criterion is reached.

Improved Cuckoo Search

In our study, we have applied the improved cuckoo search (ICS) algorithm based on the work of Valian, Mohanna, and Tavakoli (2011). There are two important parameters in the cuckoo search. These parameters take a fixed value in the traditional version of cuckoo search and are introduced in order to find globally and locally improved solutions. The first parameter is p_a which is important in the fine-tuning of solution vectors, and it is used in enhancing the convergence rate of the algorithm. The second parameter λ is the step size related to the scales of problem. If these parameters are not tuned well, the performance of the algorithm would be poor that results in large number of iterations or loss best solution.

In order to solve potential problem arisen from traditional version of algorithm, we have investigated ICS which is the focus on these parameters (p_a and λ). The most significant difference between the ICS and CS is the way of tuning p_a and λ. Fixed value parameters (p_a and λ) of the CS algorithm lead to drawbacks in computing the best solution. To improve the performance of the CS algorithm and eliminate difficulties related to tuning p_a and λ, the ICS algorithm uses variables p_a and λ. The values of p_a and λ should be dynamically changed with the number of iteration and decreased from high value to low value in order to obtain a better fine-tuning of solution vectors. Therefore, the Eqs. (16)–(18) are used as ICS operators, where *NI* and *gn* indicate the number of total iterations and the current iteration, respectively (Valian, Mohanna, and Tavakoli 2011).

$$P_a(gn) = P_{a_{max}} - \frac{gn}{NI}(P_{a_{max}} - P_{a_{min}}) \tag{16}$$

$$\alpha(gn) = \alpha_{max} \ exp \ (c \cdot gn) \tag{17}$$

$$c = \frac{1}{NI} \ ln \left(\frac{\alpha_{min}}{\alpha_{max}} \right) \tag{18}$$

Genetic Algorithm (GA)

GA is a stochastic global search technique that solves problems by imitating the processes observed during natural evolution (Kuo and Han 2011). The procedure of GA is a simulation following biological evolution behavior. GA not only adopts the spirit of creature elimination rule but also finds the approximate optimal solution after the process of coding, decoding, and constant operation (reproduction, crossover, and mutation). GA is performed in many applications such as humanitarian logistics network design (Tofighi, Torab, and Mansouri 2016), job shop scheduling (Asadzadeh 2015), prediction (Tsoukalas and Fragiadakis 2016), inventory control (Çelebi 2015), and risk of project (Pfeifera et al. 2015). Crossover and mutation are introduced as main GA operators. In the next subsection, we shall describe how to apply these operators in the ICS to achieve better performance. Crossover and mutation are considered as the main operators of GA. Several interesting methods are demonstrated in the literature (Haupt and Haupt 2004). We have used blending methods combining values of variables of two solutions into new solutions called offspring. Let S_{new} denote a variable value of an offspring which is calculated by the following formulae:

$$S_{new} = \alpha \times S_{in} + (1 - \alpha) \times S_{jn} \tag{19}$$

Where α is a random number which belongs to the interval [0, 1], S_{in} is the n^{th} variable in the mother chromosome, and S_{jn} is the nth variable in the father chromosome.

The other important operator of GA that randomly changes one or more of the gene(s) in the chromosome is mutation. The main purpose of this operator is to prevent the genetic population from converging to a local minimum and introduce to the population as new possible solutions. The mutation is carried out according to the probability of mutation, which is calculated by the formulae:

$$M_{new} = M_i + \sigma\ N\ (0, 1), \tag{20}$$

where M_i denotes a randomly selected variable of a chromosome. The term σ is the standard deviation of the normal distribution $N(0, 1)$.

Particle Swarm Optimization (PSO)

The PSO based on the behavior of birds was been proposed by Eberhart and Kennedy (1995). Generally speaking, from the view point of initiating with a population of random solutions, PSO is similar to GA. PSO has been successfully applied in many contexts and also it has been detailed by researchers. The two main equations in PSO are as follows:

$$\begin{aligned} V_i(t+1) = \omega \times V_i(t) + c_1 \times rand(n) \times (lbest_i(t) - X_i(t)) \\ + c_2 \times rand(n) \times (gbest_i(t) - X_i(t)) \end{aligned} \tag{21}$$

$$\text{and } X_i(t+1) = X_i(t) + V_i(t+1), \tag{22}$$

where ω is the inertia weight, $V_i\ (t)$ and $V_i\ (t+1)$ are the velocities of the ith particle at time t and $(t+1)$ in the population, respectively. The terms c_1 and c_2 are acceleration coefficients, $X_i\ (t)$ is the position of the ith particle and $lbest_i\ (t)$ and $gbest_i\ (t)$ are the local best particles of the ith particle and the global best particle among local bests at time t, respectively. The term $rand\ (n)$ generates a random value between 0 and 1.

Hybrid of ICS with GA

There are many studies which have introduced hybrid GA with other algorithms. But a hybrid of GA and CS is rarely found in the literature. The idea of hybrid GA and CS has been studied by Abu-Srhan and Al Daoud

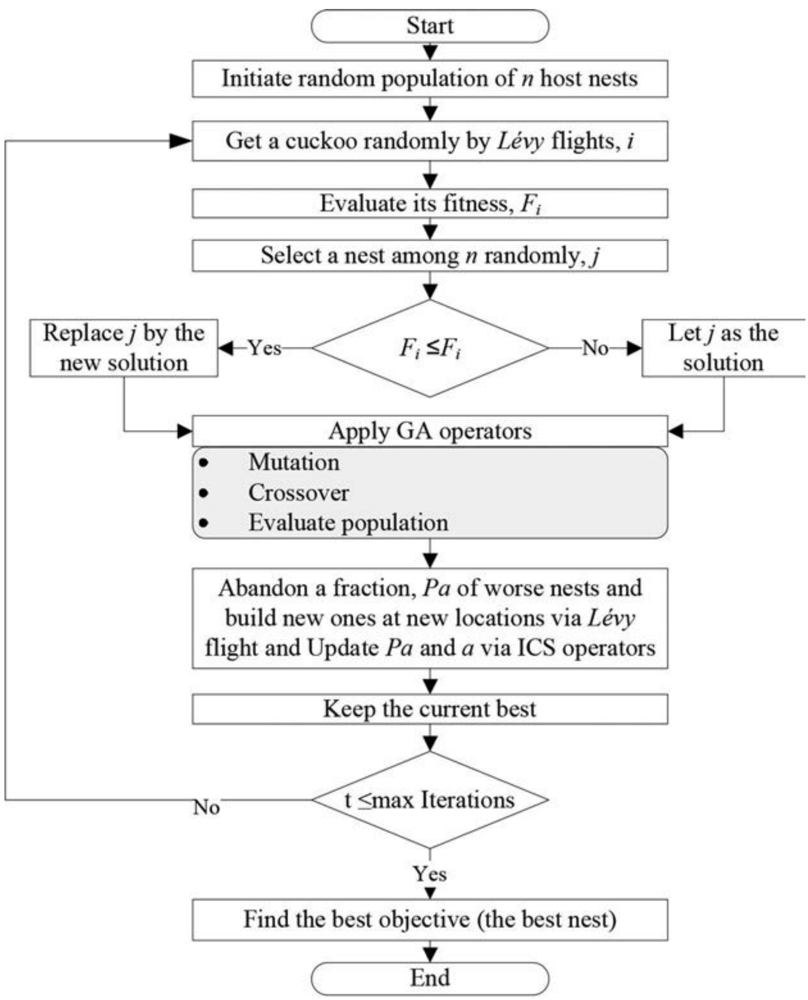

Figure 5. The flowchart of the ICSGA approach.

Table 3. Parameter description.

(1) Common parameters
(1.1) Number of population (nests) = 40
(1.2) Number of Iterations = 100
(2) CS, ICS and ICSGA
(2.1) $\beta = 1.5$
(2.1) $\alpha = 1$
(3) GA and ICSGA
(3.1) Crossover probability = 0.7
(3.2) Mutation probability = 0.03
(4) ICSGA and ICS
(4.1) $P_{a_{max}} = 1$
(4.2) $P_{a_{min}} = 0$
(5) CS
(5.1) Pa = 0.3
(6) PSO
(6.1) $\omega = 0.9$
(6.2) $c_1 = 2$
(6.3) $c_2 = 2$

(2013). They have combined the advantages of GA and ICS and have overcome the main disadvantage of GA easily becoming trapped in the local minima through the ICS. The main steps of the proposed approach are introduced as follows (Figure 5).

In Figure 5, the last step in the common CS algorithm is the rejection and replacement of a percent of the worst solutions with new randomly generated valid solution vectors. No matter which one of these procedures is carried out, a group of solutions are selected according to genetic algorithm and then mutation and crossover operators are applied accordingly. In this case, different approaches of metaheuristics are used, thus the stopping criteria must be set for the algorithm. There are some theoretical guidelines for determining the stopping time for the algorithm (Bhandari, Murthy, and Pal 2012). In this study, we have simply used the stopping criterion based on the maximum number of iterations. Hence, the maximum number of iterations is set to 100 for all algorithms. Each algorithm has its own parameters. The specific input parameters of all algorithms reported in the related literatures are specified in Table 3.

In order to make the current research, an expert system is designed and developed in a MATLAB environment and a personal computer with CPU 1.6 GHz Intel Core i5, OS X Yosemite (version 10.10.2), and a 4.0-GB 16,000 MHz DDR3 installed memory (RAM) is used to achieve all the results.

Data Analysis

In this section, the ANN with the proposed architecture is trained by 1609 data records of which 1126 records are considered for the training set, 241 for the validation set, and 242 for the test set. In the training phase, the training and validation sets are used together. The different metaheuristic methods are used to train the network and the connection weight of neurons is computed in accordance with the decreasing measure of mean square error (MSE) between network outputs and observed outputs.

As shown in Figure 6, the MSE values gained from various metaheuristics go down while the iterations of training increase. Among the early iterations, GA had the best performance and its MSE value is calculated as 0.0209, while in the recent iterations, PSO outperforms the other metaheuristic approaches and its MSE value is 0.0013. The results show that the approaches achieve good result within the range 0.0013 (MSE value of PSO)–0.0074 (MSE value of GA). PSO shows very good result when MSE value is around 0.043 in the first iteration and converges at 0.0013 in the final iteration. Among all the metaheuristic approaches, GA has the worst performance starting at 0.078 in the first iteration and converges at 0.0074 in the final iteration.

Similar to Figure 6, we have an apparently homogeneous result in the RMSE analysis as shown in Figure 7, where ICSGA outperforms other approaches while PSO shows superior performance in the last iteration. Therefore, PSO shows very good result with a RMSE value starting above 0.2071 in the first iteration and converging at 0.0363 in the final iteration. On the other hand, GA has the worst performance starting around 0.2793 in the first iteration and converges at 0.0861 in the final iteration.

Another validation criterion discussed in this section is meant to study absolute error (MAE) which is shown in Figure 8. Here, PSO shows superior

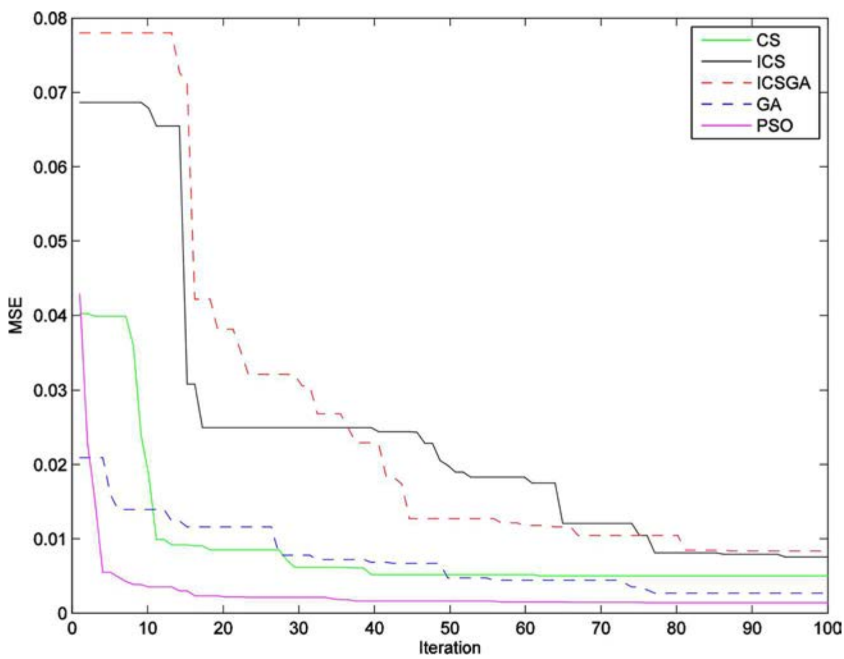

Figure 6. MSE convergence of metaheuristics.

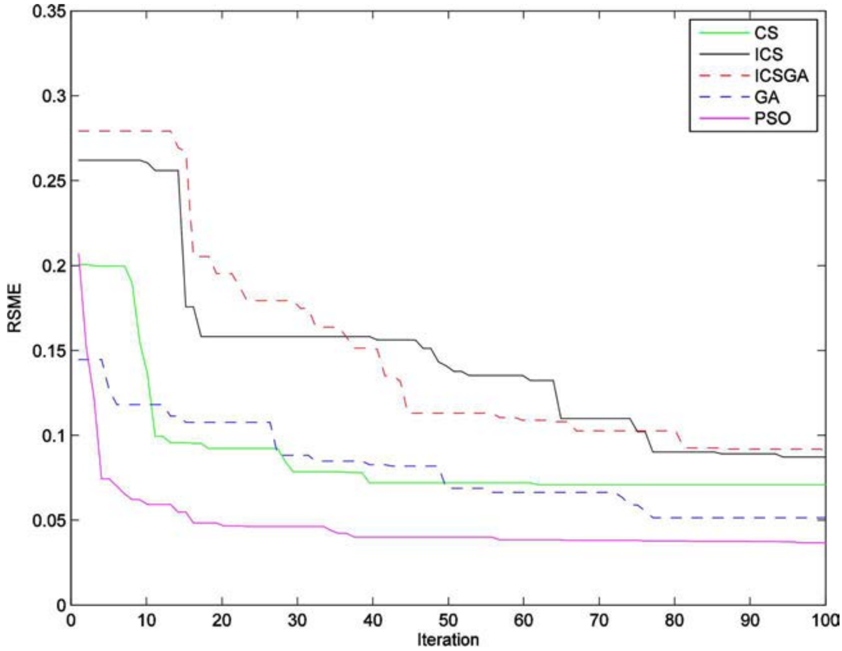

Figure 7. RMSE convergence of metaheuristics.

performance compared to other approaches. So, this criterion has similar performance to MSE and RMSE.

In order to evaluate the accuracy of the ANN, a test data set is used after training the network. The test aims to assess the accuracy of the results predicted by the system. In this section, we have just drawn diagrams on

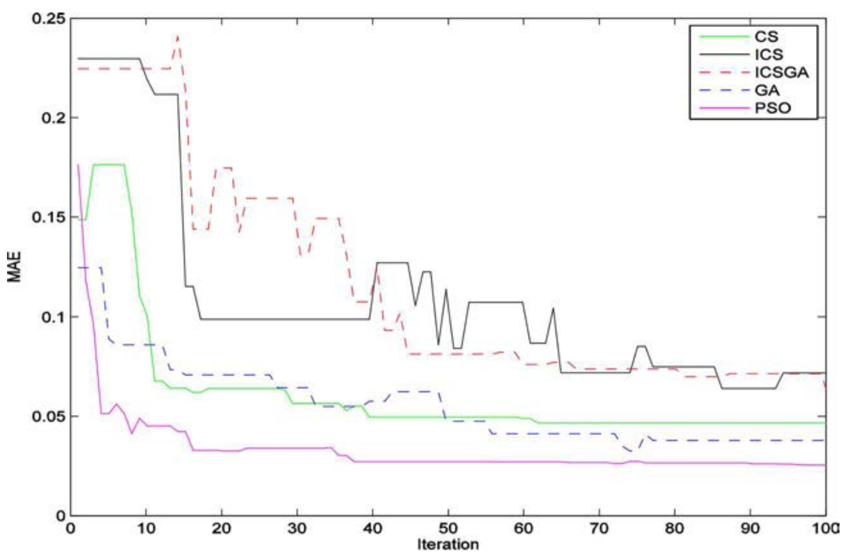

Figure 8. MAE convergence of metaheuristics.

the test data set to show the accuracy of the proposed approach. The comparison between actual and predicted close price based on the proposed approaches are demonstrated in Figure 9.

Generally speaking, regression analysis is a statistical approach for modeling the relationship between a dependent variable and one or more independent variables. The diagrams of performance of the regression coefficient (R) based on various approaches are shown in Figure 10. Based on the results, the PSO algorithm has a far better performance than other approaches. A comparisons of the performance of the proposed approaches

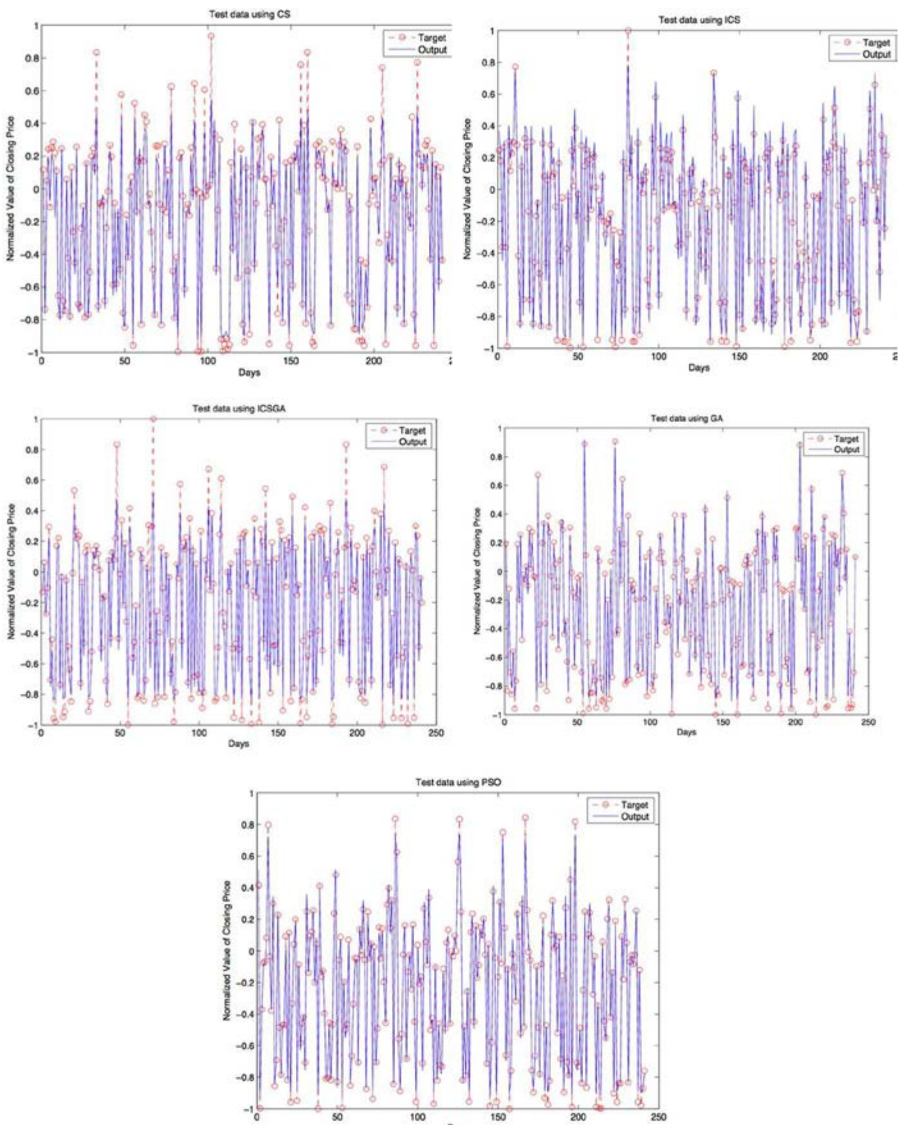

Figure 9. Normalized values of closing price versus days in various approaches.

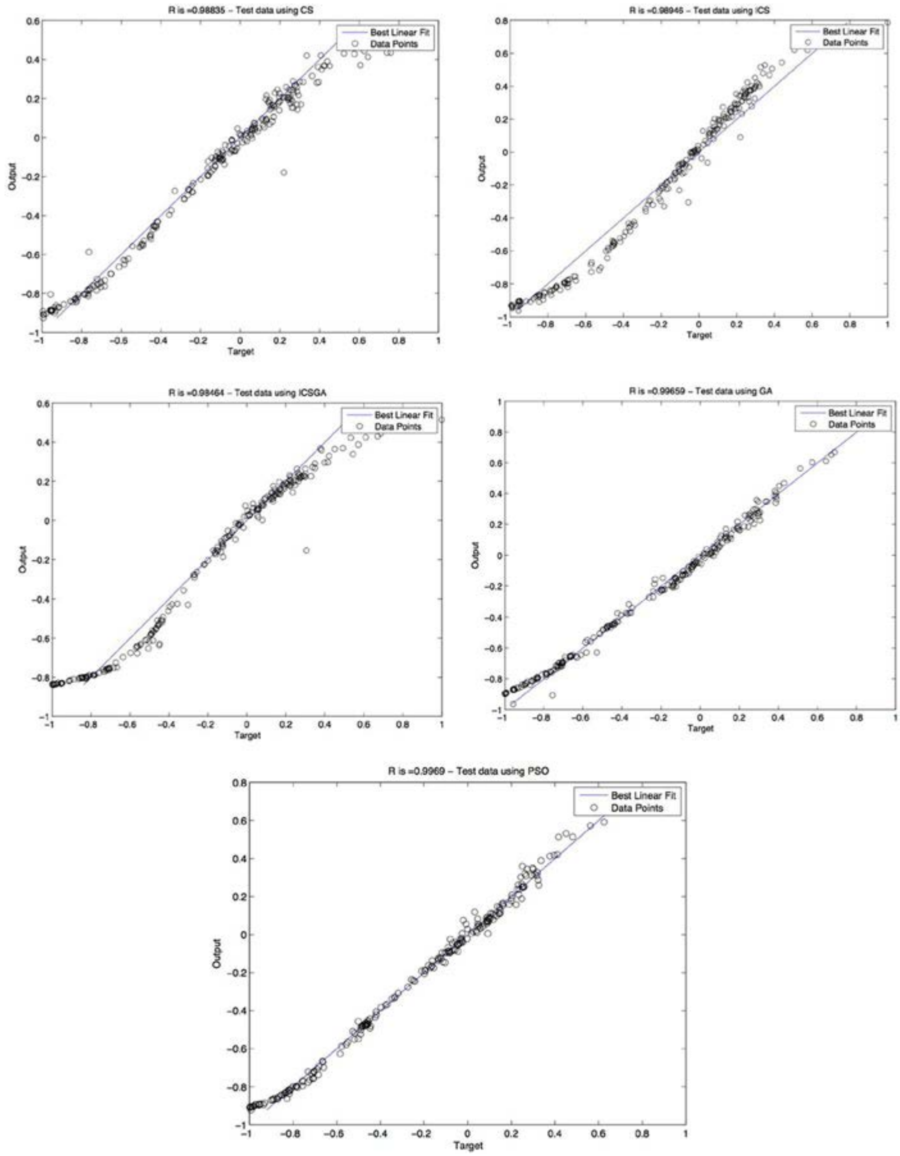

Figure 10. Regression coefficient (R) based on various approaches.

are presented in Table 4. It constitutes a summary of problem specification in which a considerable variety in the number of patterns, attributes, and classes are illustrated.

In order to obtain the score of any data type, the mean of ranking based on validation criteria (MSE, RMSE, MAE, and R) are computed in Table 4. In this table, the final score is computed based on the mean of train score, test score, validation score, and total score. For example, the first row of the aforementioned table is related to the value and rank of the train data of CS algorithm. The mean of these ranks (9, 9, 9, and 8) is computed as train

Table 4. Summary of MSE, RMSE, MAE, and R for ANN based on various approaches.

Proposed approaches	Type of data	MSE		RMSE		MAE		R	
		Value	Rank	Value	Rank	Value	Rank	Value	Rank
CS	Train	0.0050	9	0.0706	9	0.0466	9	0.9912	8
	Test	0.0065	11	0.0805	11	0.0503	11	0.9883	15
	Validation	0.0071	14	0.0843	15	0.0509	12	0.9884	14
	Total	0.0055	10	0.0743	10	0.0478	10	0.9904	10
ICS	Train	0.0076	19	0.0871	19	0.0720	20	0.9866	17
	Test	0.0068	12	0.0822	12	0.0694	17	0.9895	11
	Validation	0.0068	12	0.0827	13	0.0696	18	0.9890	12
	Total	0.0073	16	0.0857	16	0.0712	19	0.9886	13
GA	Train	0.0074	17	0.0859	17	0.0584	15	0.9866	17
	Test	0.0079	20	0.0890	20	0.0585	16	0.9855	20
	Validation	0.0071	14	0.0842	14	0.0546	13	0.9878	16
	Total	0.0074	17	0.0861	18	0.0578	14	0.9865	19
ICSGA	Train	0.0027	6	0.0517	6	0.0378	6	0.9948	6
	Test	0.0018	5	0.0420	5	0.0332	5	0.9936	7
	Validation	0.0040	8	0.0636	8	0.0436	8	0.9949	5
	Total	0.0027	6	0.0524	7	0.0380	7	0.9905	9
PSO	Train	0.0013	1	0.0363	2	0.0253	2	0.9972	1
	Test	0.0014	4	0.0371	4	0.0260	4	0.9969	4
	Validation	0.0013	1	0.0357	1	0.0245	1	0.9971	2
	Total	0.0013	1	0.0363	2	0.0253	2	0.9971	2

score of CS algorithm shown in Table 5. Additionally, the final rank is achieved from the final score, computed by the average rank of algorithm in different costs. It reveals that the result obtained by PSO is better than other approaches. Among the other four approaches, ICSGA has the smallest final score and goes to rank 2 out of 5 approaches. From Table 4 and the converging paths shown in Figures 6–8, GA approach has the worst performance compared to other approaches.

In this paper, neural network is used to improve the multivariate prediction ability based on prior knowledge of stock price prediction which is difficult to insert into initial network structures to evaluate error measurements. Event-knowledge is extracted from survey of experts of stock exchange and input them into neural networks. The adoption of a knowledge-based system requires that users, experts, and managers have a good understanding of the concepts of information resources, to form input data for stock price prediction using appropriate optimization techniques. In order to build a structural system with a knowledge repository, it is necessary to sort out tremendous amount of data while generating information to support decision making with intelligence features. This analysis suggests a manager of a firm to forecast

Table 5. The rank of algorithms obtained from different data types.

	Train score	Test score	Validation score	Total score	Final score	Final rank
CS	8.75	12	13.75	10	11.125	3
ICS	18.75	13	13.75	16	15.375	4
GA	16.5	19	14.25	17	16.6875	5
ICSGA	6	5.5	7.25	7.25	6.5	2
PSO	1.5	4	1.25	1.75	2.125	1

optimal stock price to sell the inventory to achieve optimal goal of the firm through the generation of neural decision support.

Conclusion

Over the last decade, the applications and trends of neural networks have been noticed by many researchers. In this area, application of ANN techniques in stock price prediction is an emerging topic of the academia in industries. To make the best investment decisions on financial markets, several researchers have studied the possibility of predicting the stock market fluctuations. Hence, there is no general agreement on the effectiveness of the forecasting models. This paper has proposed a new hybrid model based on heuristic optimization methodologies (CS, ICS, ICSGA, GA, and PSO) and ANN to enhance stock market forecasting performance. The basic objective of this paper is to show the effectiveness of the proposed approaches in predicting stock price and then to prove which approach is better than others. So, findings of statistical analyses indicate that the hybrid of ANNs and metaheuristic algorithms have a significant role in predicting stock price accurately. It is noticed that the PSO has superior performance to find out the best fitness functional value in the neural network. Also, the hybridized ICS and GA (ICSGA) algorithm reaches second place in this contest. Based on the simulation results obtained from the proposed approaches, it can be concluded that proposed approaches can be efficiently used in stock price prediction with a better convergence at less error rates.

Many extensions of the presented work could be aimed by future researches. In our future work, the focus will be on the hybridization of ANN with some other recently developed metaheuristics, like grey wolf optimizer (GWO) (Mirjalili, Mirjalili, and Lewis 2014), bees algorithm (BA) (Pham et al. 2016), artificial fish swarm algorithm (AFS) (Farzi 2009), symbiotic organisms search (SOS) (Cheng and Prayogo 2014), spider optimization algorithm (SOA) (Cuevas et al. 2013), shuffled frog leaping (SFL) (Eusuff, Lansey, and Pasha 2006), monkey search algorithm (MSA) (Zhao and Tang 2008), or keshtel algorithm (KA) (Hajiaghaei-Keshteli and Aminnayeri 2014), to predict stock price. Moreover, it will be a good idea to design new solution procedures that consist of function approximation and the number of neurons. Apart from this, a deep experimental investigation on the values of other parameters like number of hidden layer transfer function for input and outer layer can be studied further. Also, the sensitive analysis of proposed model can be analyzed in the near future.

References

Abdulrani, K., F. Abdmalek, and N. Siew-Chin. 2012. Nature-inspired cuckoo search algorithm for side lobe uppression in a symmetric linear antenna array. *Radioengineering* 21:865–74.

Aboueldahab, T., and M. Fakhreldin. 2011. Prediction of stock market indices using hybrid genetic algorithm/particle swarm optimization with perturbation term. Paper presented at the International Conference on Swarm Intelligence, Cergy, France, June 5.

Abu-Srhan, A., and E. Al Daoud. 2013. A hybrid algorithm using a genetic algorithm and cuckoo search algorithm to solve the traveling salesman problem and its application to multiple sequence alignment. *International Journal of Advanced Science and Technology* 61:29–38. doi:10.14257/ijast.2013.61.04

Adebayo, I. M., D. Saiang, and E. Nordlund. 2015. Stochastic assessment of pillar stability at Laisvall Mine using artificial neural network. *Tunnelling and Underground Space Technology* 49:307–19. doi:10.1016/j.tust.2015.05.003

Alarifi, A. S. N., N. S. N. Alarifi, and S. Al-Humidan. 2012. Earthquakes magnitude predication using artificial neural network in northern red sea area. *Journal of King Saud University–Science* 24:301–13. doi:10.1016/j.jksus.2011.05.002

Aminian, F., E. D. Suarez, M. Aminian, and D. T. Walz. 2006. Forecasting economic data with neural networks. *Computational Economics* 28:71–88. doi:10.1007/s10614-006-9041-7

Arulanand, N., S. Subramanian, and K. Premalatha. 2012. An enhanced cuckoo search for optimization of bloom filter in spam filtering. *Global Journal of Computer Science and Technology* 12:75–81.

Asadzadeh, L. 2015. A local search genetic algorithm for the job shop scheduling problem with intelligent agents. *Computers & Industrial Engineering* 85:376–83. doi:10.1016/j.cie.2015.04.006

Bae, J. H., K. C. Jung, J. W. Kim, and H. J. Kim. 1998. Segmentation of touching characters using an MLP. *Pattern Recognition Letters* 19:701–09. doi:10.1016/s0167-8655(98)00048-8

Benardos, A., and D. Kaliampakos. 2004. Modelling TBM performance with artificial neural networks. *Tunnelling and Underground Space Technology* 19:597–605. doi:10.1016/j.tust.2004.02.128

Bhandari, A. K., V. K. Singh, A. Kumar, and G. K. Singh. 2014. Cuckoo search algorithm and wind driven optimization based study of satellite image segmentation for multilevel thresholding using Kapur's entropy. *Expert Systems with Applications* 41:3538–60. doi:10.1016/j.eswa.2013.10.059

Bhandari, D., C. Murthy, and S. K. Pal. 2012. Variance as a stopping criterion for genetic algorithms with elitist model. *Fundamenta Informaticae* 120:145–64.

Boyacioglu, M. A., and D. Avci. 2010. An adaptive network-based fuzzy inference system (ANFIS) for the prediction of stock market return: The case of the Istanbul stock exchange. *Expert Systems with Applications* 37:7908–12. doi:10.1016/j.eswa.2010.04.045

Çelebi, D. 2015. Inventory control in a centralized distribution network using genetic algorithms: A case study. *Computers & Industrial Engineering* 87:532–39. doi:10.1016/j.cie.2015.05.035

Chang, P. C., and C. H. Liu. 2008. A TSK type fuzzy rule based system for stock price prediction. *Expert Systems with Applications* 34:135–44. doi:10.1016/j.eswa.2006.08.020

Chen, F., H. Li, Z. Xu, S. Hou, and D. Yang. 2015. User-friendly optimization approach of fed-batch fermentation conditions for the production of iturin a using artificial neural networks and support vector machine. *Electronic Journal of Biotechnology* 18:273–80. doi:10.1016/j.ejbt.2015.05.001

Chen, Y., X. Dong, and Y. Zhao. 2005. Stock index modeling using EDA based local linear wavelet neural network. In *International Conference on Neural Networks and Brain*, *2005*, 1646–1650. IEEE.

Cheng, M. Y., and D. Prayogo. 2014. Symbiotic organisms search: A new metaheuristic optimization algorithm. *Computers & Structures* 139:98–112. doi:10.1016/j.compstruc.2014.03.007

Civiciolu, P. 2013. Backtracking search optimization algorithm for numerical optimization problems. *Applied Mathematic and Computation* 219:8121–44. doi:10.1016/j.amc.2013.02.017

Cuevas, E., M. Cienfuegos, D. Zaldívar, and M. Pérez-Cisneros. 2013. A swarm optimization algorithm inspired in the behavior of the social-spider. *Expert Systems with Applications* 40:6374–84. doi:10.1016/j.eswa.2013.05.041

Darbellay, G. A., and M. Slama. 2000. Forecasting the short-term demand for electricity: Do neural networks stand a better chance? *International Journal of Forecasting* 16:71–83.

Eberhart, R. C., and J. Kennedy. 1995. A new optimizer using particle swarm theory. In *Proceedings of the Sixth International Symposium on Micro Machine and Human Science*, 39–43. New York, NY: Science and Education Publishing.

Ebrahimabadi, A., M. Azimipour, and A. Bahreini. 2015. Prediction of roadheaders' performance using artificial neural network approaches (MLP and KOSFM). *Journal of Rock Mechanics and Geotechnical Engineering* 7:573–83. doi:10.1016/j.jrmge.2015.06.008

Eusuff, M., K. Lansey, and F. Pasha. 2006. Shuffled frog-leaping algorithm: A memetic metaheuristic for discrete optimization. *Engineering Optimization* 38:129–54. doi:10.1080/03052150500384759

Farzi, S. 2009. Efficient job scheduling in grid computing with modified artificial fish swarm algorithm. *International Journal of Computer Theory and Engineering* 1:13. doi:10.7763/ijcte.2009.v1.3

Feng, S., L. Li, L. Cen, and J. Huang. 2003. Using MLP networks to design a production scheduling system. *Computers & Operations Research* 30:821–32. doi:10.1016/s0305-0548(02)00044-8

Fernandez, F. G., F. L. Redondo, I. S. Los Santos, J. L. Martinez, and S. I. Izquierdo. 2013. *Use of artificial neural networks to predict the business success or failure of start-up firms.* Rijeka, Croatia: INTECH Open Access Publisher.

Formato, R. A. 2007. Central force optimization: A new metaheuristic with applications in applied electromagnetics. *Progress in Electromagnetics Research (PIER)* 77:425–91. doi:10.2528/pier07082403

Gandomi, A. H., and A. H. Alavi. 2012. Krill herd: A new bio-inspired optimization algorithm. *Communications in Nonlinear Sciences and Numerical Simulations* 17:4831–45. doi:10.1016/j.cnsns.2012.05.010

Geem, Z. W., J. H. Kim, and G. V. Loganatha. 2001. A new heuristic optimization algorithm: Harmony search. *Simulation* 76:60–68. doi:10.1177/003754970107600201

Gemperline, P., J. R. Long, and V. G. Gregoriou. 1991. Nonlinear multivariate calibration using principal components regression and artificial neural networks. *Analytical Chemistry* 63:2313–23. doi:10.1021/ac00020a022

Göçken, M., M. Özçalıcı, A. Boru, and A. T. Dosdoğru. 2016. Integrating metaheuristics and Artificial Neural Networks for improved stock price prediction. *Expert Systems with Applications* 44:320–31. doi:10.1016/j.eswa.2015.09.029

Güler, E. Ç., B. Sankur, Y. P. Kahya, and S. Raudys. 1998. Visual classification of medical data using MLP mapping. *Computers in Biology and Medicine* 28:275–87. doi:10.1016/s0010-4825(98)00010-9

Guresen, E., G. Kayakutlu, and T. U. Daim. 2011. Using artificial neural network models in stock market index prediction. *Expert Systems with Applications* 38:10389–97. doi:10.1016/j.eswa.2011.02.068

Hajiaghaei-Keshteli, M., and M. Aminnayeri. 2014. Solving the integrated scheduling of production and rail transportation problem by keshtel algorithm. *Applied Soft Computing* 25:184–203. doi:10.1016/j.asoc.2014.09.034

Hamed, M., M. Khalafallah, and E. Hassanien. 2004. Prediction of wastewater treatment plant performance using artificial neural networks. *Environmental Modelling and Software* 19:919–28. doi:10.1016/j.envsoft.2003.10.005

Han, J., and M. Kamber. 2001. *Data mining concepts and techniques*. San Francisco, Moraga: Kaufman.

Haupt, R. L., and S. E. Haupt. 2004. *Practical genetic algorithms*. Hoboken, NJ: John Wiley & Sons.

He, S., Q. H. Wu, and J. R. Saunders. 2009. Group search optimizer: An optimization algorithm inspired by animal searching behavior. *IEEE Transactions on Evolutionary Computations* 13:973–90. doi:10.1109/tevc.2009.2011992

Holland, J. H. 1975. *Adaptation in natural and artificial systems*. Ann Arbor, MI: University of Michigan Press.

Holubar, P., L. Zani, M. Hager, W. Fröschl, Z. Radak, and R. Braun. 2002. Advanced controlling of anaerobic digestion by means of hierarchical neural networks. *Water Research* 36:2582–88. doi:10.1016/s0043-1354(01)00487-0

Husseinzadeh, K. A. 2015. A new metaheuristic for optimization: Optics inspired optimization (OIO). *Computer and Operation Research* 55:99–125. doi:10.1016/j.cor.2014.10.011

Ince, H., and T. B. Trafalis. 2007. Kernel principal component analysis and support vector machines for stock price prediction. *IIE Transactions* 39:629–37. doi:10.1080/07408170600897486

Kahrizia, A., and H. Hashemib. 2014. Neuron curve as a tool for performance evaluation of MLP and RBF architecture in first break picking of seismic data. *Journal of Applied Geophysics* 108:159–66. doi:10.1016/j.jappgeo.2014.06.012

Kamruzzaman, J., R. Begg, and R. Sarker. 2006. *Artificial neural networks in finance and manufacturing*. London, UK: Idea Group Publishing.

Kashaninejad, M., A. A. Dehghani, and M. Kashiri. 2009. Modeling of wheat soaking using two artificial neural networks (MLP and RBF). *Journal of Food Engineering* 91:602–607. doi:10.1016/j.jfoodeng.2008.10.012

Kennedy, J., and R. C. Eberhard. 1995. Particle swarm optimization. In *Proceedings of IEEE International Conference on Neural Networks*, 1942–1948. Piscataway, NJ: IEEE.

Kilic, N., B. Ekici, and S. Hartomacioglu. 2015. Determination of penetration depth at high velocity impact using finite element method and artificial neural network tools. *Defence Technology* 11:110–22. doi:10.1016/j.dt.2014.12.001

Kim, K. J., and I. Han. 2000. Genetic algorithms approach to feature discretization in artificial neural networks for the prediction of stock price index. *Expert systems with Applications* 19:125–32. doi:10.1016/s0957-4174(00)00027-0

Kuo, R. J., C. Chen, and Y. Hwang. 2001. An intelligent stock trading decision support system through integration of genetic algorithm based fuzzy neural network and artificial neural network. *Fuzzy Sets and Systems* 118:21–45. doi:10.1016/s0165-0114(98)00399-6

Kuo, R. J., and Y. S. Han. 2011. A hybrid of genetic algorithm and particle swarm optimization for solving bi-level linear programming problem – A case study on supply chain model. *Applied Mathematical Modelling* 35:3905–17. doi:10.1016/j.apm.2011.02.008

Lightbody, G., P. O'Reilly, G. W. K. Irwin, and J. McCormick. 1997. Neural modelling of chemical plant using MLP and B-spline networks. *Control Engineering Practice* 5:1501–15. doi:10.1016/s0967-0661(97)10004-1

Lin, T., and C. Tseng. 2000. Optimum design for artificial neural networks: An example in a bicycle derailleur system. *Engineering Applications of Artificial Intelligence* 13:3–14. doi:10.1016/s0952-1976(99)00045-7

Liu, H., H. Q. Tian, C. Chen, and Y. F. Li. 2013. An experimental investigation of two wavelet-MLP hybrid frameworks for wind speed prediction using GA and PSO optimization. *International Journal of Electrical Power & Energy Systems* 52:161–73. doi:10.1016/j.ijepes.2013.03.034

Majhi, R., G. Panda, B. Majhi, and G. Sahoo. 2009. Efficient prediction of stock market indices using adaptive bacterial foraging optimization (ABFO) and BFO based techniques. *Expert Systems with Applications* 36:10097–104. doi:10.1016/j.eswa.2009.01.012

Mantegna, R. N. 1994. Fast, accurate algorithm for numerical simulation of levy stable stochastic processes. *Physical Review E* 49:4677–83. doi:10.1103/physreve.49.4677

Metghalchi, S. 2011. Stock price prediction on Tehran Stock Exchange by using a model combining artificial neural network and PSO. Master's thesis, Islamic Azad University of Tehran, Tehran, Iran.

Mirjalili, S., S. M. Mirjalili, and A. Lewis. 2014. Grey wolf optimizer. *Advances in Engineering Software* 69:46–61.

Mostafa, M. M. 2010. Forecasting stock exchange movements using neural networks: Empirical evidence from Kuwait. *Expert Systems with Applications* 37:6302–09. doi:10.1016/j.eswa.2010.02.091

Mostafa, M. M. 2014. Forecasting the Suez Canal traffic: A neural network analysis. *Maritime Policy & Management* 31:139–56. doi:10.1080/0308883032000174463

Nayak, S., B. Misra, and H. Behera. 2012. Index prediction with neuro-genetic hybrid network: A comparative analysis of performance. Paper presented at the International Conference on Computing, Communication and Applications, Tamil Nadu, India, February 22–24. doi:10.1109/ICCCA.2012.6179215

Nayak, S. C., B. B. Misra, and H. S. Behera. 2015. Artificial chemical reaction optimization of neural networks for efficient prediction of stock market indices. *Ain Shams Engineering Journal.* doi:10.1016/j.asej.2015.07.015

Ng, N. O. L., and C. K. Tham. 2000. Connection admission control of ATM network using integrated MLP and fuzzy controllers. *Computer Networks* 32:61–79. doi:10.1016/s1389-1286(99)00124-3

Nguyen, H. H., and C. W. Chan. 2004. Multiple neural networks for a long term time series-forecast. *Neural Computing & Applications* 13:90–98. doi:10.1007/s00521-003-0390-z

Park, J., F. Diehl, M. J. F. Gales, M. Tomalin, and P. C. Woodland. 2011. The efficient incorporation of MLP features into automatic speech recognition systems. *Computer Speech & Language* 25:519–34. doi:10.1016/j.csl.2010.07.005

Passino, K. M. 2002. Biomimicry of bacterial foraging for distributed optimization and control. *IEEE Control Systems Magazine* 22:52–67. doi:10.1109/mcs.2002.1004010

Pfeifera, J., K. Barker, J. E. Ramirez-Marquez, and N. Morshedloua. 2015. Quantifying the risk of project delays with a genetic algorithm. *International Journal of Production Economics* 170:34–44. doi:10.1016/j.ijpe.2015.09.007

Pham, D., A. Ghanbarzadeh, E. Koc, S. Otri, S. Rahim, and M. Zaidi. 2016. The bees algorithm – a novel tool for complex optimisation. Paper presented at Intelligent Production Machines and Systems-2nd PROMS Virtual International Conference, Oxford, UK, July 3–14.

Rao, R. M., and A. V. Naresh Babu. 2013. Optimal power flow using cuckoo search optimization algorithm. *International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering* 2:4213–18.

Rezaia, A., P. Keshavarzia, and R. Mahdiye. 2014. A novel MLP network implementation in CMOL technology. *Engineering Science and Technology* 17:165–72. doi:10.1016/j.jestch.2014.04.009

Rouhani, S., and A. Zare Ravasan. 2013. ERP success prediction: An artificial neural network approach. *Scientia Iranica Transactions E: Industrial Engineering* 20:992–1001. doi:10.1016/j.scient.2012.12.006

Saad, E. W., D. V. Prokhorov, and D. C. Wunsch. 1998. Comparative study of stock trend prediction using time delay, recurrent and probabilistic neural networks. *IEEE Transactions on Neural Networks* 9:1456–70. doi:10.1109/72.728395

Sarkar, A., and P. Pandey. 2015. River water quality modelling using artificial neural network technique. *AquaticProcedia* 4:1070–77.

Shrme, A. E. 2011. Hybrid intelligent technique for automatic communication signals recognition using bees algorithm and MLP neural networks based on the efficient features. *Expert Systems with Applications* 38:6000–06. doi:10.1016/j.eswa.2010.11.021

Simon, D. 2008. Biogeography-based optimization. *IEEE Transactions on Evolutionary Computation* 12:702–13. doi:10.1109/tevc.2008.919004

Srinivas, Y., A. Stanley, D. Raj, H. Oliver, D. Muthuraj, N. Chandrasekar, and F. Abdmalek. 2012. A robust behavior of feed forward back propagation algorithm of artificial neural networks in the application of vertical electrical sounding data inversion. *Geoscience Frontier* 3:729–36. doi:10.1016/j.gsf.2012.02.003

Takia, M., Y. Ajabshirchia, S. F. Ranjbarb, A. Rohanic, and M. Matloobid. 2016. Heat transfer and MLP neural network models to predict inside environment variables and energy lost in a semi-solar greenhouse. *Energy and Buildings* 110:314–29. doi:10.1016/j.enbuild.2015.11.010

Tofighi, S., S. A. Torab, and S. A. Mansouri. 2016. Humanitarian logistics network design under mixed uncertainty. *European Journal of Operation Research* 250:239–50. doi:10.1016/j.ejor.2015.08.059

Tsoukalas, V. D., and N. G. Fragiadakis. 2016. Prediction of occupational risk in the shipbuilding industry using multivariable linear regression and genetic algorithm analysis. *Safety Science* 83:12–22. doi:10.1016/j.ssci.2015.11.010

Valian, E., S. Mohanna, and S. Tavakoli. 2011. Improved cuckoo search algorithm for feedforward neural network training. *International Journal of Artificial Intelligence & Applications* 2:36–43.

Videnova, I., D. Nedialkov, M. Dimitrova, and S. Popova. 2006. Neural networks for air pollution now casting. *Applied Artificial Intelligence* 20:493–506.

Walczak, B., and D. L. Massart. 1996. The radial basis functions-partial least squares approach as a flexible non-linear regression technique. *Analytica Chimica Acta* 331:177–85. doi:10.1016/0003-2670(96)00202-4

Wang, Y. F. 2002. Predicting stock price using fuzzy grey prediction system. *Expert Systems with Applications* 22:33–38. doi:10.1016/s0957-4174(01)00047-1

Yamashita, T., K. Hirasawa, and K. Hu. 2005. Application of multi-branch neural networks to stock market prediction. In *Proceedings of IEEE International Joint Conference on Neural Networks*, 2544–48. doi:10.1109/IJCNN.2005.1556303

Yang, X. S., and S. Deb. 2009. Cuckoo search via levy flights. In *Proceedings of World Congress on Nature & Biologically Inspired Computing*, 210–225.

Yasin, Y., F. B. H. Ahmad, M. Ghaffari-Moghaddam, and M. Khaje. 2014. Application of a hybrid artificial neural network–genetic algorithm approach to optimize the lead ions removal from a queous solutions using intercalatedtartrate-Mg–Allayereddoublehydroxides. *Environmental Nanotechnology, Monitoring & Management* 1–2:2–7. doi:10.1016/j.enmm.2014.03.001

Yilmaz, I., and O. Kaynar. 2011. Multiple regression, ANN (RBF, MLP) and ANFIS models for prediction of swell potential of clayey soils. *Expert Systems with Applications* 38:5958–66. doi:10.1016/j.eswa.2010.11.027

Yu, L., S. Wang, and K. K. Lai. 2009. A neural-network-based nonlinear metamodeling approach to financial time series forecasting. *Applied Soft Computing* 9:563–74. doi:10.1016/j.asoc.2008.08.001

Yu, L., and Y. Q. Zhang. 2005. Evolutionary fuzzy neural networks for hybrid financial prediction. *IEEE Transactions on Systems, Man and Cybernetics, Part C* 35:244–49. doi:10.1109/tsmcc.2004.841902

Yuan, C., S. Liu, and Z. Fang. 2016. Comparison of China's primary energy consumption forecasting by using ARIMA (the autoregressive integrated moving average) model and GM (1,1) model. *Energy* 100:384–90. doi:10.1016/j.energy.2016.02.001

Zahedi, J., and M. M. Rounaghi. 2015. Application of artificial neural network models and principal component analysis method in predicting stock prices on Tehran stock exchange. *Physica A:Statistical Mechanics and its Applications* 438:178–87. doi:10.1016/j.physa.2015.06.033

Zhang, Y., and L. Wu. 2009. Stock market prediction of S&P 500 via combination of improved BCO approach and BP neural network. *Expert systems with applications* 36:8849–54. doi:10.1016/j.eswa.2008.11.028

Zhao, R. Q., and W. S. Tang. 2008. Monkey algorithm for global numerical optimization. *Journal of Uncertain Systems* 2:165–76.