Neurocomputing 179 (2016) 69–87

Contents lists available at ScienceDirect

Neurocomputing

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

A self adaptive harmony search based functional link higher order ANN for non-linear data classification

Bighnaraj Naik [a,*], Janmenjoy Nayak [a], H.S. Behera [a], Ajith Abraham [b,c]

[a] Department of Computer Sc. Engg. & Information Technology, Veer Surendra Sai University of Technology, Sambalpur, Burla 768018, Odisha, India
[b] Machine Intelligence Research Labs (MIR Labs), Washington, USA
[c] IT4Innovations - Center of excellence, VSB -Technical University of Ostrava, Czech Republic

ARTICLE INFO

Article history:
Received 27 February 2015
Received in revised form
15 October 2015
Accepted 7 November 2015
Available online 10 December 2015

Keywords:
Self adaptive harmony search
Gradient descent learning
Functional link artificial neural network
Classification
Data mining
Machine intelligence

ABSTRACT

In the data classification process involving higher order ANNs, it's a herculean task to determine the optimal ANN classification model due to non-linear nature of real world datasets. To add to the woe, it is tedious to adjust the set of weights of ANNs by using appropriate learning algorithm to obtain better classification accuracy. In this paper, an improved variant of harmony search (HS), called self-adaptive harmony search (SAHS) along with gradient descent learning is used with functional link artificial neural network (FLANN) for the task of classification in data mining. Using its past experiences, SAHS adjusts the harmonies according to the maximum and minimum values in the current harmony memory. The powerful combination of this unique strategy of SAHS and searching capabilities of gradient descent search is used to obtain optimal set of weights for FLANN. The proposed method (SAHS–FLANN) is implemented in MATLAB and the results are contrasted with other alternatives (FLANN, GA based FLANN, PSO based FLANN, HS based FLANN, improved HS based FLANN and TLBO based FLANN). To illustrate its effectiveness, SAHS–FLANN is tested on various benchmark datasets from UCI machine learning repository by using 5-fold cross validation technique. Under the null-hypothesis, the proposed method is analyzed by using various statistical tests for statistical correctness of results. The performance of the SAHS–FLANN is found to be better and statistically significant in comparison with other alternatives. The SAHS–FLANN differs from HS–FLANN (HS based FLANN) by the elimination of constant parameters (bandwidth and pitch adjustment rate). Furthermore, it leads to the simplification of steps for the improvisation of weight-sets in IHS–FLANN (improved HS based FLANN) by incorporating adjustments of new weight-sets according to the weight-sets with maximum and minimum fitness.

1. Introduction

1990s saw a boom in the growth of data collection in almost every field – web, business management, e-commerce, remote sensors, microarrays gene expression, scientific simulations, production control and engineering design, transactions, stocks, bioinformatics, etc. And, since then there was a rise in the need of automated extraction of valid, unknown, novel and potentially useful information from the data in large databases which gave birth to many data analysis methodologies including data mining, business intelligence, etc. Data analysis is an analytical process of examining data to discover useful information and draw conclusions, which in turn, help in the decision making process. It integrates diversified techniques under statistics, engineering and science. Data mining is a data analysis process of identifying novel, understandable and previously unknown patterns in data, which acts as decision support system. Most tricky and challenging decision making processes in day to day human life is classification that helps to make decision from past experience. In data mining, classification is defined as a variety of data analysis process that can be used to assign important classes to unknown patterns. Classification task predicts definite class labels and constructs a model based on the training data set, which is used to classify anonymous patterns.

Many classification tasks have been proposed in recent years from emerging areas of science and engineering which comprise of document classification [1–3], sentiment classification [4–7], fault classification [8–11], text classification [12–14], image classification [15–18], gene expression classification and bio medical data classification [19–23] and others [24–30], which has given new shape, motivation and direction to the application of classification task in data mining. Although a good number of traditional classification methods are proposed by many researchers [31–35], for the first time, Zhang et al. [35] realized that artificial neural network models

* Corresponding author. Tel.: +91 9439152272.
E-mail addresses: mailtobnaik@gmail.com (B. Naik), mailforjnayak@gmail.com (J. Nayak), mailtohsbehera@gmail.com (H.S. Behera), ajith.abraham@ieee.org (A. Abraham).

http://dx.doi.org/10.1016/j.neucom.2015.11.051
0925-2312/

are alternative to various conventional classification methods which are based on statistics. The artificial neural networks (ANNs) are capable of generating complex mapping between the input and output space and thus they can form arbitrarily complex nonlinear decision boundaries. Along the way, there are already several artificial neural networks, each utilizing a different form of learning or hybridization. As compared to higher order neural network, classical neural networks (for example, MLP) suffer from slow convergence and are unable to automatically decide the optimal model of prediction for classification. In the last few years, to overcome the limitations of conventional ANNs, some researchers have focused on higher order neural network (HONN) models [36,37] for better performance. MLP is the most frequently used ANN model to solve real world complex classification problems. However, in MLP, the no. of hidden layers and number of neurons in the hidden layer change depends on the nature and complexity of the problem. The training of MLP is more complex due to the fact that the number of weight adjustments increase with the increase in number of layers and number of neurons in each layer. To address this intricacy associated with MLP, Pao [38] and Pao et al. [39] have suggested to use a single layer neural network (without hidden layer) which can be considered as an unconventional approach. The functional link higher order ANN (FLANN) proposed by them may be suitably used because of faster convergence rate, smaller computational load and less complex architecture as compared to MLP. FLANN is basically a flat network without hidden layers which makes it simple for learning and weight adjustment. Notwithstanding the fact that it has a single-layer network, it is capable to handle non-linear separable classification task as compared to MLP.

Functional link artificial neural network (FLANN) [38] utilizes the higher combination of its inputs [39,40] and has been successfully applied in a wide spectrum of applications such as pattern recognition [41,42], classification [43–45], channel equalization [46] , system identification [47–52], etc. Almost all the higher order ANNs (HONNs) including functional link higher order ANN (FLANN) are sensitive to random initialization of weight and rely on the learning algorithm adopted. Although a selection of efficient learning algorithm for HONNs help to improve the performance, but initialization of weights with optimized weights rather than random weights also plays an important role in the efficiency of HONNs. Various benchmark meta-heuristic optimization algorithms like genetic algorithm [53,54] and particle swarm optimization [55] are successfully used to address the issue of random initialization of weight in FLANN. These optimization algorithms are used to select best set of weights for FLANN models for various non-linear data analysis. Few negative aspects of these implementations are the requirement of various complicated mathematical operators like: (i) mutation and crossover operator in GA (ii) Position and velocity calculation and updation in PSO etc. The performance of these models are highly dependent upon the way of implementation of these mathematical operations (like selection of crossover operation, mutation operation and mutation rate) and any change in these factors may lead to increase in time and space complexity of the algorithm. Keeping these facts in mind, in this study, the meta-heuristic optimization algorithm: harmony search [56] and its variants (improved HS [57] and self-adaptive HS [58]) are used with FLANN model with gradient descent learning scheme for classification. The HS algorithms have few mathematical requirements as compared to earlier meta-heuristic optimization algorithms and those can be easily used for optimization of weights of FLANN model.

2. Literature survey

In this paper, an attempt has been made to design FLANN model with competitive learning based on a popular meta-heuristic optimization algorithm for classification of benchmark datasets considered from well known machine learning data repository. Prior to this, Patra et al. [59] have proposed FLANN model for prediction of financial indices and have used other Chebyshev neural network model with Chebyshev polynomial functional expansion. From the simulation studies they found that, the performance of FLANN and chFLANN are nearly equivalent. Also, the training time for FLANN and chFLANN is almost half of the MLP. It is found that FLANN and chFLANN have less complex architecture and chFLANN is better among the three ANNs (MLP, FLANN and chFLANN) in terms of better prediction capability. A FLANN based classification method with a least complex architecture as compared to MLP is anticipated by Misra and Dehuri [60] and the proposed method is found to be efficient in terms of ability of handling linearly non-separable classes by increasing the dimension of input space. In most of the cases, the performance and processing time of FLANN model is found to be better than other models. To address non-linear nature of classification problems, Dehuri et al. [61] have designed a hybrid functional link artificial neural network (HFLANN) based on genetic algorithm (GA) for optimal input feature selection by using functionally expanded selected features. They proved that, as compared to RBFN and FLANN with back propagation learning, HFLANN is found better for optimal set feature selection. A survey on FLANN is made and a PSO based back propagation learning is proposed by Dehuri and Cho [62]. They have emphasized on the basic concept of FLANN with associated basis functions, learning schemes, development of FLANNs over time and discussed a Chebyshev-FLANN with hybrid PSO-back propagation learning for classification. Patra et al. [63] suggested an efficient FLANN model for making stock price prediction of the closing price of US stocks and the proposed model is compared with MLP-based prediction model through several experiments. This proposed trigonometric FLANN has shown better performance over MLP by making more accurate predictions of stock prices. Prediction of the causing genes in gene diseases by FLANN model is proposed by Sun et al. [64] and compared with MLP and support vector machine (SVM). In this study, three classifiers (MLP, SVM, FLANN) have been implemented and the performance of FLANN classifier is found to be better than MLP and SVM. Chakravarty and Das [65] have proposed functional link neural fuzzy (FLNF) model to predict the stock market indices and compared with FLANN model in terms of root mean square error. The simulation results have been demonstrated to claim the better performance of FLNF over FLANN and they used PSO to address the the local minima issue in back propagation learning. Classification of online Indian customer behavior by using FLANN is proposed by Majhi et al. [66] which is better than the other statistical approaches. The proposed FLANN model shows superior classification performance as compared to discriminant analysis and also the authors suggested that, it can be improved further by providing psychographic and cultural information. A compact and accurate hybrid FLANN classifier (HFLNN) has been proposed by Dehuri and Cho [67] by selecting an optimal subset of favorable input features. By eliminating features with fewer or no predictive information, this method is found better as compared to FLANN and RBFN. Bebarta et al. [68] have used FLANN and its variants (Power FLANN, Laguerre FLANN, Legendre FLANN and Chebyshev FLANN) for stock price index forecasting and they have compared the results by using various performance measures (standard deviation error, squared error etc.). Mishra et al. [69] have used FLANN classification model based on bat inspired optimization and compared with FLANN and hybrid PSO based FLANN model. In this paper, bat algorithm has been used to update the weight of the FLANN, which results high classification accuracy. Simulation results show that bat algorithm based FLANN outperform FLANN and hybrid PSO based FLANN classifiers. The Chebyshev FLANN classifier with various dimension reduction strategies is suggested by Mahapatra et al. [70] for cancer classification. In this proposed method, initially PCA, FA, DFT and DCT techniques have been used to reduce the dimension of the data and then Chebyshev FLANN classifier is applied for better classification.

Table 1
Recent applications of FLANN models.

Reference	*Model used*	*Learning method employed*	*Application*
Sicuranza and Carini [76]	Recursive FLANN	Back propagation	Noise Control
George and Panga [77]	FLANN	Back propagation	Noise Control
Sicuranza and Carini [78]	Recursive FLANN	Back propagation	Noise Control
Parija et al. [79]	FLANN	Back propagation	Location management
Ali and Haweel [80]	Legendr- FLANN	Back propagation	Channel equalization
Durga and Tarun [81]	FLANN	Back propagation	Wind Power Forecasting
Durga and Tarun [81]	Legendre - FLANN	Back propagation	Wind Power Forecasting
Durga and Tarun [81]	Chebyshev - FLANN	Back propagation	Wind Power Forecasting
Cui et al. [82]	FLANN	Back propagation	Identification of Model

They have shown that, the DCT feature reduction technique along with Chebyshev FLANN classifier outperforms other alternatives. The MLP, FLANN and PSO based FLANN classification models are used and tested by Mishra et al. [71] for classification of biomedical data. They proposed an efficient dynamic classifier fusion (DCF) along with principal component analysis (PCA) scheme, which is used to extract important input features. Extracted features are then supplied to LMS classifiers with learning based on PSO based back propagation algorithm. Although MLP is a traditional ANN, surprisingly, in this study, it is found better as compared to FLANN and PSO based FLANN. Dehuri et al. [72] have proposed an IPSO (improved PSO) based FLANN classifier (IPSO–FLANN) and compared with MLP, SVM, RBFN, FLANN with GDL and fuzzy swarm net (FSN) model. Initially, a set of weight values of functional link ANN are optimized using IPSO and supplied to FLANN classifier along with functionally expanded (using trigonometric basis functions) input patterns for classification. The ISO–FLANN is simple in architecture and has better performance than MLP, SVM, FLANN with GDL and FSN. The PSO based FLANN, GA based FLANN and differential evolution (DE) based FLANN for classification task are proposed by Mili and Hamdi [73] .They compared and tested the proposed models with various expansion functions. It is found that, the proposed hybrid FLANN models are performing better in terms of accuracy and convergence as compared to FLANN. Naik et.al. [74] have addressed an efficient FLANN classifier with a hybrid-learning scheme based on PSO and GA for classification, which is relatively better than the other alternatives. They have used PSO, GA and gradient descent search to iteratively adjust the parameters of FLANN until the error is less than the required value, which helps the FLANN model to get better classification accuracy. Naik et al. [75] have proposed a honey bee mating optimization (HMBO) based learning scheme for FLANN classifier and compared with FLANN, GA based FLANN and PSO based FLANN classifiers. To select the best weights for FLANN classifiers, the proposed method mimics the iterative mating process of honeybees and strategies to select eligible drones for mating process.

Apart from these applications, many recent variants of FLANN models with hybrid learning methods has been used for a variety of applications from the period 2011 to 2015 (Table 1) [76–82]. In this paper, a FLANN model with hybrid self-adaptive harmony search and gradient descent search based learning method for classification have been proposed and compared with previously available alternatives.

The remaining part of the paper is organized as follows: background, motivation and contribution of proposed work in Section-3, preliminaries in Section-4, proposed method in Section-5, experimental setup in Section-6, results obtained in Section-7, proof of statistical significance in Section-8, conclusion in Section-9 and references.

3. Background, motivation and contribution of proposed work

In previous related works, various optimization algorithms (GA, PSO, Improved PSO, HMBO [83,84] etc.) are used to select best set of weights for FLANN models for various non-linear data classification. Although it is reported that these optimization techniques are successfully used in FLANN models to design improved models like GA based FLANN (GA–FLANN) [61], PSO based FLANN (PSO–FLANN) [62] and IPSO based FLANN (IPSO–FLANN) [72] and HBMO based FLANN (HBMO–FLANN) [75], the major negative aspects of these implementations are the requirement of various complicated mathematical operators like: (i) mutation and crossover operator in GA in GA–FLANN, (ii) position and velocity calculation in PSO in PSO–FLANN and IPSO–FLANN and (iii) crossover and mutation in HBMO in HBMO–FLANN etc. The performances of these models are highly dependent upon the way of implementation of these mathematical operation and any changes in these factors may lead to increase in time and space complexity of the algorithm.

Keeping into fact, the meta-heuristic optimization algorithm: harmony search [56] is used in FLANN model with GDL scheme for classification. Many researchers are attracted towards the study of harmony search and its applications due to the fact, HS algorithms has few mathematical requirements as compared to earlier meta-heuristic optimization algorithms and can be easily used for optimization problems. A survey is made by Manjarres et al. [85] on application of harmony search algorithms till the year 2012 from the scientific database of Elsevier, IEEE and Springer. They shown that, various works are published in the area of diverse application of HS which includes engineering, water system management, medical, robotics, control, power and energy, cross application and others. Starting from the development of HS, it has been a keen interest among the diversified researchers and has been used in various real life applications [85].

Inspired from successful applications of harmony search algorithms, in this paper, an attempt has been made to address the intricacy in adjusting the set of weights of the FLANN model by using appropriate learning algorithm. Recently, a FLANN model with harmony search algorithm is reported by Naik et al. [86], in which an attempt has been made to address the above issues. The authors have claimed for better performance of HS–FLANN over FLANN, GA–FLANN and PSO–FLANN. This proposed HS–FLANN utilizes searching ability of harmony search effectively to obtain optimized set of weights for FLANN and requires less mathematical computation.

However, the performance of this model can be improved by eliminating constant parameters (bw, PAR) in HS algorithm and incorporating dynamically changes in PAR and bw, with iteration number. The mechanism of dynamically decreasing of bw with iteration is inspired from the strategy of decreasing the learning rate of neural networks dynamically [87]. The improved harmony search algorithm [57] is free from the fixed values of PAR and bw, by dynamically changing bw and PAR with iteration number. Incorporating these, an IHS based FLANN model (IHS–FLANN) [88] with GDL is used for better performance. Employing these strategies has found considerable influence on the quality of solutions as compared with previously available alternatives.

Further, the pitch adjustment step during improvisation of weight-sets has been made easier by using self adaptive harmony search (SAHS) [58]. The SAHS is a modification of IHS in order to make pitch adjustment step simpler. In the proposed SAHS based FLANN (SAHS–FLANN), the pitch adjustment step in IHS have been modified to incorporate better utilization of its own experiences, by updating the new harmonies according to the maximum and minimum values in the HM. Here, the objective is to simplify the pitch adjustment step by introducing a novel strategy of adjusting new harmony by using maximum and minimum values in HM encountered so far, thereby eliminating bw altogether from HS procedure.

In this study, the problem solving approach of the self adaptive harmony search along with learning ability of the GDL is used to obtain the optimal set of weights of FLANN model. This paper introduces a self adaptive harmony search based learning scheme with FLANN model for non-liner data classification. The main contributions of the proposed scheme, which differs from earlier works are: (1) it keeps away from the use of constant parameters (PAR and bw) on harmony search algorithm based learning scheme (HS–FLANN) while adjusting weights and (2) it makes use of better utilization of its own experiences by updating the new weight-sets according to the maximum and minimum values in the HM, thereby eliminating bw altogether from improve harmony search based learning scheme (IHS–FLANN). This adjustment method for weight-sets eliminates intricacy of parameter setting and also improves the quality of weight-sets. A good number of experiments are carefully conducted by setting suggested values [58] to control parameters. Also an investigation is made to note the impact of these control the parameters to obtain a better set of values.

4. Preliminaries

4.1. Functional link artificial neural network

The training of MLP model happens to more complex due to the fact that the number of weight adjustments increase with the increase in no. of layers and no. of neurons in the each layer. In contrast to MLP, FLANN is basically a flat network without hidden layers which makes FLANN model simpler for learning and weight adjustment. However, the use of single layer neural network as an alternative to MLP with multiple layer neurons is linear in nature and often fails to solve the real life complex problems including classification task in data mining which are extremely non-linear in nature. In order to fill the gap between linearity in the single layer neural network and the multi-layer neural networks which are complex and computationally intensive in nature, FLANN is designed as a single layer network (without hidden layers) with functional expansion mechanism. The functional expansion mechanism in FLANN effectively enlarges the dimension of the input vectors and the enhanced feature space provides better discriminating ability of the input patterns. Various mathematical functions (sine, cosine, log, etc.) are used for functional expansion.

Basically, in FLANN, the input pattern goes through a transformation (known as functional expansion) of input data by which dimension of input space increases artificially. Then the extended input data is used to train the feed forward network. Mathematical functions, such as sine, cosine, log, etc. are used in functional expansion to transform an original input pattern to it extended version. If the no. of attributes of an input pattern increases, then the number of input terms during functional expansion increases rapidly. Fig. 1 depicts the basic architecture of FLANN.

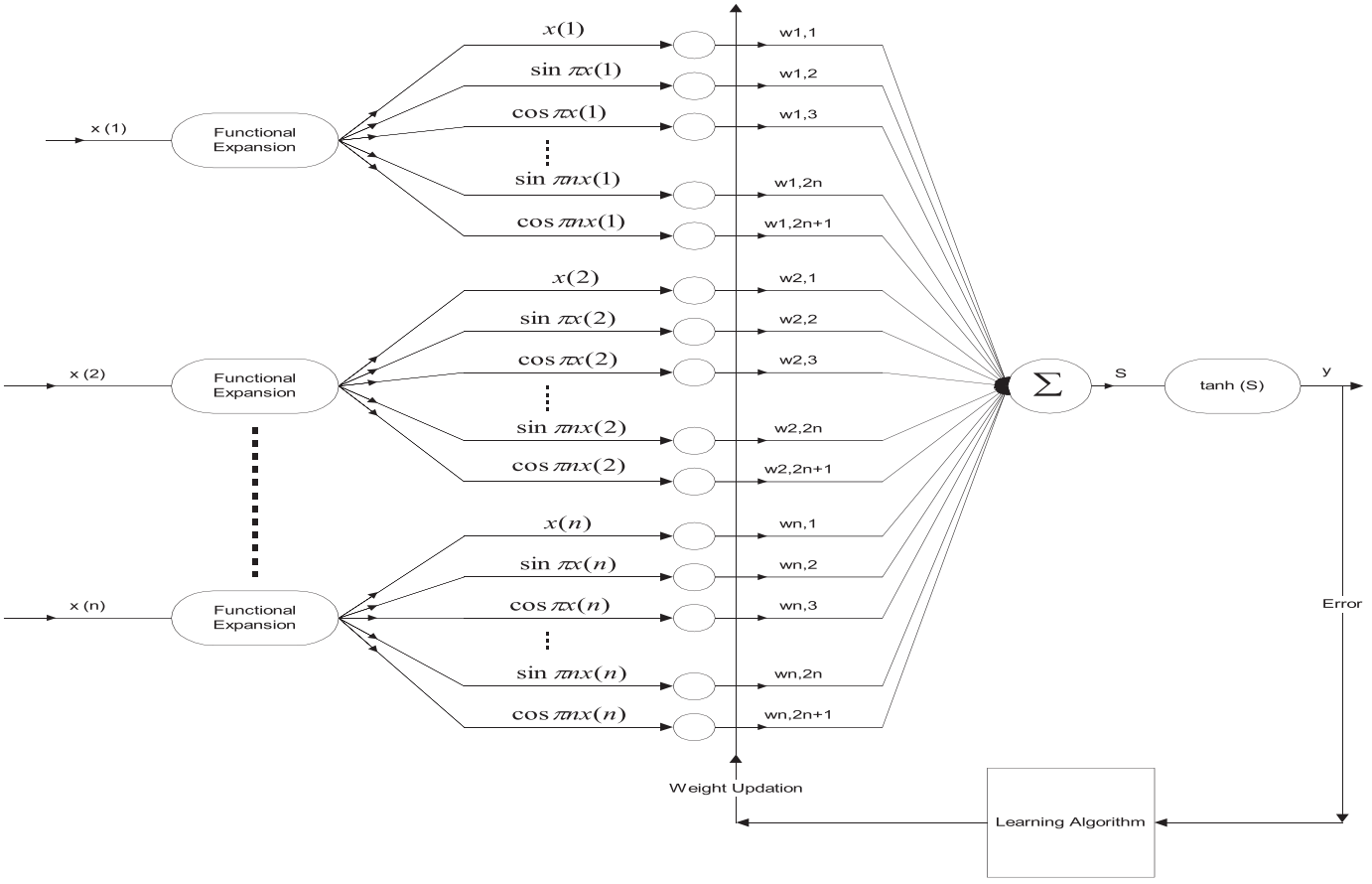


Fig. 1. Functional link artificial neural network architecture.

Let '*x*' be a dataset in a form of matrix of order *mxn*. The functionally expanded values for dataset *x* can be generated by using Eq. (1), where $x_i(j)$ stands for *j*th attribute value of *i*th pattern.

$$\varphi(x_i(j)) = \{x_i(j),\ \cos \Pi x_i(j),\ \sin \Pi x_i(j),\ \cos 2\Pi x_i(j), \sin 2\Pi x_i(j) \ldots \cos n\Pi x_i(j),\ \sin n\Pi x_i(j)\} \quad (1)$$

For each input attribute value of a pattern $x_i(j)$, $2n+1$ no. of functionally expanded values are generated, intern, for a single input pattern x_i , $(n^*(2n+1))$ no. of expanded values are generated. In Eq. (1), value of i is ranged from $i=1$ to m and value of j is ranged from $j=1$ to n, where m and n are no. of input pattern and no. of attribute values of each input pattern respectively except the class label. The functional expansion for dataset *x* is represented by using Eq. (2).

$$\varphi = \left\{ \{\varphi(x_1(1)), \varphi(x_1(2)) \ldots \varphi(x_1(n))\}^T, \{\varphi(x_2(1)), \varphi(x_2(2)) \ldots \varphi(x_2(n))\}^T \ldots \ldots \{\varphi(x_m(1)), \varphi(x_m(2)) \ldots \varphi(x_m(n))\}^T \right\} \quad (2)$$

The above functionally expanded values 'φ' are the input to FLANN classifier for training phase. Prior to this, weights of FLANN must be set randomly. As each input pattern is transformed to $n^*(2n+1)$ no. of functionally expanded values, weights for each individual pattern can be set accordingly (Eq. (3)).

$$W_i = \{w_{i,1}, w_{i,2}, \ldots w_{i,2n+1}\}, \text{for } i = 1 \text{ to } n \quad (3)$$

Here W_i is the weight vector initialized randomly for a single input pattern. Hence, initialization of weight for input patterns of dataset '*x*' can be viewed as a weight vector$W = W_1, W_2 \ldots W_m{}^T$. The dataset '*x*' is supplied to FLANN in terms of functionally expanded values and the net output is obtained as follows:

- First, S is calculated as $S = \varphi X W = s_1, s_2 \ldots s_m$
- Then, the net output Y is computed as $Y = f(S) = f(s_1), f(s_2) \ldots f(s_m) = \{y_1, y_2 \ldots y_m\} = \tanh(s_1), \tanh(s_2) \ldots \tanh(s_m)$. Here tanh is used as an activation function and net output y_i is for input pattern x_i.

Based on net output and given target value, error of FLANN is calculated and a suitable learning method is adapted to adjust weight values of FLANN. An efficient learning method, known as GDL has been presented in Section 4.2.

4.2. Gradient descent learning

Gradient descent learning is one of the most commonly used training method, in which weights are changed in such a way that network error is declined as rapidly as possible. The learning of FLANN model using Gradient descent method with error of the network is described below.

- Error of *k*th input pattern is generated as $e(k) = Y(k) - t(k)$ which is used to compute error term $\delta(k) = \left(\frac{1-y_k{}^2}{2}\right) \times e(k)$, for $k = 1,2 \ldots m$, where m is the number of input pattern in a dataset.
- Then, weight factor of 'ΔW' can be computed as $\Delta W_q = \left(\frac{\sum_{i=1}^{L} 2 \times \mu \times \varphi_i \times \delta_i}{L}\right)$, for $q = 1,2 \ldots L \times (2n+1)$ Here $\varphi = (\varphi_1, \varphi_2 \ldots \varphi_L)$, $e = (e_1, e_2 \ldots e_L)$ and $\delta = (\delta_1, \delta_2 \ldots \delta_L)$ are the vectors which represent set of functional expansions, set of errors and set of error terms respectively, where L is the number of input patterns.
- Finally, weight updation is done as $w_{new} = w + \Delta W$, where $w = (w_1, w_2 \ldots w_{L \times (2n+1)})$ and $\Delta W = (\Delta W_1, \Delta W_2 \ldots \Delta W_{L \times (2n+1)})$.

Basically, a good learning algorithm helps the ANN models for faster convergence. Further, a use of competitive optimization technique can not only improve the performance of a learning algorithm in terms of fast convergence, but also enhances the accuracy of an ANN based classifier.

4.3. Variants of harmony search

The harmony search (HS) [56] is a meta-heuristic algorithm inspired by musical process of searching for a perfect shape of harmony. The algorithm is based on natural musical processes in which a musician searches for a better state of harmony by the tuning pitch of each musical instrument, such as during jazz improvisation. Music improvisation by pitch adjustment is analogous to local and global search process to find better solution in any optimization techniques.

4.3.1. Harmony search

This section contains brief review on working procedure of the harmony search algorithm. In general, basic steps of harmony search can be expressed as follows:

Step 1 Initialize a harmony memory (HM) with randomly generated solution vectors (Harmonies).
Step 2 Repeat Steps 3 and 4 until no further significant growth in fitness of solution vector is noticed or the maximum no. of iterations is reached.
Step 3 Improvise HM to get new harmony memory (NHM).
Step 4 Update the HM based on comparison between solution vectors of HM and NHM in terms of fitness. If any harmony in HM is less fit than harmony in NHM, then harmony in HM will be excluded by adding harmony from NHM.
Step 5 Exit.

Basically, the HM is a group of pre-defined number of solution vectors similar to a population of particle in PSO or chromosome in GA. Initially HM is initialized with random solution vectors and gradually, solution vectors in HM are improved by using step-2 of harmony search procedure known as HM improvisation step. This step is entirely controlled by the parameters such as harmony memory consideration rate (HMCR), pitch adjustment rate (PAR) and bandwidth (bw).

In HS, HMCR is set between 0 and 1.The searching procedure behaves as purely random search if the HMCR is set to 0 and a value 1 for HMCR specifies 100% of previous solution vectors from HM are taken into consideration for next generation, which means, there is no chance to improve the harmony from outside the HM. In this way, HMCR keeps the balance between exploration and exploitation. Another parameter PAR determines the rate of adjustment of solution vectors based on bandwidth (bw) which is usually a variable which behaves as step size.

The HMCR and PAR determine memory consideration probability (MCP), pitch adjustment probability (PAP) and random probability (RP) as follows:

$$MCP = HMCR * (1 - PAR) * 100$$

$$PAP = HMCR * PAR * 100$$

$$RP = 100 - MCP - PAP$$

Basically, improvisation of HM is governed by these parameters (MCP, PAP, and RP).

Example: If HMCR=0.99 and PAR=0.45 then MCP=0.9*(1−0.45)*100=49.5 and PAP=0.9*0.45*100=40.5 and RP=100−49.5−40.5=10. Which means, during harmony improvisation phase (step-2), 49.5% of solution vectors from previous HM are migrated to new NHM, 40.5% of solution vectors are gone through

pitch adjustment and then included into NHM and 10% of solution vectors are gone through modification by adding randomly generated values with existing solution vector in HM.

In HS, the bw and PAR are fixed and pitch adjustment is done according to Eq. (4).

$$HM_i(t+1) = \begin{cases} HM_i(t+1) = HM_j(t) - rand() * bwif rand(1) < 0.5 \\ HM_i(t+1) = HM_j(t) + rand() * bwif rand(1) > 0.5 \end{cases} \quad (4)$$

In Eq. (4), '$HM_i(t+1)$' is the next ith harmony at time $(t+1)$ and '$HM_j(t)$' is the jth randomly selected harmony for pitch adjustment at time 't'.

In recent years, many HS variants (Table 2) have been proposed by the researchers by incorporating some modifications to the basic HS algorithm [56]. The major variants are: Improved HS and Self Adaptive HS and their hybridization with other optimization technique for better performance.

4.3.2. Improved harmony search

Improved harmony search (IHS) [57] is an initial variant of HS which employs a novel strategy for generation of new solution vectors that not only enhances accuracy but also improves the convergence rate of basic HS algorithm. The authors have claimed for better performance of IHS over HS by eliminating constant parameters (bw, PAR) in HS algorithm and incorporating dynamically changes in PAR and bw with iteration number. The IHS is free from the fixed values of PAR and bw by decreasing bw and increasing PAR with an iteration number and found to be considerable influence on the quality of solutions. The mechanism of dynamically decrement of bw with the iteration is inspired from the strategy of decreasing the learning rate of neural networks dynamically [87].

Unlike HS, the bw and PAR are not fixed and these values change according to HS iterations which can be achieved by using Eqs. (5) and (6).

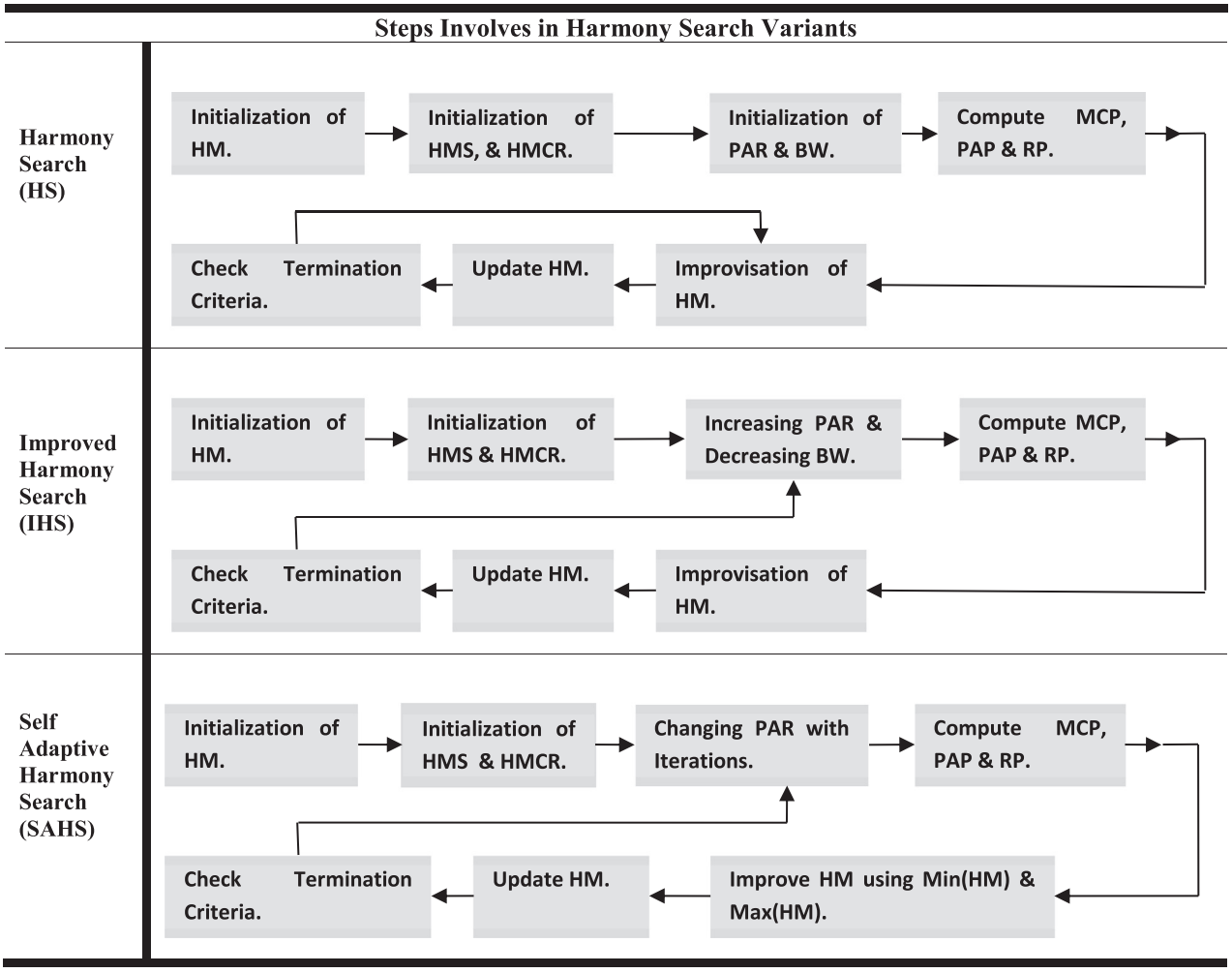
$$bw(iter) = bw_{max} \times exp\left(\frac{ln\frac{bw_{min}}{bw_{max}}}{N} \times iter\right) \quad (5)$$

In Eq. (5), '$bw(iter)$' is the bandwidth in particular iteration 'iter'.'bw_{min}'and'bw_{max}' are the minimum and maximum bandwidth respectively and 'N' is the number of solution vectors in the population.

$$PAR(iter) = PAR_{min} + \frac{PAR_{max} - PAR_{min}}{N} \times iter \quad (6)$$

In Eq. (6), '$PAR(iter)$' is the pitch adjustment rate in particular iteration 'iter'. The 'PAR_{min}' and 'PAR_{max}' are the minimum and maximum pitch adjustment rate and 'N' is the number of solution vectors in the population.

Table 2
Harmony search variants.



4.3.3. Self adaptive harmony search

Self adaptive harmony search (SAHS) [58] is a modification of IHS in order to make pitch adjustment step simpler. In SAHS, the pitch adjustment step in IHS have been modified to incorporate better utilization of its own experiences, by updating the new harmonies according to the maximum and minimum values in the HM. Here, the objective is to simplify pitch adjustment step by introducing a new strategy of adjusting new harmony by using maximum and minimum value in HM encountered so far, thereby eliminating bw altogether from HS procedure.

Like IHS, in SAHS, the PAR changes with HS iterations. The SAHS is different from IHS only in pitch adjustment mechanism as illustrated in Eq. (7). In SAHS, the HM are adjusted by the following Equations:

$$HM_i(t+1)=\begin{cases} HM_i(t+1)=HM_j(t)+[\max(HM)-HM_j(t)]\times rand() & if rand(1)<0.5 \\ HM_i(t+1)=HM_j(t)-[HM_j(t)-\min(HM)]\times rand() & if rand(1)>0.5 \end{cases} \tag{7}$$

Here '$HM_i(t+1)$' is the next ith harmony at time $t+1$, '$HM_j(t)$' is the jth randomly selected harmony at time 't', 'min(HM)' and 'max (HM)' are the minimum and maximum values of entire harmony memory (HM) respectively and 'rand(1)' is a uniform number in the [0, 1] range without 1.

5. Proposed method

In this work, a study has been made on the performance of harmony search algorithm and its different variants (i.e. HS, improved HS and self adaptive HS) and an attempt has been made to use the problem solving strategies of these variants to improve the performance of FLANN classifier. This study is mainly focused on self adaptive HS (SAHS) and its use to obtain better FLANN classifier. The problem solving strategy of self adaptive HS along with GDL to find best set of weights for FLANN model for classification task. The objective is to investigate the performances of SAHS to enhance classification accuracy of FLANN classifier as compared to GA, PSO, HS, IHS and teaching learning based optimization (TLBO) [89]. The pseudo codes developed during implementation of SAHS based FLANN with GDL (SAHS–FLANN) is presented in next section.

The simulation results and the comparison of performance of these hybrids FLANN classifiers (FLANN, GA based FLANN, PSO based FLANN, HS based FLANN, Improved HS based FLANN, TLBO based FLANN and Self-Adaptive HS based FLANN) are presented and discussed.

5.1. Self adaptive harmony search based gradient descent learning FLANN (SAHS–FLANN)

Initially, the HM is initialized with 'n' no. of weight-sets for FLANN. Each weight-set is a possible candidate set of weight of FLANN for classification of a particular dataset. Each individual weight-set in HM can be defined as:

$$W_i=(w_{i,1},w_{i,2}\ldots w_{m\times a\times(2\times k+1)}) \tag{8}$$

In Eq. (8), '$(2\times k+1)$'is the number of functionally expanded values for a single value in input pattern (for a chosen value of k), 'a' is the number of values (attributes) in a single input pattern; 'm' is the number of patterns in the dataset and 'n' is the number of weight-sets in the population HM. The set of weight-sets in the HM (Population) is represented in Eq. (9).

$$HM=(W_1,W_2\ldots W_n) \tag{9}$$

Each weight-set W_i is set to FLANN individually and the FLANN model is trained with a dataset. Based on output of the FLANN and given target value, the error of the network is obtained. For a specific dataset, the root mean square error (RMSE) (Eq. (10)) for each weight-set W_i is computed by using output of the FLANN and given target value (Algorithm 4). Based on RMSEs, fitness of the weight-sets is computed by using Eq. (11).

The Root Mean Square Error (RMSE) of predicted output values $\hat{y}_i$ of a target variable y_i is computed for n different predictions as follows:

$$\mathrm{RMSE}=\sqrt{\sum_{i=1}^{n}\frac{(y_i-\hat{y}_i)^2}{n}} \tag{10}$$

$$F_{W_i}=1/RMSE_i \tag{11}$$

In Eq. (11), W_i is the ith weight-set in the population, $RMSE_i$ is the root mean square error of ith weight-set and F_{W_i} is the fitness of ith weight-set W_i.

After evaluation of fitness values for each weight-sets in HM, the HM goes through HM improvisation process based on self adaptive harmony search. During this, the parameter HMS and HMCR are set and based on that MCP, PAP and RP are computed (Algorithm-1). Basically, harmony search procedure is governed by these parameters. Algorithm-2 represents pseudo codes for harmony memory improvisation process in which, initially best and worst harmony are fetched from HM based on maximum and minimum fitness respectively. After that, among all weight-sets (harmonies) in HM, some are selected randomly with a probability of MCP. Here the objective is to migrate some weight-sets (harmonies) from HM into NHM, which serve as new harmonies. For improvement in the weight-sets through pitch adjustment, some of the weight-sets are selected randomly from HM and then adjusted based on max(HM) and min(HM). The max(HM) and min(HM) are the weight-sets having highest and lowest fitness value respectively. With a probability of RP, some weight-sets are selected randomly and added to NHM by suitably adding or subtracting a random value on it.

After improvisation of HM, the HM is updated which is based on comparison of fitness of weight-sets in HM and NHM. If the fitness of ith weight-set in HM is less than fitness of ith weight-set in NHM, then HM (i,:) will be replaced by NHM(i,:) else HM(i,:) serves as new harmony for next iteration. The pseudo codes for HM updation procedure are represented in Algorithm-3. These processes are continued until maximum iteration is reached or increase in fitness of weight-sets in HM is not significant. The complete flow of execution can be realized in Fig. 2.

Algorithm – 1. Self-adaptive-harmony-search-FLANN(SAHS–FLANN) procedure.

```
% Initialization of HMS (Harmony memory size (Number of Har monies)), HMCR (Harmony memory consideration rate) and PAR (Pitch adjustment rate) .
    HMS=40;
    HMCR=0.9;
    PARmin=0.01;
    PARmax=0.9;
% Randomly initialize a harmony memory (HM) with size HMS.
HM=-1 + (1 - -1).*rand(HMS, LHM);% LHM is the length of each weight-sets (harmonies).
Iter=0;
While (1)
    % Improvisation of harmony
    PAR=PARmin+((PARmax-PARmin)/HMS)*iter;% Changes of PAR in Iterations.
    % Compute MCP(memory consideration probability), PAP(pitch adjustment probability) and RP(randomization probability).
    MCP=HMCR*(1-PAR)*100;
    PAP=HMCR*PAR*100;
```

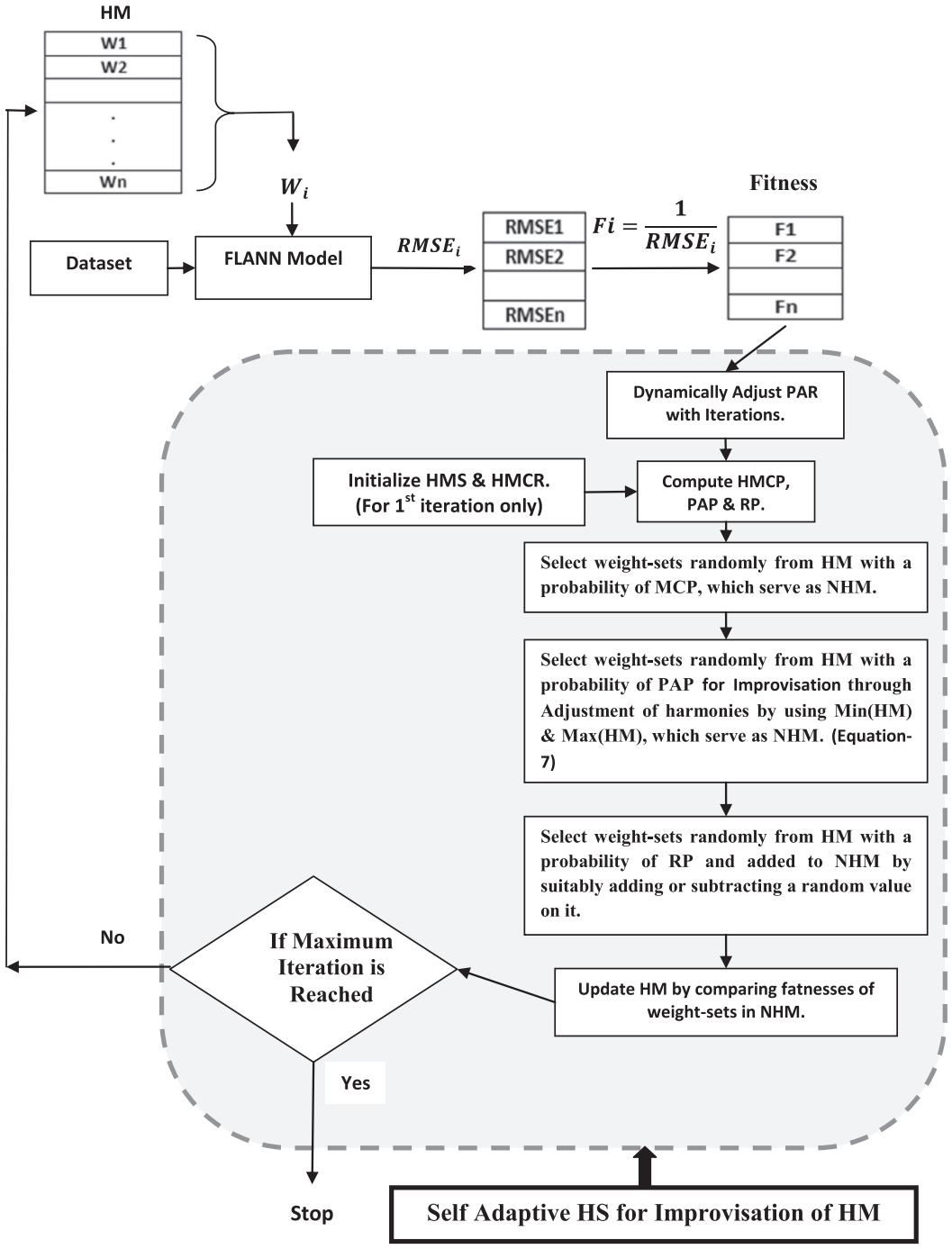


Fig. 2. Overview of proposed scheme.

```
    RP=100-MCP-PAP;
    NHM=ImprovizationOfHarmonyMemory (HM, HMS, MCP ,
    PAP, RP);
    % Updation of HM
    HM=UpdateHarmonyMemory(HM,NHM);
    % Check for termination criteria.
    if (iter > =MAX_ITERATION)
        break;
    end if
    iter=iter+1;
End While
```

Algorithm – 2. ImprovizationOfHarmonyMemory procedure.

```
Function NHM=ImprovizationOfHarmonyMemory (HM,
    HMS, MCP , PAP, RP)
        % Find out harmony with minimum and maximum
    fitness from HM and term as HMmin and HMmax respectively.
        for i=1:1:HMS
            w=HM(i,:);
            F(i,1)=fitfromtrain(φ, w, t, μ);
```

```
        end
        [mx,mxi]=max(F);
        HMmax=HM(mxi,:);
        [mn,mni]=min(Fhs);
        HMmin=HM(mni,:);

        for i=1:1:HMS
            r=rand(1)*100;
            % Select jth weight-set randomly from HM with a
            probability of MCP which serve as NHM.
            If (1<=r && r<=MCP)
                j=f loor ( mod((rand(1)*1000),
                HMS ) )+1;
                NHM(i,:)=HM(j,:);
            Endif
            % Select jth weight-set randomly from HM with a
            probability of PAP for the pitch adjustment to
            improve quality of weight-sets in HM which
            serves as NHM.
            if(MCP+1<=r && r<=MCP+PAP)
                    j=floor(mod((rand(1)*1000),
                    HMS))+1;
                    r1=rand(1);
                    if(r1<=0.5)
                            for k=1:1:lhm,
                            here lhm is the
                            length of HM.
                                NHM(i,k)=HM
(j,k)+(HMmax(1,k)-HM(j,k))*rand(1);
                            end
                    else
                            for k=1:1:lhm
                                NHM(i,k)=HM
(j,k)-(HM(j,k)-HMmin(1,k))*rand(1);
                            end
                    end
            end
            % Select jth weight-set randomly from harmony
        memory with a probability of RP which serve as NHM. In this
        phase, a jth weight-set is selected randomly from HM and added
        to NHM by suitably adding or subtracting a random value on it.
            If (MCP+PAP+1<=r && r<= MCP+PAP
+RP)
                    j=floor(mod((rand(1)*1000),
HMS))+1;
                    NHM(i,:)=HM(j,:)+(-0.1+(0.1
- -0.1).*rand(1));
            endif
        endfor
end
```

Algorithm – 3. UpdateHarmonyMemory Procedure.

```
Function HM= UpdateHarmonyMemory (HM,NHM)
        % If the new harmony (weight-sets) in NHM is
    better than the harmony in the HM, then add the new harmony
    into the HM by excluding the worst harmony from the HM.
        for i=1:1:HMS
            w=HM(i,:);
            F1(i,1)=fitfromtrain (φ,w,t, μ);
        endfor
        for i=1:1:HMS
            w=NHM(i,:);
            F2(i,1)=fitfromtrain (φ,w,t, μ);
        endfor
        for i=1:1:lf, here lf is the length of F1 or F2
            if (F1(i,1)<F2(i,1))
                    HM(i,:)=NHM(i,:);
            end if
        endfor
end
```

Algorithm – 4. Fitfromtrain Procedure.

Function F=**fitfromtrain** (φ, w, t, μ)

S=φ. w; here 'φ' is the functionally expanded input data , 'x_i' be the selected weight-set from the population and 'S' is a vector.

Then the output vector 'y' is computed by using tanh activation function as y=tanh(S).

The errors of the network 'e' is computed by using target vector' t ' and output vector ' y' as e=t−y;

If $\varphi=(\varphi_1,\varphi_2\ldots\varphi_L)$, $e=(e_1,e_2\ldots e_L)$ and $\delta=(\delta_1,\delta_2\ldots\delta_L)$ are the vectors which represent set of functional expansions, set of errors and set of error respectively, then weight factor of w 'ΔW' is Computed as follows: $\Delta W_q=\left(\frac{\sum_{i=1}^{L}2\times\mu\times\varphi_i\times\delta_i}{L}\right)$.

Compute root mean square error (RMSE) by using Eq. (10) *from target vector 't' and output vector 'y'.*

.Compute fitness 'F' of the network instance of FLANN model as F=1/RMSE. (Eq. (11))

Finally, weight factor 'ΔW' is used to update the weights.

. end

6. Experimental setup

In this section, the environment for simulation, the dataset used for training and testing phase and parameter settings for proposed method during simulation are presented.

All the classification methods (FLANN, GA based FLANN, PSO based FLANN, HS based FLANN, IHS based FLANN, TLBO based FLANN and SAHS–FLANN) are implemented in MATLAB (Version 9.0) in a system with Window XP operating system. After obtaining the results of simulation, statistical analysis has been carried out for statistical correctness of results. The benchmark datasets (Table 3) used for classification are originated from UCI machine learning repository [90] and processed by KEEL software [91]. The detail descriptions about all these dataset can be obtained at 'http://archive.ics.uci.edu/ml/' and 'http://keel.es/'.

6.1. Parameter setting used for simulation

6.1.1. FLANN parameter settings

During the learning of FLANN model, the GDL method is used by setting 'μ' to 0.13. The value of 'μ' is obtained by testing the models in the range 0–3. Each value in the input pattern is expanded to 11 no. of functionally expanded input values by setting $n=5$. (As FLANN model suggests to generate $2n+1$ no. of functionally expanded input values for a single value in the input pattern).

6.1.2. Harmony search algorithms' parameter settings

The improvisation step in HS is entirely controlled by parameters: HMS, HMCR, PAR and bw. The HMS is usually same as population size in any meta-heuristic algorithm like GA, PSO etc. The value of HMS is set by the user depending upon the problem being solved. The value of HMCR is set between 0 to 1. The searching procedure behaves as purely random search if the HMCR is set to 0, which means, no previous harmonies in harmony memory is included in

Table 3
Data set information.

Dataset	Number of pattern	Number of features (excluding class label)	Number of classes
Monk 2	256	06	02
Iris	150	04	03
Heart	256	13	02
Hayesroth	160	04	03
Wine	178	13	03
Ionosphere	351	33	02
Hepatitis	80	19	02
Pima	768	08	02
New Thyroid	215	05	03
Bupa	345	06	02
Dermatology	256	34	06

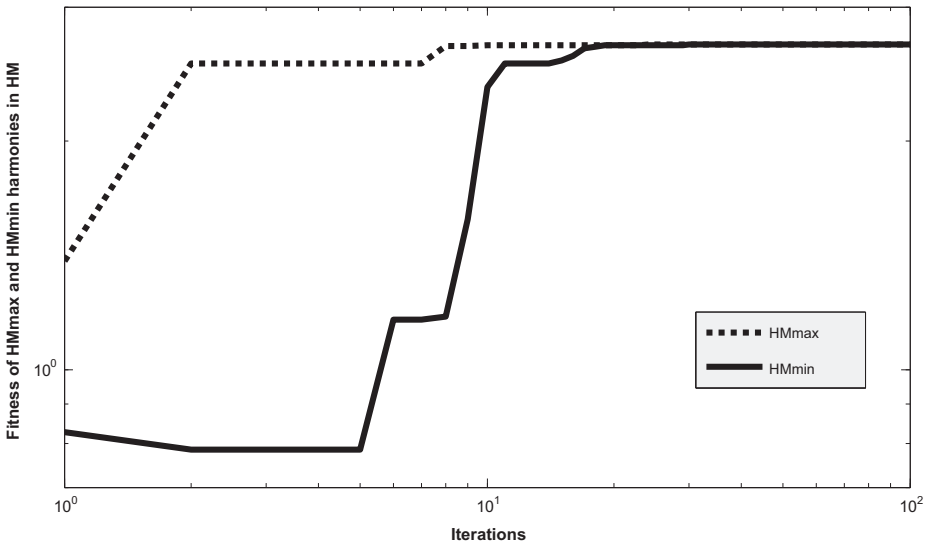


Fig. 3. Changes in HMmax and HMmin in various iterations in SAHS–FLANN.

new harmony memory. A value of 1 for HMCR specifies 100% of previous solution vectors from HM are taken into consideration for next generation, which means, there is no chance to improve the harmony from outside the HM. In this way, HMCR keeps the balance between exploration and exploitation. Another parameter PAR determines rate of adjustment of solution vectors based on bw, which is usually a variable which behaves as step size.

The HS suggests to set 0.9, 0.3 and 0.01 for the parameters HMCR, PAR and bw respectively [56]. This value of HMCR and PAR determines MCP, PAP and RP as follows: MCP=HMCR*(1−PAR) *100; PAP=HMCR*PAR*100 and RP=100−MCP−PAP. Basically, Improvisation of HM is governed by these parameters (MCP, PAP and RP). Example: if HMCR=0.9 and PAR=0.3 then MCP=0.9*(1−0.3)*100=63 and PAP=0.9*0.3*100=27 and RP=100−63−27=10. Which means, during harmony improvisation phase (step-2), from previous HM, 63% of solution vectors are migrated to NHM without any changes, 27% of solution vectors are gone through pitch adjustment and then included into NHM and 10% of solution vectors are gone through modification by adding randomly generated values with existing solution vector in HM.

In IHS [57], the value for parameters HMS and HMCR are set similar to HS. Whereas, the parameters: PAR and bw are not fixed. It is advised to set 0.9, 0.01, 0.99, 0.0001 and $\frac{1}{20\times(UB-LB)}$ for HMCR, PARmin , PARmax , BWmin and BWmax respectively.

In SAHS [58], the parameters HMS and HMCR are set similar to HS and IHS, whereas the parameter bw is completely removed. It avoids pitch adjustments through bw by incorporating better utilization of its own experiences, by updating the new harmonies according to the maximum (HMmax) and minimum (HMmin) values in the HM. It is recommended to set 0.99, 0, and 1 for HMCR, PARmin and PARmax respectively. Values of HMmax and HMmin changes in various iterations are depending on the current harmonies in harmony memory. The changes in fitness of HMmax and HMmin in various iterations of the proposed method are noticed and presented in Fig. 3.

In this study, we have set all the parameters according to the suggested values during simulations. The set of parameter settings in this study during the experiments are listed in Table 4.

7. Results obtained

In this section, the obtained classification accuracies (Eq. (12)) of different methods for various benchmark datasets are presented and a comparison has been made among them. These classification accuracies (Tables 6 and 7) are obtained individually for training and testing phase. In Eq. (12), CM is the confusion matrix which represents the number of well classified and miss-classified patterns after the classification operation and n is the number of class. In the confusion matrix CM, all diagonal elements represent total number of well classified patterns and non diagonal elements represent number of miss-classified patterns under each class.

$$Classification\ Accuracy = \frac{\sum_{i=1}^{n}\sum_{j=1,\, i==j}^{n} CM_{i,j}}{\sum_{i=1}^{n}\sum_{j=1}^{n} CM_{i,j}} \times 100\,\% \qquad (12)$$

Table 4
Harmony search algorithms' parameter settings.

Methods	HMS	HMCR	PAR	PAR_{min}	PAR_{max}	Bw	Bw_{min}	Bw_{max}
HS–FLANN	40	0.9	0.3	–	–	0.0001	–	–
IHS–FLANN	40	0.9	–	0.01	0.9	–	0.0001	$\frac{1}{20\times(UB-LB)}$
SAHS–FLANN	40	0.9	–	0.01	0.9	–	–	–

Table 5
Datasets in 5-fold for cross validation.

Dataset	*Data files*	*Number of pattern*	*Task*	*Number of pattern in class-1*	*Number of pattern in class-2*	*Number of pattern in class-3*
New Thyroid	newthyroid-5-1tra.dat	172	Training	120	28	24
	newthyroid-5-1tst.dat	43	Testing	30	07	06
	newthyroid-5-2tra.dat	172	Training	120	28	24
	newthyroid-5-2tst.dat	43	Testing	30	07	06
	newthyroid-5-3tra.dat	172	Training	120	28	24
	newthyroid-5-3tst.dat	43	Testing	30	07	06
	newthyroid-5-4tra.dat	172	Training	120	28	24
	newthyroid-5-4tst.dat	43	Testing	30	07	06
	newthyroid-5-5tra.dat	172	Training	120	28	24
	newthyroid-5-5tst.dat	43	Testing	30	07	06

Table 6
Comparison of results among FLANN, GA–FLANN, PSO–FLANN and HS–FLANN.

Datasets	Classification accuracies of classifiers in %							
	FLANN		GA–FLANN		PSO–FLANN		HS–FLANN	
	Training	Testing	Training	Testing	Training	Testing	Training	Testing
Monk 2	93.828	92.043	96.545	93.199	97.453	95.466	97.914	96.537
Iris	96.847	97.368	97.13	98.166	97.352	98.65	97.857	99.472
Heart	88.963	78.481	89.407	79.074	89.778	79.852	89.917	80.222
Hayesroth	90.359	82.313	91.063	83.562	91.266	83.937	91.547	85.063
Wine	92.76	93.186	94.368	95.536	97.762	95.627	97.597	95.570
Ionosphere	79.482	80.927	87.336	89.152	92.372	90.18	91.552	90.069
Hepatitis	73.519	70.593	80.275	75.826	80.028	75.42	82.481	76.273
Pima	78.416	78.76	78.64	78.80	80.126	79.47	80.683	80.581
Thyroid	93.918	76.558	94.198	77.535	94.302	78.791	94.407	79.256
Bupa	72.16	72.76	74.321	75.5	76.384	76.75	76.318	76.358
Dermatology	96.358	92.442	96.946	93.859	97.011	94.08	97	93.872

In this paper, all the dataset used for classification are prepared for cross validation by using 5-folds cross validation technique [92,93]. During the preparation of datasets for 5-fold cross validation, 5 pairs of dataset sample are created and each pair contains datasets for training and testing phase. For example (Table 5), the 'newthyroid-5-1tra.dat' and 'newthyroid-5-1tst.dat' data are a pair of datasets sample of New Thyroid dataset which is used for training and testing phase for a single run respectively. As 5-fold cross validation technique is employed, the New Thyroid dataset contains 5 such pair of dataset sample for training and testing the algorithms. The 5-fold cross validated dataset for New Thyroid dataset is presented in Table 5. All other datasets are prepared for 5-fold cross validation in the same way and collected from KEEL dataset repository. The average classification accuracies on 5-fold cross validation dataset during training and testing phase are listed in Tables 6 and 7. In these tables, the average of classification accuracies of the algorithm on 'newthyroid-5-1tra.dat', 'newthyroid-5-2tra.dat', 'newthyroid-5-3tra.dat', 'newthyroid-5-4tra.dat' and 'newthyroid-5-5tra.dat' are presented as the classification accuracy in training phase for New Thyroid dataset. Similarly, the average of classification accuracies of the algorithms on 'newthyroid-5-1tst.dat', 'newthyroid-5-2tst.dat', 'newthyroid-5-3tst.dat', 'newthyroid-5-4tst.dat' and 'newthyroid-5-5tst.dat' are presented as the classification accuracy in testing phase.

Table 6 describes the comparison of classification accuracies of FLANN, GA based FLANN (GA–FLANN), PSO based FLANN (PSO–FLANN) and HS based FLANN (HS–FLANN) classifiers and Table 7 represents comparison of other 4 classifiers: HS–FLANN, IHS–FLANN , TLBO–FLANN and SAHS–FLANN. Also we have compared the results of the proposed method with recently developed meta-heuristic algorithm (teaching learning based optimization [88]) based FLANN model. In this study, the performance of GA, PSO, HS, IHS, TLBO and SAHS are analyzed in order to know the improvement of harmonies (weight-sets) in the population by these algorithms in different generation. The changes in fitness of weight-sets in different generations are observed in all the 11 no. of datasets and the Figs. 4–14 demonstrate the improvements of fitness of weight-sets in the population.

Also few comparisons are made on these methods based on their learning capabilities from training phase. All the methods are trained with various datasets and ability of these learning models are studied in terms of RMSE and fitness of the best resultant weight-sets. Table 8 compares all the hybrid models based on fitness values of the weight-sets obtained for Monk 2, Iris, Heart, Hayesroth, Wine and Ionosphere datasets. Similarly Table 9 compares these hybrid models based on fitness value on Hepatitis, Pima, New Thyroid, Bupa and Dermatology datasets. Finally, all the

Table 7
Comparison of results among HS–FLANN, IHS–FLANN, TLBO–FLANN and SAHS–FLANN.

Datasets	Classification accuracies of classifiers in %							
	HS–FLANN		IHS–FLANN		TLBO–FLANN		SAHS–FLANN	
	Training	Testing	Training	Testing	Training	Testing	Training	Testing
Monk 2	97.914	96.537	97.929	96.552	98.209	97.382	98	96.634
Iris	97.857	99.472	97.871	99.695	97.861	99.493	97.869	99.541
Heart	89.917	80.222	89.924	80.275	89.826	79.891	89.932	80.295
Hayesroth	91.547	85.063	91.557	85.193	91.825	85.523	91.602	85.26
Wine	97.597	95.570	97.902	95.63	97.915	95.622	97.927	95.783
Ionosphere	91.552	90.069	91.893	90.173	92.517	90.284	92.735	90.672
Hepatitis	82.481	76.273	82.638	76.334	82.577	76.29	82.533	76.294
Pima	80.683	80.581	80.835	80.593	81.0	80.794	80.738	80.587
Thyroid	94.407	79.256	94.437	79.263	94.413	79.26	94.426	79.261
Bupa	76.318	76.358	76.475	76.925	76.6	77.083	76.618	77.426
Dermatology	97	93.872	97.046	94.382	97.138	94.55	97.176	94.762

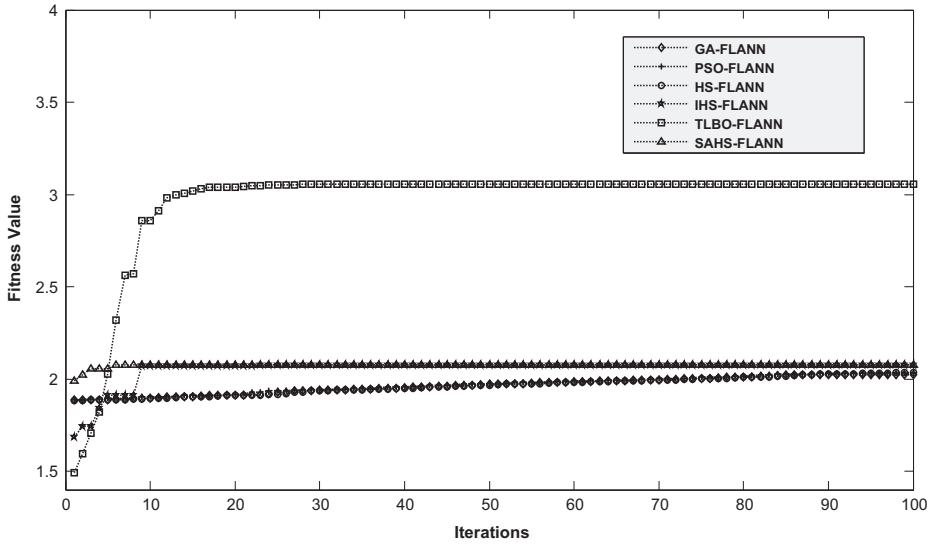


Fig. 4. Improvements in fitness of population in different iterations observed in MONK2 Dataset.

hybrid models based on the average RMSE obtained on all 11 no.s of datasets (Table 10).

8. Proof of statistical significance and correctness

In this section, the statistical comparison of classifiers over multiple data sets [94] is presented to argue the projected method is statistically better and significantly different from other alternative classifiers by using Friedman test [95,96]. List of datasets on which these tests have been carried out are presented in Table 11.

The Friedman test is a non-parametric statistical method which computes average ranks of algorithms by using Eq. (13) and compares them. In Table 11, the ranks of each classifier on various datasets are shown in brackets. Based on r_i^j, the average ranks of seven classifiers is found from Eq. (13).

$$R_j = \frac{1}{N}\sum_i r_i^j \tag{13}$$

The average ranks for all classifiers are found as follows:

$\{R_1 = 7,\quad R_2 = 5.909,\quad R_3 = 4.636,\quad R_4 = 4.273,$
$R_5 = 2.364,\quad R_6 = 2.091,\quad R_7 = 1.727\}.$

The ${X_F}^2$ value is computed from average rank R_j of each classifier by using Eq. (14). In this study, we got the value of ${X_F}^2$ as 58. From the value of ${X_F}^2$, the Friedman statistics F_F is computed by Eq. (15) and found as 72.5. In our case, for the 7 no. of classifiers and 11 no. of datasets, the $F_F = 72.5$ with $7-1=6$ and $(7-1)^*(11-1)=60$ degrees of freedom and the crucial value $=3.12$ is obtained from suitably selecting $\alpha = 0.01$.

The null-hypothesis (H_0) is clearly rejected as critical value 3.12 is less than F_F statistic (72.5).

H_0 : All the classifier has same rank, hence they are equivalent.

$$X_F^2 = (12N/k(k+1))\left(\sum_j R_j^2 - \frac{k(k+1)^2}{4}\right) \tag{14}$$

$$F_F = \left((N-1)X_F^2\right)/\left(N(K-1)-X_F^2\right) \tag{15}$$

After the rejection of null-hypothesis from Friedman test, to evaluate performance by pair wise comparison of proposed classifier with another classifier based on z-score value and *p*-value, the post-hoc test has been carried out by using the Holm procedure [94,97,98]. By using Holm [99] and Hochberg procedure [100], the *z*-value is obtained from the Eq. (16) and based on *z*-value, *p*-value is computed from the table of the normal distribution.

$$z = (R_i - R_j)/\sqrt{\frac{k(k+1)}{6N}} \tag{16}$$

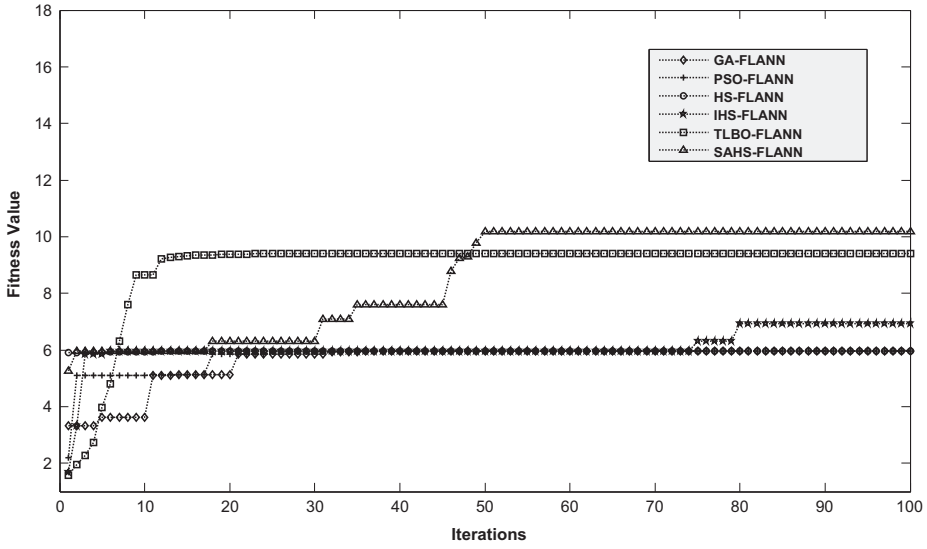


Fig. 5. Improvements in fitness of population in different iterations observed in IRIS dataset.

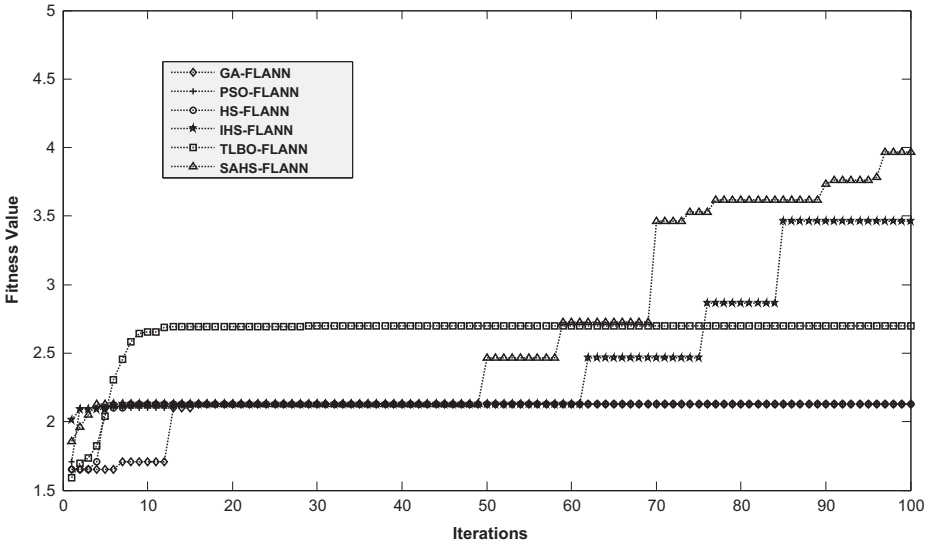


Fig. 6. Improvements in fitness of population in different iterations observed in HEART dataset.

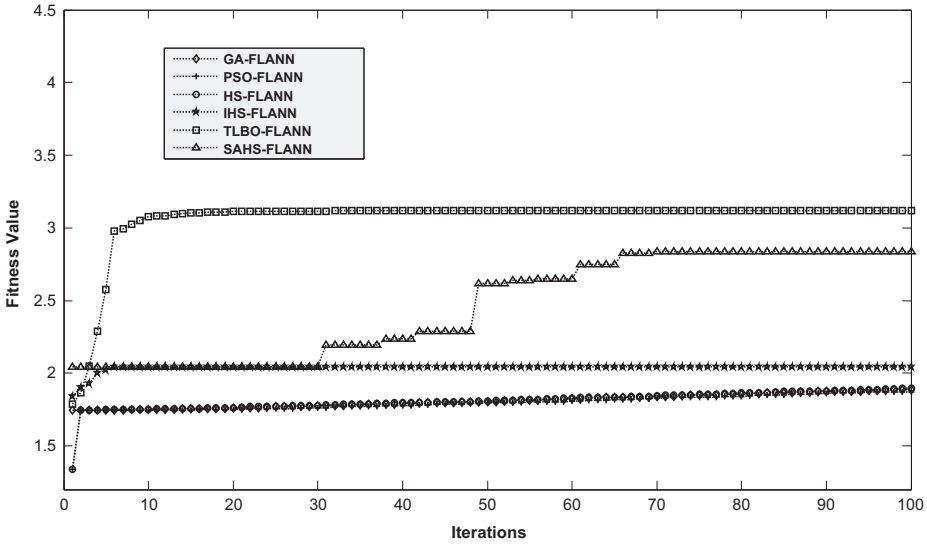


Fig. 7. Improvements in fitness of population in different iterations observed in HAYESROTH dataset.

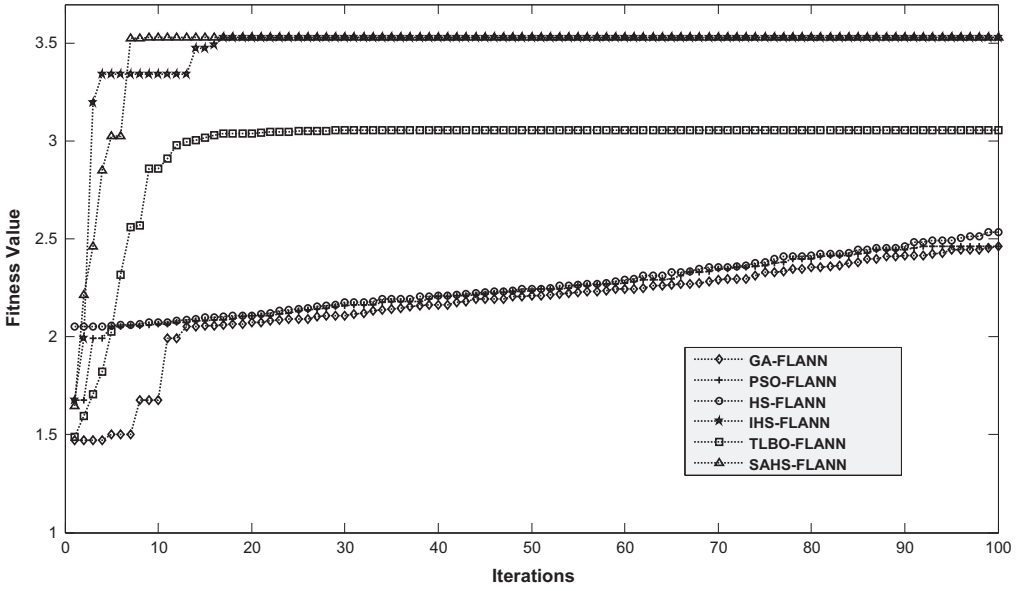


Fig. 8. Improvements in fitness of population in different iterations observed in WINE dataset.

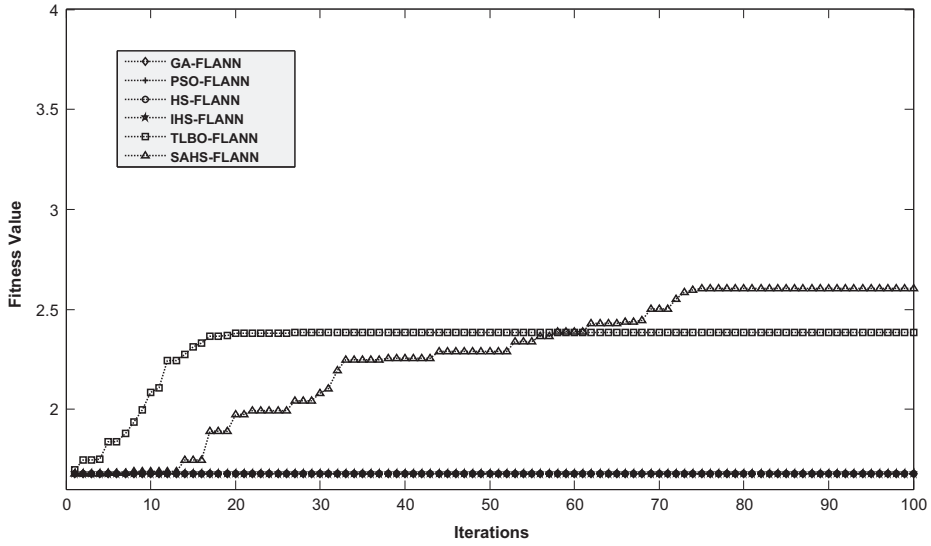


Fig. 9. Improvements in fitness of population in different iterations observed in IONOSPHERE dataset.

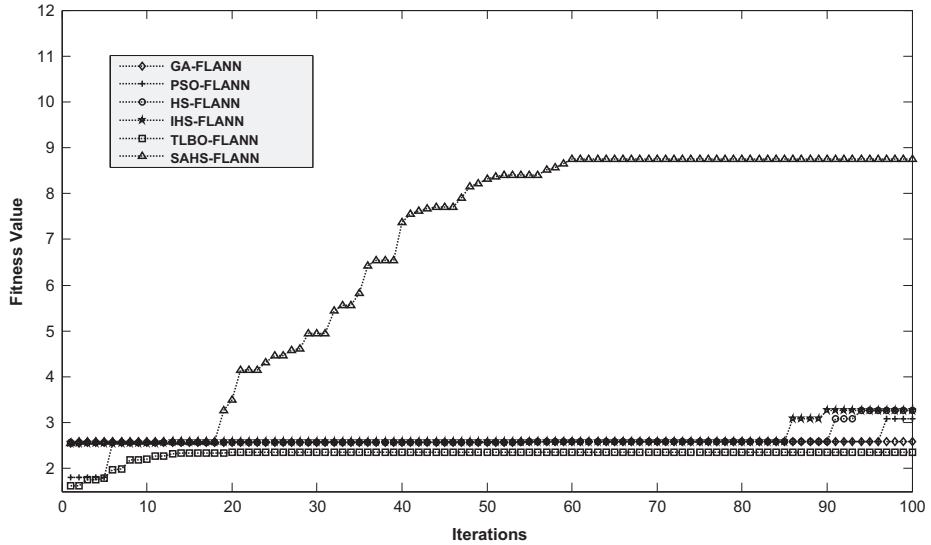


Fig. 10. Improvements in fitness of population in different iterations observed in HEPATITIS dataset.

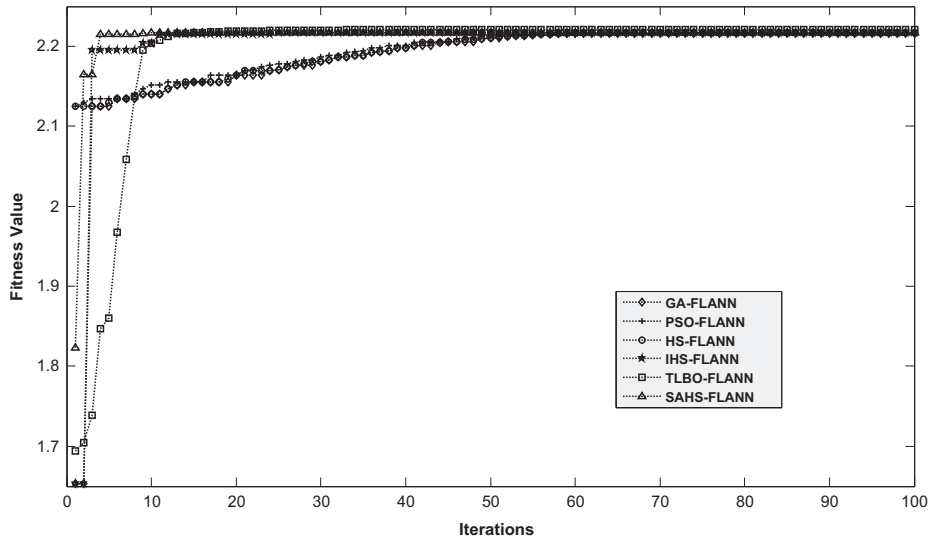


Fig. 11. Improvements in fitness of population in different iterations observed in PIMA dataset.

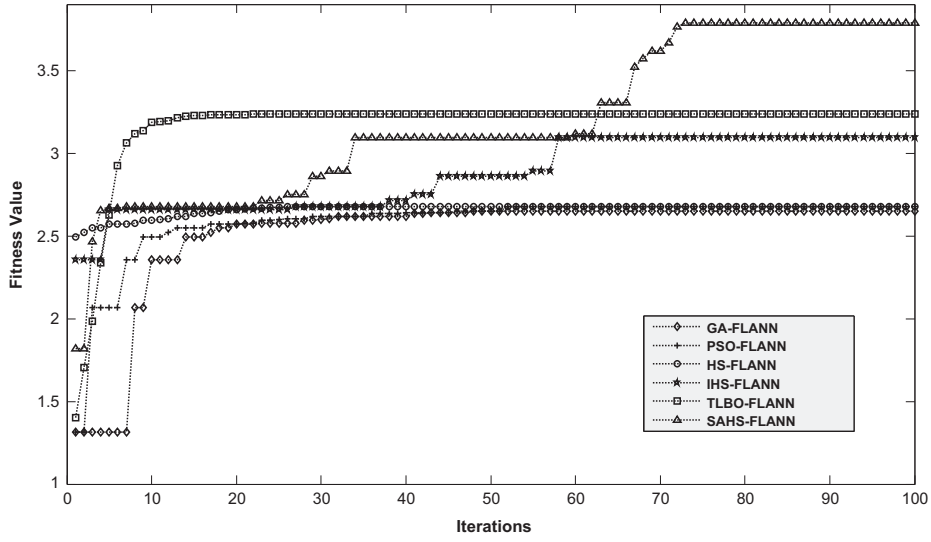


Fig. 12. Improvements in fitness of population in different iterations observed in NEW THYROID dataset.

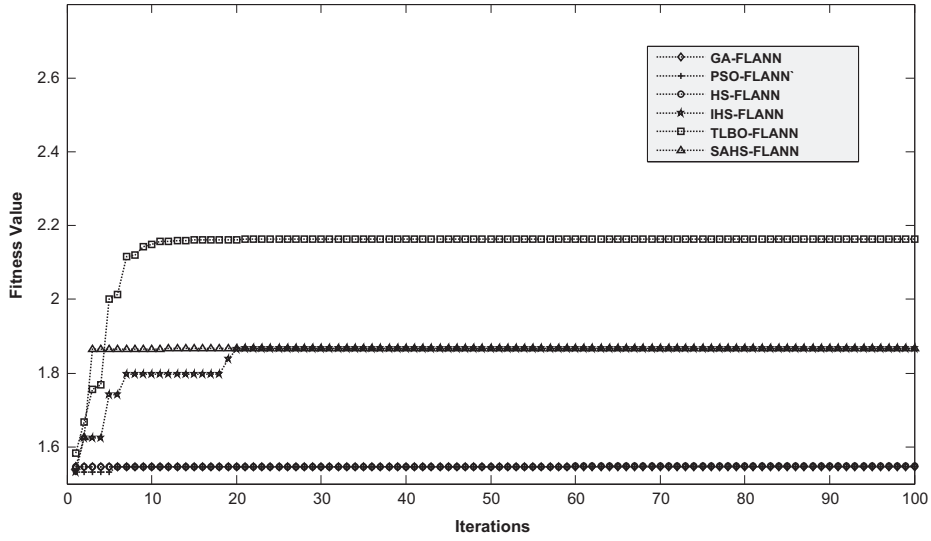


Fig. 13. Improvements in fitness of population in different iterations observed in BUPA dataset.

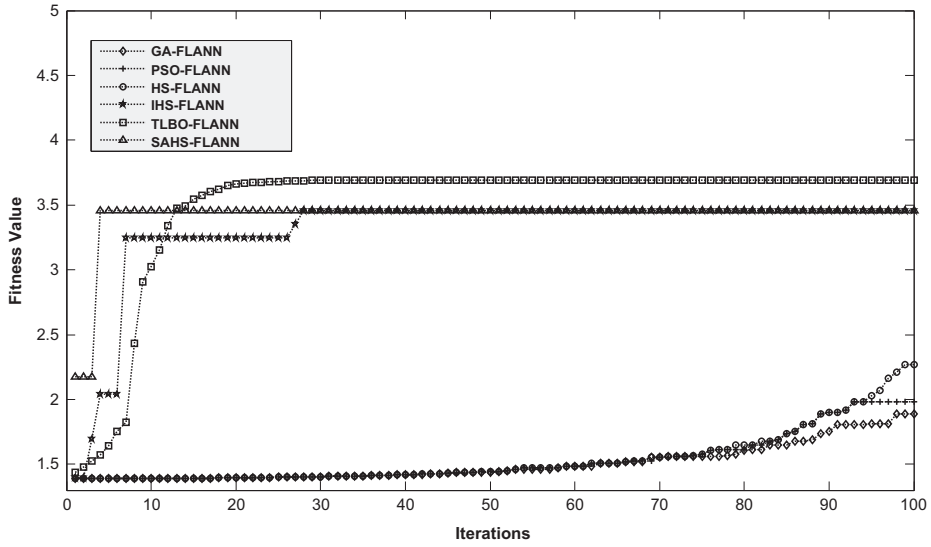


Fig. 14. Improvements in fitness of population in different iterations observed in DERMATOLOGY dataset.

Table 8
Comparison of hybrid models based on fitness value on Monk 2, Iris, Heart, Hayesroth, Wine and Ionosphere datasets.

Datasets	Fitness values obtained by various hybrid models on datasets					
	Monk 2	Iris	Heart	Hayesroth	Wine	Ionosphere
GA–FLANN	2.024735	5.972679	2.128121	1.888252	2.462398	1.677268
PSO–FLANN	2.033361	5.97268	2.127275	1.869618	2.462398	1.67727
HS–FLANN	2.037116	5.97268	2.128124	1.895258	2.534043	1.67727
IHS–FLANN	2.075259	6.937149	3.46403	2.045696	3.530592	1.67727
TLBO–FLANN	3.057657	9.411738	2.696475	3.118014	3.057658	2.382388
SAHS–FLANN	2.075259	10.18941	3.966872	2.838424	3.530592	2.602899

Table 9
Comparison of hybrid models based on fitness value on Hepatitis, Pima, New Thyroid, Bupa and Dermatology datasets.

Datasets	Fitness values obtained by various hybrid models on datasets				
	Hepatitis	Pima	New Thyroid	Bupa	Dermatology
GA–FLANN	2.58331	2.216361	2.651259	1.54692	1.888967
PSO–FLANN	3.090341	2.216474	2.675806	1.547116	1.981951
HS–FLANN	3.276094	2.217241	2.675806	1.548419	2.267907
IHS–FLANN	3.276094	2.217241	3.093559	1.86651	3.459351
TLBO–FLANN	2.349537	2.220777	3.238955	2.162731	3.690652
SAHS–FLANN	8.754259	2.217241	3.788515	1.86651	3.459203

Table 10
Comparison among hybrid models based on average RMSE on all 11 no. of datasets.

	Various hybrid models					
	GA–FLANN	PSO–FLANN	HS–FLANN	IHS–FLANN	TLBO–FLANN	SAHS–FLANN
Average RMSE on 11 datasets	0.459494	0.451444	0.442128	0.380665	0.344513	0.318715

Here z is the z-score value, k is the number of classifiers, N is the number of datasets and R_i and R_j are average rank of ith and jth classifier respectively. Table 12 presents comparison of results of all 7 classifiers based on z-value, p-value and $\frac{\alpha}{(k-i)}$, where 'i' is the classifier's number. By using the Holm test, when we have compared the p_i – values with $\frac{\alpha}{(k-i)}$ values, it is observed that, almost in all the cases p_i is less than $\frac{\alpha}{(k-i)}$ (except TLBO– FLANN and IHS–FLANN). Hence, it is clear that the null-hypothesis is rejected. Thus, the proposed classifier 'SAHS–FLANN' is statistically better and significantly different from other classifiers (except TLBO–FLANN and IHS–FLANN). While comparisons with TLBO–FLANN and IHS–FLANN, the SAHS–FLANN is found better than TLBO–FLANN and IHS–FLANN in performance but not significantly better than them. Out of seven classifiers, the null-hypothesis is rejected for five; hence the null-hypothesis can be clearly rejected on majority basis.

Table 11
Ranks of classifiers on various dataset based on the classification accuracy on training and testing set.

Datasets	Average classification accuracies of classifiers in %						
	FLANN	GA–FLANN	PSO–FLANN	HS–FLANN	IHS–FLANN	TLBO–FLANN	SAHS–LANN
Monk 2	92.9355 (7)	94.872 (6)	96.4595 (5)	97.2255 (4)	97.2405 (3)	97.7955 (1)	97.317 (2)
Iris	97.1075 (7)	97.648 (6)	98.001 (5)	98.6645 (4)	98.783 (1)	98.677 (3)	98.705 (2)
Heart	83.722 (7)	84.2405 (6)	84.815 (5)	85.0695 (3)	85.0995 (2)	84.8585 (4)	85.1135 (1)
Hayesroth	86.336 (7)	87.3125 (6)	87.6015 (5)	88.305 (4)	88.375 (3)	88.674 (1)	88.431 (2)
Wine	92.973 (7)	94.952 (6)	96.6945 (4)	96.5835 (5)	96.766 (3)	96.7685 (2)	96.855 (1)
Ionosphere	80.2045 (7)	88.244 (6)	91.276 (3)	90.8105 (5)	91.033 (4)	91.4 (2)	91.7035 (1)
Hepatitis	72.056 (7)	78.0505 (5)	77.724 (6)	79.377 (4)	79.486 (1)	79.4335 (2)	79.4135 (3)
Pima	78.588 (7)	78.72 (6)	79.798 (5)	80.632 (4)	80.714 (2)	80.897 (1)	80.6625 (3)
Thyroid	85.238 (7)	85.8665 (6)	86.5465 (5)	86.8315 (4)	86.85 (1)	86.8365 (3)	86.8435 (2)
Bupa	72.46 (7)	74.9105 (6)	76.567 (4)	76.338 (5)	76.7 (3)	76.8415 (2)	77.022 (1)
Dermatology	94.4 (7)	95.4025 (6)	95.5455 (4)	95.436 (5)	95.714 (3)	95.844 (2)	95.969 (1)
Friedman's rank in average	**7**	**5.909091**	**4.636364**	**4.272727**	**2.363636**	**2.090909**	**1.727273**

Table 12
Result of Holm and Hochberg procedure.

i	*Classifiers*	*z-values*	*p-values*	$\frac{\alpha}{(k-i)}$
1	SAHS–FLANN : FLANN	5.7243	5.200583e-9	0.0016666
2	SAHS–FLANN : GA–FLANN	4.54	0.000003	0.002
3	SAHS–FLANN : PSO–FLANN	3.1582	0.000794	0.0025
4	SAHS–FLANN : HS–FLANN	2.763	0.002864	0.0033
5	SAHS–FLANN : IHS–FLANN	0.6908	0.244846	0.005
6	SAHS–FLANN : TLBO–FLANN	0.394	0.34679	0.01

9. Conclusion

This paper has introduced a self adaptive harmony search based learning scheme for FLANN model for non-liner data classification. The main contribution of the proposed scheme which differs from earlier works are: i) it avoids the use of constant parameters (PAR and bw) in harmony search algorithm based learning scheme (HS–FLANN) while adjusting weights and ii) it incorporates better utilization of its own experiences, by updating the new weight-sets according to the maximum and minimum values in the HM, thereby eliminating bw altogether from improve harmony search based learning scheme (IHS–FLANN). This automatic adjustment method for weight-set avoids intricacy of parameter settings and also improves the qualities of weight-sets. A set of experiments are carefully conducted by setting suggested values to control parameters. Also we have investigated the impact of these control parameters to obtain a better set of values. The proposed SAHS–FLANN is analyzed under null-hypothesis by using various statistical methods like Friedman test, Holm and Hochberg procedure. As a result, a clear rejection of null-hypothesis is noticed in Friedman test and Holm and Hochberg procedure. Further, the performance of the SAHS–FLANN can be improved by fine-tuning of various parameters like HMCR, PAR and bw appropriately and can be tested over other set of benchmark real life datasets. From rigorous tests under various statistical methods (Friedman test, Holm and Hochberg procedure), we claim for the proposed SAHS–FLANN is better and outperforms other alternatives. Also the proposed method can be computed with a low cost due to less complex architecture of FLANN and self adaptive HS requires less mathematical computation and is free from complicated operators.

However, we have observed few pitfalls in the proposed method. In the proposed scheme, the sine and cosine trigonometric functions are used for functional expansion of input data. It effectively enlarges the dimension of the input vectors and creates enhanced feature space which provides better discriminating ability of the input patterns of the selected datasets. However, a particular mathematical function considered for function expansion may not always be fit to the entire dataset. In such cases, selection of suitable function from range of mathematical functions for expansion of the input data plays considerably an important role. As the dimensions of many real life problems are high, the dimensional enlargement to a huge extent is not appropriate. Hence, it is advisable to choose a small set of alternative functions, which can map the function to the desired extent with significant improvement in the output. Limitations of the proposed method lie in functional expansion mechanism in case of datasets with large dimension. Presently we are on an experimental vision for the proposed method with further difficult problems, particularly in higher dimensions. The future work is comprised of integration of other new and improved variants of SAHS with other higher order neural networks in diverse applications of data mining.

Acknowledgments

This work is supported by Department of Science and Technology (DST), Ministry of Science and Technology, New Delhi, Govt. of India, under Grants no. DST/INSPIREFellowship/2013/585.

The authors would like to thank the editor and anonymous reviewers for their valuable comments and necessary suggestions those have helped to give more clarity to the paper.

Reference

[1] D. Mladenic, J. Brank, M. Grobelnik, Document classification, Encyclopedia of Machine Learning (2010) 289–293, ISBN 978-0-387-30768-8.
[2] P. Macioek, G. Dobrowolski, Using shallow semantic analysis and graph modelling for document classification, Int. J. Data Min. Model. Manag. 5 (2) (2013) 123–137.
[3] P. Yang, W. Gao, Q. Tan, K. Wong, A link-bridged topic model for cross-domain document classification, Inf. Process. Manag. 49 (6) (2013) 1181–1193.
[4] Z. Zhang, Q. Ye, Y. Li, R. Law, Sentiment classification of online cantonese reviews by supervised machine learning approaches, Int. J. Web Eng. Technol. 5 (4) (2009) 382–397.
[5] P. Yin, H. Wang, L. Zheng, Sentiment classification of Chinese online reviews: analysing and improving supervised machine learning, Int. J. Web Eng. Technol. 7 (4) (2012) 381–398.
[6] Z. Hao, J. Cheng, R. Cai, W. Wen, L. Wang, Chinese sentiment classification based on the sentiment drop point, Emerging Intelligent Computing Technology and Applications, Communications in Computer and Information Science, 375, 2013, pp. 55–60.
[7] M.S. Hajmohammadi, R. Ibrahim, A. Selamat, Bi-view semi-supervised active learning for cross-lingual sentiment classification, Inf. Process. Manag. 50 (5) (2014) 718–732.
[8] J. Upendar, C.P. Gupta, G.K. Singh, Modified PSO and wavelet transform-based fault classification on transmission systems, Int. J. Bio-Inspired Comput. 2 (6) (2010) 395–403.
[9] B. Bhalja, R.P. Maheshwari, A new fault detection, classification and location scheme for transmission line, Int. J. Power Energy Convers. 2 (4) (2011) 353–364.

[10] F. Yu, Z. Zhi-song, W. Xiao-ping, Research on model of circuit fault classification based on rough sets and SVM, Advances in Computer Science and Information Engineering, Advances in Intelligent and Soft Computing, 168, 2012, pp. 433–439.
[11] Z. He, S. Lin, Y. Deng, X. Li, Q. Qian, A rough membership neural network approach for fault classification in transmission lines, Int. J. Electr. Power Energy Syst. 61 (2014) 429–439.
[12] T. Joachims, Text classification, Learning to Classify Text Using Support Vector Machines, The Springer International Series in Engineering and Computer Science, 668, 2002, pp. 7–33.
[13] M.A. Wajeed, T. Adilakshmi, Supervised and semi-supervised learning in text classification using enhanced KNN algorithm: a comparative study of supervised and semi-supervised classification in text categorisation, Int. J. Intell. Syst. Technol. Appl. 11 (3/4) (2012) 179–195.
[14] A.K. Uysal, S. Gunal, Text classification using genetic algorithm oriented latent semantic features, Expert. Syst. Appl. 41 (13) (2014) 5938–5947.
[15] A. Tolambiya, S. Venkataraman, P.K. Kalra, Content-based image classification with wavelet relevance vector machines, Soft Comput. 14 (2) (2010) 137.
[16] P.S. Hiremath, P. Bannigidad, Identification and classification of cocci bacterial cells in digital microscopic images, Int. J. Comput. Biol. Drug Des. 4 (3) (2011) 262–273.
[17] D. Sriramkumar, R. Malmathanraj, R. Mohan, S. Umamaheswari, Mammogram tumour classification using modified segmentation techniques, Int. J. Biomed. Eng. Technol. 13 (3) (2013) 218–239.
[18] K. Mei, P. Dong, H. Lei, J. Fan, A distributed approach for large-scale classifier training and image classification, Neurocomputing 144 (2014) 304–317.
[19] K. Kim, S. Cho, DNA gene expression classification with ensemble classifiers optimized by speciated genetic algorithm, Pattern Recognition and Machine Intelligence, Lecture Notes in Computer Science, 3776, 2005, pp. 649–653.
[20] K. Kianmehr, M. Alshalalfa, R. Alhajj, Fuzzy clustering-based discretization for gene expression classification, Knowl. Inf. Syst. 24 (3) (2010) 441–465.
[21] S. Lee, E. Lee, K.H. Lee, D. Lee, Predicting disease phenotypes based on the molecular networks with condition-responsive correlation', Int. J. Data Min. Bioinforma. 5 (2) (2011) 131–142.
[22] Ed Keedwell, A. Narayanan, Gene expression rule discovery and multi-objective ROC analysis using a neural-genetic hybrid, Int. J. Data Min. Bioinforma. 7 (4) (2013) 376–396.
[23] C.E. Gillies, M. Siadat, N.V. Patel, G.D. Wilson, A simulation to analyze feature selection methods utilizing gene ontology for gene expression classification, J. Biomed. Inform. 46 (6) (2013) 1044–1059.
[24] Z. Andreopoulou, G. Tsekouropoulos, T. Koutroumanidis, M. Vlachopoulou, B. Manos, Typology for e-business activities in the agricultural sector, Int. J. Bus. Inf. Syst. 3 (3) (2008) 231–251.
[25] B.K. Sarkar, S.S. Sana, K. Chaudhuri, Accuracy-based learning classification system, Int. J. Inf. Decis. Sci. 2 (1) (2010) 68–86.
[26] I.K. Valavanis, G.M. Spyrou, K.S. Nikita, A comparative study of multi-classification methods for protein fold recognition, Int. J. Comput. Intell. Bioinforma. Syst. Biol. 1 (3) (2010) 332–346.
[27] M.Z. Solesvik, Collaborative knowledge management: case studies from ship design, Int. J. Bus. Inf. Syst. 8 (2) (2011) 131–145.
[28] P. Kumar, K.I. Varma, A. Sureka, Fuzzy based clustering algorithm for privacy preserving data mining, Int. J. Bus. Inf. Syst. 7 (1) (2011) 27–40.
[29] P. Mulay, P.A. Kulkarni, Knowledge augmentation via incremental clustering: new technology for effective knowledge management, Int. J. Bus. Inf. Syst. 12 (1) (2013) 68–87.
[30] D.T.C. Lai, J.M. Garibaldi, A preliminary study on automatic breast cancer data classification using semi-supervised fuzzy c-means, Int. J. Biomed. Eng. Technol. 13 (4) (2013) 303–322.
[31] J.R. Quinlan, C4. 5: Programs For Machine Learning, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1993.
[32] Y. Yung, M.J. Shaw, Introduction to fuzzy decision tree, Fuzzy Net Syst. 69 (1) (1995) 125–139.
[33] Y. Hamamoto, S. Uchimura, S. Tomita, A bootstrap technique for nearest neighbour classifier design, IEEE Trans. Pattern Anal. Mach. Intell. 19 (1) (1997) 73–79.
[34] R.R. Yager, An extension of the naive Bayesian classifier, Inf. Sci. 176 (5) (2006) 577–588.
[35] G.P. Zhang, Neural networks for classification: a survey, IEEE Trans. Syst. Man Cybern. Part C: Appl. Rev. 30 (4) (2000) 451–462.
[36] N. Redding, A. Kowalczyk, T. Downs, Constructive high-order network algorithm that is polynomial time, Neural Netw. 6 (1993) 997–1010.
[37] A. Goel, S. Saxena, S. Bhanot, Modified functional link artificial neural network, Int. J. Electr. Comput. Eng. 1 (1) (2006) 22–30.
[38] Y.H. Pao, Adaptive pattern recognition and neural networks, Addison-Wesley Pub., USA, 1989.
[39] Y.H. Pao, Y. Takefuji, Functional-link net computing: theory, system architecture, and functionalities, Computer 25 (1992) 76–79.
[40] J.C. Patra, A.C. Kot, Nonlinear dynamic system identification using Chebyshev functional link artificial neural networks, IEEE Trans. Syst. Man Cybern. Part B: Cybern. 32 (2002) 505–511.
[41] M. Klaseen, Y.H. Pao, The functional link net in structural pattern recognition, in: Proceedings of the TENCON 90. 1990 IEEE Reg. 10 Conference on Computer and Communication Systems, 2, 1990, pp. 567–571.
[42] G.H. Park, Y.H. Pao, Unconstrained word-based approach for off-line script recognition using density-based random-vector functional-link net, Neurocomputing 31 (2000) 45–65.
[43] L.M. Liu, M.T. Manry, F. Amar, M.S. Dawson, and A.K. Fung, Image classification in remote sensing using functional link neural networks, in: Proceedings of the IEEE Southwest Symposium on Image Analysis and Interpretation, 1994, pp. 54–58.
[44] P.P. Raghu, R. Poongodi, B. Yegnanarayana, A combined neural network approach for texture classification, Neural Netw. 8 (6) (1995) 975–987.
[45] I.-A. Abu-Mahfouz, A comparative study of three artificial neural networks for the detection and classification of gear faults, Int. J. Gen. Syst. 34 (2005) 261–277.
[46] J.C. Patra, R.N. Pal, A functional link artificial neural network for adaptive channel equalization, Signal Process. 43 (1995) 181–195.
[47] J. Teeter, C. Mo-Yuen, Application of functional link neural network to HVAC thermal dynamic system identification, IEEE Trans. Ind. Electron. 45 (1998) 170–176.
[48] H.M. Abbas, System identification using optimally designed functional link networks via a fast orthogonal search technique, J. Comput. 4 (2) (2009) 147–153.
[49] S.J. Nanda, G. Panda, B. Majhi, P. Tah, Improved identification of nonlinear MIMO plants using new hybrid FLANN-AIS Model, in: Proceedings of the IEEE International Conference on Advance Computing (IACC), 2009, pp. 141–146.
[50] J.C. Patra, C. Bornand, Nonlinear dynamic system identification using legendre neural network, in: Proceedings of the 2010 International Joint Conference on Neural Networks (IJCNN), 2010, pp. 1–7.
[51] S. Emrani, S.M.A. Salehizadeh, A. Dirafzoon, M.B. Menhaj, Individual particle optimized functional link neural network for real time identification of nonlinear dynamic systems, in: Proceedings of the 5th IEEE Conference on Industrial Electronics and Applications (ICIEA), 2010, pp. 35–40.
[52] R. Majhi, G. Panda, G. Sahoo, Development and performance evaluation of FLANN based model for forecasting of stock markets, Expert. Syst. Appl. 36 (2009) 6800–6808.
[53] J.H. Holland, Genetic algorithms, Scientific American, 1992 July, pp. 66–72.
[54] D.E. Goldberg, Genetic Algorithms in Search, Optimization, and Machine Learning, Addison-Wesley, Reading, MA, 1989, ISBN : 0201157675.
[55] J. Kennedy, R. Eberhart, Particle swarm optimization, in: Proceedings of the 1995 IEEE International Conference on Neural Networks, 4, 1995, pp. 1942–1948.
[56] Z.W. Geem, J.-H. Kim, G.V. Loganathan, A new heuristic optimization algorithm: harmony search, Simulation 76 (2) (2001) 60–68.
[57] M. Mahdavi, M. Fesanghary, E. Damangir, An improved harmony search algorithm for solving optimization problems, Appl. Math. Comput. 188 (2007) 1567–1579.
[58] C.-M. Wang, Y.-F. Huang, Self-adaptive harmony search algorithm for optimization, Expert Syst. Appl. 37 (2010) 2826–2837.
[59] J.C. Patra, W. Lim, P. Meher, E. Ang, Financial prediction of major indices using computational efficient artificial neural networks, in: Proceedings of the IEEE International Joint Conference on Neural Networks, Canada, 2006 July 16–21, pp. 2114–2120.
[60] B.B. Mishra, S. Dehuri, Functional link artificial neural network for classification task in data mining, J. Comput. Sci. 3 (12) (2007) 948–955 , Science Publications.
[61] S. Dehuri, B.B. Mishra, S. Cho, Genetic feature selection for optimal functional link artificial neural network in classification, in: Proceedings of the 9th International Conference on Intelligent Data Engineering and Automated Learning, Springer-Verlag, Berlin, Heidelberg, 2008, pp. 156–163.
[62] S. Dehuri, S. Cho, A comprehensive survey on functional link neural networks and an adaptive PSO–BP learning for CFLNN, Neural Comput. Appl. 19 (2) (2009) 187–205.
[63] J.C. Patra, W. Lim, N. Thanh and P. Meher, Computationally efficient FLANN-based intelligent stock price prediction system, in: Proceedings of the IEEE International Joint Conference on Neural Networks, Atlanta, Georgia, USA, 2009 June 14–19, pp. 2431–2438.
[64] J. Sun, J. Patra, W. Lim, and Y. Li, Functional link artificial neural network-based disease gene prediction, in: Proceedings of the IEEE International Joint Conference on Neural Networks, Atlanta, Georgia, USA, 2009 June 14–19, pp. 3003–3010.
[65] S. Chakravarty, P.K. Dash, Forecasting stock market indices using hybrid network, in: Proceedings of the IEEE World Congress on Nature & Biologically Inspired Computing, 2009, pp. 1225–1230.
[66] R. Majhi, S. Pandu, B. Panda, B. Majhi, G. Panda, Classification of consumer behavior using functional link artificial neural network, in: Proceedings of the IEEE International Conference on Advances in Computer Engineering, 2010, pp. 323–325.
[67] S. Dehuri, S.-B. Cho, Evolutionarily optimized features in functional link neural network for classification, Expert Syst. Appl. 37 (2010) 4379–4391.
[68] D.K. Bebarta, A.K. Rout, B. Biswal, P.K. Das, Forecasting and classification of Indian stocks using different polynomial functional link artificial neural networks, in: Proceedings of the India Conference (INDICON), 2012, pp. 178–182.
[69] S. Mishra, K. Shaw, D. Mishra, A new meta-heuristic bat inspired classification approach for microarray data, Procedia Technol. 4 (2012) 802–806 , C3IT.
[70] R. Mahapatra, K. Shaw, D. Mishra, Reduced feature based efficient cancer classification using single layer neural network, Procedia Technol. 6 (2012) 180–187.
[71] S. Mishra, K. Shaw, D. Mishra, S. Patnaik, An enhanced classifier fusion model for classifying biomedical data, Int. J. Comput. Vis. Robot. 3 (1/2) (2012) 129–137.

[72] S. Dehuri, R. Roy, S. Cho, A. Ghosh, An improved swarm optimized functional link artificial neural network (ISO-FLANN) for classification, J. Syst. Softw. (2012) 1333–1345.
[73] F. Mili, H. Hamdi, A comparative study of expansion functions for evolutionary hybrid functional link artificial neural networks for data mining and classification, in: Proceedings of the International Conference on Computer Applications Technology (ICCAT), 2013, pp. 1–8.
[74] B. Naik, J. Nayak, H.S. Behera, A Novel FLANN with a Hybrid PSO and GA based gradient descent learning for classification, in: Proceedings of the 3rd International Conference on Frontiers of Intelligent Computing: Theory and Applications (FICTA). Advances in Intelligent Systems and Computing, 327, 1, 2015, pp. 745–754.
[75] JR: B. Naik, J. Nayak, H.S. Behera, A honey bee mating optimization based gradient descent learning – FLANN (HBMO-GDL-FLANN) for classification, in: Proceedings of the 49th Annual Convention of the Computer Society of India CSI - Emerging ICT for Bridging the Future, Advances in Intelligent Systems and Computing, 338, 2015, pp. 211–220, doi: 10.1007/978-3-319-13731-5_24.
[76] G.L. Sicuranza, A. Carini, A generalized FLANN filter for nonlinear active noise control, IEEE Trans. Audio, Speech, Lang. Process. 19 (8) (2011) 2412–2417.
[77] N.V. George, G. Panda, A particle-swarm-optimization-based decentralized nonlinear active noise control system, IEEE Trans. Instrum. Meas. 61 (12) (2012) 3378–3386.
[78] G.L. Sicuranza, A. Carini, On the BIBO stability condition of adaptive recursive FLANN filters with application to nonlinear active noise control, IEEE Trans. Audio, Speech, Lang. Process. 20 (1) (2012) 234–245.
[79] S. Parija, P.K. Sahu, S.K. Nanda, S.S. Singh, A functional link artificial neural network for location management in cellular network, in: Proceedings of the International Conference on Information Communication and Embedded Systems (ICICES), 2013, pp. 1160–1164.
[80] H.H. Ali, M.T. Haweel, Legendre based equalization for nonlinear wireless communication channels, in: Proceedings of the international Electronics, Communications and Photonics Conference (SIECPC), Saudi, 2013, pp. 1–4.
[81] A.V. Durga Ganesh Reddy, L. Tarun Varma, Wind power forecasting without using historical data, in: Proceedings of the International Conference on Advances in Electrical Engineering (ICAEE), 2014, pp. 1–3.
[82] M. Cui, H. Liu, Z. Li, Y. Tang, X. Guan, Identification of Hammerstein model using functional link artificial neural network, Neurocomputing 142 (2014) 419–428.
[83] H.A. Abbass, A monogenous MBO approach to satisfiability, in: Proceedings of the International Conference on Computational Intelligence for Modelling Control and Automation, CIMCA 2001, Las Vegas, NV, USA, 2001.
[84] H.A. Abbass, Marriage in honey-bee optimization (MBO): a haplometrosis polygynous swarming approach. In: Proceedings of the Congress on Evolutionary Computation (CEC 2001), Seoul, Korea, 2001, pp. 207–214.
[85] D. Manjarres, I. Landa-Torres, S. Gil-Lopez, J. DelSer, M.N. Bilbao, S. Salcedo-Sanz, Z.W. Geem, A survey on applications of the harmony search algorithm, Eng. Appl. Artif. Intell. 26 (2013) 1818–1831.
[86] B. Naik, J. Nayak, H.S. Behera and A. Abraham, A harmony search based gradient descent learning-FLANN (HS-GDL-FLANN) for classification, Computational Intelligence in Data Mining - Volume 2, Proceedings of the International Conference on CIDM, 20–21, December 2014.
[87] S. Haykin, Neural Networks: A Comprehensive Foundation, 3rd ed., Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 2007.
[88] B. Naik, J. Nayak, H.S. Behera, A. Abraham, An improved harmony search-based functional linkhigher order ANN for nonlinear data classification, Int. J. Intell. Syst. Des. Comput. (2015) (in press).
[89] R.V. Rao, V.J. Savsani, D.P. Vakharia, Teaching–learning-based optimization: a novel method for constrained mechanical design optimization problems, Comput.-Aided Des. 43 (2011) 303–315.
[90] K. Bache, M. Lichman, UCI Machine Learning Repository, University of California, School of Information and Computer Science, Irvine, CA, 2013.
[91] J. Alcalá-Fdez, A. Fernandez, J. Luengo, J. Derrac, S. García, L. Sánchez, F. Herrera, KEEL data-mining software tool: data set repository, integration of algorithms and experimental analysis framework, J. Mult.-Valued Log. Soft Comput. 17 (2–3) (2011) 255–287.
[92] S. Larson, The shrinkage of the coefficient of multiple correlation, J. Educ. Psychol. 22 (1931) 45–55.
[93] F. Mosteller, J.W. Turkey, Data Analysis, Including Statistics. In Handbook of Social Psychology, Addison-Wesley, Reading, MA, 1968.
[94] J. Demsar, Statistical comparisons of classifiers over multiple data sets, J. Mach. Learn. Res. 7 (2006) 1–30.
[95] M. Friedman, The use of ranks to avoid the assumption of normality implicit in the analysis of variance, J. Am. Stat. Assoc. 32 (1937) 675–701.
[96] M.A. Friedman, Comparison of alternative tests of significance for the problem of m rankings, Ann. Math. Stat. 11 (1940) 86–92.
[97] J. Luengo, S. Garcia, F. Herrera, A study on the use of statistical tests for experimentation with neural networks: analysis of parametric test conditions and non-parametric tests, Expert Syst. Appl. 36 (2009) 7798–7808.
[98] S. Garcia, A. Fernandez, J. Luengo, F. Herrera, Advanced nonparametric tests for multiple comparisons in the design of experiments in computational intelligence and data mining: experimental analysis of power, Inf. Sci. 180 (2010) 2044–2064.
[99] Holm, A simple sequentially rejective multiple test procedure, Scand. J. Stat. 6 (1979) 65–70.
[100] Y. Hochberg, A sharper Bonferroni procedure for multiple tests of significance, Biometrika 75 (1988) 800–803.

Bighnaraj Naik: Bighnaraj Naik is working as an Assistant Professor in the Department of Master in Computer Application, Veer Surendra Sai University of Technology (VSSUT), Burla, Odisha, India. Currently, he is a Ph.D. Research Scholar in the Department of Computer Sc. Engineering and Information Technology, Veer Surendra Sai University of Technology, Burla, Odisha, India. He received his Bachelor in Engineering from National Institute of Science and Technology, Odisha, India and Master in Technology from Institute of Technical Education and Research, Odisha, India. His area of interest includes Data Mining, Machine Learning and Soft Computing.

Janmenjoy Nayak: Janmenjoy Nayak is a Research Fellow in the Department of Computer Sc. Engineering and Information Technology, Veer Surendra Sai University of Technology, Burla, Odisha, India. He completed his M.Tech. (Gold Medalist) degree in Computer Science from Fakir Mohan University, Balasore, Odisha, India and M. Sc. (Gold Medalist) degree in Computer Science from Ravenshaw University, Cuttack, Odisha, India . He has been awarded INSPIRE fellowship of Department of Science and Technology, Ministry of Science and Technology, Govt. of India for pursuing Ph. D. in Computer Science and Engineering at Veer Surendra Sai University of Technology (VSSUT), Burla, Sambalpur, Odisha, India . His area of interest includes Data Mining, Artificial Intelligence, Pattern Recognition and Soft Computing.

Dr. H.S. Behera: Himansu Sekhar Behera is working as an Associate Professor in the Department of Computer Sc. Engineering and Information Technology, Veer Surendra Sai University of Technology (VSSUT), Burla, Odisha. He has received M.Tech from N.I.T, Rourkela (formerly R.E.C) and Ph.D. in Engineering from Biju Pattnaik University of Technology (BPUT), Odisha respectively. He has published more than 80 research papers in international journals and conferences, edited 11 books and is acting as a member of the editorial / reviewer board of various international journals. He is a proficient in the field of Computer Science Engineering and served the capacity of program chair, tutorial chair and act as advisory member of committees of many national and international conferences. His research interest includes Data Mining, Soft Computing, Machine Intelligence, Evolutionary Computation and Distributed Systems.

Prof. Dr. Ajith Abraham: Current Director of Machine Intelligence Research Labs (MIR Labs), which has members from more than **100** countries. Dr. Abraham's research and development experience includes more than 23 years in the industry and academia. He received the M.S. degree from Nanyang Technological University (NTU), Singapore, and the Ph.D. degree in Computer Science from Monash University, Melbourne, Australia. He works in a multi-disciplinary environment involving machine (network) intelligence, cyber security, sensor networks, Web intelligence, scheduling, data mining and applied to various real world problems. He has given more than **60** conference plenary lectures/tutorials and invited seminars/lectures in over 75 Universities around the globe. He is an author/co-author of *900+* publications and some of the works have also won best paper awards at International conferences and also received several citations. Some articles are available in the ScienceDirect Top *25 hottest articles*. He serves/has served the editorial board of over 50 International journals and has also guest edited 40 special issues on various topics.