Parameter Tuning of Neural Network for Financial Time Series Forecasting

Zeinab Fallahshojaei[1] and Mehdi Sadeghzadeh[2]
[1]Department of Computer Engineering, Buin Zahra Branch, Islamic Azad University, Buin Zahra, Iran
[2]Department of Computer Engineering, Mahshahr Branch, Islamic Azad University, Mahshahr, Iran

Abstract: *One of the most challengeable problems in pattern recognition domain is financial time series forecasting which aims to exactly estimate the cost value variations of a particular object in future. One of the best well-known financial time series prediction methods is Neural Network (NN) but it suffers from parameter tuning such as number of neuron in hidden layer, learning rate and number of periods that should be forecasted. To solve the problem, this paper proposes a new meta-heuristic-based parameter tuning scheme which is based on Harmony Search (HS). To improve the exploration and exploitation rates of HS, the control parameters of HS are adapted during the generations. Evaluation of the proposed method on several financial times series datasets shows the efficiency of the improved HS on parameter setting of NN for time series prediction.*

Keywords: *Financial times series forecasting, parameter setting, NN, HS, parameter adaptation.*

Received November 1, 2015; accepted March 20, 2018

1. Introduction

1.1. Problem Statement

Financial time series forecasting is a stochastic process which is based on the variation of financial random variable value over the periods (daily, weekly, monthly and annually). To estimate the financial time series (stock prices, currency exchange rates, price indices, and so on) in future, a prediction model of previously generated values of financial random variables is produced. The prediction model can generate the fittest financial values to the future time series if the prediction model captures the completely correct relationships among the financial variables of empirical dataset. As the predicted model of financial time series can be used in financial policies of a government, generation of a precise model is needed. For this purpose, several pattern recognition methods are introduced for financial time series forecasting.

1.2. Related Works

The first studies within the financial time series prediction framework are classic statistical approaches that are based on regression, correlation and spectral analysis [3, 8, 11]. While the regression-based prediction method [3] showed an inefficient performance, the results of correlation and spectral-based prediction methods [8, 11] proved their successfully against the regression-based approach. Also, several developments on time series forecasting framework were carried out in parallel [11]. The next most well-known and correlation-based prediction method is Auto regression-Integrated Moving-Average (ARIMA) which is introduced under the Box–Jenkins forecasting approach [1] and contains polynomial and seasonal trends. However this model is more computationally and functionality effective, it keeps the drawbacks of correlation based method. In another research, financial time series prediction was done by using a combination of wavelet transform, neural networks and statistical time series analytical techniques [2].

A well-known method of the intellectual forecasting group is Neural Network (NN). A main challenge of NN approach is parameter setting in which different values of parameters (such as number of neuron in hidden layer, learning rate and so on) take different effects on the performance of NN and so, the accuracy of NN highly depends on the initial parameter setting. A solution to obtain the optimal parameter setting of NN is meta-heuristic approach which discovers the search space (optimization problem) in order to achieve the sufficiently good solution. A well-known meta-heuristic-based method is Harmony Search (HS) which is based on the improvisation process of musician. One the most significant development on HS is Improved Harmony Search (IHS) that is proposed in [13]. IHS adapts some parameters dynamically in order to enhance the fine-tuning properties of the HS algorithm. A main challenge of IHS is exploration rate which may be leads the HIS into the local optima.

In this paper, IHS [13] is improved to get the best optimal solutions for the main parameters of NN as the follow:

a) Preprocessing of initial population of harmony vectors using the Differential Evolution (DE) in order to reduce the computational complexity of HIS.

b) Leading of the harmony vectors to the global best harmony vector in each cycle. After the improvement of HS, the best harmony vector which contains the optimal setting of parameter is employed in NN in order to estimate the financial time series with the high accuracy.

The rest of the paper is organized as follows. Introduction of HS, IHS, DE and MLP are provided in section 2. Section 3 illustrates the proposed method. Experimental results and discussion are given in Section 4, and finally, conclusion at section 5.

2. Background

2.1. Harmony Search (HS)

HS it is inspired by improvisation process of music player to find the best solution in optimization problem [6]. The process of finding the best harmony in music is similar to process of finding optimal solution in meta-heuristic algorithms and so, these two similar processes can be compared. Musician searches for a perfect state of harmony through improvisation process. On the other hand, in optimization problem, optimal solution should be achieved under the given objectives and limitations [6]. In improvisation process, a musician has three choices: first one is playing any notes exactly from memory (musician's memory); the second one is playing a note near to aforementioned notes of memory; and the last one is playing new or random notes. Likewise in HS algorithm, the value of a decision variable is selected according to one of the following rules:

a) Selecting a value from Harmony Memory (HM).
b) Choosing a value near to values of HM.
c) Selecting a random value in the possible range of values.

HS has several properties that differentiate it from other meta-heuristics such as [6]:

a) In HS, a new solution is generated according to all vectors of population while in Genetic Algorithm (GA), just two solutions (parent) are considered to generate new solution.
b) Each decision variable of harmony vector is considered, independently.
c) Defining of continuous values for decision variables without any loss the precision. In the HS algorithm, there are some components such as Harmony Memory (HM), HM Considering Rate (HMCR) and Pitch-Adjusting Rate (PAR) which are described as follows.

HM: in the HS, each candidate solution is known as a Harmony Vector (HV) which is represented as a set of real values in range of [0,1]. Furthermore, length of HV is equal to the number of dimension of a specific problem. HM is a matrix of HVs which are generated randomly in a solution space (1). HM Size (HMS) determines the number of HVs in HM.

$$HS = \begin{bmatrix} x_1^1 & x_2^1 & \cdots & x_n^1 \\ x_1^2 & x_2^2 & \cdots & x_n^2 \\ \vdots & \vdots & \ddots & \vdots \\ x_1^{HMS} & x_2^{HMS} & \cdots & x_n^{HMS} \end{bmatrix} \quad (1)$$

HMCR: HM Considering Rate (HMCR) is a probability of selecting a value from the HM (memory consideration) in order to update the value of a decision variable of a harmony vector. This state is analogous to the first choice in improvisation process. According to (2), with the probability HMCR, the new value of a decision variable is selected from the HM, while with probability (1-HMCR), the value of a decision variable is updated to a random value which is selected out of the HM but in the possible range of values. Note that the probability (1-HMCR) demonstrates the last choice of improvisation process. If the probability HMCR is satisfied then the intensification rate of HS is increased while with the probability (1-HMCR), the diversification rate of HS is increased. Note that the intensification ability helps the algorithm to tune the local search process while the diversification ability allows the HS to find unvisited solutions in the search space [6]. It is important that the value of HMCR is set to a high digit because in the music domain, each musician has a specific methodology and follows its method in the most melodies.

$$X'_{1,j} = \begin{cases} HM_{\lceil rand \times HMS \rceil, j} & \text{With Probability HMCR} \\ rand \in \left[\vec{X}_{min}, \vec{X}_{max}\right] & \text{With Probability}(1 - HMCR) \end{cases} \quad (2)$$

Where $\lceil rand \times HMS \rceil, j$ randomly selects the i^{th} row in the j^{th} column. $rand \in \left[\vec{X}_{min}, \vec{X}_{max}\right]$ selects a random value from interval $\left[\vec{X}_{min}, \vec{X}_{max}\right]$ which are the lower and upper bounds of the HM. Note that index '1' in $X'_{1,j}$ shows the index of harmony vector.

PAR: Pitch-Adjusting Rate (PAR) is a probability value to determine the rate of small changes in values of HV. In the other words, if the random generated value is lower than the HMCR, the condition of PAR (3) will checked, otherwise the PAR condition is not checked. This probability is used to model the second rule in the improvisation process of a musician. The value of PAR is low because a musician follows his/her method and selects rarely a random harmony to explore new tune.

$$X'_{1,j} = \begin{cases} X'_{1,j} \mp rand(0,1).bw & \text{With Probability PAR} \\ X'_{1,j} & \text{With Probability } (1 - PAR) \end{cases} \quad (3)$$

An arbitrary distance bandwidth (*bw*) shows small changes in values of variables. The pitch adjustment tunes the value of each decision variable within the predefined bandwidth in order to enhance the diversification [6].

2.2. Improved Harmony Search (IHS)

The fine-tuning features and convergence rate of HS algorithm are affected by the values of PAR and *bw* [13]. One of the improvements on the HS algorithm is IHS which is proposed in [13]. IHS focuses on the dynamically adaptation of the PAR and *bw* parameters during the optimization process. According to their theorem, the value of PAR is linearly increased using the following Equation:

$$PAR(gn) = PAR_{min} + \frac{PAR_{max} - PAR_{min}}{NI} \times gn \quad (4)$$

Where gn is the generation number, $PAR(gn)$ is the value of PAR in each generation, the PAR_{min} and PAR_{max} are the minimum and maximum pitch adjustment rates, respectively, and NI is the maximum number of generations. For the parameter *bw*, following equations exponentially decrease the value of *bw* in each iteration:

$$bw(gn) = bw_{max} . exp(c.gn) \quad (5)$$

$$C = \frac{Ln(\frac{bw_{min}}{bw_{max}})}{NI} \quad (6)$$

Where $bw(gn)$ is the value of *bw* in each generation, bw_{min} and bw_{max} are the minimum and maximum bandwidth, and the other parameters are defined as the previous. This kind of adaptation technique causes that the final best solution is improved in final generations which algorithm converged to optimal solution [13]. So, the convergence rate and fine-tuning properties of HS algorithm are improved.

2.3. Differential Evolution (DE)

Differential Evolution (DE) is a population-based approach which attempts to iteratively improve a solution according to the fitness function defined in a particular problem [4]. In DE, new solution $\overrightarrow{X'_\iota} = [x'_{i,1}, x'_{i,2}, x'_{i,3}, \cdots, x'_{i,n}]$ is generated using the combination of four particles where one of them is the candidate solution $\overrightarrow{X_\iota} = [x_{i,1}, x_{i,2}, x_{i,3}, x_{i,4}, \ldots, x_{i,n}]$ and the others ($\overrightarrow{X_a} = [x_{a,1}, x_{a,2}, x_{a,3}, x_{a,4}, \ldots, x_{a,n}]$, $\overrightarrow{X_b} = [x_{b,1}, x_{b,2}, x_{b,3}, x_{b,4}, \ldots, x_{b,n}]$ and $\overrightarrow{X_c} = [x_{c,1}, x_{c,2}, x_{c,3}, x_{c,4}, \ldots, x_{c,n}]$) are selected randomly from the population. For the combination of them, two main phases should be passed:

1) Difference-vector based mutation using (7) in order to generate trivial solution $\overrightarrow{V_\iota} = [v_{i,1}, v_{i,2}, v_{i,3}, v_{i,4}, \ldots, v_{i,n}]$.
2) Using the crossover operator (8) to generate new solution $\overrightarrow{X'_\iota}$:

$$v_{i,j} = x_{a,j} + F.(x_{b,j} - x_{c,j}) \quad (7)$$

Where F is the mutation factor that lies in interval [0.4,1]. The other parameters are defined as the previous. After construction of the trivial solution $\overrightarrow{V_\iota}$, candidate solution $\overrightarrow{X_\iota}$ and the trivial solution $\overrightarrow{V_\iota}$ are passed into crossover box for the recombination and generation of new solution $\overrightarrow{X'_\iota}$ as follows:

$$x'_{i,j} = \begin{cases} v_{i,j}, & \text{if } R < CR \\ x_{i,j}, & \text{otherwise} \end{cases} \quad (8)$$

Where R is a uniformly distributed random value from interval [0,1] and CR is the predefined crossover rate. After the generation of new solution $\overrightarrow{X'_\iota}$, the candidate solution would be replaced by the offspring if the fitness value of the offspring is better than the candidate solution. DE has some advantages such as:

1) Simplicity of implementation.
2) Few parameter setting.
3) Low time and space complexities.

In this paper, we focus on employing of IHS and DE to improve the convergence rate and fine-tuning features of HS. It should be noted that in this paper, the exploration rate of HS algorithm is increased by adding the Global Best (G_{best}) position to (10).

2.4. Multi-Layer Perceptron (MLP)

MLP is one of the best types of Neural Network (ANN) that consists of multi layers of nodes connected to each other. In MLP, each layer is fully connected to the next layer. The learning scheme of MLP is based on the backpropagation approach where actual outputs are compared with the desired outputs and the error rate between them is applied on the updating function of weights in order to form an appropriate pattern of training data [9]. For p^{th} input pattern, square error rate for all nodes inside the output layer of network is defined as follows:

$$E_p = \frac{1}{2}(d^p - y^p)^2 = \frac{1}{2}\sum_{j=1}^{s}\left(d_j^p - y_j^p\right)^2 \quad (9)$$

Where d^p and y^p are the desired output vector and actual output vector, respectively, d_j^p and y_j^p are the desired and actual outputs for the j^{th} node of output layer, respectively, s is the size of output vector. Thus, the total square error rate for the p patterns is computed as:

$$E = \sum_{p=1}^{P} E_p = \frac{1}{2}\sum_{p=1}^{P}\sum_{j=1}^{s}\left(d_j^p - y_j^p\right)^2 \quad (10)$$

In the network, the edge defined as the connection between the consequence layers takes a weight value which is updated iteratively to decrease the error rate E [9]. For this purpose, the weights are updated according to the following Equations:

$$w_{ij}(t+1) = w_{ij}(t) + \eta \times \Delta w_{ij}(t) + \alpha \times \Delta w_{ij}(t-1) \quad (11)$$

$$\Delta w_{ij}(t) = -\left(\frac{\partial E_p}{\partial w_{ij}(t)}\right) \quad (12)$$

Where η is learning rate, a is moment coefficient, $w_{ij}(t)$ is weight of edge from node i to j at time t and $w_{ij}(t+1)$ is is weight of edge from node i to j at the next time. The learning process of MLP will be finished when the error rate is less than the threshold or the maximum number of iterations is reached. The main challenge of NN model is on setting of learning rate, number of neurons inside the hidden layer and number of periods that should be forecasted. Due to the important role of these parameters on learning scheme of NN, they highly effect on the estimation results of NN and so, should be tuned. One of the best approaches to set the parameters of NN model is meta-heuristic algorithms such as HS. The details of the proposed method will be described as follows.

3. Proposed Method

This paper presents an improvement on parameter tuning of NN model based on the HS method which is improved in this paper. Since the previously reported results of NN on financial time series forecasting problem is not satisfactory, the proposed scheme of HS-based parameter setting of NN is evaluated financial time series forecasting problem. In this scheme, the length of each harmony vector is set to the number of NN parameters which should be tuned. The proposed values in each decision variable of harmony vector refer to corresponding parameter of NN. As the mentioned before, these parameters are learning rate, number of neurons inside the hidden layer and number of periods that should be forecasted.

Based on the mentioned problems of HS, several improvements on HS algorithm are considered as:

a) Improving of the convergence rate of HS.
b) Improving of fine-tuning features of HS and HIS.
c) Improving of the exploration rate of HS and IHS.

To achieve these goals, three innovations on the HS are proposed as:

a) Differential Evolution Harmony Search (DEHS).
b) Parameter Adaptation of DEHS (DEIHS).
c) Global-based DEIHS (DEGIHS).

The contributions are describes in detail as follows.

3.1. Differential Evolution Harmony Search (DEHS)

In this paper, we introduce a different facet of combination of DE and HS which is completely different from [5]. To combine the DE and HS, DE is used as the preprocessing step on the worse harmony vectors of initial population of HM. It means that after generation of initial population, all harmony vectors are evaluated, then, they are ranked according to their fitness values and finally, the initial population is divided into two subpopulations:

a) Subpopulation of harmony vectors with most significant fitness values (good subpopulation).
b) Subpopulation of harmony vectors with less significant fitness values (worse subpopulation).

After generation of subpopulations, the DE evolutionary algorithm is applied on the worse subpopulation to improve its harmony vectors. After the improving of worse subpopulation, it is replaced by the improved subpopulation and so, a population with optimal harmony vectors is subjected to undergo the evolutionary operations of HS. The improved population contains optimal solutions which are located in the neighboring of the optimum solution and so, the convergence rate of HS to the global optimum solution is increased. So, the first goal of this paper is met using DEHS. It is important that the run time of the proposed DEHS is lower than the original HS.

3.2. Parameter Adaptation of DEHS (DEIHS)

As the mentioned before, two parameters of HS, PAR and *bw*, play important roles in finding optimal solution in HS and need to be enhanced using the fine-tuning features of mathematical techniques. To improve the PAR and *bw* parameters of DEHS, the same idea of parameter adaptation scheme in IHS [13] is added to DEHS. Adding parameter adaptation scheme to DEHS helps the proposed algorithm to fine tune the global optimal solution and so, the second goal of this paper is achieved.

- Global-based DEIHS (DEGIHS): The third contribution of this paper is focused on the generation of novel harmony vector according to the Global Best (G_{best}) vector of the HM. In traditional HS, a novel harmony vector is generated by considering of all memory if the probability of HMCR is satisfied. This kind of generation forces high exploration rate to the HS. So, it would be possible that the HS jumps over the optimum solution and total number of generation is raised. The HS algorithm requires a technique to control this exploration rate. To achieve this goal, the position of the G_{best} particle is considered in the generation of novel harmony vector. In the other words, if the probability of HMCR is satisfied, novel harmony vector $\overrightarrow{X'_i} = [x'_{i,1}, x'_{i,2}, x'_{i,3}, \cdots, x'_{i,n}]$ is generated using the following Equation:

$$x'_{i,j} = x_{best,j} + (rand - \tfrac{1}{2}) \times HM_{[rand \times HMS],j} \quad (13)$$

Where $x'_{i,j}$ is the j^{th} dimension of novel vector $\overrightarrow{X'_i}$, variable $x_{best,j}$ is the j^{th} dimension of the best harmony vector $\overrightarrow{X_{best}} = [x_{best,1}, x_{best,2}, x_{best,3}, \cdots, x_{best,n}]$, and the *rand* operator generates randomly numbers which lie in interval [0,1]. The other parameters are defined as the previous. Formulation of the probability of (1-HMCR) is defined as the previous, this innovation controls the exploration rate of HS. According to this idea, the new generated harmony vector is located around the best harmony vector in each cycle and so, all vectors are led to the best harmony vector and the exploration rate of HS is tuned. The pseudo code of the proposed method is shown in Algorithm 1. To evaluate the harmony vectors, fitness function of the proposed method is NN model in which values of NN's parameters proposed by each harmony vector are passed into the NN function to order to train and test of NN model based on the proposed values. Therefore, in Algorithm 2 error rate of financial time series forecasting is computed. Note that the type of NN model is set to the Multi-Layer Perceptron (MLP). Flowchart of the propose method is demonstrated n Figure 1.

Algorithm 1. DEGIHS procedure

Input: HS parameter
Output: Best harmony vector
Initial HM
for solution :=1 to HMS do
Generate random Harmony vector in range [0,1].
Call the NN function as the fitness function to compute the fitness of solution.
end for
Call the DE procedure
for I :=1 to maximum iteration do
Adaptation of PAR and bw parameters in this iteration.
$X'(I) \leftarrow$ *select randomly a harmony vector from HM.*
for F :=1 to no. of features do
if rand(0,1) < HMCR then
$X'_F(I) = x_{best,F}(I) + (rand - \frac{1}{2}) \times HM_{[rand \times HMS],F}$
if rand(0,1) < PAR then
$X'_F(I) = X'_F(I) + rand(0,1).bw(I)$
end if
else
$X'_F(I) \leftarrow$ *randomly select any pitch within bounds.*
end if
end for
calculate the fitness of new vector $X'_F(I)$ *using NN function.*
if fitness $(X'_F(I))$ *is better than the fitness (worst vector) then*
replace new vector $X'_F(I)$ *with the worst one*
end if
end for
passing of best harmony vector to the NN function for financial time series forecasting.

Algorithm 2. NN function

Input: best values of NN's parameters proposed in harmony vector.
Output: error rate of financial time series forecasting.
1: set the input values to the corresponding parameters.
2: construction of NN model.
3: training of NN model on training data.
4: testing of NN on testing data.
5: computation of error rate between the estimated and real values of series.

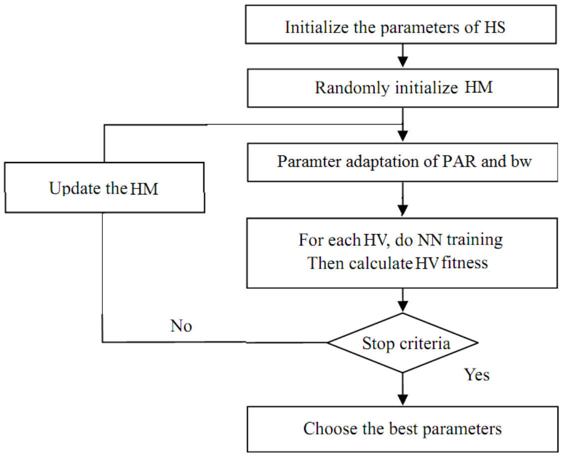

Figure 1. Flowchart of the proposed method.

4. Experimental Results

The main goal of the proposed method is improving the results of financial time series forecasting. For this purpose, the proposed method is evaluated on several financial time series datasets such as S&P 500 index, Dow Jones index, NASDAQ Composite index, NIKKEI index and Russell index which are collected from [7]. The information of each dataset is presented in the next Section. Note that the 80% of each dataset is used for training and the remained 20% of entire dataset is employed for testing step. The proposed method is compared with several meta-heuristic and evolutionary methods such as Genetic Algorithm (GA) [10], Particle Swam Optimization (PSO) [12], simple HS and NN with fixed parameter values (without parameter tuning). Note that all of the GA, PSO and simple HS are employed on parameter tuning of NN. The parameter setting is mentioned in Tables 1-5.

4.1. Time Series Dataset

S&P 500 index: this financial dataset is identified with the full name of Standard and Poor's 500 and relates to the stock exchange of USA. The original dataset is collect from 1950 until now but the financial data of 2013 to 2015 which contains 2020 records is used in this testing.

Dow Jones index: the full name of this dataset is Dow Jones Industrial Average which collected from 1896 until now by the Brazil Central Bank. Financial data of 2013 to 2015 which contains 150 records is employed in this testing.

NASDAQ Composite index: it is one of the most important financial datasets of USA stock index which contains the stock market information of all local and international companies of NASDAQ list from 2003 until now but the financial data from 2013 to 2015 with 3025 records is just employed.

NIKKEI index: this dataset contains the entire of stock market information of Japan from 1950 until now but the proposed method is tested on financial data from 2013 to 2015 with 507 records.

Russell index: this dataset is identified with the full name of Russell 1000 which is collected from 1992 until now by the Yahoo Finance group. The used dataset is selected from 2013 to 2015 with 285 records.

4.2. Performance Measure

To evaluate the results of the proposed method, two measures are employed such as Root Mean Square Error (RMSE) and Mean Absolute Difference (MAD), They are measures of the deviation between actual and predicted values. The smaller values of RMSE and MAD, the closer are the predicted values to that of the actual values. The measures are defined as follows:

$$RMSE = \sqrt{\frac{\sum_{i=1}^{N}(T_i - A_i)^2}{N}} \quad (14)$$

$$MAD = \frac{\sum_{i=1}^{N}|T_i - A_i|}{N} \quad (15)$$

Where in both of Equations (14) and (15), N refers to the number of testing set, T_i is the estimated value and A_i is the actual value.

Table 1. Parameter setting of simple NN (without parameter tuning).

Parameter	Value
No. of neurons inside the hidden layer	5
Learning rate	0.001
No. of periods for estimation	Daily

Table 2. Parameter setting of the proposed method (improved HS).

Parameter	Value
No. of population	20
Length of harmony vector	3
HMCR	0.9
PAR_{min}	0.4
PAR_{max}	0.9
bw_{min}	0.0001
bw_{max}	0.01
No. of iteration	100
MF	0.9
CF	0.9

Table 3. Parameter setting of simple HS.

Parameter	Value
No. of population	20
Length of harmony vector	3
HMCR	0.9
PAR	0.4
Bw	0.0001
No. of iteration	100

Table 4. Parameter setting of GA.

Parameter	Value
No. of population	20
Length of chromosome	3
Crossover rate	0.8
Mutation rate	0.1
No. of iteration	100

Table 5. Parameter setting of PSO.

Parameter	Value
No. of population	20
Length of particle	3
C_1	2
C_2	2
W	1
No. of iteration	100

4.3. Comparison

The estimation results of all mentioned methods on the Dow Jones dataset is reported in Figure 2-6. In each plot the horizontal axis refers to the set of sequential days for estimation and the vertical axis refers to the stock price. In all plots of this figure, the black curve shows the actual values of stock price in 30 sequential days and the colored curves shows the estimation results of all method on Dow jones dataset. Note that the plot in Figure 3 is related to the simple NN without the parameter tuning. Plots in figures 4-6 are related to HS-NN based, PSO-based and GA-based methods respectively. According to this figures, the fittest values to the actual values are obtained by the proposed method.

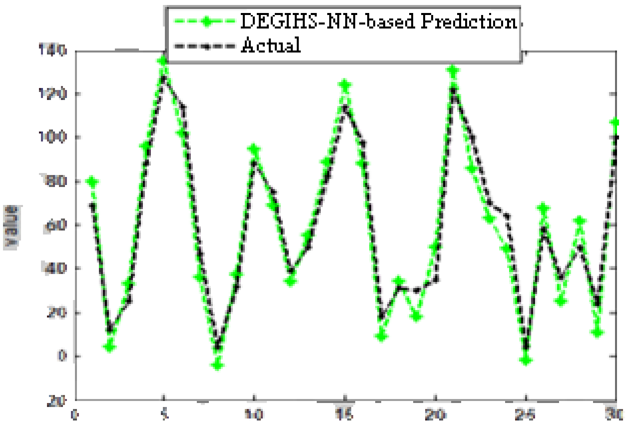

Figure 2. Estimation results of proposed method (DEGIHS) on Dow Jones financial dataset.

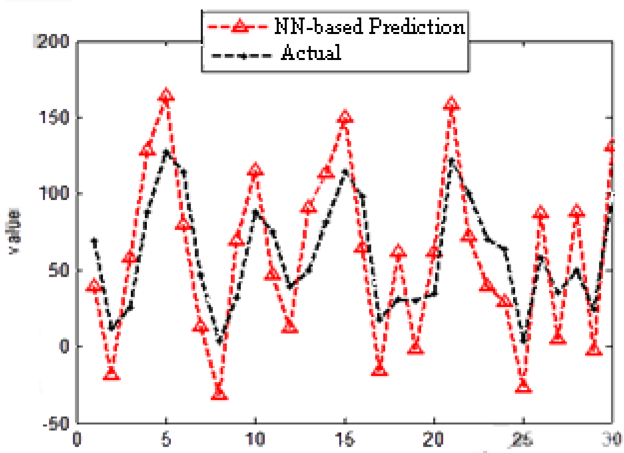

Figure 3. Estimation results of NN-based method on Dow Jones financial dataset.

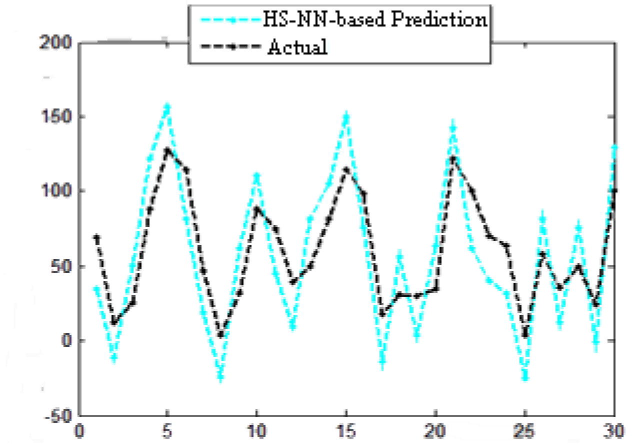

Figure 4. Estimation results of HS-NN-based method on Dow Jones financial dataset.

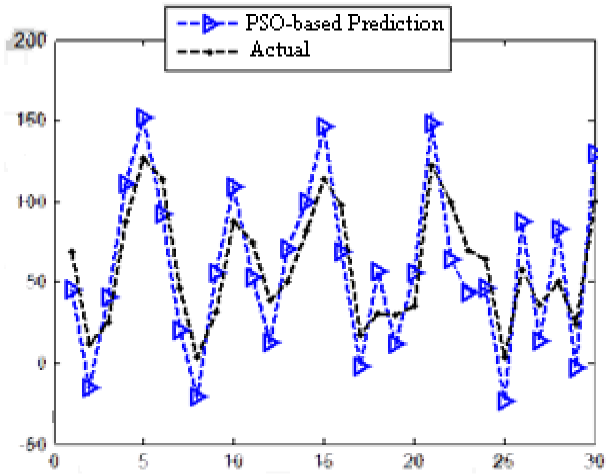

Figure 5. Estimation results of PSO-based method on Dow Jones financial dataset.

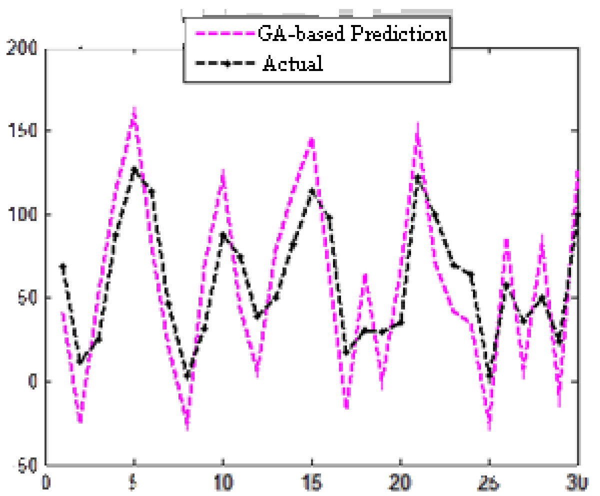

Figure 6. Estimation results of GA-based method on Dow Jones financial dataset.

According to the presented estimation results, the best financial time series forecasting results are obtained using the improved HS-based parameter tuning of NN that proves the effect of improved HS-based parameter tuning on the performance of NN. The reason of the success of the proposed method refers to the search strategy of the HS in which, it is able to efficiently reach the global optimum solution. For instance, the proposed G_{best} parameter on harmony vector generation equation helps the algorithm to jump over the local and so, the HS method gets the global optimum solution, quickly. Also, parameter adaptation scheme of HS algorithm causes that the algorithm spends less time on jumping to the global optimum point when the algorithm is near to the global solution in the search space and so, the convergence rate of the HS will be improved.

Table 6. Comparison of all methods on RMSE and MAD measures. The bold values mean the best values.

Meth-od	Dataset									
	S&P 500		Dow Jones		NASDAQ Composite		NIKKEI		Russell	
	RMSE	MAD	RMSE	MAD	RMSE	MAD	RMSE	MAD	RMSE	MAD
Simpl-e NN	10.6	6.9	11,9	7.5	11.7	7.5	10.3	6.2	11.6	7.4
GA-NN	10.4	6.3	11.2	7.0	11.6	7.4	10.0	4.9	10.6	6.3
PSO-NN	8.6	5.7	8.9	5..9	7.5	3.7	6.8	3.0	8.8	5.9
HS-NN	9.3	4.2	8.6	5.8	9.5	4.3	7.3	3.8	7.9	4.1
DEGIHS-NN	**5.2**	**2.4**	**4.9**	**2.1**	**5.7**	**3.2**	**3.8**	**1.8**	**4.0**	**2.3**

Table 6 presents the results of all methods on RMSE and MAD measures. In this table, for each dataset the two mentioned metrics are considered. According to Table 6, the proposed method with the name of DEGIHS-NN obtains the best results on all measures on all datasets. After proposed method HS-NN, PSO-NN, GA-NN and simple NN have better performance respectively. Note that sometimes in some data set PSO-NN is better than HS-NN. Since the worse harmony vectors are improved before the running the HS procedure, the population of harmony vectors will be passed the initial search steps and thus the difficulty of search in the proposed method is decreased. According to the presented results, the DEGIHS-NN can be considered as the best solutions to set the optimal values to NN's parameters. Results show important improvements in all 5 datasets.

5. Conclusions

In this paper, the main challenge of NN model for financial time series forecasting problem was studied that comes from the parameter setting of NN. To solve the problem, this paper was focused on utilization of HS method to find and set the optimal values to NN's parameters. One of the best meta-heuristics algorithms is the HS which is inspired by the improvisation process of a musician. In this paper, new developments on HS were proposed, called DEGIHS. The contributions of DEGIHS are listed as:

1) Adding the DE approach to the HS algorithm (DEHS) as a preprocessing step in order to improve the weak harmony vectors.
2) Improving the fine-tuning features of the DEHS

using the adaptation of PAR and bw parameters (DEIHS).

3) Generation of new harmony vectors according to the position of the G_{best} solution (DEGIHS).

After that, the best harmony vector proposed by DEGIHS function is applied on NN model for financial time series forecasting. The obtained results of the proposed method on several empirical datasets prove the superiority of the proposed method in comparison to the similar method.

References

[1] Box G. Jenkins G., *Time Series Analysis: Forecasting and Control*, Holden Day, 1970.

[2] Bozic J. and Babic D., "Financial Time Series Forecasting Using Hybrid Wavelet-Neural Model," *The International Arab Journal of Information Technology*, vol. 15, no. 1, pp. 50-57, 2018.

[3] Brillinger D., *Time Series. Data Analysis and Theory*, Holt, Rinehart and Winston Inc., 1975.

[4] Das S. and Suganthan P., "Differential Evolution: A Survey of the State-of-the-Art," *IEEE Transactions on Evolutionary Computation*, vol. 15, no. 1, pp. 4-31, 2011.

[5] Gao X., Wang X., and Ovaska S., "Modified Harmony Search Methods for Uni-Modal and Multi-Modal Optimization," *in Proceedings of 8th International Conference on Hybrid Intelligent Systems*, Barcelona, pp. 65-72, 2008.

[6] Geem Z., Kim J., and Loganathan G., "A New Heuristic Optimization Algorithm: Harmony Search. Simulation," *Journals SAGE*, vol. 76, no. 2, pp. 60-68, 2001.

[7] Global Stock Markets, https://www.quandl.com/c/markets/global-stock-markets, Last Visited, 2015.

[8] Hannan E., *Multiple Time Series*, John Wiley and Sons, 1979.

[9] Haykin S., *Neural Networks*, Macmillan College Publishing Company, 1999.

[10] Holland J., "Genetic Algorithms," *Scientific American*, vol. 267, no. 1, pp. 66-72, 1992.

[11] Kendall M. and Stuart A., *the Advanced Theory of Statistics. Design and Analysis, and Time Series*, Charles Griffin and Company, 1968.

[12] Kennedy J., *Particle Swarm Optimization in Encyclopedia of Machine Learning*, Springer, 2010.

[13] Mahdavi M., Fesanghary M., and Damangir E., "An Improved Harmony Search Algorithm For Solving Optimization Problems," *Applied Mathematics and Computation*, vol. 188, no. 2, pp. 1567-1579, 2007.

Zeinab Fallahshojaei was born in Lahijan, Iran, in 1988. He received the B.Sc. degree in Computer engineering from the Azad University of Lahijan, Iran, in 2012, and the M.Sc degree in computer engineering from Islamic Azad University of Buin Zahra, Iran in 2016.

Mehdi Sadeghzadeh is currently, an assistant professor of the Department of Computer Engineering, Mahshahr Branch, Islamic Azad University. He Received B.Sc. degree in Computer engineering from the Amirkabir University of Technology, Tehran, Iran, and the M.Sc degree in Computer engineering from Tarbaiat Modarres University, Tehran, Iran, and Ph.D. degree in Computer engineering from the Islamic Azad University, Science and Research Branch, Tehran, Iran. His current research interests include data mining, soft computing, image processing and distributed systems.