Expert Systems With Applications 44 (2016) 320–331

Contents lists available at ScienceDirect

Expert Systems With Applications

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

Integrating metaheuristics and Artificial Neural Networks for improved stock price prediction

Mustafa Göçken[a,*], Mehmet Özçalıcı[b], Aslı Boru[a], Ayşe Tuğba Dosdoğru[c]

[a] Adana Science and Technology University, Industrial Engineering Department, Yeşiloba Yerleşkesi, 01180, Adana, Turkey
[b] Kilis 7 Aralık University, Business and Administration Department, Kilis, Turkey
[c] Gaziantep University, Industrial Engineering Department, Gaziantep, Turkey

ARTICLE INFO

Keywords:
Artificial Neural Network
Genetic Algorithm
Harmony Search Algorithm
Stock market price

ABSTRACT

Stock market price is one of the most important indicators of a country's economic growth. That's why determining the exact movements of stock market price is considerably regarded. However, complex and uncertain behaviors of stock market make exact determination impossible and hence strong forecasting models are deeply desirable for investors' financial decision making process. This study aims at evaluating the effectiveness of using technical indicators, such as simple moving average of close price, momentum close price, etc. in Turkish stock market. To capture the relationship between the technical indicators and the stock market for the period under investigation, hybrid Artificial Neural Network (ANN) models, which consist in exploiting capabilities of Harmony Search (HS) and Genetic Algorithm (GA), are used for selecting the most relevant technical indicators. In addition, this study simultaneously searches the most appropriate number of hidden neurons in hidden layer and in this respect; proposed models mitigate well-known problem of overfitting/underfitting of ANN. The comparison for each proposed model is done in four viewpoints: loss functions, return from investment analysis, buy and hold analysis, and graphical analysis. According to the statistical and financial performance of these models, HS based ANN model is found as a dominant model for stock market forecasting.

1. Introduction

A stock market is a public market to trade the company's stocks and derivative at an approved stock price (Preethi & Santhi, 2012). Stock market provides opportunities for brokers and companies to make investments on neutral ground and is one of the primary indicators of a economic condition of the country (Perwej & Perwej, 2012). However, stock market is characterized by nonlinearities, discontinuities, and high-frequency multi-polynomial components because it is interacted with many factors such as political events, general economic conditions, and traders' expectations (Hadavandi, Shavandi, & Ghanbari, 2010). Also, the fast data processing of these events with the help of improved technology and communication systems has caused the stock prices to fluctuate very fast. Thus many banks, financial institutions, large scale investors and stock brokers have to buy and sell stocks within the shortest possible time and time span of even a few hours between buying and selling is not unusual (Bonde & Khaled, 2012). Robust and agile stock market is also highly desirable in the field of finance, engineering and mathematics due to high return possibility. It is generally seen as a peak investment outlet. For these purposes, many researchers have been investigated the predictability of the stock market by using of fundamental analysis, technical analysis, time series prediction, and machine learning methods (Prasanna & Ezhilmaran, 2013). Besides, most of the companies have created new methods for evaluating financial data and investment decisions (Sureshkumar & Elango, 2012). Among them, ANN approach has been thought as the best forecasting method with a high level of validity in the fields of stock market forecasting. However, some critical points of ANN structure should be carefully analyzed. The definition what constitutes an optimal set of ANN input variables can be considered one of the main problems in ANN structure because the choice of input variables directly affects the forecasting accuracy. Secondly, number of neurons (or units, nodes) in hidden layer is also so important for ANN. It is an adjustable part in ANN but unfortunately, there is no unique method for fixing the optimum number of neurons in hidden layer for a particular problem. Therefore, researchers prefer generally to use trial and error method for this purpose. In this paper, we proposed hybrid methodology for determining input variable and the number of neurons in hidden

* Corresponding author. Tel.: +90 322 455 0000–2226.
E-mail addresses: mgocken@adanabtu.edu.tr (M. Göçken), mozcalici@kilis.edu.tr (M. Özçalıcı), aboru@adanabtu.edu.tr (A. Boru), dosdogru@gantep.edu.tr (A.T. Dosdoğru).

http://dx.doi.org/10.1016/j.eswa.2015.09.029
0957-4174/

layer. GA and HS are used as a tool for improving ANN's forecasting performance. In literature, GA is often used with ANN for the purpose of training the network, feature subset selection, and architecture optimization. However, HS is generally not used with ANN for these purposes. Therefore, our study is created alternative solution methods for stock market forecasting with better solutions. The contribution is structured as follows. Section 2 describes the related works. Then, we start describing solution methodology in Section 3. Section 4 deals with results and discussions. Finally, Section 5 is devoted to conclusions.

2. Literature

The importance of Turkish stock market has increased substantially with the establishment of the Istanbul Stock Exchange in 1986. Since its establishment in 1986, the ISE has followed a fast pace growth in terms of trading volume, market capitalization, number of listed corporations and foreign investment (Adaoglu, 2000). Also, ISE characterized with high volatility in the market and such volatility attracts many local and foreign investors as it provides high return possibility (Cinko & Avci, 2009). Hence, forecasting stock market movement has been the objective of the vast research papers applying different techniques. Among them, ANN is featured as being data driven and, hence, does not require assumptions concerning data. With such a feature ANN is a suitable technique in handling nonlinear, highly complex and dynamic data of stock markets (Karymshakov & Abdykaparov, 2012). In the literature, ANN is clearly explained by Egeli, Ozturan, and Badur (2003). The authors used six different ANNs which includes multi-layer perceptron (MLP) and generalized feed forward to predict ISE market index value. Authors used previous day's index value, previous day's TL/USD exchange rate, previous day's overnight interest rate and 5 dummy variables each representing the working days of the week as inputs. The results showed that for each ANN model, the highest accuracies were obtained with 1 hidden layer and also ANN models give more accurate results than the ones based on moving averages. Guresen, Kayakutlu, and Daim (2011) compared ANN models including MLP, dynamic ANN, and the hybrid neural networks. It is observed that classical ANN model MLP gives more reliable results than the other models used in this comparison. Kara, Boyacioglu, and Baykan (2011) revealed that ANN works better than Support Vector Machine in predicting the direction of stock price movement in the ISE. In the study, parameters of ANN models such as number of neurons in the hidden layer were determined empirically. Also, ten technical indicators were selected as feature subsets by the review of domain experts and prior researches. Şenol and Özturan (2008) statistically demonstrated that ANN outperforms Logistic Regression methodology. In the study, ANN was used to predict the stock price behavior in terms of its direction. The best results were obtained for ANN model with three inputs, 11 hidden neurons in the single hidden layer and one output with three indicators, relative strength index of 14 days, stochastic indicator for 14 days, and stochastic moving average. Yildiz, Yalama, and Coskun (2008) utilized ANN for forecasting the direction of the ISE National-100 using the highest and lowest prices paid during the day, the closing price, the exchange rate (as the US dollar), and response rates as an input variables. The results of the previous studies show that accuracy of stock market prediction is generally between 60% and 76%, and hence more robust ANN model is needed to increase prediction accuracy in Turkish stock market.

Having highly functional stock markets and exchanges is incredibly valuable all over the world is a well-known fact. Therefore, so many types of ANN models are developed to search out more efficient forecasting model. Chiu and Chuang (2003) showed that ANN has ability for predicting tendency of Taiwan stock market. Five different ANN models were developed to decide the number of input neurons and hidden neuron. Also, the classification technique and clustering method were used under framework of ANN with quantitative and qualitative factors. Similarly, Aldin, Dehnavi, and Entezari (2012) used ANN for stock price index forecasting on the Taiwan Stock Exchange. Closing price, the high and low price index were converted into technical indicators for predicting the position of stock price movements. In the study, neuron numbers in the hidden layer was determined empirically. Dastgir and Enghiad (2012) evaluated Iran Stock Market by focusing on forecasting Tehran Stock Exchange Price Index which is the most significant index of Iran Stock Market. In the study, two hidden layers were used with many combinations of architecture. The number of neurons in each hidden layer was changed from one to sixteen. Results of the study revealed that ANN model with three hidden neurons on the first hidden layer and four hidden neurons on the second achieved the best performance in Iran Stock Market. Ruxanda and Badea (2014) presented different configured ANNs and compared them in terms of forecasting errors while making predictions on Bucharest Stock Market Index. Input variables were set based on a stepwise forward regression. Adebiyi, Adewumi, and Ayo (2014) found that 10 inputs obtained from the New York Stock Exchange including open price, low price, high price, close price, and volume traded, 17 hidden neurons, and one output neuron give more accurate results in ANN model. Laboissiere, Fernandes, and Lage (2015) used ANN to predict the maximum and minimum day stock prices of Brazilian power distribution companies. In the study, correlation analysis was used to select input variable and different ANN architectures were tested empirically. The best results were found with one hidden layer and only five hidden neurons. Zahedi and Rounaghi (2015) applied ANN and principal component analysis to predict stock price on Tehran Stock Exchange. The results of the study show that ANN model has superiority over its rivals. Also, principal component analysis method can accurately predict stock price on Tehran Stock Exchange using 20 accounting variables.

In this paper we review studies in the ANN literature which have been used for stock market forecasting, results revealed that a different combination of attribute sets was experimented with different ANN model parameter values and each study provides satisfying result in existing condition but ANN architecture is very important which directly affects system performance essentially. Hence, most previous studies were focused on the improvement of the ANN architecture. However, there are few studies on the input variable selection from predetermined data set and there is no clear methodology available for variable selection and determining number of hidden neurons in hidden layer. Therefore, the basic idea that lies behind the proposed models is not only selecting the most relevant input variables that are to be used by ANN models but also setting the number of neurons in hidden layer by manipulating ANN structure via metaheuristics. Thus, proposed models based on GA and HS are applied to improve forecasting accuracy and stability of ANN.

3. Solution methodology

3.1. Technical indicators

This section describes input variable selection methodology. For each case, 45 technical indicators are considered as input variables. Technical indicators are effective tools to characterize the real market situation. Using technical indicators can be more informative than using pure prices (Nikfarjam, Emadzadeh, & Muthaiyah, 2010) and it is very practical way for stock analysts and fund managers to analyze stock market. On the other hand, this technique may not be a good alternative solution for common investors because too many technical indicators are available to be considered as prediction factors and the most commonly used technical indicators are ordinarily not understandable. Therefore, selection of the useful technical indicators accurately is the key issue to make a profit for those stock market investors (Wei & Cheng, 2012). However, no method is successful enough to consistently beat the market. Every stock index or stock has unique characteristics. That means, say feature A might play an important role in predicting future prices of stock X while feature B might be regarded as redundant for that stock. For that reason it seems that it is not possible to say that "feature A is a good predictor for every stock". Different features must be used for prediction attempt in different time periods and/or different stocks. In our study, technical indicators are applied as the input variables of ANN to forecast the stock market index. GA and HS are integrated with ANN not only for optimizing the architecture of ANN but also for determining the indicators that has the most significant effect on the forecasting performance. The underlying logic for using GA and HS for variable selection is to evaluate the usefulness of indicators and eliminate irrelevant ones to simplify the proposed model. It should be noted that there is no limit for the number of indicators to be considered by GA and HS. In Table 1, all of the technical indicators considered in this study together with the final indicator selection results of GA and HS algorithms are illustrated. Note that, shaded variables are selected by none of the optimization methods.

3.2. *Artificial Neural Network*

ANN is a computational network which attempt to simulate, in a gross manner, the networks of nerve cell (neurons) of the biological (human or animal) central nervous system (Graupe, 2007). The information processing and physical structure of the brain is partially emulated with a web of neural connections (Li, 1994) which has great capacity in modeling nonlinear systems. Also, ANN is known with good generalization capabilities and is substantially robust against noisy or missing data (Versace, Bhatt, Hinds, & Shiffer, 2004). On the other hand, it is difficult to design ANN model for a particular forecasting problem. Modeling issues should be considered carefully. Determining the appropriate architecture such as number of the input variables, hidden layers and hidden neurons in each layer can be considered as a critical factor (Vaisla & Bhatt, 2010). For example, number of hidden layers and neurons in each hidden layer is proportional to the ability of the network to approximate more complicated functions. However, this does not infer that complicated structures of networks will always perform better (Perwej & Perwej, 2012). If the network has too many hidden neurons, it will follow the noise in the data due to over parameterization leading to poor generalization for untrained data (Subasi & Erçelebi, 2005). On the other hand, network with too few hidden neurons would be incapable of differentiating between complex patterns leading to only a linear estimate of the actual trend (Kuruş, Kılıç, & Uçan, 2013). To eliminate these hesitations, three different forecasting models are proposed in this study and their performances are compared. The parameters of each proposed model are given in Table 2.

From Table 2, it is apparently seen that as their name imply, the parameters of the first two proposed models are set by using HS and GA, respectively. Note that the third model does not employ any optimization method. Thus, third model directly uses all considered features for training the ANN. It should be noted that 10 neurons in the hidden layer are selected arbitrarily for the third model. The general architecture of proposed models is demonstrated in Fig. 1.

In Fig. 1, p is the input pattern, b_1 is the vector of bias weights on the hidden neurons, and W_1 is the weight matrix between 0th (i.e. input) layer and 1st (i.e. hidden) layer. a_1 is the vector containing the outputs from the hidden neurons, and n_1 is the vector containing net-inputs going into the hidden neurons. a_2 is the column-vector coming from the second output layer, and n_2 is the column-vector containing the net inputs going into the output layer. W_2 is the synaptic weight matrix between the 1st (i.e. hidden) layer and the 2nd (i.e. output) layer and b_2 is the column-vector containing the bias inputs of the output neurons. Each row of W_2 matrix contains the synaptic weights for the corresponding output neuron (Ahmed, Jafri, Ahmad, & Khan, 2007). Firstly, the neuron receives information from the environment and then this information multiplied by the corresponding weights is added together and used as a parameter within an activation (transfer) function (Haider & Hanif, 2009). The transfer functions are used to prevent outputs from reaching very large values that can 'paralyze' ANN structure (Duch & Jankowski, 1999). For hidden layer, suitable transfer function is

Table 1
Initial feature pool and final result of selection status.

	Technical indicators	HS Is selected? (0: No) (1: Yes)	GA Is selected? (0: No) (1: Yes)
1	Today's close−previous close price	0	0
2	Previous close price	1	1
3	Previous highest price	1	1
4	Previous lowest price	1	1
5	Previous open price	0	0
6	5 day simple moving average of close price	0	0
7	6 day simple moving average of close price	0	1
8	10 day simple moving average of close price	0	0
9	20 day simple moving average of close price	0	0
10	5 day exponential moving average of close price	0	0
11	6 day exponential moving average of close price	1	0
12	10 day exponential moving average of close price	1	1
13	20 day exponential moving average of close price	1	1
14	5 day triangular moving average of close price	1	1
15	6 day triangular moving average of close price	0	1
16	10 day triangular moving average of close price	1	1
17	20 day triangular moving average of close price	1	1
18	Accumulation/distribution oscillator	0	1
19	Close price moving average convergence/divergence	0	1
20	9-period exponential moving average of MACD	0	1
21	Acceleration opening price	0	1
22	Acceleration highest price	1	0
23	Acceleration lowest price	0	1
24	Acceleration close price	1	1
25	Momentum open price	1	1
26	Momentum highest price	1	1
27	Momentum lowest price	0	0
28	Momentum close price	0	1
29	Chaikin volatility	0	1
30	Fast stochastic %*K*	0	0
31	Fast stochastic %*D*	1	1
32	Slow stochastic %*K*	0	0
33	Slow stochastic %*D*	1	0
34	William's %*R*	0	0
35	Relative strength index	1	1
36	Bollinger middle band	1	1
37	Bollinger higher band	1	1
38	Bollinger lower band	1	0
39	Highest high	1	0
40	Lowest low	1	1
41	Median price	1	1
42	Price rate of change	1	0
43	Typical price	0	0
44	Weighted close	0	0
45	William's accumulation/distribution	0	0

Table 2
Parameters of the models.

Parameters	HS-ANN model	GA-ANN model	Regular ANN model
Size of the input layer	Determined by HS = 23	Determined by GA = 26	45
Number of neurons in hidden layer	Determined by HS = 17	Determined by GA = 2	10
Transfer function in hidden layer	Sigmoid tangent	Sigmoid tangent	Sigmoid tangent
Transfer function in output layer	Pure-linear transfer function	Pure-linear transfer function	Pure-linear transfer function
Training function	Levenberg–Marquardt training algorithm	Levenberg–Marquardt training algorithm	Levenberg–Marquardt training algorithm

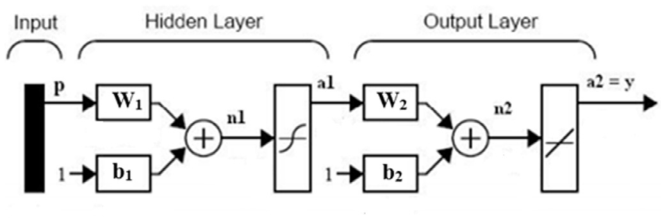

Fig. 1. Architecture of the proposed neural network (Ahmed et al., 2007).

particularly needed to introduce non-linearity into the network because it gives the power to capture nonlinear relationship between input and output (Ravichandran, Thirunavukarasu, Nallaswamy, & Babu, 2005). In this part, tangent sigmoid transfer function is applied in hidden layer. However, the use of sigmoid units at the outputs can limit the range of possible outputs to the range attainable by the sigmoid, and this would be undesirable in some cases (Bishop, 1995). Hereby, a pure linear function is selected in output layer. The pure linear transfer function calculates the neuron's output by simply returning the value passed to it. After ANN model is constructed, training of ANN is the next important step of the forecasting model. Training of ANN is an iterative process like weights and bias of the network. In this paper, proposed ANN-based forecasting models are trained by Levenberg–Marquardt (LM) algorithm with optimum network parameters. LM is a trust region based method with hyper-spherical trust region (Burney, Jilani, & Ardil, 2005) and is used as an intermediate optimization algorithm between the Gauss–Newton (GN) method and gradient descent algorithm. Also, LM addresses the limitations of each of those techniques (Kermani, Schiffman, & Nagle, 2005). When the current solution is far from a local minimum, the algorithm behaves like a gradient descent method: slow, but guaranteed to converge. When the current solution is close to a local minimum, it becomes GN method and exhibits fast convergence (Lourakis & Argyros, 2005). However, it is important to note that LM is very efficient when training networks which have up to a few hundred weights (Hagan & Menhaj, 1994).

3.3. GA-ANN forecasting model

When building an ANN, a number of parameters should be considered and unlimited ways are available to construct ANN. In the literature, particularly, input variable selection remains an important part of ANN model development, due to the negative impact that poor selection can have on the performance of ANNs during training and deployment post-development (Mańdziuk & Jaruszewicz, 2011). In this study, we used GA to overcome drawbacks of the input variable selection. GA is a general adaptive optimization search methodology based on a direct analogy to Darwinian natural selection and genetics in biological systems (Huang & Wang, 2006). GA ensures the development of new and better populations among different species during evolution. Although most standard meta-heuristic algorithms used only information from a single individual, GA uses information of a population of individuals (solutions) when they conduct their search for better solutions (Pardalos, Pitsoulis, Mavridou, & Resende, 1995). Additionally, it is important to note that GA has proved its success in search and optimization problems. Its ability to exploit the information accumulated about an initially unknown search space in order to bias subsequent searches into useful subspaces can be given as the main reason for the success. This is the key feature, particularly in large, complex, and poorly understood search spaces, where classical search tools (enumerative, heuristic, etc.) are inappropriate, offering a valid approach to problems requiring efficient and effective search techniques (Martínez & Lozano, 2008).

Fig. 2 depicts a synthetic scheme of the GA based selection of input variables. The advantage of the GA-ANN model lies in synergy between the GA, which is used for the selection of the variables to be used, and ANN, exploiting the selected variables. In the same manner, GA is used to determine number of neurons in hidden layer because inadequate neurons can restrict the relationship or too many neurons can cause overtraining. Certainly, getting the correct balance between numbers of neurons directly affect forecasting accuracy in models.

Basically, GA-ANN model is shown in Fig. 3 and can be summarized as follows. It divides dataset as training and testing dataset. Furthermore, training data set is also divided into the subsets to give the ANN generalization ability. For this purpose we evaluated the candidate solutions on different subsets and obtained a mean of mean squared error (MSE). The minimization of this error is performed by GA. Calculating MSE continues until stopping criteria is satisfied. Also, GA has several genetic operators that can be modified to improve the performance of particular implementations, namely representation, selection, crossover, and mutation. These procedures are given in the next sections.

3.3.1. Chromosome representation

Chromosome representation is the first and the most important operator obtained by encoding of a chromosome to represent a solution. In literature, binary encoding is the most commonly used in GA and gives many possible chromosomes even with small number of alleles (Rajasekaran & Pai, 2003). Similarly, binary encoding is used in this study and chromosome is considered to be composed of two parts: (1) variable selection and (2) determination of the number of hidden neurons (Fig. 4). If a variable is selected, gene is coded as 1, otherwise 0. Similarly, if a node is selected, gene is coded as 1, otherwise 0. Total length of the chromosome is determined as the sum of the total number of variables considered and the total number of neurons considered.

GA starts with a randomly generated initial population. Initial population consists of a number of chromosomes that represent the number of variables and the number of hidden neurons. After all fitness values for the whole initial population are obtained, the chromosomes evolve through successive iterations called generations. To enhance diversity of the generation and to generate the population of the next generation, GA operators such as selection, crossover and mutation are activated.

3.3.2. Selection operator

Individuals, called parents, are selected based on a selection rule to generate new, better solutions for next generations. In this study, stochastic uniform selection which lays out a line is used. This method chooses an individual according to its scaled fitness value.

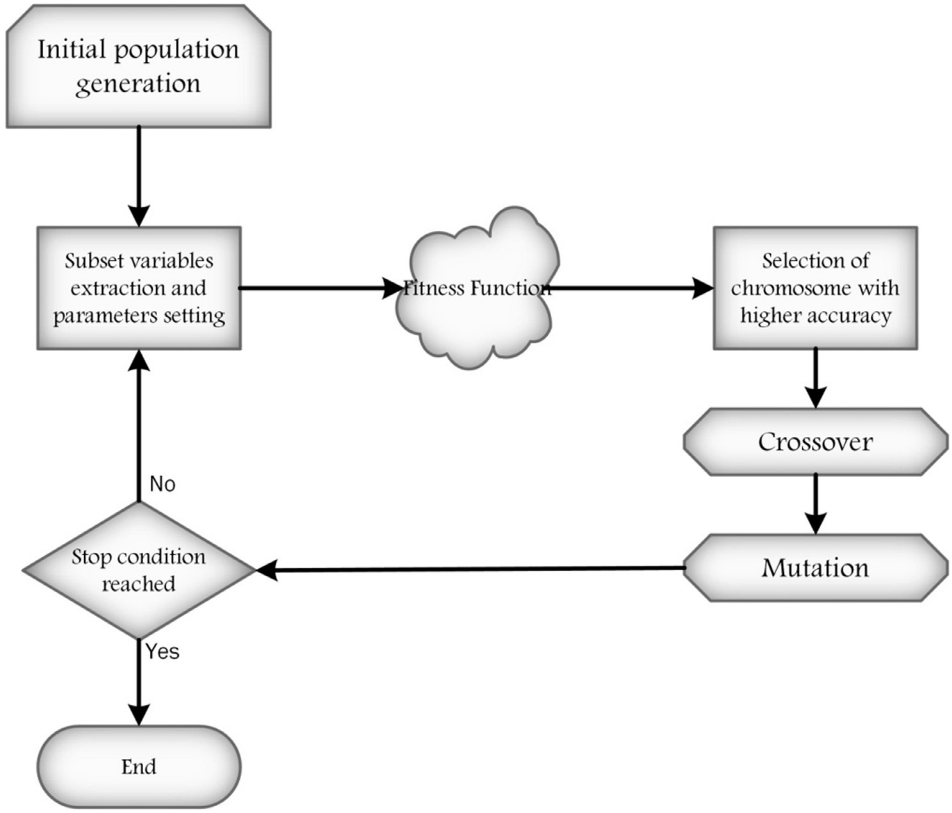

Fig. 2. Flow-chart representing the GAs based variable selection system (Cateni, Colla, & Vannucci, 2011).

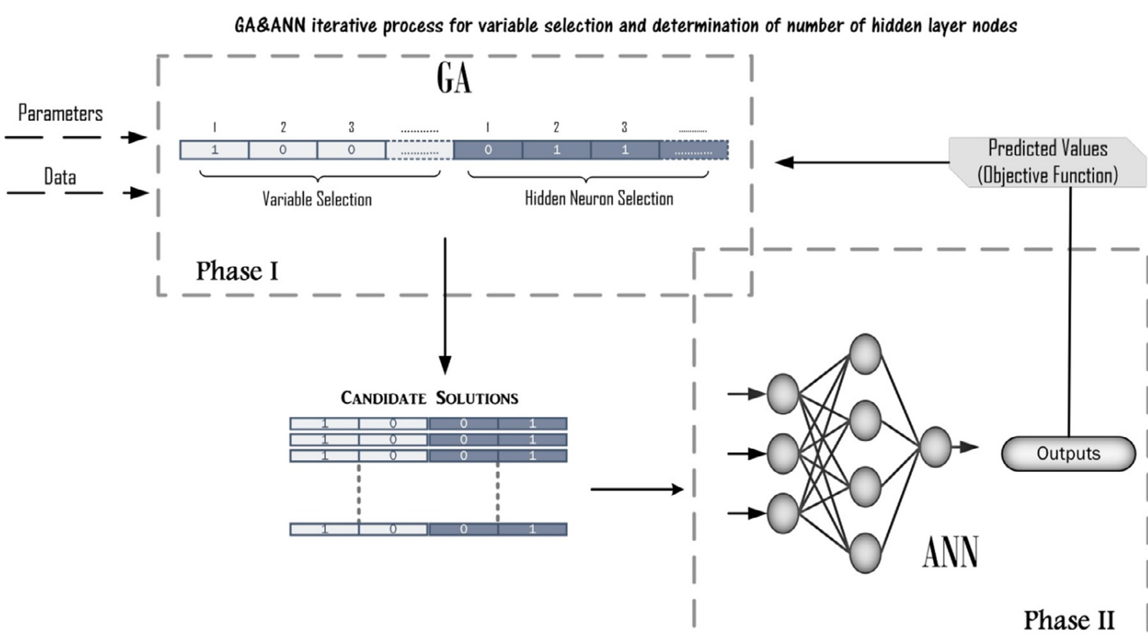

Fig. 3. GA-ANN iterative process for variable selection and determination of number of hidden layer neurons.

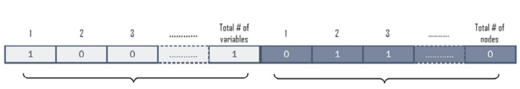

Fig. 4. Chromosome representation.

3.3.3. Crossover and mutation operators

In crossover operator, two chromosomes are randomly selected and their chromosome strings are randomly cut to produce new chromosomes. In this respect, a pair of parents is firstly randomly selected from the mating pool. Secondly, a point, called crossover site, along their common length is randomly selected, and the information after the crossover site of the two parent strings are swapped, thus creating two new children (Otman & Jaafar, 2011). An illustrative crossover operator utilized in this study, is shown in Fig. 5.

Then, mutation operator is applied to provide a small amount of random search. Without mutation, offspring chromosomes would be limited to only the genes available within the initial population.

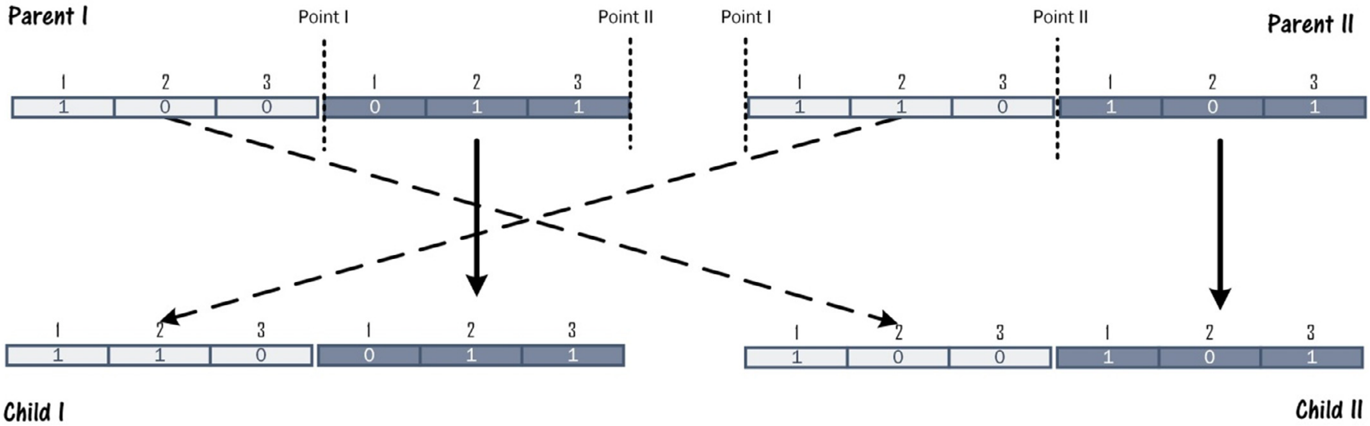

Fig. 5. Illustrative example of crossover operator.

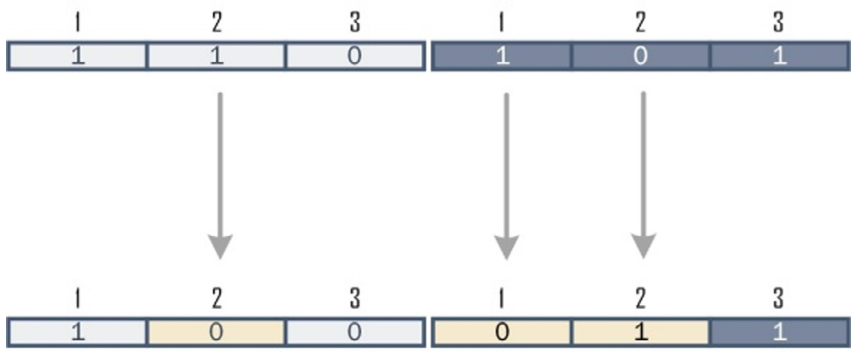

Fig. 6. Illustrative example of mutation operator.

Table 3
Parameters of the GA.

Elite count	2
Crossover fraction	0.8
# of generations	100
Population size	20

Mutation should be able to introduce new genetic material as well as modify existing one (Fig. 6). With these new gene values, the GA may be able to arrive at a better solution than was previously possible (Kougias & Theodosiou, 2010).

After mutation operator, the candidate solutions obtained by GA proceeds to the phase II (fitness function prediction). Best two individuals are saved for the next generations. This iterative process is repeated over many generations. The run of GA terminates when the termination criterion is satisfied. The best individual ever encountered during the run is typically designated as the result of the run. The parameters of the GA are given in Table 3.

3.4. HS-ANN forecasting model

HS is based on the improvization process of musicians in a band. In HS algorithm, multiple harmonics groups can be used in parallel. Proper parallelism usually leads to better implantation with higher efficiency (Geem, 2006). The good combination of parallelism with elitism as well as a fine balance of intensification and diversification is the key to the success of the HS algorithm, and in fact, to the success of any meta-heuristic algorithms (Yang, 2009). HS is simple in concept, few in parameters, easy in implementation, imposes fewer mathematical requirements. Therefore, HS has been successfully applied as an optimization method in many scientific and engineering fields and was reported to be competitive alternative to many rivals (Mahdavia, Fesanghary, & Damangir, 2007). In this paper, we proposed HS-ANN model for determining the most relevant input variables and the number of neurons in hidden layer. The first step in HS-ANN model is to divide dataset as training and testing dataset. Furthermore, training data set is also divided into the subsets to give the ANN generalization ability. For this purpose we evaluated the candidate solutions on 5 different subsets and obtained a mean of MSE. The minimization of this error is performed by HS. Details of the proposed HS-ANN model are shown in Fig. 7.

To apply HS, the problem should be formulated in the optimization environment, having an objective function and constraints as Eq. (1) (Mahdavia et al., 2007; Yadav, Kumar, Panda, & Chang, 2012):

$$\text{Minimize (or Maximize) } f(x)$$
$$\text{Subject to } x_i \in X_i \; i = 1, 2, 3, \ldots, N \tag{1}$$

where $f(x)$ is the objective function with x as the solution vector composed of decision variable x_i, and X_i is the set of a possible range of values for each decision variable x_i ($Lx_i \le X_i \le Ux_i$), where Lx_i and Ux_i are the lower and upper bounds for each decision variable, respectively. In addition, the values of different parameters of the HS algorithm also have to be specified. These parameters include harmony memory size (HMS), harmony memory considering rate (HMCR), pitch-adjusting rate (PAR).

3.4.1. Initialize the harmony memory (HM)

The initial HM consists of an HMS number of randomly generated solution vectors. Each component of the solution vector in HM is initialized using the uniformly distributed random number between the lower and upper bounds of the corresponding decision variable $[Lx_i, Ux_i]$, where $1 \le i \le N$. The ith component of the jth solution

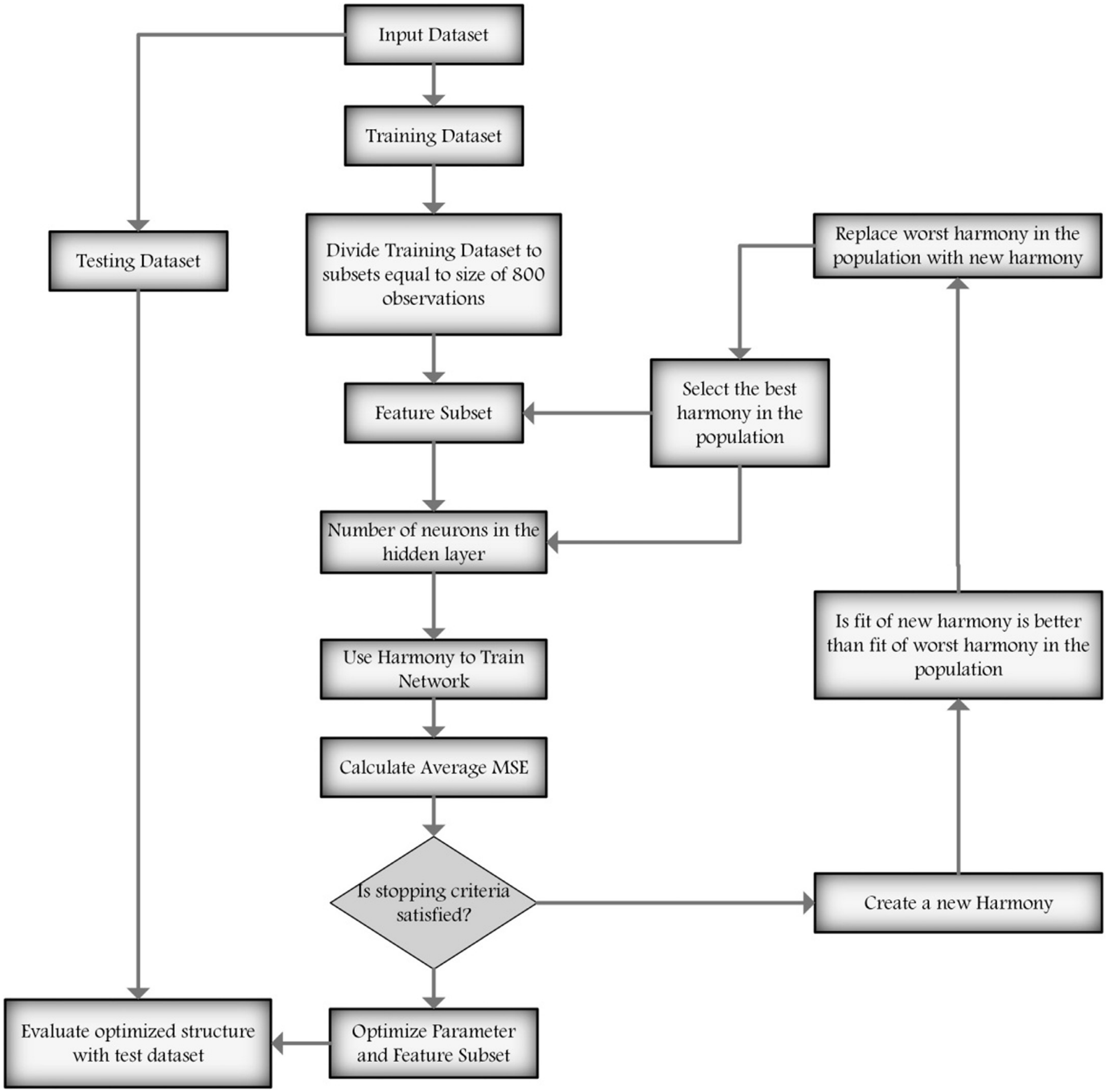

Fig. 7. Flow-chart representing the ANNs based variable selection system.

vector is given by Eq. (2)

$$x_i^j = Lx_i + (Ux_i - Lx_i).\mathrm{rand}[0,1] \tag{2}$$

where $j = 1,2,3,\ldots,$ HMS and rand[0, 1] is a uniformly distributed random number between 0 and 1. Each row consists of a randomly generated solution vector, and the objective function value for the jth solution vector is denoted by $f(x^j)$. The matrix formed is governed by Eq. (3).

$$\begin{aligned} HM(j, 1:N) &= x^j \\ HM(j, N+1) &= f\left(x^j\right) \end{aligned} \tag{3}$$

The HM with the size of $HMS \times (N+1)$ can be represented by a matrix, as

$$HM = \begin{pmatrix} x_1^1 & x_2^1 & x_3^1 & \cdots & x_N^1 & f(x^1) \\ & \vdots & & \ddots & & \vdots \\ x_1^{HMS} & x_2^{HMS} & x_3^{HMS} & \cdots & x_N^{HMS} & f(x^{HMS}) \end{pmatrix}$$

In this study, each row of the matrix of HM coincides with a solution. First value of each row gives the number of hidden layer neurons. The rest gives the information of whether the considered variable is selected or not. The last value of the row gives the objective function of the related row (see Fig. 8).

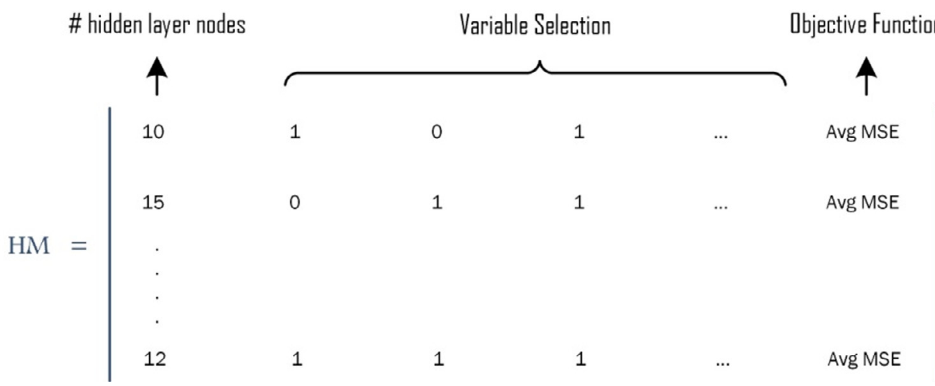

Fig. 8. Representation of the HM matrix.

Table 4
Parameters of the HS.

HMS	100
bw	0.2
HMCR	0.95
PAR	0.3
Max iteration	10,000

Table 5
Descriptive statistics of testing and training dataset.

	Train dataset	Test dataset
Mean	50082.53	74373.31
Standard deviation	14858.80	4494.85
Length (Days)	4000	160

3.4.2. *Improvise a new harmony from the HM*

Then, new harmony is improvised which is the essence of the HS algorithm. In improvization, the HS generates a new harmony vector, $x' = x'_1,\ x'_2, \ldots .x'_N$, using the following rules: memory consideration, pitch adjustment, and random selection. The original HS algorithm consists of three operations for considering the computational intelligence or randomness as Eq. (4):

$$x'_i \in \begin{cases} x'_i \in x_i^1, x_i^2, \ldots, x_i^{HMS}, & (HMCR) \\ x'_i \in X_i, & (1 - HMCR) \end{cases} \tag{4}$$

In this step, a random number is generated. If this value is less than HMCR, value of 1 is chosen, else value of 0 is chosen. After the memory consideration, each decision variable is evaluated to determine whether pitch adjustment is necessary or not. This evaluation is carried out with PAR parameter which is the probability of pitch adjusting and identified as Eq. (5):

$$x'_i = \begin{cases} x'_i \pm \text{rand}(0,1) \times bw \ \text{with probability}\ PAR \\ x'_i \ \text{with probability}\ (1 - PAR) \end{cases} \tag{5}$$

where *bw* is the range of X_i, rand(0,1) is random number between 0–1. If the values are 1 in both of PAR and HMCR, the value is chosen for the new harmony. Other values are selected as 0. In this step, the value of x'_i is chosen randomly. The value of x'_i is in the range of X_i and it has probability of HMCR. Details are given in Fig. 9.

3.4.3. *Generation of new HM*

After selecting the new values, the objective function value is calculated for new harmony vector. If this value is better than the worst harmony vector in the harmony matrix, it is then included in the matrix, while the worst one is taken out of the matrix. Then, harmony memory matrix is sorted in descending order by the objective function value. These are repeated until the termination criterion which is the pre-selected maximum number of cycles is satisfied. Parameters of the HS are given in the Table 4.

4. Results and discussions

The main purpose of this study is to propose new hybrid stock price forecasting models to get more accurate and reliable forecasting. In the first section, variables are determined for ANN model. In regular ANN model, we used all the variables considered. However, in GA-ANN and HS-ANN models; we reduce the input variable set to an optimal subset. Among 45 relevant input variables, GA-ANN selected 26 variables as the optimal input variable subset while HS-ANN selected 23 input variables as the optimal variable subset. Similarly, the optimal number of neurons in hidden layer is specified. It should be noted that for both GA-ANN and HS-ANN forecasting model, the number of hidden layer is considered to be 1. In regular model, 10 neurons in hidden layer are predetermined arbitrarily. However, both GA-ANN and HS-ANN models characterized their own number of neurons in the hidden layer. While GA is selected only 2, HS has 17 neurons in hidden layer.

To construct ANN model; suitable training and testing samples should also be selected. The first issue here is to split the data into two separate sets, the training and testing data sets. Although there is no general solution to this problem, several factors such as the problem characteristic, the data type, and the size of the available data should be considered in making this decision. In this study, price information of BIST100 index between 08/06/2005 and 27/05/2013 (4000 observations) used as training dataset, and 28/05/2013 and 20/09/2013 (160 observations) used as testing the performance of forecasting models. Details of training and testing data set are given in Table 5.

Nine different loss functions namely mean absolute error (MAE), root mean square error (RMSE), mean absolute relative error (MARE), mean squared relative error (MSRE), root mean squared relative error (RMSRE), mean absolute percent error (MAPE), mean squared percentage error (MSPE), and root mean squared percentage error (RMSPE) are used to evaluate the performance of training and testing data sets. Resulting values of these loss functions are summarized in Table 6. Actual prices and predicted prices are compared for each forecasting models.

According to the results, HS-ANN model outperformed other forecasting models in terms of all statistical loss functions. It should be noted that regular ANN model produced the highest forecasting errors. In general loss function values are smaller at training performance. However this performance is not representative because, training set is used in the training of the model. In order to assess the performance of forecasting models, we have to test it with a new dataset. This performance is called as the testing performance and is a true indicator of the forecasting performance. Note that, all of these indicators have the smaller-the-better characteristic.

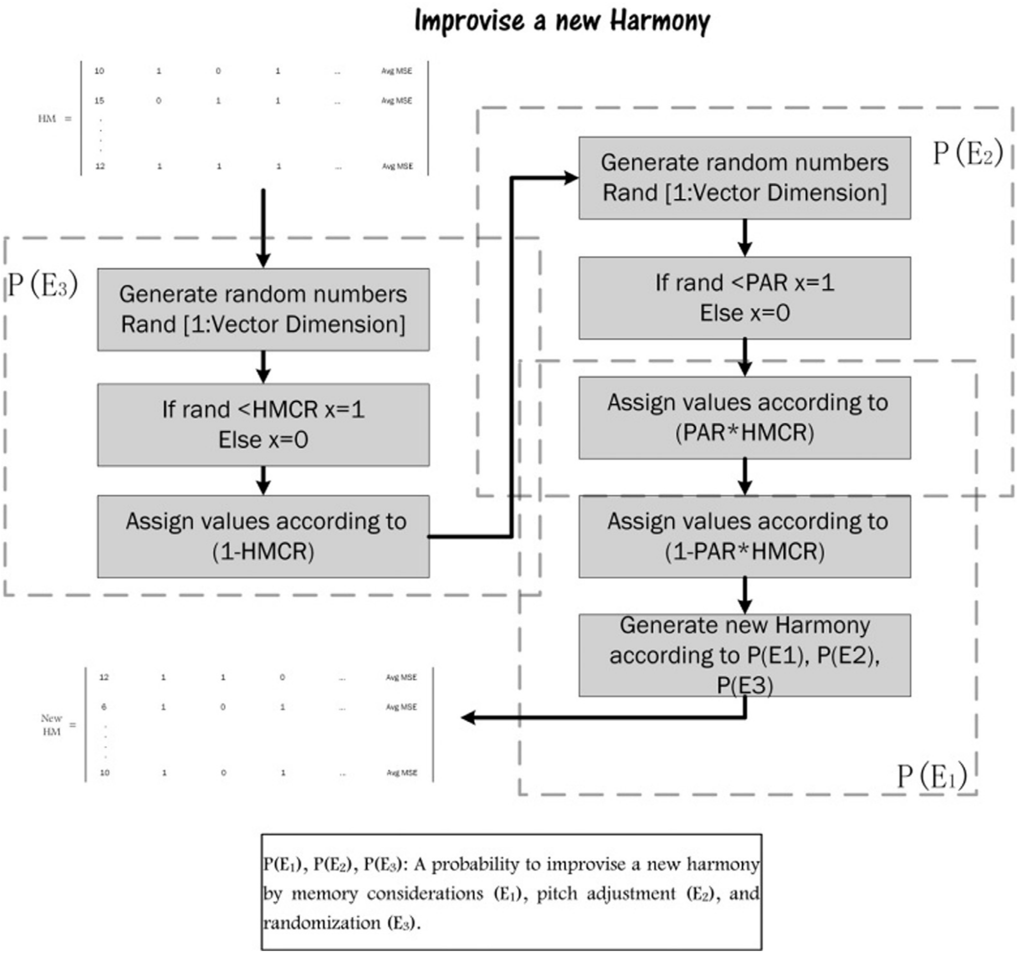

Fig. 9. New harmony improvization concept for the proposed HS-ANN forecasting model.

Table 6
Training and testing statistics of models.

	$e_i = p_i - a_i$ p: predicted price, a: actual price	Training HS-ANN	Training GA-ANN	Training regular ANN	Testing HS-ANN	Testing GA-ANN	Testing regular ANN
MAE	$\frac{1}{n}\sum_{i=1}^{n}\lvert e_i\rvert$	944.7658	978.9924	429.3262	2597.321	2950.251	2951.554
MSE	$\frac{1}{n}\sum_{i=1}^{n} e_i^2$	2475056	2916503	362116,2	11236305	12202954	14516650
RMSE	$\sqrt{\frac{1}{n}\sum_{i=1}^{n} e_i^2}$	1573.231	1707.777	601.7609	3352.06	3493.273	3810.072
MARE	$\frac{1}{n}\sum_{i=1}^{n}\lvert\frac{e_i}{a_i}\rvert$	0.019524	0.018384	0.008916	0.033814	0.038628	0.038191
MSRE	$\frac{1}{n}\sum_{i=1}^{n}\lvert\frac{e_i}{a_i}\rvert^2$	0.000772	0.000681	0.000155	0.00181	0.001995	0.002256
RMSRE	$\sqrt{\frac{1}{n}\sum_{i=1}^{n}\lvert\frac{e_i}{a_i}\rvert^2}$	0.027787	0.026087	0.012447	0.042541	0.044671	0.047493
MAPE	$\frac{100}{n}\sum_{i=1}^{n}\lvert\frac{e_i}{a_i}\rvert$	1.95244	1.838397	0.891558	3.381416	3.862837	3.819056
MSPE	$\frac{100}{n}\sum_{i=1}^{n}\lvert\frac{e_i}{a_i}\rvert^2$	7.720985	6.805103	1.549378	18.09704	19.95454	22.55602
RMSPE	$\sqrt{\frac{100}{n}\sum_{i=1}^{n}\lvert\frac{e_i}{a_i}\rvert^2}$	2.778666	2.608659	1.24474	4.254061	4.46705	4.749318

Regular ANN produced the smallest errors on training dataset. However in testing performance it produces the highest errors in almost all indicators except for MARE which means that regular ANN memorizes the training set and lost the generalization ability. This situation is also known as overfitting issue. When the performance of HS-ANN and GA-ANN models are examined, it becomes clear that in training dataset except for the first three indicators, HS-ANN model produced higher errors in comparison to GA-ANN model. However, in testing period HS-ANN model outperformed GA-ANN model which means it produced less error. This also means that HS-ANN has better generalization ability than GA-ANN model. Among the statistical loss functions MAPE has the most human friendly characteristic and gives the error in percent. Note that, HS-ANN model produced 3.38% error which is an acceptable error rate.

Fig. 10 depicts actual and forecasted prices as a time series. Every figure consists of two parts. In the upper side predicted and

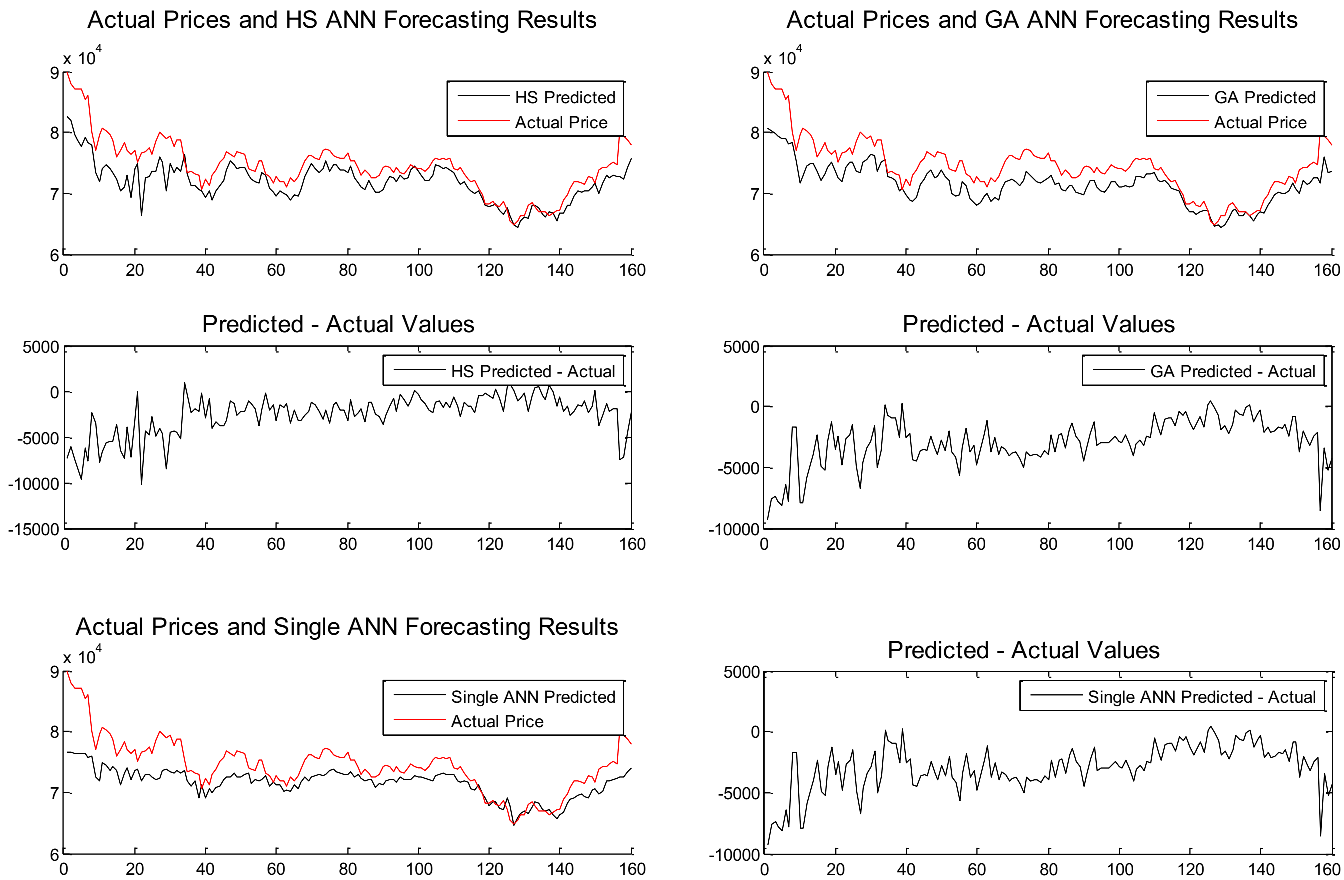

Fig. 10. Actual versus predicted prices as time series.

actual prices are illustrated while in the lower side predicted minus actual values are shown. Difference values are useful to get a general vision of how well the forecasting model is. It should be noted that in a perfect prediction difference graph the values should lie around zero. Since deviations from zero indicate a deviation from a good prediction it could be said that HS-ANN prediction performance is better than that of others. In GA and regular ANN models difference lines are far away from zero line.

Statistical performance measurements do not have much meaning for practical investors. Financial performance of forecasting model must also be examined to evaluate forecasting model. We can simulate buying and selling behaviors of a typical investor. An investor will buy stocks from the market if he/she expects an increase in prices. Similarly if an investor anticipates a decrease in prices then he/she will try to sell his/her financial assets in order to prevent a potential loss. We simulated the above mentioned simple trading logic using the prediction results (or predicted values) of the proposed models. A trading algorithm is developed to trade-on-paper along testing period. This algorithm returned the paper profits obtained from transactions. We neglected the trading costs and taxes to simplify the calculations.

Similarly, financial performances of the proposed models are compared with a passive trading strategy. In buy and hold strategy, an investor buys stocks from the beginning of trading period price and sells all of its assets from end of trading period closing price. The return in percent from this transaction is calculated as follow (Eq. (6)):

$$r = \frac{P_{t-n} - P_t}{P_{t-n}} \tag{6}$$

Table 7
A Comparison of the proposed models' performances with a passive trading strategy.

	HS-ANN model	GA-ANN model	Regular ANN model
Return from investment	0.060406	0.011221	−0.20626
Buy and hold	−0.13405	−0.13405	−0.13405

where P_{t-n} and P_t represents the first-day and last day closing price of stock index in testing period.

The result of trading strategy yields a loss of 13.41%. HS-ANN model yields a return of 6.04% profit while GA-ANN model returns only 1.12% of profit during 160 trading sessions. By the way, regular ANN yields a loss of 20.63% as can be seen from Table 7.

5. Conclusion

Over the years, researchers around the world have been studying to forecast the stock market price as precisely as possible to reach the best investment decisions. However, there is no consensus on the effectiveness of forecasting models and hence, research on improving the effectiveness of forecasting models has been continued. This paper has proposed a new hybrid model, based on a heuristic optimization methodology (HS or GA) and ANN, to improve stock market forecasting performance in terms of statistical and financial terms.

With development of the hybrid ANN models we show that structuring ANN has become easy in implementation because our proposed models have great capability in variable selection and determining the number of neurons in hidden layer. In order to select the most relevant technical indicators, we firstly set predetermined 45 variables and at the end of the analysis 26 and 23 variable are specified as non-redundant by GA and HS models, respectively. That means the complexity of the variable selection is reduced to almost its half. In addition, determining the optimum number of neurons in hidden layer eliminates the overfitting or underfitting problems of ANN models.

Based on the results, the average stock price forecasting performance of the HS-ANN (MAPE = 3.38) is significantly better than that of GA-ANN (MAPE = 3.86) model and the regular ANN model (MAPE = 3.81). It should be noted that MAPE values with proposed models are about 10% lower than the ones reported in existing studies. Furthermore, MAPE results with the proposed models look promising in emerging markets. Also, trading performances are quite impressive in proposed models. HS-ANN (%6.04) and GA-ANN (%1.12) models yield higher returns in comparison with regular ANN model (−%20.63). Even operating in a bear market (buy and hold return is −%13.41), forecasting models accomplished to yield a positive return.

Although proposed hybrid models of predicting stock market prices using the GA and HS give remarkable results, this study has some limitations. First, number of hidden layer is fixed at 1. Although training becomes excessively time-consuming with increasing number of hidden layers, the performance of the model can change with the number of hidden layer. The second limitation is predetermined transfer and training functions because combinations of training function and transfer functions may affect quality of ANN models.

HS-ANN and GA-ANN can be used successfully to forecast the stock market price movement in different stock markets. The other good direction for future research would be to consider other parameters which may affect the ANN architecture such as number of hidden layer, type of transfer function. Variants of the HS such as improved HS, global-best HS are possibly used to increase in the accuracy of the forecasting. Similarly, the effects of various GA including many different forms of selection, crossover, and mutation operators can be examined as a part of the approach used in this study.

References

Adaoglu, C. (2000). Instability in the dividend policy of the Istanbul Stock Exchange (ISE) corporations: evidence from an emerging market. *Emerging Markets Review, 1*(3), 252–270.

Adebiyi, A. A., Adewumi, A. O., & Ayo, C. K. (2014). Comparison of ARIMA and artificial neural networks models for stock price prediction. *Journal of Applied Mathematics*, 1–7. doi:10.1155/2014/614342.

Ahmed, J., Jafri, M. N., Ahmad, J., & Khan, M. I. (2007). Design and implementation of a neural network for real-time object tracking. *World Academy of Science, Engineering and Technology, International Journal of Computer, Electrical, Automation, Control and Information Engineering, 1*(6), 1816–1819.

Aldin, M. M., Dehnavi, H. D., & Entezari, S. (2012). Evaluating the employment of technical indicators in predicting stock price index variations using artificial neural networks (case study: Tehran Stock Exchange). *International Journal of Business and Management, 7*(15), 25–34.

Bishop, C. M. (1995). *Neural networks for pattern recognition*. Oxford: Clarendon Press.

Bonde, G., & Khaled, R. (2012). Stock price prediction using genetic algorithms and evolution strategies. In *Proceedings of the 2012 international conference on genetic and evolutionary methods* (pp. 10–15).

Burney, S. M. A., Jilani, T. A., & Ardil, C. (2005). Levenberg–Marquardt algorithm for Karachi Stock Exchange share rates forecasting. *International Journal of Computational Intelligence, 1*(3), 144–149.

Cateni, S., Colla, V., & Vannucci, M. (2011). A genetic algorithm-based approach for selecting input variables and setting relevant network parameters of a SOM-based classifier. *International Journal of Simulation Systems, Science & Technology, 12*(2), 30–37 UKSim 4th European modeling symposium on mathematical modeling and computer simulation.

Chiu, D. Y., & Chuang, K. P. (2003). Applying artificial neural network and Chinese news classification techniques to Taiwan Stock Market. *Tamsui Oxford Journal of Mathematical Sciences, 19*(2), 201–215.

Cinko, M., & Avci, E. (2009). Examining the day of the week effect in Istanbul Stock Exchange (ISE). *International Business & Economics Research Journal, 8*(11), 1–5.

Dastgir, M., & Enghiad, M. H. (2012). Short-term prediction of Tehran Stock Exchange Price Index (TEPIX): using artificial neural network (ANN). *Quarterly Journal of Securities Exchange, 4*(14), 237–261.

Duch, W., & Jankowski, N. (1999). Survey of neural transfer functions. *Neural Computing Surveys, 2*(1), 163–212.

Egeli, B., Ozturan, M., & Badur, B. (2003). Stock market prediction using artificial neural networks. In *Proceedings of the 3rd Hawaii international conference on business* (pp. 1–8). Hawaii, USA.

Geem, Z. W. (2006). Optimal cost design of water distribution networks using harmony search. *Engineering Optimization, 38*(3), 259–280.

Graupe, D. (2007). Principles of artificial neural networks. *Advanced series on circuits and systems, World Scientific Publishing, Singapore City (Vol. 6)* (2nd ed.).

Guresen, E., Kayakutlu, G., & Daim, T. U. (2011). Using artificial neural network models in stock market index prediction. *Expert Systems with Applications, 38*(8), 10389–10397.

Hadavandi, E., Shavandi, H., & Ghanbari, A. (2010). Integration of genetic fuzzy systems and artificial neural networks for stock price forecasting. *Knowledge-Based Systems, 23*(8), 800–808.

Hagan, M. T., & Menhaj, M. B. (1994). Training feedforward networks with the Marquardt algorithm. *IEEE Transactions on Neural Networks, 5*(6), 989–993.

Haider, A., & Hanif, M. N. (2009). Inflation forecasting in Pakistan using artificial neural networks. *Pakistan Economic and Social Review, 47*(1), 123–138.

Huang, C. L., & Wang, C. J. (2006). A GA-based feature selection and parameters optimization for support vector machines. *Expert Systems with Applications, 31*(2), 231–240.

Kara, Y., Boyacioglu, M. A., & Baykan, O. K. (2011). Predicting direction of stock price index movement using artificial neural networks and support vector machines: the sample of the Istanbul Stock Exchange. *Expert Systems with Applications, 38*(5), 5311–5319.

Karymshakov, K., & Abdykaparov, Y. (2012). Forecasting stock index movement with artificial neural networks: the case of Istanbul Stock Exchange. *Trakya University Journal of Social Science, 14*(2), 231–242.

Kermani, B. G., Schiffman, S. S., & Nagle, H. T. (2005). Performance of the Levenberg–Marquardt neural network training method in electronic nose applications. *Sensors and Actuators B: Chemical, 110*(1), 13–22.

Kougias, I., & Theodosiou, N. (2010). A new music-inspired harmony based optimization algorithm. Theory and applications. In *Proceedings of international conference on protection and restoration of the environment X*. Corfu.

Kuruş, O. A., Kılıç, N., & Uçan, O. N. (2013). Hermitian transform approach in classification of ECG signals. *İstanbul Aydın Üniversitesi Dergisi, 2*(7), 89–101.

Laboissiere, L. A., Fernandes, R. A. S., & Lage, G. G. (2015). maximum and minimum stock price forecasting of Brazilian power distribution companies based on artificial neural networks. *Applied Soft Computing, 35*, 66–74.

Li, E. Y. (1994). Artificial neural networks and their business applications. *Information & Management, 27*(5), 303–313.

Lourakis, M. I. A., & Argyros, A. A. (2005). Is Levenberg–Marquardt the most efficient optimization algorithm for implementing bundle adjustment? In *Proceedings of the 10th IEEE international conference on computer vision* (pp. 1526–1531).

Mahdavia, M., Fesanghary, M., & Damangir, E. (2007). An improved harmony search algorithm for solving optimization problems. *Applied Mathematics and Computation, 188*(2), 1567–1579.

Mańdziuk, J., & Jaruszewicz, M. (2011). Neuro-genetic system for stock index prediction. *Journal of Intelligent and Fuzzy Systems, 22*(2), 93–123.

Martínez, C. G., & Lozano, M. (2008). *Local search based on genetic algorithms, advances in metaheuristics for hard optimization* (pp. 199–221). Berlin Heidelberg: Springer.

Nikfarjam, A., Emadzadeh, E., & Muthaiyah, S. (2010). Text mining approaches for stock market prediction. In *Proceedings of Computer and Automation Engineering (ICCAE): 4* (pp. 256–260).

Otman, A., & Jaafar, A. (2011). A comparative study of adaptive crossover operators for genetic algorithms to resolve the traveling salesman problem. *International Journal of Computer Applications, 31*(11), 49–57.

Pardalos, P. M., Pitsoulis, L., Mavridou, T., & Resende, M. G. C. (1995). *Parallel search for combinatorial optimization: genetic algorithms simulated annealing, tabu search and grasp, parallel algorithms for irregularly structured problems* (pp. 317–331). Berlin Heidelberg: Springer.

Perwej, Y., & Perwej, A. (2012). Prediction of the Bombay Stock Exchange (BSE) market returns using artificial neural network and genetic algorithm. *Journal of Intelligent Learning Systems and Applications, 4*(2), 108–119.

Prasanna, S., & Ezhilmaran, D. (2013). An analysis on stock market prediction using data mining techniques. *International Journal of Computer Science & Engineering Technology, 4*(2), 49–51.

Preethi, G., & Santhi, B. (2012). Stock market forecasting techniques: a survey. *Journal of Theoretical & Applied Information Technology, 46*(1), 24–30.

Rajasekaran, S., & Pai, G. A. V. (2003). *Neural networks, fuzzy logic and genetic algorithm: synthesis and applications* (15th ed.) PHI Learning Pvt. Ltd.

Ravichandran, K. S., Thirunavukarasu, P., Nallaswamy, R., & Babu, R. (2005). Estimation of return on investment in share market through ANN. *Journal of Theoretical and Applied Information Technology*, 44–54.

Ruxanda, G., & Badea, L. M. (2014). Configuring artificial neural networks for stock market predictions. *Technological and Economic Development of Economy, 20*(1), 116–132.

Şenol, D., & Özturan, M. (2008). Stock Price Direction prediction using artificial neural network approach: the case of Turkey. *Journal of Artificial Intelligence, 1*(2), 70–77.

Subasi, A., & Erçelebi, E. (2005). Classification of EEG signals using neural network and logistic regression. *Computer Methods and Programs in Biomedicine, 78*(2), 87–99.

Sureshkumar, K. K., & Elango, N. M. (2012). Performance analysis of stock price prediction using artificial neural network. *Global Journal of Computer Science and Technology, 12*(1), 18–26.

Vaisla, K. S., & Bhatt, A. K. (2010). An analysis of the performance of artificial neural network technique for stock market forecasting. *International Journal on Computer Science and Engineering, 2*(6), 2104–2109.

Versace, M., Bhatt, R., Hinds, O., & Shiffer, M. (2004). Predicting the exchange traded fund DIA with a combination of genetic algorithms and neural networks. *Expert Systems with Applications, 27*(3), 417–425.

Wei, L. Y., & Cheng, C. H. (2012). A hybrid recurrent neural networks model based on synthesis features to forecast the Taiwan Stock Market. *International Journal of Innovative Computing Information and Control, 8*(8), 5559–5571.

Yadav, P., Kumar, R., Panda, S. K., & Chang, C. S. (2012). An intelligent tuned harmony search algorithm for optimization. *Information Sciences, 196*, 47–72.

Yang, X. S. (2009). Harmony search as a metaheuristic algorithm, in music-inspired harmony search algorithm: theory and applications. In Z. W. Geem (Ed.), *Studies in computational intelligence* (pp. 1–14). Berlin: Springer.

Yildiz, B., Yalama, A., & Coskun, M. (2008). Forecasting the Istanbul Stock Exchange national 100 index using an artificial neural network. *World Academy of Science, Engineering and Technology, 22*, 36–39.

Zahedi, J., & Rounaghi, M. M. (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*, 178–187.