



Multi-strategy learning and deep harmony memory improvisation for self-organizing neurons

Shafaatunnur Hasan^{1,2} · Siti Mariyam Shamsuddin^{1,2}

Published online: 14 March 2018
© Springer-Verlag GmbH Germany, part of Springer Nature 2018

Abstract

This study proposes a concept of representation learning by implementing multi-strategy deep learning harmony memory improvisation for selecting the best harmony of self-organizing neurons. Representation learning is a set of methods that allows a machine to be fed with raw data and to automatically discover the representations needed for detection or classification. In our study, the deep multi-strategy learning involves the convolution of the self-organizing neurons with deep harmony memories improvisation in self-organizing and representation of map learning. The convolution of self-organizing neurons and harmony memory optimize the representation neurons' weights by generating the optimal best matching unit which is represented as fitness function of $f_1(x)$ and $f_2(x)$. While $f_1[g(f_1''(x))]$ and $f_2[g(f_2'(x))]$ represent the New Harmony fitness function. The best fitness function, $f_{best}(x)$ is selected based on the $f_1(x)$ and $f_2(x)$ performance which will be later stored in the harmony memory vector. The position vector of a particle is subjected to the Newtonian mechanics constant acceleration during the interval Δt . The search space of self-organizing map with Newton-based particle swarm algorithm particles depends on the width area, $\sigma_\alpha(t)$ of organizing neurons lattice structure. Our proposed methods are experimented on various biomedical datasets. The findings indicate that the proposed methods provide better quantization error for clustering and good classification accuracy with statistical measurement validations.

Keywords Representation learning · Machine learning · Deep learning · Meta-heuristic algorithms · Harmony search algorithm · Classification and clustering problems

1 Introduction

Learning is an improvement process of human behavior through experience. The key factor of a learning system involves the ability to reason, store and retrieve information from memory. In computational or machine learning (ML) approaches, a learning system is defined as a goal-guided process of the learner's knowledge by exploring the learner's experience and prior knowledge, so-called inferential theory of learning (ITL) (Michalski 1994). ITL is the basis of multi-

strategy learning systems where multiple types of inference mechanisms are integrated into one system. The objective of multi-strategy learning systems or deep learning strategy in our context is to achieve the ability to learn different types of knowledge from different types of inputs. In this context, a multi-strategy learning system can benefit from adapting different types of knowledge and input using multiple algorithm architecture. Transforming the initial learner's knowledge to satisfy the learning goals is the main process of learning and is mostly applied in clustering, classification, prediction and optimization. The typical problems commonly arise in classification include learning classification rules (training a classifier) from a set of examples (the training set) and testing the performance of the learned classifier from new input or environment (testing a test dataset).

ML techniques are used by many researchers as alternative solutions to solve the aforementioned problems. Among ML methods, artificial neural networks (ANNs), fuzzy sets, genetic algorithms (GAs), swarm intelligence (SI) and rough set methods are commonly used. ANNs are also

Communicated by V. Loia.

✉ Shafaatunnur Hasan
shafaatunnur@utm.my
Siti Mariyam Shamsuddin
mariyam@utm.my

¹ UTM Big Data Centre, Universiti Teknologi Malaysia, 81310 Skudai, Johor, Malaysia

² Faculty of Computing, Universiti Teknologi Malaysia, 81310 Skudai, Johor, Malaysia

known as neurocomputers, connectionist networks or parallel distributed processors and widely used ML methods (Negnevitsky 2005; Haykin 1999). It mimics the biological characteristics of the human brain which involves artificial neurons that can portray complex behavior. Simple neurons are connected together to form a connected network. Though it does not have to be adaptive, its advantages arise with proper algorithms to update the weights of the connections to produce a desired output. Therefore, ANN and meta-heuristics methodologies have each been proven effective in solving certain classes of learning problems. For example, neural networks are excellent at mapping input vectors to outputs, and meta-heuristics algorithms are very good at optimization (Kennedy and Eberhart 1995).

Meta-heuristics algorithms are based on optimization techniques such as evolutionary algorithms (EAs) and swarm intelligence (SI). Genetic algorithms (GA) are one of the techniques used in EA and are inspired by biological evolution phenomena such as inheritance, mutation, selection and crossover, while SI methods such as particle swarm optimization (PSO) and ant colony optimization (ACO) are inspired by the behavior of bird flocks, bee swarms, ant colonies and fish schools. Harmony search algorithm (HSA) was introduced by Geem et al. (2001) inspired from the improvisation process of Jazz musicians. The aim of the HSA is to find the perfect state of harmony, and the search process in optimization can be compared to a musician's improvisation process. The success of modern meta-heuristics algorithms relies on a good balance between the *intensification* and *diversification* (Yang 2010).

In this paper, deep improvisation of meta-heuristic algorithms based on HSA, PSO and Newton-based PSO with particles exploration/exploitation are proposed for multiple representation of self-organizing mapping (SOM) structure. The aim of this multiple representation learning is to investigate the efficiency of the multi-strategy and deep learning of SOM architecture for solving clustering and classification problems. Multiple representations learning of these algorithms will assist the organizing neurons in finding the optimal best matching unit (BMU) with a good set of weights for learning and generalization.

The remainder of this paper is organized as follows: Sect. 2 gives the scenario of multi-strategy learning representation in ANN follows by the preliminary theory on PSO and HSA in Sect. 3. Section 4 discusses our proposed algorithm on Multi-Strategy SOM Deep Learning Process with Meta-heuristic algorithms. Section 5 describes the experimental protocols for the proposed methods, while Sect. 6 gives the Results and Post-Analytics of the proposed methods. The conclusions are summarized in Sect. 7. Figure 1 provides the structure of this paper.

2 Multi-strategy learning representation in artificial neural network

Artificial Neural Networks (ANN) have emerged as an important tool for classification. There are many types of ANNs that have been used as classifiers because of their similarity with a simple brain model. These include backpropagation (BP), multilayer perceptron (MLP), self-organizing map (SOM), learning vector quantization (LVQ), radial basis function (RBF), and adaptive resonance theory (ART) networks. ANN have been developed as an alternative to statistical methods, which require assumptions that cannot be always satisfied. BP algorithm is one of the most popular classifiers used for training (Kennedy et al. 2001). However, BP learning suffers from a number of weaknesses such as slow convergence and local minima. Thus, many significant research efforts have been applying meta-heuristic algorithms or nature-inspired (NI) algorithms, such as evolutionary computation (EC) and SI techniques, for addressing ANN training issues.

Meta-heuristic algorithms or Nature-inspired Computing (NIC) is an emerging computing paradigm that draws on the principles of self-organization and complex systems (Jiming and Tsui 2006). NIC algorithms are autonomous, distributed, emergent, adaptive, and self-organized (Kennedy and Eberhart 1995). NIC methodologies have been applied to optimize ANN architectures. There are three main attributes of ANN architectures: network connection weights, network architecture (network topology, transfer function), and network learning algorithms. Most of the previous researches related to ANNs have focused on the network weights and topological structure. For example, the weights and/or topological structure are encoded as a GA chromosome. The selection of a fitness function is problem dependent. For a classification problem, the rate of misclassified patterns can be viewed as the fitness value. Meta-heuristic algorithms can be used in cases with non-differentiable processing element (PE) transfer functions and when no gradient information is available. The disadvantages of GA-based learning include performance being heavily dependent on the selection of the parameters together with difficult to represent weights and genetic operators. Therefore, several papers have reported using meta-heuristic algorithms such as PSO to replace GA (Bahesti et al. 2013; Beheshti and Shamsuddin 2013, 2014). Early studies on the hybridization of meta-heuristic algorithms with ANNs, especially PSO-MLP was proposed by Kennedy and Eberhart (1995). Recently, the enhancement of standard PSO with MLP was proposed by Beheshti et al. (2014).

The hybridization of the SOMPSO approach was first introduced by Xiao et al. (2003) and Xiao et al. (2004) for better clustering of gene datasets. These authors used SOM learning and PSO to optimize the SOM weights. However,

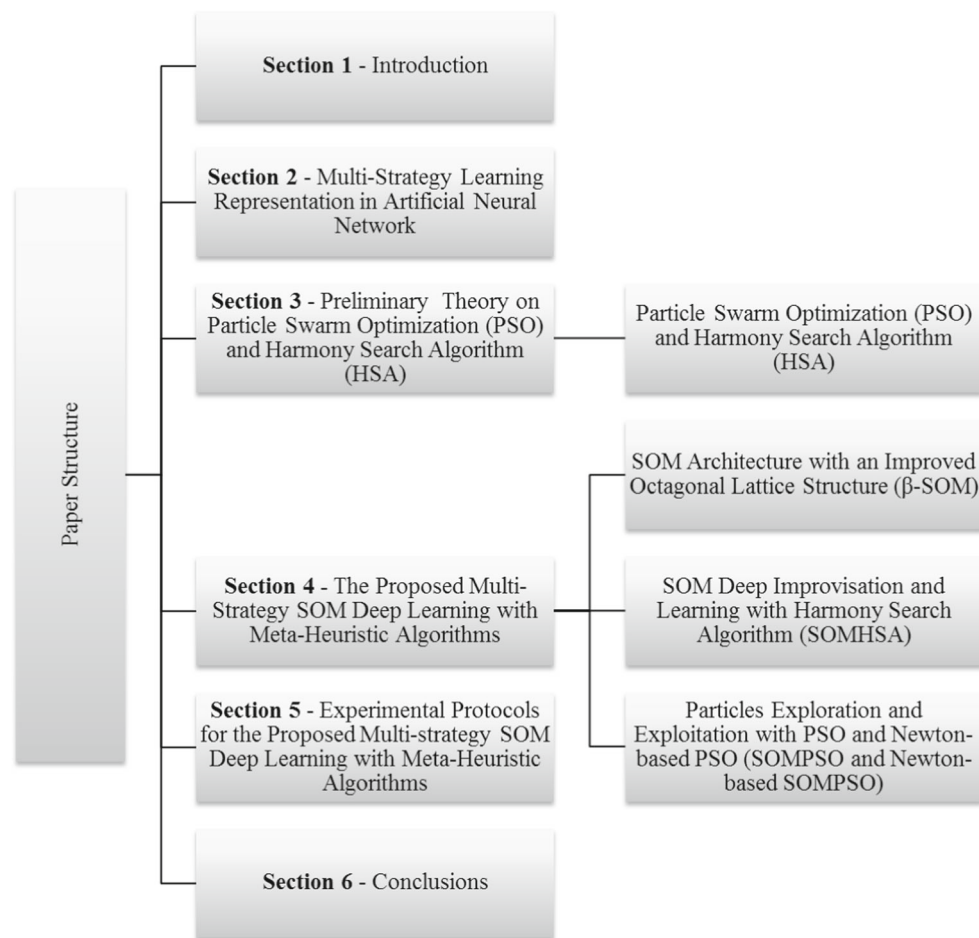


Fig. 1 Paper structure

the effectiveness of the combination of SOMPSO without a conscience factor was poorer than using SOM alone. This outcome is due to the use of conscience factor, which is a valuable as a competitive learning technique that reduces the number of epochs necessary to produce a robust solution. PSO was proposed for unsupervised learning in SOM, namely the self-organizing swarm (SOSWARM) (Brabazon and O'Neill 2006). The author explores the PSO parameters to adapt in a SOM where a fixed neighborhood and PSO velocity update are used in updating the weights. The study highlights some interesting features in SOM that can be explored, including the combination of SOMPSO, which can be further tested in various distance measurements and neighborhood structures, specifically in reducing the lattice size. On the other hand, Ozcift et al. (2009) used PSO for SOM optimization by reducing the neighborhood size and speeding up the training process. The author stated that the lattice size is related to the SOM clustering quality.

According to Hasan and Shamsuddin (2011) and Hasan (2010), this optimization technique has successfully reduced the number of nodes that find the BMU for a particular input.

With a larger lattice size, more nodes are considered for BMU calculation, which causes higher operating costs for the algorithm (Ozcift et al. 2009). However, a reduced lattice size will suffer loss of clustering information, which leads to the multi-strategy SOM with PSO for classification problems (Hasan and Shamsuddin 2011; Hasan 2010). The enhanced hexagonal lattice structure gives a wider exploration area in the training process, especially in BMU searching, which preserves the clustering quality and provides better accuracy for most standard UCI datasets.

Harmony search algorithm (HSA) (Yang 2010) has become an active research area because of its simplicity and higher efficiency. These characteristics make it easier to hybridize HSAs with ANN (Lee et al. 2016) and other meta-heuristic algorithms such as PSO (Omran and Mahdavi 2008). The enhancement of HSA, Improved Harmony Search (IHS) (Mahdavi et al. 2007), is used in solving engineering optimization problems. Further developments such as global harmony search (GHS) (Omran and Mahdavi 2008) algorithms perform better than IHS. Although GHS performed better than IHS, GHS is sometimes worse than basic HS

if the number of decision variables is big (Geem 2009a). Finally, the self-adaptive global best harmony search (SGHS) algorithm has been proposed for continuous optimization problems (Pan et al. 2010).

A Modified IHS (MIHS) (Kattan et al. 2010) has been applied to train a neural network. The algorithm is similar to IHS, except for the best to worst (BtW) ratio that is set for termination criterion. Subsequently, Kulluk et al. (2012) applied a SGHS algorithm for training neural networks in classification problems. In this study, the authors used MLP and BP with different variants of HSA including IHS, MIHS and GHS, for comparative study. The proposed algorithm performed better than the others in terms of accuracies with reasonable training time. The authors suggested implementing the algorithm in training neural network models such as SOM, LVQ and ART networks.

Harmony Search has also been used in clustering problems (Mahdavi et al. 2008). The author introduced harmony clustering method, HClust, and integrated the method with k-means clustering for web documents. The hybrid clustering method outperformed both HClust and k-means, whereas Amiri et al. (2010) applied HSA with k-means algorithm and Alia and Mandava (2011) implemented HSA with fuzzy and hard c-means for clustering problems. According to the authors, k-means and c-means algorithms are simple and easy to implement. However, the number of clusters must be defined in advance, and the algorithms are always trapped at local optima. Thus, these studies used HSA to assist k-means and c-means in finding the initial cluster center. Unlike SOM, the initial cluster center can be defined randomly. However, the structure of SOM network depends on the neighborhood lattice representation. A larger lattice area means more chances for a neuron to be updated but with high computational cost.

Meta-heuristic algorithms have been accredited as powerful and efficient to solve optimization problems than deterministic optimization algorithms. Meta-heuristic algorithms can be an alternative method to produce acceptable solutions by trial and error to a complex problem in a reasonably practical time (Yang 2010). The key factors of meta-heuristic algorithms performances are: intensification and diversification, or exploitation and exploration. The diversification via randomization avoids the solutions being trapped at local optima, while increases the diversity of the solutions. The good combination of these two major components will usually ensure that the global optimality is achieved. The next section provides brief introduction on the PSO and HSA prior to detail explanation on the proposed methods of multi-strategy and deep learning of SOM architecture with meta-heuristic algorithms.

3 Preliminary theory on particle swarm optimization (PSO) and harmony search algorithm (HSA)

In this section, we provide preliminary theory on PSO and HSA for better understanding of our proposed methods in the next subsections.

3.1 Particle swarm optimization (PSO)

Particle Swarm Optimization (PSO) is introduced by Kennedy and Eberhart (1995). It is inspired by birds flocking behavior in searching for food. Each bird updates its personal position based on its velocities to the food source. The nearest position of a bird to the food source will be the landmark for other birds to find the food. The standard PSO algorithm is given as below.

To explain how PSO works in solving an optimization problem, we assume to choose D as continuous variables x_1, \dots, x_D to maximize a function

$$f(x_1, \dots, x_D). \quad (1)$$

Suppose that we also create a swarm of $i = 1, \dots, N$ particles. At all points in time, each particle i has

1. A current position X_i or $X_n = (x_{i1}, \dots, x_{iD})$,
2. A record of the direction it followed to get to that position V_i or $V_n = (v_{i1}, \dots, v_{iD})$,
3. A record of its own best previous position $P_i = (P_{i1}, \dots, P_{iN})$,
4. A record of the best previous position of any members in its group $p_g = (p_{g1}, \dots, p_{gN})$.

Given the current position of each particle, as well as the other information, the problem is to determine the change direction of the particles. As mentioned above, this is done by referring to each particle's own experience and its companions. Its own experience includes the direction it came from V_i and its own best from previous position. The experience of others is represented by the best previous position for any members in its group. This suggests that each particle might move in:

- a. the same direction that it comes from V_i ,
- b. the direction of its best previous position $P_i - X_i$,
- c. the direction of the best previous position of any members in its group $p_g - X_i$.

The algorithm supposes that the actual direction of change for particle i will be a weighted combination of:

$$V_n = w \times V_n + C1 * r1 * (p_i - X_n) + C2 * r2 * (P_g - X_n), \quad (2)$$

where

r_1 and r_2 are uniform $[0,1]$ random numbers,
 $C_1 > 0$ and $C_2 > 0$ are constant called the *cognitive* and *social* parameters, and
 $w > 0$ is a constant called the *inertia* parameter.

n and $n + 1$ are successive index periods (generations), and given the direction of change, the new position of the particle will simply be:

$$X_n = X_n + V_n. \quad (3)$$

Given the initial values of X_i , V_i , P_i and P_g , Eqs. (2) and (3) will determine the subsequent path that each particle in the swarm will follow. To avoid particles from flying beyond the boundary, the velocities on each dimension are clamped to a maximum velocity, V_{\max} . If the sum of accelerations causes the velocity on that dimension to exceed V_{\max} , predefined parameter, then the velocity is limited to V_{\max} . For clear representation, the standard PSO algorithm is illustrated in Fig. 2.

3.2 Harmony search algorithm (HSA)

HSA is introduced by Geem et al. (2001) which inspired by the analogy of jazz improvisation. Each musician plays a different type of musical instrument. The musicians keep updating the harmony until the perfect state of harmony is obtained. In HSA, Harmony Memory (HM) is a matrix which stores solution vectors that ensures good harmonies are considered as elements of new solution vectors. The number of solution vectors in harmony memory is called the Harmony Memory Size (HMS). Figure 3 depicts the flowchart of the basic HSA model.

As seen in the figure, there are four main steps involved. For Step 1, the HM is initialized. The initial HM consists of a certain number of randomly generated solutions for the optimization problem under consideration. For an n dimension problem, an HM with the size of HMS can be represented as follows:

$$HM = \begin{bmatrix} x_1^1, x_2^1, \dots, x_n^1 \\ x_1^2, x_2^2, x_n^2 \\ \vdots \\ x_1^{HMS}, x_2^{HMS}, x_n^{HMS} \end{bmatrix} \quad (4)$$

where $(x_1^i, x_2^i, \dots, x_n^i)$, $(i = 1, 2, \dots, HMS)$, is a candidate solution.

HMS is typically set to be between 10 and 100. In Step 2, a new solution, $(x_1', x_2', \dots, x_n')$, is improvised from the HM. Each component of this solution x_j' is obtained based on the

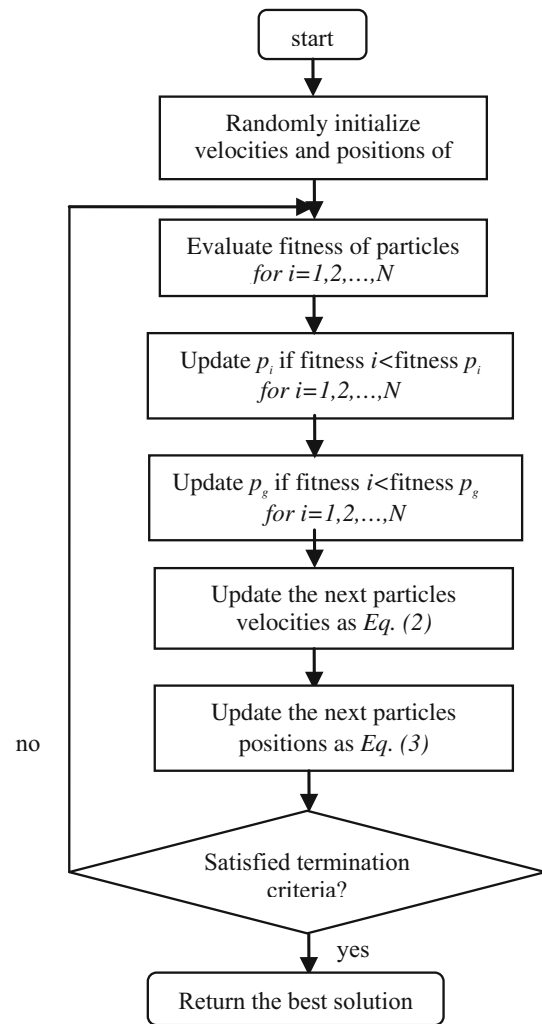
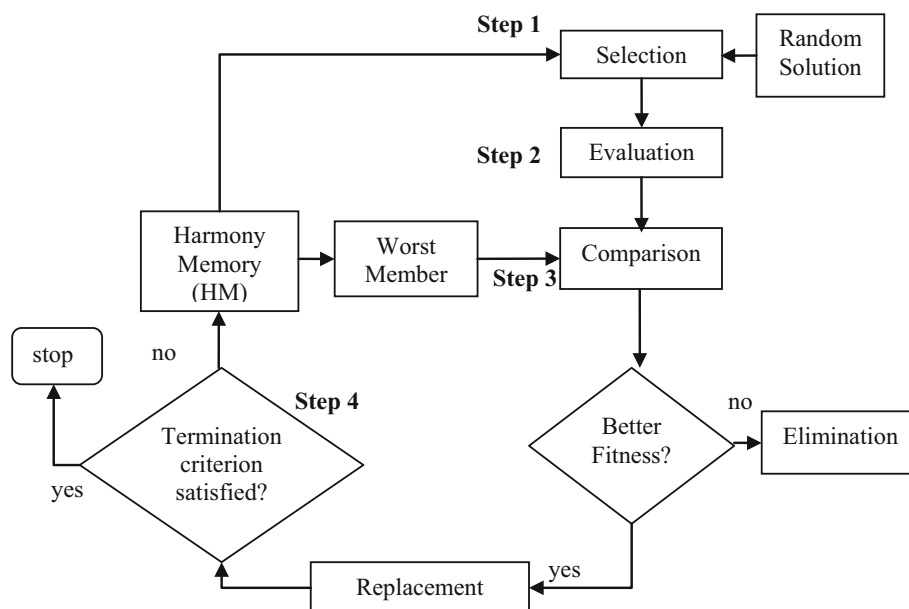


Fig. 2 PSO algorithm

harmony memory consideration rate (HMCR). The HMCR, r_{accept} is defined as the probability of selecting a component from the HM members, and $1-HMCR$ is, therefore, the probability of generating it randomly. Based on previous studies, typically, $r_{accept} = 0.7-0.95$ (Yang 2009). Once the rate is too low, it may converge extremely slowly. Otherwise, the pitches in the harmony memory are over exploited and lead to inaccurate solutions. If x_j' comes from the HM, it can be further changed according to the pitching adjusting rate (PAR). The PAR, r_{pa} determines the possibility of changing a candidate from the HM. A low pitch adjusting rate with a narrow bandwidth can slow the convergence of HSA because of the limitation in the exploration phases, as it covers only a small subspace of the search space. In contrast, a very high pitch adjusting rate with a wide bandwidth may cause the solution to scatter around some potential optima as in a random search. Thus, r_{pa} is normally set to values between $[0.1, 0.5]$ (Yang 2009). After a new solution from Step 2 is evaluated, the HM is compared and sorted in Step 3 to find the new

Fig. 3 HSA method. .
(Reproduced with permission
from Geem 2009b)



solution vector. It will replace the worst member in the HM if it yields a better fitness. Otherwise, it will be eliminated. Finally, in Step 4, the process keeps repeating and stopping until a termination criterion is satisfied.

4 The proposed multi-strategy SOM deep learning with meta-heuristic algorithms

In this section, the proposed multi-strategy SOM deep learning with meta-heuristic algorithms is presented. The proposed multi-strategy involves the deep improvising and map learning of SOM architecture with HSA (SOMHSA) and wider exploration/exploitation of Newton-based PSO (SOMPSO). We call our proposed methods as “Deep learning” since it involves deep learning of neurons organization in multi-strategy SOM architectures for obtaining optimal solutions. The process involves both global and local searching in finding optimal best matching unit (BMU) that can give a good set of weights for better mapping and labeling. For SOM architecture, an improved octagonal lattice structure is formulated to provide wider neurons explorations for better visualization. Figure 4 provides a journey to understand how our proposed multi-strategy SOM deep learning with meta-heuristic algorithms being developed. Detail explanation of each box is given in the following section.

4.1 SOM architecture with an improved octagonal lattice structure (β -SOM)

Self-organizing map (SOM) was first introduced by Von der Malsburg (1973) and presented by Kohonen (2001). The goal of SOM network is to map high dimensional input signal into

a simpler low dimensional discrete map. SOM is based on competitive learning, where the output nodes compete among themselves to be the winning node and the only node to be activated by a particular input observation (Haykin 1999). Generally, SOM learning algorithm is synonym with the clustering concept due to the adaptation process which produces a group of output patterns. In the SOM architecture, the adaptation process is crucial for updating the neurons weights in the neighborhood lattice area. As in Fig. 5, the best neurons are chosen as a winner which so-called BMU and the nearest neighbors will be updated until the best solution is met. Thus, the adaptation process is important in boosting the SOM performance in terms of the quality of network mapping, convergence and generalization.

The SOM learning generally uses rectangular, triangular and hexagonal neighborhood area. In this study, an octagonal-based lattice structure is developed to enhance the SOM learning capabilities with wider nodes exploration for clustering and classification problems. The SOM learning with standard octagonal SOM lattice is denoted as α -SOM and SOM learning with the improved octagonal SOM lattice is represented as β -SOM. Unlike α -SOM, the β -SOM generates neighborhood width, $(\sigma_{\beta}(t))$, four times wider than α -SOM $(\sigma_{\alpha}(t))$, as in Eqs. (5) and (6), respectively.

$$\sigma_{\beta}(t) = 32 \times \sigma_0(t)^2 \times (\sqrt{2} - 1), \quad (5)$$

$$\sigma_{\alpha}(t) = 8 \times \sigma_0(t)^2 \times (\sqrt{2} - 1), \quad (6)$$

Meanwhile, the improved octagonal neighborhood lattice area, $\sigma_{\alpha}(t)$ and $\sigma_{\beta}(t)$ consists of eight (8) important points, $P_i(x, y)$:

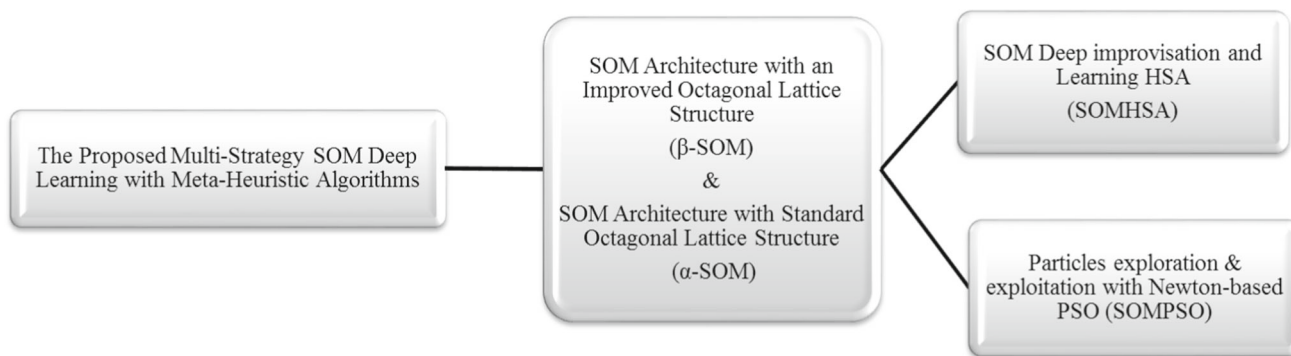


Fig. 4 Schematic representation of the proposed methods

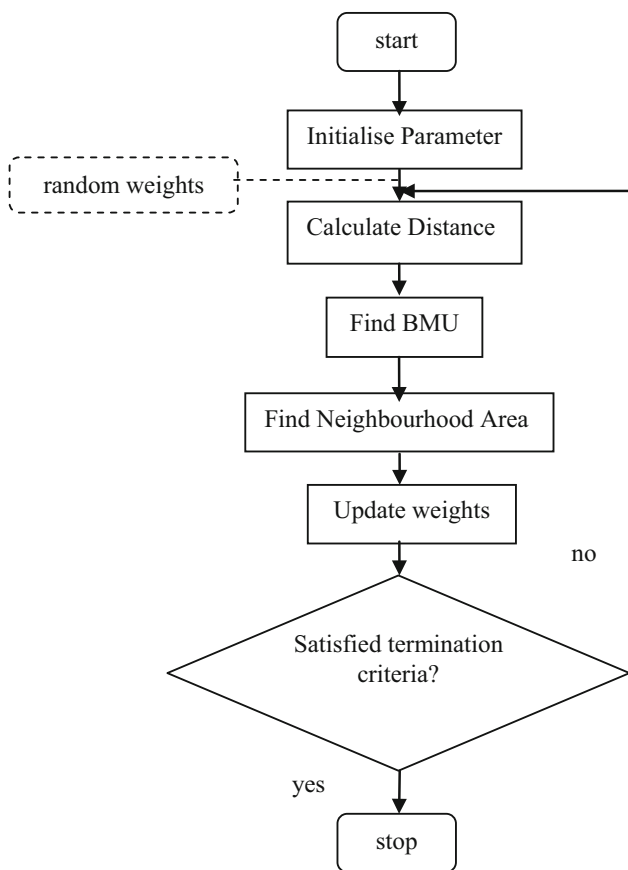


Fig. 5 SOM learning algorithm

$$\begin{aligned}
 &P_{top}(x, y), P_{bottom}(x, y), \\
 &P_{diagonal_right_corner}(x, y), P_{diagonal_left_corner}(x, y), \\
 &P_{right_corner}(x, y), P_{left_corner}(x, y), \\
 &P_{diagonal_bottom_right_corner}(x, y), \\
 &P_{diagonal_bottom_left_corner}(x, y),
 \end{aligned}$$

Figure 6 illustrates the improved octagonal lattice area, $\sigma_\beta(t)$ which $P_i(x, y)$ correspond to the $P_{center}(x, y)$.

The above lattice structures (standard and improved lattice structures) are used for optimizing the architectures of SOMHSA and SOMPPO. This can be achieved by searching the ideal winning nodes through deep learning optimizations with meta-heuristic algorithms, and this will be further discussed in the following subsection.

4.2 SOM deep improvisation and learning with harmony search algorithm (SOMHSA)

In this study, deep harmony improvisation of HSA for SOM mapping learning is implemented by finding the *BMU* of the best harmony solution, so-called best harmony fitness solution, HMS_{best} , and it is denoted as $f(x)$. $f(x)$ is evaluated based on harmony fitness solution, HMS_1 and HMS_2 , or $f_1(x)$ and $f_2(x)$. In order to produce deep improvisation scheme, $f(x)$, *BMU* is selected based on the improved octagonal SOM (β -SOM) in $f_1(x)$, while *BMU* are chosen according to the HSA improvisation in $f_2(x)$. Table 1 provides the description of HSA abbreviation used in this study.

HSA parameters include harmony memory solution (*HMS*), harmony memory consideration rate (*HMCR*), pitch adjusting rate (*PAR*), termination criterion, learning rate and radius. The *HMS* corresponds to the number of neurons in a 2-D mapping structure, while harmony memory (*HM*) consists of *HMS* ($HMS \in HM$) and decision variables which are set randomly between lower bound (LB) and upper bound (UB). In this context, decision variables contain input vector and weights vectors.

For *HM* of $HMS = 2 \times$ the mapping dimension, a New Harmony solution and *BMU* are chosen according to the HSA improvisation which is based on three rules: memory considering, pitch adjusting and random choosing. These rules are used for searching a new harmony fitness solution, HMS_2 , and it is denoted as $f_2(x)$. The first rule on memory consideration is implemented with two conditions: (1) the decision variables are less than the harmony memory accepting rate (r_{accept}) and (2) the condition of *PAR* (r_{pa}) is

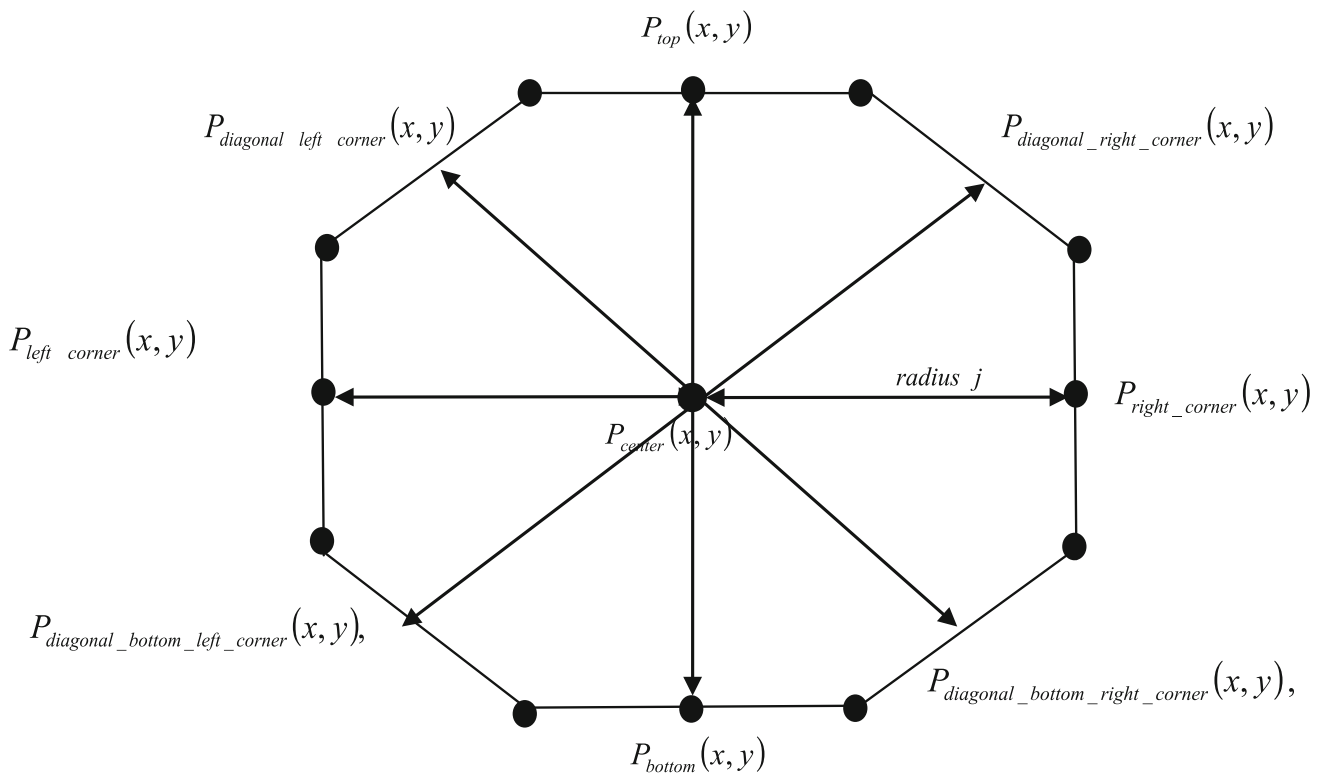


Fig. 6 The improved octagonal lattice area, $\sigma_\beta(t)$

Table 1 HSA notations

Abbreviation	Description
<i>HM</i>	Harmony memory
<i>HMS</i>	Harmony memory solution
<i>HMS_{best}</i>	Best harmony fitness solutions $\rightarrow f(x)$
<i>HMS₁</i>	A harmony fitness solution $\rightarrow f_1(x)$
<i>HMS₂</i>	A harmony fitness solution $\rightarrow f_2(x)$
<i>HMCR</i>	Harmony memory consideration rate
<i>PAR</i>	Pitch adjusting rate

employed within the pitch limits (*bw*). Otherwise, New Harmony will be generated randomly. Later, the best harmony fitness solution, *HMS_{best}*, $f(x)$, is evaluated based on the $f_1(x)$, and $f_2(x)$. From the best harmony fitness solution, $f(x)$, the BMU and the weights (w_k) of the best solution are chosen to update the weights. The updating procedure involves the implementation of an improved octagonal lattice width area ($\sigma_\beta(t)$), denoted as $HMS_{\sigma_\beta(t)}$ as given in Eq. (9). The procedure stops whenever the termination criterion (epoch) is met. The illustration of the Deep Harmony memory improvising for SOM mapping architecture is given Fig. 7 together with their generated matrices.

$$\text{HM} = \begin{bmatrix}
 \text{HMS}^1 = \begin{bmatrix} x_1 w_1^{0,0}, x_2 w_2^{0,0}, \dots, x_n w_n^{0,0} \\ x_1 w_1^{0,1}, x_2 w_2^{0,1}, \dots, x_n w_n^{0,1} \\ \vdots \\ x_1 w_1^{n,n}, x_2 w_2^{n,n}, \dots, x_n w_n^{n,n} \end{bmatrix} \\
 \text{HMS}^2 = \begin{bmatrix} x_1 w_1^{0,0}, x_2 w_2^{0,0}, \dots, x_n w_n^{0,0} \\ x_1 w_1^{0,1}, x_2 w_2^{0,1}, \dots, x_n w_n^{0,1} \\ \vdots \\ x_1 w_1^{n,n}, x_2 w_2^{n,n}, \dots, x_n w_n^{n,n} \end{bmatrix} \\
 \vdots \\
 \text{HMS}^n = \begin{bmatrix} x_1 w_1^{0,0}, x_2 w_2^{0,0}, \dots, x_n w_n^{0,0} \\ x_1 w_1^{0,1}, x_2 w_2^{0,1}, \dots, x_n w_n^{0,1} \\ \vdots \\ x_1 w_1^{n,n}, x_2 w_2^{n,n}, \dots, x_n w_n^{n,n} \end{bmatrix}
 \end{bmatrix}$$

(a) Matrices for Deep Harmony Memory Improvisation

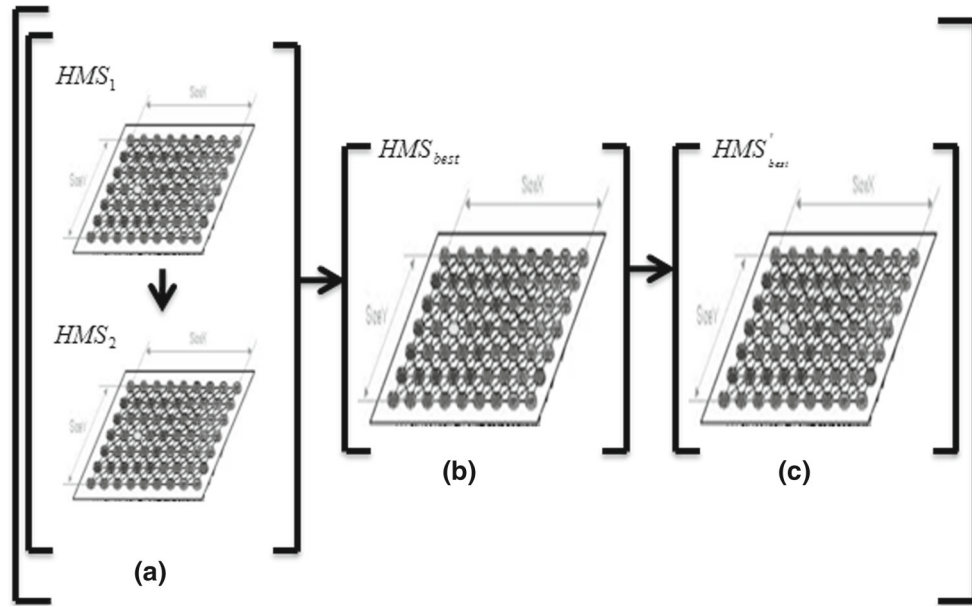


Fig. 7 Deep harmony memory improvisation (HM) representation of SOMHSA for a HMS_1 and HMS_2 , b HMS_{best} and c HMS'_{best}

$$HM' = \begin{bmatrix} HMS^1 = \begin{bmatrix} c^{0,0} \\ c^{0,1} \\ \vdots \\ c^{n,n} \end{bmatrix} \\ HMS^2 = \begin{bmatrix} c^{0,0} \\ c^{0,1} \\ \vdots \\ c^{n,n} \end{bmatrix} \\ \vdots \\ HMS^n = \begin{bmatrix} c^{0,0} \\ c^{0,1} \\ \vdots \\ c^{n,n} \end{bmatrix} \end{bmatrix}$$

(b) Matrices for the best centroid for each of the Deep HMS Improvisation

$$HM'' = \begin{bmatrix} c^{0,0} \\ c^{0,1} \\ \vdots \\ c^{n,n} \end{bmatrix}$$

(c) Matrices for the best of the best centroid in Deep HMS Improvisation

For each fitness solution, the Euclidean distance is computed to obtain the minimum distance. The distance is calculated between the input vector and the weights vector as below.

$$f_1(x) = \sqrt{\sum_{i=0}^{i=n} (V_i - W_{ij})^2},$$

$$f_2(x) = \sqrt{\sum_{i=0}^{i=n} (V_i - W_{ij})^2}, \tag{7}$$

where

$$x = (x_1, \dots, x_j)^i$$

V = Input vector,
 W_j = Weights vector

and $f_1(x) = f_2(x) = \arg_j \min D(V - W_j)$ overall output nodes.

The procedure to find the best fitness solutions of HMS'_{best} , $f(x)$, $x = (x_1, \dots, x_j)^i$, is illustrated below:

if $(f_1(x) < f_2(x))$ then

$$f(x) = f_1(x),$$

else

if $(f_2(x) < f_1(x))$,

then $f(x) = f_2(x)$.

The best fitness solutions keep the *BMU* weights (w_k) for each input vector (\mathbf{v}), and the radius of the neighborhood area $HMS_{\sigma_\beta(t)}$ is reducing using the exponential decay function,

$$\sigma(t) = \sigma_0 \exp\left(-\frac{t}{\lambda}\right), t = 1, 2, 3, \dots \tag{8}$$

where

σ_0 is the initial radius,
 λ is maximum iteration, and
 t is current iteration.

The neighborhood area ($HMS_{\sigma_\beta(t)}$) is defined as,

$$HMS_{\sigma_\beta(t)} = 32 (\sigma_0(t))^2 (\sqrt{2} - 1), \tag{9}$$

where $HMS_{\sigma_\beta(t)}$ is the octagonal lattice area and $\sigma_0(t)$ is the initial neighborhood radius at iteration t . The learning rate $L(t)$ updates the weight as in Eq. (10).

$$L(t) = L_0 \exp\left(-\frac{t}{\lambda}\right), t = 1, 2, 3, \dots \tag{10}$$

where

L_0 = initial learning rate, and

$$\Theta(t) = \exp\left(-\frac{dist(t)^2}{HMS_{\sigma_\beta(t)}}\right), t = 1, 2, 3, \dots \tag{11}$$

and

$\Theta(t)$ takes into account the neighborhood area $HMS_{\sigma_\beta(t)}$, and the average distance ($dist(t)$) of nodes in the neighborhood to obtain the winning node.

For updating the *HM*:

$$x(t+1) = x(t) + \Theta(t) L(t) (V(t) - x(t)), \tag{12}$$

where

$L(t)$ is learning rate, and
 $\Theta(t)$ is the influence of a node's distance from winning node.

Figure 8 illustrates the pseudo-code of the SOMHSA, respectively.

4.3 Particles exploration and exploitation with PSO and Newton-based PSO

To see the significance of the multi-strategy learning of SOM with other meta-heuristic algorithms, deep particles exploration and exploitation of PSO and Newton-based PSO with SOM learning (SOMPSO and Newton-based SOMPSO) are proposed to obtain better output mapping and labeling. Figure 9 illustrates the pseudo-code of PSO and Newton-based-PSO for exploration and exploitation of SOM architecture. The PSO velocity and position of each particle are represented as 2-D mapping dimension; and the computations are as follows:

$$v_{id}(t+1) = w v_{id}(t) + C_1 rand(p_{id}(t) - x_{id}(t)) + C_2 rand(p_{gd}(t) - x_{id}(t)), \tag{13}$$

$$w = ((w_{max} - w_{min})/iter \max) iter, \tag{14}$$

$$x_{id}(t+1) = x_{id} + v_{id}(t+1), \tag{15}$$

where

C_1 and C_2 are acceleration coefficients and both parameters are set to 1.0,

$rand$ is uniformly random number in the interval of [0, 1].

N is the number of particles,

$\vec{X}_i = (x_{i1}, x_{i2}, \dots, x_{id})$ and $\vec{V}_i = (v_{i1}, v_{i2}, \dots, v_{id})$ represents the position and velocity of i^{th} particle respectively,

$\vec{P}_i = (p_{i1}, p_{i2}, \dots, p_{id})$ is the personal best position found by the i^{th} particle, and $\vec{P}_g = (p_{g1}, p_{g2}, \dots, p_{gd})$ is the local best position achieved by the entire swarm.

In PSO, the cognition and social terms move a particle toward good solutions based on the particle experience and the best solution found by the swarm in the search space. However, in Newtonian's mechanics, the position vector of a particle is subjected to the acceleration as in Eq. (16):

$$x_2 = x_1 \left(-\nabla f^{-1}\right) x_1 f(x_1), \tag{16}$$

$$\frac{df}{dx} \approx \frac{f(x+\epsilon) - f(x)}{\epsilon}$$

where x_1 and x_2 are initial and final position, α and v_1 represent the particle's acceleration and velocity, respectively. These terms are applied for updating the next particle position in the Newton-based-SOMPSO. In other words, the cognition and social terms in PSO is used as a particle acceleration to update the next particle position, $x_{id}(t+1)$ as shown in Eq. (17):

Fig. 8 SOMHSA pseudo-code

```

Initialize the SOMHSA parameters and fitness function  $f_1(x), f_n(x), x = (x_1, K, x_j)^i$ 
While ( $t_3 < epoch$ )
  Begin
    Generate initial harmony (random weights) for  $HMS_1$ ,
     $x_j^1 = LB_j + rand \times (UB_j - LB_j), \quad (j = 1, 2K, N),$ 
  While ( $t_1 < f_1(x)$ )
    For each ( $j = 1, 2K, N$ ), do
      Calculate harmony fitness solution,  $f_1(x), x = (x_1, K, x_j)^i$ 
      Update HM,  $x(t+1) = x(t) + \Theta(t) L(t) (V(t) - x(t))$ 
    End
    Store initial BMU in HM
  End
  While ( $t_2 < f_n(x)$ )
  Begin
    Generate new harmony by accepting new harmony (solutions)
    For each ( $j = 1, 2K, N$ ), do
      If ( $rand < r_{accept}$ ),
        Choose an existing harmony randomly
         $x_j^i = x_j^i$ 
        where, ( $i = 1, 2K, HMS$ ), ( $j = 1, 2K, N$ ),
        UB = Upper bound and LB = Lower bound
      Else If ( $rand < r_{pa}$ ),
        Adjust the pitch randomly within a bandwidth
         $x_j^i = x_j^{current} \pm rand \times Bw,$ 
        where,  $X_j^i$  = new solution (pitch),
         $X_j^{current}$  = current solution (pitch),
        Bw = bandwidth,
        rand = random numbers between [0,1].
      Else
        Generate new harmony via randomisation
         $x_j^i = LB_j + rand \times (UB_j - LB_j),$ 
      End
    End
    Calculate new harmony (NH) fitness solution,  $f_n(x), x = (x_1, K, x_j)^i,$ 
    Store BMU NH in HM,
    Find the  $HMS_{best} \cdot f(x)$  in HM,
    If ( $f_n(x) < f_1(x)$ ),
      Set the  $HMS_{best}$  as the NH fitness solution,  $f(x) = f_n(x)$ 
    Else
      Set the  $HMS_{best}$  as the initial harmony fitness solution,  $f(x) = f_1(x)$ 
    End
    Set the BMU,  $W_k$  and neighborhood area,  $HMS_{\sigma_\beta(t)}$ 
    Update HM for  $HMS'_{best}, x(t+1) = x(t) + \Theta(t) L(t) (V(t) - x(t))$ 
  End

```

Fig. 9 SOMPSO and Newton-based SOMPSO pseudo-code

Step 1: Start.
Step 2: Initialize the population array of particle representing random solutions for d dimensional problem space in SOM network.
Step 3: Evaluate fitness values of particles.
Step 4: Update P_i if particle fitness value $i <$ particle best fitness value P_p , for $i = 1, 2, \dots, N$.
Step 5: Set P_i as best matching unit, BMU.
Step 6: Set P_g as local neighbourhood area.
Step 7: Update P_g if distance from BMU $P_i <$ distance from local neighbourhood area P_g for $i = 1, 2, \dots, N$.
Step 8: Update the next velocities of particles as Eq. (13)
Step 9: Update the next positions of particles as Eq. (15) for SOMPSO or Update the next positions of particles as Eq. (17) for Newton-based SOMPSO
Step 10: Repeat steps 3 to 9 until the stop criterion is reached.

$$x_{id}(t + 1) = x_{id} + v_{id}(t) + C_1 \times rand \times \left(-\nabla f^{-1}(x_{id}(t)) f(x_{id}(t)) \right) \quad (17)$$

5 Experimental protocols for the proposed multi-strategy SOM deep learning with meta-heuristic algorithms

The clarification of the experimental setup, performance measurements and parameter setting will be given in Sects. 5.1 and 5.2, respectively.

5.1 Experimental setup and performance measurement

In this study, biomedical datasets from the KEEL database (Alcalá-Fdez et al. 2011) have been implemented in the clustering and classification problems (see Table 2). For each of the datasets, min–max normalization is employed during training and testing with tenfold cross-validation. Consequently, the clustering and classification performance are evaluated.

Subsequent to the training and testing procedure, the results are evaluated using clustering and classification performance measurements. For clustering performance, quantization error (QE) is used to describe how accurately the neurons respond to the given dataset. For example, if the reference vector of the best matching unit (BMU) calculated for a given testing vector x_i is exactly the same x_i , the error in precision is 0. The equation is as follows:

$$E_q = \frac{1}{N} \sum_{k=1}^N \|x_k(t) - w_{mk}(t)\|, \quad (18)$$

where w_{mk} is the weight for the input vector x_k and the BMU m , at time t .

Table 2 Biomedical dataset information

Dataset	No. of instances	No. of attributes	Class name
Appendicitis	107	7	Negative Positive
Heart	270	13	Absence Presence
Hepatitis	155	19	Die Survive
Pima Indian Diabetes	768	8	Negative Positive
Mammographic	961	5	Benign Malignant
Wisconsin Breast Cancer	699	9	Benign Malignant
New Thyroid	215	5	Hyperthyroidism Hypothyroidism Normal

cluster cohesion (CC) is defined as the average sum of distances from cluster members to the cluster center.

$$CC_i = \frac{1}{|C_i|} \sum_{x \in C_i} dist(x, c_i), \quad (19)$$

where C_i denotes the i^{th} cluster, c_i is the center and $|C_i|$ is the magnitude of cluster C_i .

Upon completion, clusters and similarity between objects can be justified. However, the objects (data) that belong to the cluster (prototype vectors) are unknown. Thus, we proceed with testing data at the classification phase.

As seen on Table 3, the accuracy (ACC) and F measure is used as performance measurements for classification tasks.

Table 3 Classification performance measurements

Performance measurements	Abbreviation	Formula
Accuracy	<i>ACC</i>	$\frac{(TP+TN)}{(TP+TN+FP+FN)}$
F measure	<i>F1</i>	$\frac{2TP}{(2TP+FP+FN)}$

Table 4 Parameter setting for the proposed SOM deep learning models

Parameters	SOMHSA	SOMPSO	Newton-based SOMPSO
Learning strategy	Multi-strategy and deep harmony improvisation	Multi-strategy and wider particles exploration and exploitation	Multi-strategy and wider particles exploration and exploitation
Lattice structure/local neighborhood	Improved Octagonal	Square	Octagonal
Learning rate	0.5	N/A	N/A
HMCR	0.85	N/A	N/A
PAR	0.45	N/A	N/A
Bandwidth	0.5	N/A	N/A
HMS	Number of nodes	N/A	N/A
Particles	N/A	Number of nodes	Number of nodes
Decision variable /value range	[0,1]/[- 1,1]	[0,1]/[- 1,1]	[0,1]/ [- 1,1]
W_{\min}	N/A	0.4	0.4
W_{\max}	N/A	0.9	0.9
<i>C1</i>	N/A	1.0	1.0
<i>C2</i>	N/A	2.0	2.0
Input/problem dimension	Number of features		
Output dimension	2D-mapping dimension (<i>X, Y</i>)		
Mapping dimension (<i>X, Y</i>)	(10,10) / (5,5)		
Epoch	1000/3000/5000		

The *F* measure takes into account the probability of true positive (*TP*), false positive (*FP*) and false negative (*FN*) predictive values. In other words, the harmonic average of the precision and recall. While *ACC* measurement applies all predictive values including the true negative (*TN*). The best score is 1 and worst scores is 0, for both *ACC* and *F* measure.

5.2 Parameter setting

In this study, three SOM deep learning strategies are implemented using different types of lattice or local neighborhood structures for the proposed hybridization models of the multi-strategy learning. These include deep harmony improvising for SOM mapping learning (SOMHSA), SOM with PSO (SOMPSO) and SOM with Newton-based PSO (Newton-based SOMPSO) for particles wider exploration and exploitation (refer to Table 4). The number of *HMS* and particles are determined according to the number of nodes in tandem with the 2-D output dimension. Meanwhile, the decision variable or value range is set up according to the min–max normalization for each dataset. For instance, the

appendicitis, mammographic and Wisconsin dataset are normalized in the range [0,1], while other datasets are bounded to [- 1,1]. The number of epoch is set according to the number of sample and features dataset: appendicitis datasets are set for 1000 epoch, new thyroid, heart and hepatitis for 3000 epoch and Pima Indian, Wisconsin for 5000 epoch. The predefined number of epoch is given to avoid the overtraining of the network that can lead to the instability of the network generalization. The experimental results and analysis of the proposed SOM deep learning algorithms are described in Sect. 6.

6 Experimental results and analysis

The experimental results and analysis on clustering and classification of the proposed SOM deep learning models in multi-strategy learning environment are given together with the statistical analysis. Section 6.1 provides the clustering analysis of SOMPSO, SOMHSA and Newton-based SOMPSO, while Sect. 6.2 gives the classification analysis of SOMPSO, SOMHSA, Newton-based SOMPSO, standard

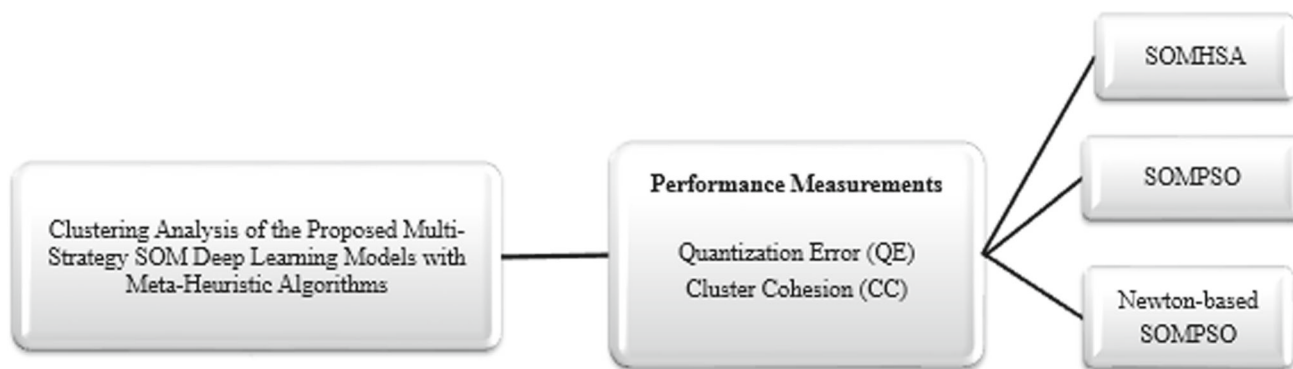


Fig. 10 A schematic view of clustering performance measurement

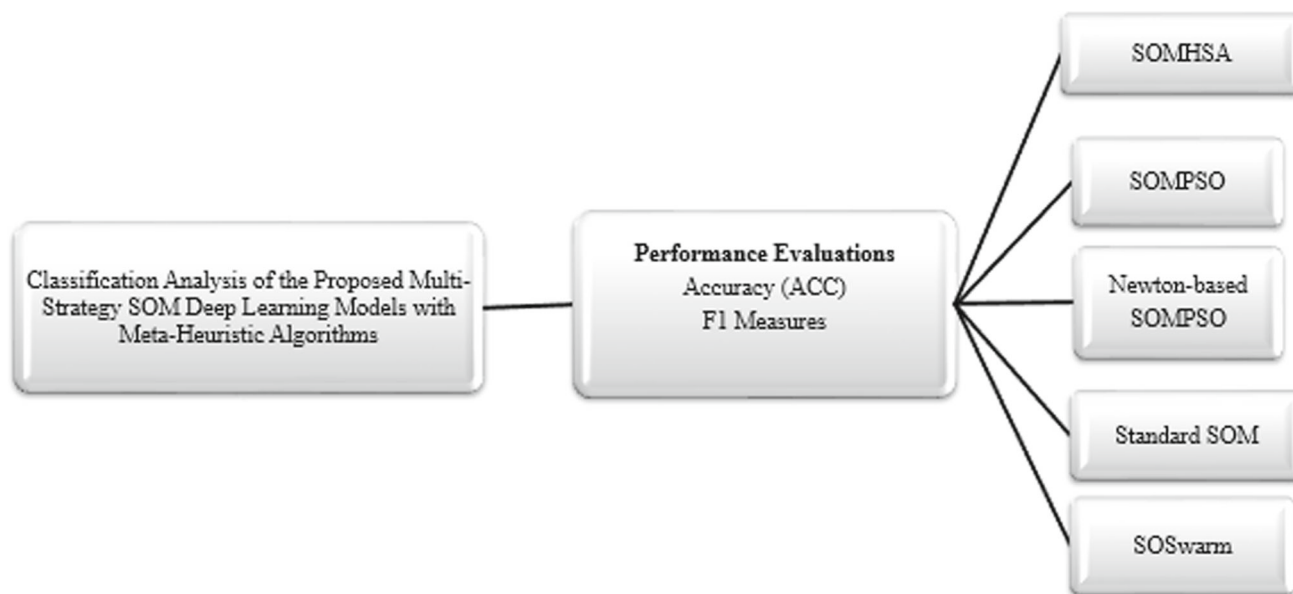


Fig. 11 A schematic view of classification performance measurement

SOM and self-organizing Swarm (SoSwarm) (O'Neill and Brabazon 2008). Finally, Sect. 6.3 provides the statistical analysis of the proposed SOM deep learning models in multi-strategy learning. Figures 10 and 11 provide the schematic diagram for better presentation in understanding how the performance evaluations are being conducted in this paper.

6.1 Clustering analysis of the proposed multi-strategy SOM deep learning models with meta-heuristic algorithms

Table 5 shows the clustering analysis of the proposed SOM deep learning models of SOMPSO, SOMHSA and Newton-based SOMPSO in the multi-strategy learning environment. The results are evaluated based on the average of quantization error (QE) and average cluster cohesion (CC). The best result of the performance evaluations (PE) is shown in bold. The performance of the SOMHSA is better than

SOMPSO and Newton-based SOMPSO for Pima Indian, mammographic and new thyroid datasets in terms of cluster cohesion (CC) and quantization error (QE). While for the appendicitis and hepatitis datasets, the results are similar in terms of CC : $CC = 0.033$ and $CC = 0.059$, respectively. SOMHSA is capable of preserving better mapping structure and correlation than SOMPSO and Newton-based SOMPSO. Meanwhile, SOMPSO and Newton-based SOMPSO generate quite poor QE and CC , as shown in hepatitis dataset with $CC = 1.184$ and $CC = 3.146$. This is due to the poor topological mapping of SOMPSO and Newton-based SOMPSO. However, square local neighborhood structure of SOMPSO performs better than the octagonal structure of Newton-based SOMPSO in terms of QE and CC . This is due to the broader particles exploration and exploitation in SOMPSO on fix local neighborhood structure. In Newton-based SOMPSO, the octagonal-based local neighborhood structure decreases gradually at time, t .

Table 5 Clustering analysis of the proposed SOM deep learning

Dataset	Performance evaluations	SOMPSO	SOMHSA	Newton-based SOMPSO
Appendicitis	QE	0.161	0.135	0.172
	CC	0.153	0.033	0.344
Heart	QE	0.836	0.078	0.144
	CC	0.119	0.013	1.881
Hepatitis	QE	0.083	0.187	0.105
	CC	1.184	0.059	3.146
Pima	QE	0.177	0.125	0.22
	CC	0.62	0.02	1.352
Mammographic	QE	0.243	0.138	0.294
	CC	0.343	0.0001	0.862
New Thyroid	QE	0.302	0.163	0.3823
	CC	0.709	0.0068	1.400
Wisconsin	QE	0.135	0.1099	0.146
	CC	0.329	0.01	0.545

To further verify the efficiency of the proposed methods, the measurements on the classification performance with ten-fold cross validations are also investigated and this is given in the next section.

6.2 Classification analysis of the proposed SOM deep learning models in multi-strategy with meta-heuristic algorithms

The accuracy (ACC) performance is evaluated based on the average performance of TP , TN , FP and FN predictive value. TP equals to numbers of positive cases are correctly classified, TN measures the proportion of negatives that are correctly identified, FP is defined as numbers of negative cases are wrongly classified as positive cases and FN is numbers of positive cases are wrongly classified as negative cases. As in Table 6, the high ACC are generated by SOMHSA for appendicitis dataset with an accuracy of $ACC = 88.0\%$, followed by the hepatitis with $ACC = 89.0\%$, and Wisconsin with $ACC = 97.0\%$ similar to standard SOM. Newton-based SOMPSO generates an accuracy of 68% for Pima Indian dataset and similar accuracy as SOMPSO and standard SOM for mammographic dataset which is $ACC = 70.0\%$. Standard SOM produces better results than the proposed SOM deep learning models in heart dataset with $ACC = 73.0\%$, and new thyroid with $ACC = 81.0\%$. In conclusion, the SOMHSA produces high accuracy (ACC) performance compared to SOMPSO and Newton-based SOMPSO based on the tested datasets.

However, the ACC performance is appropriate for balance dataset since the frequency of the imbalance dataset is not comparable between positive and negative class. Thus, the performance is also validated with F measure. The F measure is assessed based on the positive cases which cor-

responds to the TP (numbers of positive cases are correctly classified), FP predicted value (numbers of negative cases are wrongly classified as positive cases) and FN (numbers of positive cases are wrongly classified as negative cases). Furthermore, the F measure takes into consideration the influence of positive and negative cases which wrongly classified toward the positive cases.

As in Table 7, SOMHSA achieves high $F1$ for all datasets compared to SOMHSA Newton-based SOMPSO and standard SOM. The appendicitis datasets achieve high $F1$ with $F1 = 0.92$, hepatitis with $F1 = 0.69$, Pima Indian with $F1 = 0.55$, heart with $F1 = 0.70$ and Wisconsin with $F1 = 0.96$, new thyroid with $F1 = 0.75$ and mammographic with $F1 = 0.68$, respectively.

From these performance measurements, the proposed SOM deep learning models of SOMHSA produce high ACC and $F1$ compared to Newton-based SOMPSO, SOMPSO and standard SOM for almost all testing datasets. The SOMPSO, Newton-based SOMPSO and standard SOM outperformed SOMHSA on Pima and mammographic datasets in terms of ACC . Meanwhile, SOMHSA produces promising result for all datasets in F measures. The F measures performance is quantified based on the probabilities of FP , FN and TP predictive values. The classifiers performance toward positive cases can be evaluated using the F measures. In this context, F measures is beneficial for validating the proposed models and robust for categorizing the minority class. Unlike ACC , the F measures do not taking account the majority class, TN (number of negative cases which correctly classified).

For further evaluation, the proposed models are compared to the previous work by O'Neill and Brabazon (2008), namely Self-Organizing Swarm (SOSwarm) as shown in Table 8. The comparison is based on the average, best and SD of ACC with three datasets (Pima, new thyroid and

Table 6 Accuracy (ACC) of the proposed SOM deep learning models

Dataset	Deep learning models	ACC			
		Average	Worst	Best	SD
Appendicitis	SOMPSO	0.82	0.70	0.91	0.08
	SOMHSA	0.88	0.73	1.00	0.10
	Newton-based SOMPSO	0.80	0.70	0.91	0.06
	Standard SOM	0.87	0.71	1.00	0.10
Heart	SOMPSO	0.66	0.59	0.70	0.04
	SOMHSA	0.73	0.59	0.85	0.09
	Newton-based SOMPSO	0.68	0.63	0.85	0.07
	Standard SOM	0.78	0.67	0.89	0.07
Hepatitis	SOMPSO	0.85	0.67	1.00	0.10
	SOMHSA	0.89	0.63	1.00	0.12
	Newton-based SOMPSO	0.86	0.67	1.00	0.10
	Standard SOM	0.87	0.71	1.00	0.11
Pima	SOMPSO	0.67	0.64	0.70	0.03
	SOMHSA	0.67	0.62	0.77	0.05
	Newton-based SOMPSO	0.68	0.62	0.71	0.03
	Standard SOM	0.68	0.56	0.75	0.05
New Thyroid	SOMPSO	0.75	0.68	0.81	0.04
	SOMHSA	0.81	0.62	0.95	0.11
	Newton-based SOMPSO	0.76	0.68	0.90	0.07
	Standard SOM	0.87	0.71	1.00	0.10
Mammographic	SOMPSO	0.70	0.59	0.86	0.08
	SOMHSA	0.69	0.67	0.73	0.03
	Newton-based SOMPSO	0.70	0.60	0.90	0.10
	Standard SOM	0.70	0.52	0.81	0.08
Wisconsin	SOMPSO	0.79	0.67	0.94	0.08
	SOMHSA	0.97	0.93	1.00	0.03
	Newton-based SOMPSO	0.80	0.70	0.91	0.06
	Standard SOM	0.97	0.94	0.99	0.02

Wisconsin). As seen on Table 8, the SOMHSA outperforms SOMPSO, Newton-based SOMPSO and SOSwarm on Wisconsin dataset. Meanwhile, SOSwarm produces better result in terms of ACC on Pima and new thyroid datasets. The ACC result illustrates the concept of *no free lunch theorem* (David and William 1997) since no such algorithm works better for the whole datasets. Thus, ACC is suitable for measuring the balance dataset.

In the next section, we further validated the proposed models using statistical analysis to examine the significance of our findings for robust evaluation and verification.

6.3 Statistical analysis of the proposed SOM deep learning models in multi-strategy learning

Friedman Test is conducted to test whether k random samples drawn from K population have the same mean. In this context, N samples are obtained from k random samples

of sensitivity (S_n), positive predictive value (PPV), negative predictive value (NPV), accuracy (ACC), performance coefficient (PC) and average site performance (ASP) performance measurements with n datasets of appendicitis, heart, hepatitis, Pima Indian, mammographic, new thyroid and Wisconsin. Each dataset (n_i), ($i = 1, 2, \dots, k$) is drawn dependently with K populations from SOMHSA, SOMPSO and Newton-based SOMPSO models. As in Table 9, *Friedman* test illustrates the mean, standard deviation (SD), minimum (min), maximum (max) and median percentiles statistics based on N samples.

Based on the statistics, the *Friedman* test results are significantly different among the SOMHSA, SOMPSO and Newton-based SOMPSO with chi-square, $\chi^2(2) = 25.078$, and significance level of $p = 0$. For better illustration in observing the performance between 2-related samples from dependent populations, a *Wilcoxon* Signed-Rank test is implemented as post-hoc test.

Table 7 *F* measures of the proposed SOM deep learning models

Dataset	Deep learning models	<i>F</i> 1 measures			
		Average	Worst	Best	SD
Appendicitis	SOMPSO	0.89	0.82	0.95	0.05
	SOMHSA	0.92	0.77	1.00	0.07
	Newton-based SOMPSO	0.88	0.80	0.95	0.04
	Standard SOM	0.42	0.29	0.50	0.09
Heart	SOMPSO	0.53	0.42	0.69	0.08
	SOMHSA	0.70	0.52	0.82	0.09
	Newton-based SOMPSO	0.59	0.29	0.82	0.13
	Standard SOM	0.37	0.29	0.44	0.04
Hepatitis	SOMPSO	0.10	0.00	0.67	0.25
	SOMHSA	0.69	0.00	1.00	0.41
	Newton-based SOMPSO	0.17	0.00	0.67	0.31
	Standard SOM	0.35	0	0.50	0.19
Pima	SOMPSO	0.38	0.22	0.60	0.14
	SOMHSA	0.55	0.51	0.61	0.03
	Newton-based SOMPSO	0.34	0.00	0.57	0.18
	Standard SOM	0.26	0.18	0.35	0.05
New Thyroid	SOMPSO	0.39	0.00	0.60	0.17
	SOMHSA	0.75	0.56	0.92	0.12
	Newton-based SOMPSO	0.34	0.00	0.80	0.26
	Standard SOM	0.42	0.29	0.50	0.09
Mammographic	SOMPSO	0.66	0.48	0.86	0.12
	SOMHSA	0.68	0.65	0.75	0.03
	Newton-based SOMPSO	0.64	0.38	0.90	0.16
	Standard SOM	0.36	0.02	0.49	0.08
Wisconsin	SOMPSO	0.57	0.15	0.91	0.23
	SOMHSA	0.96	0.90	1.00	0.04
	Newton-based SOMPSO	0.65	0.37	0.88	0.17
	Standard SOM	0.48	0.45	0.50	0.02

Table 8 Accuracy (*ACC*) of the proposed SOM deep learning models

Dataset	Deep learning models	<i>ACC</i>		
		Average	Best	SD
Pima	SOMPSO	0.67	0.70	0.03
	SOMHSA	0.67	0.77	0.05
	Newton-based SOMPSO	0.68	0.71	0.03
	SOSwarm	0.69	0.80	0.03
New Thyroid	SOMPSO	0.75	0.81	0.04
	SOMHSA	0.81	0.95	0.11
	Newton-based SOMPSO	0.76	0.90	0.07
	SOSwarm	0.91	1.00	0.038
Wisconsin	SOMPSO	0.79	0.94	0.08
	SOMHSA	0.97	1.00	0.03
	Newton-based SOMPSO	0.80	0.91	0.06
	SOSwarm	0.96	1.00	0.014

Table 9 Friedman test statistics of the proposed SOM deep learning models

Proposed SOM deep learning models	N	Mean	SD	Min	Max	Percentiles		
						25th	50th (median)	75th
SOMHSA	49	0.7565	0.15623	0.37	0.99	0.66	0.73	0.93
SOMPSO	49	0.6153	0.25303	0.07	1	0.445	0.68	0.815
Newton-based SOMPSO	49	0.6220	0.24755	0.13	1	0.4600	0.70	0.80

Table 10 Wilcoxon signed ranks test of the proposed SOM deep learning models

Proposed SOM deep learning models	Ranks		Mean rank	Sum of ranks	Z-score	Significance level, p (2-tailed)	
SOMPSO (C) – SOMHSA (B)	Negative	34a	a. C < B	25.72	874.5	– 4.430a	0
	Positive	10b	b. C > B	11.55			
	Ties	5c	c. C = B				
Newton-based SOMPSO (D) – SOMHSA (B)	Negative	37a	a. D < B	27.58	1020.5	– 4.438a	0
	Positive	11b	b. D > B	14.14			
	Ties	1c	c. D = B				

For better illustration, each of the SOMHSA, SOMPSO and Newton-based SOMPSO models are labeled as B, C, D as in Table 10. The statistical test presents the result in terms of ranks (positive, negative and ties), mean ranks, sum ranks, Z – score and two-tailed significance level (p). Positive and negative ranks indicate that there exists a significant difference between the N samples, while ties rank shows no significant difference between the N samples.

Therefore, in this statistical test, the Wilcoxon Signed-Rank test shows that SOMHSA is statistically significant to SOMPSO with Z – score of $Z = -4.430$ and two-tailed significance level of $p = 0$, respectively, while Newton-based SOMPSO is statistically significant to SOMHSA with $Z = -4.438$ and $p = 0$, while SOMHSA is significantly different with higher rank than SOMPSO and Newton-based SOMPSO, derived from the negative ranks of ($C < B$) and ($D < B$).

7 Conclusions

We have proposed the multi-strategy approaches and deep SOM learning and improvisation with SOMHSA, SOMPSO and Newton-based SOMPSO for better mapping and labeling in clustering and classification problems. The overall performance of the deep harmony improvisation of SOMHSA indicates the competitiveness of the Newton-based SOMPSO and SOMPSO in terms of clustering and classifier performances,

respectively. HSA improvisation scheme provides better harmony diversification and intensification with improved octagonal neighborhood lattice structure than Newtonian-based PSO with standard octagonal local neighborhood (Newton-based SOMPSO), and standard PSO based on fixed square local neighborhood (SOMPSO). This shows that the wider with deep exploration and exploitations of the search space with improved lattice structure gives better performances in terms of clustering and classification. Furthermore, the proposed models with deep learning mechanism on classification performance have reduced the bias toward the majority class compare to the standard SOM. However, SOMPSO and Newton-based SOMPSO still need some improvement in hepatitis and new thyroid dataset.

Thus, the multi-level learning schemes with deep learning neural network architecture will be considered in the future to deal with the imbalance datasets of real world problems. Our proposed multi-strategy SOM deep mapping learning can be applied on multi-dimensional unstructured big data problems. However, the major challenge is dealing with the big data pre-processing and multi-decision solutions especially in real-time business analytics.

Acknowledgements The authors would like to thank Ministry of Higher Education (MOHE) and Universiti Teknologi Malaysia (UTM) for their support in Research and Development, *UTM Big Data Centre* and the *Soft Computing Research Group (SCRG)* for the inspiration in making this study a success. This work is partially supported by the UTM Research University Grant Scheme (17H62) and FRGS (4F786).

Compliance with ethical standards

Conflict of interest The authors declare that they have no conflict of interest.

Ethical approval This paper does not contain any studies with human participants or animals performed by any of the authors.

Informed consent Informed consent was obtained from all individual participants included in the study.

References

- Alcalá-Fdez J, Fernandez A, Luengo J, Derrac J, García S, Sánchez L, Herrera F (2011) KEEL data-mining software tool: data set repository, integration of algorithms and experimental analysis framework. *J Mult Valued Log Soft Comput* 17(2–3):255–287
- Alia O, Mandava R (2011) The variants of the harmony search algorithm: an overview. *Artif Intell Rev* 36(1):49–68
- Amiri B, Hossain L, Mosavi SE (2010) Application of harmony search algorithm on clustering. In *World congress on engineering and computer science*, vols 1 and 2, pp 460–465
- Beheshti Z, Shamsuddin SM (2013) A review of population-based meta-heuristic algorithms. *Int J Adv Soft Comput Appl* 5(1):1–35
- Beheshti Z, Shamsuddin SMH (2014) CAPSO: centripetal accelerated particle swarm optimization. *Inf Sci* 258:54–79
- Bahesti Z, Shamsuddin SM, Hasan S (2013) MPSO: median oriented particle swarm optimization. *Appl Math Comput* 219:5817–5836
- Beheshti Z, Shamsuddin SMH, Beheshti E, Yuhani SS (2014) Enhancement of artificial neural network learning using centripetal accelerated particle swarm optimization for medical diseases diagnosis. *Soft Comput* 18(11):2253–2270
- Brabazon A, O'Neill M (2006) *Biologically inspired algorithms for financial modeling*. Springer, Berlin
- David HW, William GM (1997) No free lunch theorem for optimization. *IEEE Trans Evolut Comput* 1(1):67–82
- Geem ZW (2009a) Particle-swarm harmony search for water network design. *Eng Optim* 41(4):297–311
- Geem ZW (2009b) *Music inspired harmony search algorithm: theory and applications*. Springer, New York
- Geem ZW, Kim JH, Loganathan GV (2001) A new heuristic optimization algorithm: harmony search. *Simulation* 76(2):60–68
- Hasan S (2010) Enhanced self-organizing map with particle swarm optimization for classification problems. Master thesis, Universiti Teknologi Malaysia
- Hasan S, Shamsuddin SM (2011) Multistrategy self-organizing map learning for classification problems. *Comput Intell Neurosci* 2011(121787):11. <https://doi.org/10.1155/2011/121787>
- Haykin S (1999) *Neural networks: a comprehensive foundation*, 2nd edn. Prentice Hall, Upper Saddle River
- Jiming L, Tsui KC (2006) Toward nature-inspired computing. *Commun ACM* 49(10):59–64
- Kattan A, Abdullah R, Salam RA (2010) Harmony search based supervised training of artificial neural networks. In: *UKSim-AMSS first international conference on intelligent systems, modelling and simulation*, pp 105–110
- Kennedy J, Eberhart (1995) Particle swarm optimization. In: *Proceedings of IEEE international conference on neural networks*. IEEE Press, Piscataway, NJ, pp 1942–1948
- Kennedy J, Eberhart RC, Shi Y (2001) *Swarm intelligence*. Morgan Kaufmann, San Francisco
- Kohonen T (2001) *Self-organizing maps*. Springer series in information sciences, vol 30, 3rd extended edition. Springer, Berlin
- Kulluk S, Ozbakir L, Baykasoglu A (2012) Training neural networks with harmony search algorithms for classification problems. *Eng Appl Artif Intell* 25(1):11–19
- Lee Anzy, Geem Zong Woo, Suh Kyung-Duck (2016) Determination of optimal weights of an artificial neural networks by using the harmony search algorithm: application to breakwater armor stones. *Appl Sci* 6(164):1–17
- Mahdavi M, Fesanghary M, Damangir E (2007) An improved harmony search algorithm for solving optimization problems. *Appl Math Comput* 188(2):1567–1579
- Mahdavi M, Chehreghani MH, Abolhassani H, Forsati R (2008) Novel meta-heuristic algorithms for clustering web documents. *Appl Math Comput* 201(1–2):441–451
- Michalski RS (1994) *Machine learning: a multistrategy approach*. In: Michalski RS, Tecuci G (eds) *Inferential learning theory as conceptual basis for multistrategy learning*, vol IV. Morgan Kaufmann Publishers, San Mateo
- Negnevitsky M (2005) *Artificial intelligence: a guide to intelligent systems*, 2nd edn. Addison Wesley, Harlow
- Omran MGH, Mahdavi M (2008) Global-best harmony search. *Appl Math Comput* 198(2):643–656
- O'Neill M, Brabazon A (2008) Self-organising swarm (SOSwarm). *Soft Comput* 12(11):1073–1080
- Ozcift A, Kaya M, Gulden A, Karabulut M (2009) Swarm optimized organizing map (SWOM): a swarm intelligence based optimization of self-organizing map. *Expert Syst Appl* 36(7):10640–10648
- Pan Q-K, Suganthan MPN, Tasgetiren F, Liang JJ (2010) A self-adaptive global best harmony search algorithm for continuous optimization problems. *Appl Math Comput* 216(3):830–848
- Von der Malsburg C (1973) Self-organization of orientation sensitive cells in the striate cortex. *Kybernetik* 14:85–100
- Xiao X, Dow ER, Eberhart R, Miled ZB, Oppelt RJ (2003) Gene-clustering using self-organizing maps and particle swarm optimization. *IEEE international parallel and distributed processing symposium (IPDPS)*. IEEE Press
- Xiao X, Dow ER, Eberhart R, Miled ZB, Oppelt RJ (2004) A hybrid self-organizing maps and particle swarm optimization approach. *Concurr Comput Pract Exp* 16(9):895–915
- Yang XS (2009) Music-inspired Harmony search algorithm. In: Geem Z (ed) *Harmony search as a metaheuristic algorithm*. Springer, Berlin, pp 1–14
- Yang XS (2010) *Engineering optimization: an introduction with meta-heuristic applications*. Wiley, Hoboken