International Journal of Swarm Intelligence and Evolutionary Computation

ISSN: 2090-4908

Research Article Open Access

Memetic Harmony Search Algorithm Based on Multi-objective Differential Evolution of Evolving Spiking Neural Networks

Abdulrazak Yahya Saleh*, Siti Mariyam Shamsuddin and Haza Nuzly Abdull Hamed

UTM Big Data Centre, Universiti Teknologi Malaysia

Abstract

Spiking neural network (SNN) plays an essential role in classification problems. Although there are many models of SNN, Evolving Spiking Neural Network (ESNN) is widely used in many recent research works. Evolutionary algorithms, mainly differential evolution (DE) have been used for enhancing ESNN algorithm. However, many real-world optimization problems include several contradictory objectives. Rather than single optimization, Multi-Objective Optimization (MOO) can be utilized as a set of optimal solutions to solve these problems. In this paper, Harmony Search (HS) and memetic approach was used to improve the performance of MOO with ESNN. Consequently, Memetic Harmony Search Multi-Objective Differential Evolution with Evolving Spiking Neural Network (MEHSMODE-ESNN) was applied to improve ESNN structure and accuracy rates. Standard data sets from the UCI machine learning are used for evaluating the performance of this enhanced multi objective hybrid model. The experimental results have proved that the Memetic Harmony Search Multi-Objective Differential Evolution with Evolving Spiking Neural Network (MEHSMODE-ESNN) gives better results in terms of accuracy and network structure.

Keywords: Evolving spiking neural networks, Harmony search, Multi-objective optimization.

Introduction

Classification of patterns is vital to several data mining processes. Classification is one of the most commonly obverse processing tasks for a decision support system [1]. There are many areas in life which need classification such as medical diagnoses, medicine, science, industry, speech recognition and handwritten character recognition. Among feasible classifiers, Spiking neural network (SNN) plays an essential role in biological information processing [2].

Although there are many models of SNN, Evolving Spiking Neural Network (ESNN) is widely used in many recent research works. ESNN has several attractive advantages [3] including: simplicity, efficiency, trained by a fast one-pass learning algorithm. The evolving nature of model can be updated whenever new data becomes accessible with no requirement to retrain earlier existing samples. However, ESNN model is affected by its choice of parameters. The right selection of parameters allows the network to evolve towards reaching the best structure thus guaranteeing the best output. Of all the issues that require exploration in ESNN determining the optimal number of pre-synaptic neurons for a given data set is the most important one [4] For these reasons, achieving an optimized trade-off between accuracy and the network structure is needed to find the best combination of parameters and pre-synaptic neurons.

Optimization has been used for enhancing ESNN algorithm. Multi-Objective Optimization (MOO) can be utilized as a set of optimal solutions to solve these problems. Every MOO solution appears to be a new trade-off between the objectives. The key objective of MOO is to improve ESNN optimal solutions of both structure and classification accuracy. MOO approach is preferred to traditional learning algorithms for the following reasons. First, as a result of using MOO a good performance of these learning algorithms can be achieved [5]. Second, various objectives are taken into consideration in the generation of multiple learning models. For example: accuracy, complexity [6-8], interpretability and accuracy [9], multiple error measures [10]. Third, it is superior to building learning ensembles to use models [8-12]. The important goal of MOO algorithm is to find a set of solutions from which the best one is chosen. Based on Tan et al. [13], the ability of evolutionary algorithms (EAs) to search for optimal solutions gives it the priority to be selected in MOO problems. EAs have the ability to explore different parts of the related algorithm in the optimal set because of the population-based algorithms.

In spite of the fact that one Multi-Objective Evolutionary Algorithms (MOEAs) was used and only for SpikeProp learning, most of the related work which use Multi-Objective Genetic Algorithms (MOGAs) is by Jin et al. [14]. Both classification performance and connectivity of SNN with latency coding are optimized by MOGA. During optimization, both delay and weight between the two neurons are evolved. Furthermore, they minimize the classification error in percentage or the root mean square error for optimizing performance, and minimize the number of connections or the sum of delays for connectivity to explore the objectives influence on the connectivity and performance of SNNs. This is a very motivating finding. However, more experiments must be performed to verify this observation. No conclusion can be made on classification error should be used for classification. They have also revealed that complexities of the SNNs are equivalent, when the number of connections or the sum of delays is utilized to optimize connectivity. This study needs improvement to enhance its promising results.

Moreover, one of the EAs i.e., harmony search (HS) algorithm was utilized to overcome problems of convergence rate at finding the global minimum of DE [15-17]. Subsequently, back propagation (BP) was used to speed up convergence known as a memetic approach.

***Corresponding author:** Abdulrazak Yahya Saleh, UTM Big Data Centre, Universiti Teknologi Malaysia, UTM Skudai, Johor, Malysia, Tel: 601137090573, Email: abdulrazakalhababi@gmail.com

Received January 19, 2016; **Accepted** February 22, 2016; **Published** February 27, 2016

Citation: Saleh AY, Shamsuddin SM, Hamed HNA (2016) Memetic Harmony Search Algorithm Based on Multi-objective Differential Evolution of Evolving Spiking Neural Networks. Int J Swarm Intel Evol Comput 5: 130. doi: 10.4172/2090-4908.1000130

Unlike previous study mentioned earlier in Jin [14] and other single objective studies [18,19], this paper deals with an improved multi objective method to obtain simple and accurate ESNN. The proposed method evolves toward optimal values defined by several objectives with model accuracy and ESNN's structure to improve performance for classification problems. The remaining parts of this paper are organized as follows: Methods including: Evolving Spiking Neural Network, Multi Objective Differential Evolution are presented in section 2. In addition, section 3 clarifies the proposed method Memetic Harmony Search Multi-Objective Differential Evolution with Evolving Spiking Neural Network (MEHSMODE-ESNN) used in this paper, Experimental design is discussed in section 4, section 5 explains in detail the results and discussion, and finally, section 6 concludes the paper with future works.

Methods

This section reviews the vital foundation of evolving spiking neural network (ESNN) and discusses the evolutionary algorithms that have been utilized for enhancement. In the first part, an introduction of ESNN, neuron coding, learning method and ESNN design, and its algorithm is presented. The second part focuses on the algorithms which are used for improvement of ESNN. The concepts and methods of multi objective optimization (MOO) are highlighted. After that, the literature focuses on the working of differential evolution, harmony search (HS) algorithm and memetic approach which is used to improve and enhance the performance of classification .

Evolving spiking neural network (ESNN)

Currently, several enhancements of SNN have been proposed. Wysoski improved one of these new models known as Evolving Spiking Neural Network (ESNN) [20]. Generally, ESNN used the principles of evolving connectionist systems (ECOS) where neurons are created incrementally [21,22]. ESNN can learn data gradually by one-pass propagation of the data through creating and merging spiking neurons [23]. making it possible to attain very fast learning in an ESNN [24]. The learning of ESNN has many good advantages as it can be applied incrementally, is adaptive and theoretically 'lifelong'. Therefore, the system can learn any new pattern via creating new output neurons, connecting them to the input neurons and merging with the similar ones [25-27]. This model stands on two principles: possibility of establishment of new classes and the merging of the similarities. The encoding method which is used for ESNN is the population as explained in Bohte [26]. The population distributes a single input value to multiple input neurons denotes as M. Each input neuron holds a firing time as input spikes. Firing times can be calculated which represent the input neuron *e* using the intersection of Gaussian function. Equations 1 and 2 have been used to calculate the centre and the width respectively with the variable interval of [Emin, Emax]. The width of each Gaussian receptive field is controlled by the parameter β.

$$R = E_{\min} + (2*e-3)/2*(E_{\max} - E_{\min})/(M-2)$$

$$\sigma = 1/\beta\,(E_{\max} - E_{\min})/(M-2) where 1 \le \beta \le 2$$

```
step 1: Compute firing time of pre-synaptic neuron f from input sample
step 2: Prepare the initialization of neuron repository R
Step 3: Determine ESNN parameters (Sim, Mod and C) between the range [0,1] for each of them
step 4: for all input sample e related to the same output class do
step 5: Determine weight for all pre-synaptic neuron where:
    $w_f = Mod_e^{order(f)}$
step 6: Calculate
    $PSP_{\max}(e) = \sum w_{fe} * Mod_e^{order(f)}$
step 7: Acquire PSP threshold value $\theta = PSP_{\max}(e) * C$
step 8: if the trained weight vector <= Sim of trained weight in R then
step 9: Merge threshold value weight and with most similar neuron
step 10: $w = \frac{w(new) + w*T}{T+1}$
step 11: $\theta = \frac{\theta(new) + \theta*T}{T+1}$
where N is number of merge before
step 12: else
step 13: Add new neuron to output neuron repository R
step 14: end if
step 15: end for (Repeat to all input samples for other output class)
```

Thorpe model [28] is similar to the Fast Integrate and Fire Model used in our paper. Thorpe's model shows that the earliest spikes received by a neuron will get a better weight depending on the later spikes. The Post-Synaptic Potential (PSP) will fire and become disabled only if it beats the threshold value. The computation of PSP of neuron *e* can be shown in Equation 3,

$$\mu_e = \begin{cases} 0 & \text{if fired} \\ \sum Wfe * Mod_e^{order(f)} & \text{otherwise} \end{cases}$$

where W_{fe} is considered as the weight of pre-synaptic neuron *f*; Mod *e* is the parameter factor of modulation with an interval of [0,1] and order (*f*) denotes the rank of spike emitted by the neuron. The order (*f*) starts with 0 if it spikes first among all pre-synaptic neurons and increases according to the firing time. Moreover, each training sample creates a new output neuron in the One-pass Learning algorithm. ESNN training steps can be understood from algorithm 1 as illustrated above in section 2. Figure 1 depicts a simplified architecture of ESNN model which was explained in more detail in Schliebs [3].

Figure: 1 A simplified architecture of ESNN [27].

The training starts with initialization of three ESNN parameters - modulation factor (*Mod*), proportion factor (*C*) and similarity

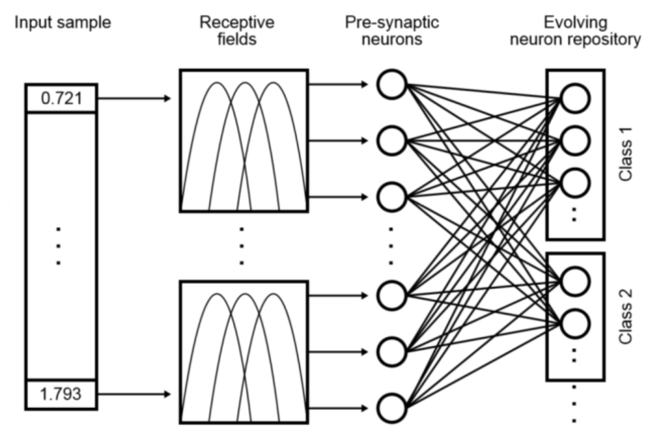


Figure 1: A simplified architecture of ESNN [1].

value (*Sim*) in the interval [0, 1]. *Mod* is the modulation factor of the Thorpe neural model. The firing threshold θ is calculated as based on proportion factor (*C*) with a value between [0, 1]. As the training process continues, every sample produces an output neuron. The similarity of output neurons is calculated according to the Euclidean distance between the weight vectors of the neurons. The parameter *Sim* controls the similarity distance. In the training stage, the output neuron stores the computed weight of all pre-synaptic neurons, a threshold value to determine when the output neuron will spike and the class label to which the input sample belongs. In the testing stage, the multiple pre-synaptic neurons encode each testing sample to spikes. After that, the PSP of the output class neurons is calculated. Once the neuron attains definite amount of spikes and the PSP exceeds the threshold value, it fires an output spike and becomes disabled. The testing sample belongs to the output class defined by the output neuron that fires first among all output neurons.

Multi-objective differential evolution (MODE)

This section discusses MOO and Evolutionary algorithms. First, the concept of MOO is introduced followed by a discussion of the essential points of MOO methods. Then, the concept of the Evolutionary Algorithm (EA) mainly Differential Evolution (DE) is clarified. After that, a brief discussion of the MOEA for DE enhancement is explained. Finally, the related works of this study are discussed in-depth.

Multi-objective optimization (MOO): MOO adds a new concept to the previous concepts of optimization. The process of systematically optimizing a set of objective functions at the same time is known as multi objective optimization (MOO) or vector optimization [29]. According to Lu [30] the optimal solutions obtained by individual optimization of the objectives are not a feasible solution to the multi-objective problem. Most practical optimization problems need the synchronized optimization of more than one objective function.

Methods of MOO algorithms: Typical methods aggregate the objectives into one objective function by using decision making before search. Conversely, the parameters of this function are assorted by the optimizer systematically. A number of optimization runs with various parameter settings are conducted with the intention of reaching a set of solutions. Essentially, this process is independent of the primary optimization algorithm. A number of examples of this class of techniques are the weighting method [31], the constraint method [31], goal programming [32], In exchange for the different methods; these three common methods are briefly discussed here.

Weighting Method: All objectives are multiplied by weighting coefficient (w_i) After that, all new functions are added together to get a single cost function. Finally, Single Objective (SO) method can be used to solve this new single cost function. Mathematically, the latest function is written as

$$f(x)=\sum^{I} w_i f_i(x)$$

where $0 \le w_i \le 1$ and $\sum_{i=1}^{I} w_i = 1$

Constraint method: Constraint Method depends on transforming *k-1* of the *k* objectives into constraints. The remaining objective, which can be selected subjectively, is the objective function of the resulting SO:

Maximize $y = f(x) = f_h(x)$

Subject to $e_i(x) = f_i(x) \ge \varepsilon_i$, $(1 \le i \le I, i \ne h)$, $x \in X_f$

The lower bounds, ε_i , are the parameters that are diverse by the optimizer to find multiple optimal solutions.

Goal programming method: According to Philipson and Ravindran [33], Goal Programming Method is considered as the most well-known method of solving MOPs. It was initially enhanced by Charnes [34] and Ijiri [35]. This method depends on ranking the goals as created by the designer. Finally, the SO function is considered as the minimization of the deviations from these goals.

EA algorithms: EAs have been utilized to optimize ESNNs, which have been used to solve the problems of optimization learning. The right classification measures are defined by the number of true positives (TPs) and the number of true negatives (TNs). Additionally, the misclassifications made by the classifier are determined by the number of false positives (FPs) and the number of false negatives (FNs). Many EAs types have been used for optimization, but DE algorithm is used in this study for many reasons, which will be explained in sections below:

Differential evolution (DE): In global optimization, DE is considered as one of the mainly powerful tools of global optimization in EAs [36]. Compared to some other competitive optimizers, DE has several strong advantages: it is much simpler to implement, has much better performance, has a small number of control parameters and has low space complexity [37]. DE, which belongs to the EAs, uses three steps: mutation, crossover and selection. In the stage of initialization, a population of individual candidates, each of dimension $N_{variables}$ (number of decision variable in the optimization problem), is randomly generated over the feasible region. A typical value of an individual is about 5-10 times $V_{ariables}$ in order to guarantee that DE has enough to work with. The fitness of every individual candidate is evaluated. Out of these individual candidates, one is randomly selected as the target candidate. The DE algorithm contains three parameters: size of population $N_{individuals}$, mutation constant *F* and crossover constant *CR*.

The MODE algorithm: According to Schaffer in Schaffer [38], Vector evaluated genetic algorithm (VEGA) is considered as the first multi-objective proposed in the literature. VEGA is an adapted single-objective genetic algorithm with a modified selection method. Many literature surveys found provide more detail about the history and enhancement of multi objective with GA like Zitzler et al. [39] and Deb [40]. Unlike GAs, where binary encoding can be utilized, DE solutions are coded with real values. More recently, evolutionary algorithms have also been adapted to solve optimization problems with multiple conflicting objectives by approximating the Pareto set in such problems such as in Qasem, Shamsuddin [41]. A complete tutorial on evolutionary multi-objective optimization with DE can be found in Mezura-Montes [42].

Harmony search algorithm

The HS algorithm was created by Geem [43] and is related to finding the harmony while playing music. In music improvisation there are three possible methods: playing the primary music, playing similarly to the original music and playing music randomly. Three measures were put forward by Geem to explain these methods above, namely: harmony memory consideration, randomization and pitch adjustment [44].

The HS algorithm involves a number of optimization parameters, for instance: harmony memory (HM), harmony memory considering rate (HMCR), harmony memory size (HMS) and pitch adjusting rate (PAR). To understand how the parameters work, HM basically saves the possible vectors which are available in the space. HMS pinpoints how many vectors it stores. Thus, a new vector is generated by choosing the components of diverse vectors randomly in the HM [45].

The use of HS is necessary because the best harmonies will affect the new harmony memory. The effective value for the parameter recognized as harmony memory considering or accepting rate (r_{accept}) should be in the period [0,1]. Normally values of parameters are determined as r_{accep} =0.7 to approximately 0.95. Only if other harmonies are not ascertained well, will possibly incorrect solutions be presented [44-46].

The second method that can generate good solutions is known as the pitch adjustment technique. Parameters such as pitch adjusting rate ($r_{pa)}$ and pitch bandwidth ($b_{range)}$ are mostly responsible for generating new solutions from existing ones. Theoretically, the pitch can be adjusted linearly or non-linearly, except in practice, linear adjustment is used. So we have

$$X_{new} = X_{old} + b_{range} * \varepsilon \tag{6}$$

Where X_{old} is the current pitch or solution from the HM and X_{new} is the new pitch after the pitch adjusting action. This mostly generates a new solution by adding a minor random amount in the range [-1,1] [47].

The third method is randomization, which shows the ability of increasing the diversity of solutions similar to achieving global optimality. The pseudo-code shown below summarizes the three components of HS algorithms (HSAs) [48]. The probability of randomization can be indicated as

$$P_{random} = 1 - r_{accept} \tag{7}$$

and the actual probability of adjusting pitches is given by

$$P_{pitch} = r_{accept} * r_{pa} \tag{8}$$

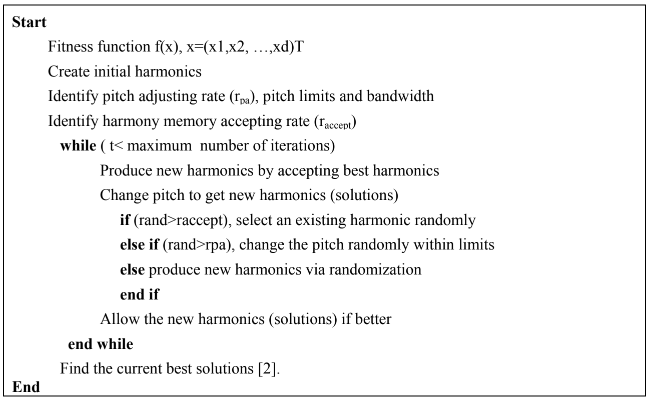
Figure 2 shows the pseudo-code of standard HSA showing each decision variable of HS.

Figure 3 demonstrates the flowchart for training of HS. The optimization process is made for every harmony as below:

1) Determine parameters and initialize the HM by random solutions.

2) Form a new harmony by using the HSA method and select which value will be set to each decision variable in the harmony (solution).

I. **New Harmony formation**: the new result from existing ones is created randomly with probability of **1 –** $\boldsymbol{r_{accept}}$ or instead with probability equal to $\boldsymbol{r_{accept}}$.



```
Start
      Fitness function f(x), x=(x1,x2, …,xd)T
      Create initial harmonics
      Identify pitch adjusting rate ($r_{pa}$), pitch limits and bandwidth
      Identify harmony memory accepting rate ($r_{accept}$)
        while ( t< maximum  number of iterations)
              Produce new harmonics by accepting best harmonics
              Change pitch to get new harmonics (solutions)
                 if (rand>raccept), select an existing harmonic randomly
                 else if (rand>rpa), change the pitch randomly within limits
                 else produce new harmonics via randomization
                 end if
              Allow the new harmonics (solutions) if better
         end while
        Find the current best solutions [2].
End
```

Figure 2: Pseudo-code of standard HSA [3].

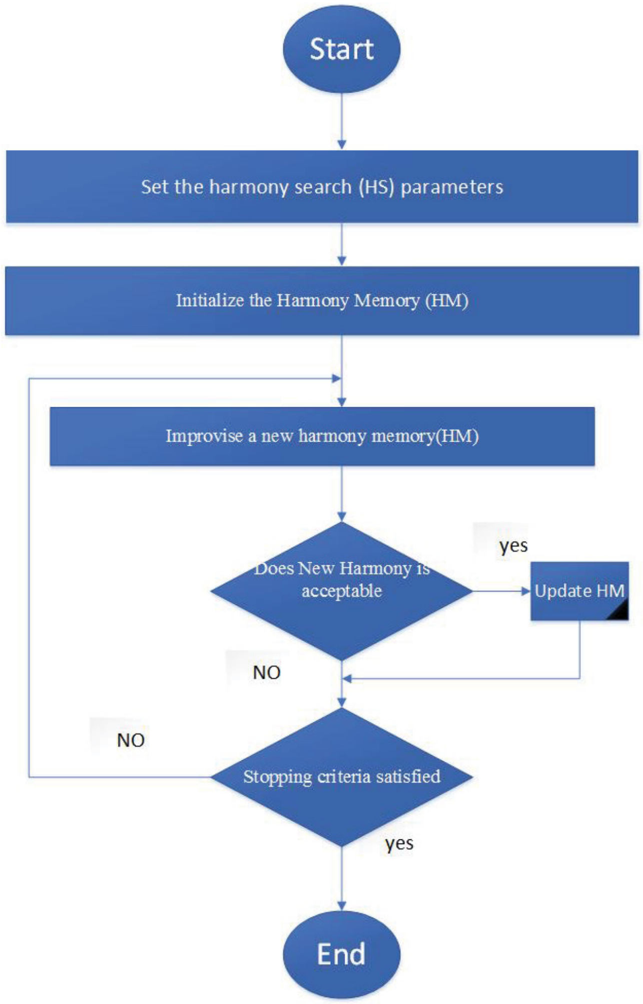

Start
Set the harmony search (HS) parameters
Initialize the Harmony Memory (HM)
Improvise a new harmony memory(HM)
Does New Harmony is acceptable
yes
Update HM
NO
NO
Stopping criteria satisfied
yes
End

Figure 3: Flowchart for training of HS.

II. **Adjustment**: the New Harmony features are updated through the probability of $\boldsymbol{r_{pa}}$.

III. **Selection**: the best harmony in the HM is chosen if the termination condition is true.

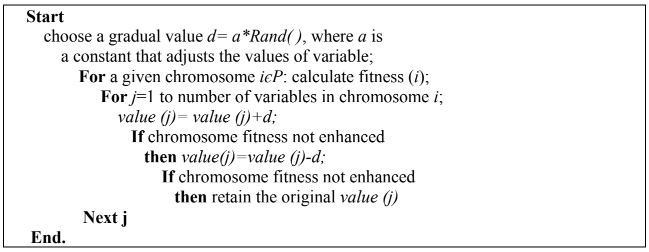
Many studies have been done to compare HS with other algorithms. One of these studies proves that HS is faster than PSO. Moreover, this study proves that HS has a significant convergence rate to meet the best solution [49]. In addition, HS can be integrated with other optimization algorithms such as GA, DE, PSO and many others in many real-world optimization problems.

Furthermore, Ameli [50] utilized a hybridization model for the tools of artificial such as HS, Tabu search (TS), simulated annealing (SA) and GA for probabilistic neural networks (PNNs).

HS has been used effectively in many problems of optimization and has numerous benefits compared to further traditional algorithms of optimization; these advantages are summed up below:

a. No need for more computational effort in achieving the results for HSA.

b. No need for derivative information for HSA for the reason that the use of stochastic random seeking strategy.



```
Start
  choose a gradual value $d= a*Rand( )$, where $a$ is
    a constant that adjusts the values of variable;
      For a given chromosome $i \in P$: calculate fitness ($i$);
        For $j$=1 to number of variables in chromosome $i$;
          value (j)= value (j)+d;
            If chromosome fitness not enhanced
              then value(j)=value (j)-d;
                If chromosome fitness not enhanced
                  then retain the original value (j)
      Next j
End.
```

Figure 4: Pseudo-code of MA [4] .

c. Integrates every existing solution vector in creating a new solution, which led to success of better results and robustness of the HS method [51].

Memetic technique

The memetic algorithm (MA) was established by Moscato [52]. This algorithm was motivated by the principles of natural evolution of Darwin and Dawkins' notion of a meme. Moscato considered MA as a type of population-based integrated GA with individual learning. MAs are now used in several areas, including hybrid evolutionary algorithms.

MAs are used in order to tackle MOPs [53]. The benefit of MAs in EAs is that they use local search for the presented problem [54]. The pseudo-code of MA local search is shown in Figure 4.

The Proposed Hybridization of Memetic Harmony Search Multi-objective DE based ESNN (MEHSMODE-ESNN)

This section discusses the proposed algorithm called MEHSMODE-ESNN which is a memetic harmony search multi-objective optimization approach for ESNN training. This algorithm will simultaneously determine the ESNN structure (pre-synaptic neurons) and its corresponding model parameters by treating this problem as a multi-objective optimization problem. In MEHSMODE-ESNN, the backpropagation (BP) method is utilized to solve the problem of convergence of the normal algorithm.

The pre-synaptic neuron of ESNN is represented as a candidate. HS and MOO are combined to carry out fitness evaluation and mating selection schemes.

MEHSMODE-ESNN starts by collecting, normalizing and reading the dataset. The maximum number of iterations are then set and the candidate size is computed. In addition, the pre-synaptic neurons and parameters of ESNN are determined randomly. A population of the hybrid method is then generated and initialized. Every candidate is evaluated for each iteration based on enhanced Differential Evolution using harmony Search algorithm. The proposed method stops after the maximum iterations is reached. Algorithm 2 describes the pseudo-code of MEHSMODE-ESNN.

Algorithm 2. Pseudo-code of MEHSMODE-ESNN

```
Initialize algorithm parameters: CR, F, HMCR, PAR and fitness function
Generate the initial harmony population HMS (t) at t=0 where t is the number
  of the actual iteration and HM as vector represents ESNN
  while not reaching NI do (NI is maximum of iteration)
    for all HM vector do
    use ESNN algorithm to find the HM vector fitness.
    achieve results of pre-synaptic neurons and ESNN parameters.
    evaluate the HM vector of population HMS (t) according to the fitness value.
    determine the best HM vector according to the best parents' values.
     improvise a new harmony
     for j ϵ 1, ..., Nvariables do
     randomly select any variable-i pitch in HM
     randomly adjust $u_j$ within a small bandwidth alpha α
     select any pitch within upper UBj and lower bounds $LO_j$
     end for
     if $v_j$ is better than the worst harmony in HM, $x_{worst}$, then
     replace $x_{worst}$ with $v_j$ in HM, then sort HM
     end if
     for j ϵ 1, ..., Nvariables do
     if rand ≤ CR or $j==j_{rand}$) then
     $u_j = xj;r_0 + F * (x_j;r_1 - x_j;r_2)$;
     else
     uj = xj;i;
     end if
     end for
    Applying local search (BP) to each of the harmony
     Perform local search ( BP ) algorithm
     evaluate the harmony HS on the basis of fitness functions
     update pre-synaptic neuron vector and ESNN parameters vector
     sort population HMS (t+1) according to their fitness values.
     t=t+1.
     end for
    end while
```

Experimental Design

This section presents the experiments of study on hybrid learning of ESNN network based on memetic harmony search multi-objective method. Many techniques can be used for validation, but k-fold cross-validation is used in this paper. The advantage of k-fold cross-validation over hold-out validation is that all observations are used for both training and testing, and each observation is used for testing exactly once. 10-fold cross-validation is commonly used [55]. In order to evaluate the effectiveness of the proposed method, a detailed empirical study is carried out on seven different data sets which is explained in detail below.

Data preparation

Several real-world data sets from UCI repository are used in this study to represent some of the most challenging problems in machine learning.

Many researchers have used these data sets as a benchmark in validating the performance of their algorithms. The key characteristics of these problems and their associated learning tasks are summarized in Tables 1 and 2.

Parameter	Description
Maximum number of iterations	The round number at which the optimization process will be stopped when it is met.
HM size	The solution (harmony) number which will be stored in the HM.
Number of decision variables	A number of decision variables are included in every solution.
HM consideration rate (r_{accept})	The rate of decision variables in the harmony are taken into account in the new harmony.
PAR (r_{pa})	The probability changing the decision variables values by adding a certain value.

Table 1: Description of parameters of the HSA algorithm.

Dataset	Attributes	Classes	Samples
Appendicitis	7	2	106
Haberman	3	2	306
Heart	13	2	297
Hepatitis	19	2	155
Ionosphere	34	2	351
Iris	4	3	150
Liver	6	2	345

Table 2: Description of data sets.

Parameter	MEHSMODE -ESNN
Population size	30
Maximal number of iterations	1000
(Mutation) constant F	0.9
Crossover constant CR	0.8
Dimensionality of problem	2
Harmony memory Consideration rate HMCR	0.9
Pitch adjusting rate PAR	0.3
Momentum	0.7
Maximum error	0.001
Learning rate	0.2

Table 3: Parameter settings for MEHSMODE -ESNN algorithm.

The learning phase of the proposed MEHSMODE-ESNN

Initially, the data set has been divided randomly into ten subsets of equal size. One subset is used as the testing data set, and the other nine subsets are used as the training data sets. The training and testing processes are repeated so that all the subsets are used as a testing data set. The training process of the hybrid MEHSMODE-ESNN is explained in section 3. There are many important parameters which can control the result of training in MEHSMODE-ESNN. The various parameter settings of the proposed method are tabulated in Table 3.

The performance of the algorithms is evaluated by conducting analysis on ten evaluations. In this experiment, the evolutionary process of the proposed algorithm is analyzed and the performance is evaluated accordingly.

The objective of this evaluation is to establish the effectiveness of the proposed method in designing ESNN network. This involves both the training of the ESNN network and the evolved structure (pre-synaptic neurons) of the ESNN network. The capabilities of the proposed method have been investigated through a comparison with each other that has been applied for classification problems. In order to evaluate the classification performance of the proposed method, the comparisons are conducted and compared with the standard ESNN, DE-ESNN, DEPT-ESNN, MODE-ESNN and HSMODE-ESNN methods. The results of all proposed methods in terms of all measures are presented in Tables 4-6. In these tables, the best results are highlighted in bold font. The results for all data sets involved are analysed based on structure of ESNN (number of pre-synaptic neurons), parameters values (Mod, Sim and Threshold), sensitivity (SEN), specificity (SPEC), geometric means (GM) and classification accuracy (ACC) for all data sets. The results of the proposed methods for each data set are analysed and presented in the following section.

Results and Discussion

This section displays the results of the proposed method MEHSMODE-ESNN compared to other methods. The results of MEHSMODE-ESNN are measured in terms of number of pre-synaptic neurons, SEN, SPE, GM and ACC. The experiments are run 10 times in the training and testing for all data sets.

Data set	ESNN	DE-ESNN	DEPT-ESNN	MODE-ESNN	HSMODE-ESNN	MEHSMODE-ESNN
Appendicitis	0.75	0.75	0.2953	0.5518	0.4471	0.3508
Haberman	0.75	0.75	0.3905	0.0899	0.7124	0.6544
Heart	0.75	0.75	0.6875	0.7761	0.6372	0.641
Hepatitis	0.75	0.75	0.2004	0.5728	0.4266	0.5535
Ionosphere	0.75	0.75	0.2833	0.7877	0.4405	0.3273
Iris	0.75	0.75	0.5596	0.4824	0.6451	0.5635
Liver	0.75	0.75	0.6977	0.78	0.2887	0.5754

Table 4: Comparison of results of all proposed algorithms in terms of the modulation factor parameter (Mod) for 10-fold cross-validation.

Data set	ESNN	DE-ESNN	DEPT-ESNN	MODE-ESNN	HSMODE-ESNN	MEHSMODE-ESNN
Appendicitis	0.9	0.9	0.7347	0.9535	0.6533	0.4804
Haberman	0.9	0.9	0.5292	0.7991	0.1968	0.6644
Heart	0.9	0.9	0.5337	0.1877	0.7498	0.8272
Hepatitis	0.9	0.9	0.5708	0.6988	0.6405	0.4088
Ionosphere	0.9	0.9	0.3099	0.5623	0.3791	0.4759
Iris	0.9	0.9	0.2789	0.6005	0.6073	0.8162
Liver	0.9	0.9	0.5916	0.6839	0.1569	0.7711

Table 5: Comparison of results of all proposed algorithms in terms of the similarity value parameter (Sim) for ten-fold cross-validation.

Data set	ESNN	DE-ESNN	DEPT-ESNN	MODE-ESNN	HSMODE-ESNN	MEHSMODE-ESNN
Appendicitis	0.1	0.1	0.336	0.4196	0.5728	0.4255
Haberman	0.1	0.1	0.4319	0.8234	0.6696	0.5888
Heart	0.1	0.1	0.8038	0.5456	0.6803	0.741
Hepatitis	0.1	0.1	0.2579	0.5088	0.2685	0.5622
Ionosphere	0.1	0.1	0.555	0.5645	0.3611	0.4548
Iris	0.1	0.1	0.8034	0.4941	0.5346	0.4833
Liver	0.1	0.1	0.5642	0.4824	0.6521	0.4306

Table 6: Comparison of results of all proposed algorithms in terms of the proportion factor parameter (Threshold) for ten-fold cross-validation.

Table 4 Shows the average results of all proposed algorithms in terms of parameter Mod. The Mod parameter importance comes from its representation for the connection weight in ESNN model. If a high value was selected, it means most weights will have a connection value which reflects a well presented weight patterns according to their output class. On the contrary, selecting too low values leads to ending up with most connections assigned with the zero weight value due to the nature of weight computation. The results in Table 4 show that HSMODE-ESNN produced the smallest value, on average, while DE-ESNN produced the highest value, on average. The findings show the ability of the proposed methods to produce the optimum values of Mod parameter which is different from data set to another. The nature of data set can affect the value of parameter. This clear effect is due to the correlation between parameter optimization and classification accuracy as integrated using the known Wrapper approach. Knowledge discovery from these results of Mod parameter shows that there is an important impact of high values of decision of the output classes of instances.

From the opposite position, the value of parameter Sim is important for the model ESNN. Higher values of Sim means more neurons with the similarity range that are merged while lower values means else. DE-ESNN produced the smallest value on all data sets and as well as on average, while MODE-ESNN produced the highest value on average. These results suggest that the proposed methods give the best values for Sim parameter in the range (0.5266, 0.5483) on average. For almost all data sets the results are shown in Table 5. It is noticed that no specific value can be applied to all problems because each problem has its own combination of parameters. Generally, each data set requires specific analysis to understand the problem it represents. Fast decision making demands a comprehensive understanding. Knowledge discovery from these results shows that the high values of Sim parameter leads to establishing better network architecture. This is due to the effect of the parameter Sim in controlling the merge rate of output neurons.

Table 6 Shows that the average results of all proposed algorithms in terms of parameter Threshold. The importance of the Threshold parameter is in its function of controlling the PSP threshold. The results in Table 6 show that DEPT-ESNN produced the smallest value on average with 0.45, whereas DE-ESNN provided the highest value on average with 0.75. The different values of Threshold parameter from data set to another for each proposed method reflects the influence of nature of data set and the hybrid method which control the process of the algorithm. Knowledge discovery from these results of Threshold parameter shows that there is a main influence of parameter values of making the decision. Higher values mean more spikes and time for this decision. However, if there are smaller values, few spikes are enough to fire an output spike and determine the class of instance.

The findings of the parameter models affect the evolution of ESNN. However, the impact of pre-synaptic neurons, which is considered to be the structure of ESNN in this study, needs to be analyzed, as in Figure 5.

Figure 6 shows the structure of the ESNN (pre-synaptic neurons) based on all proposed algorithms. As shown in Figure 6, MODE-ESNN has yielded the best structure on the appendicitis, Haberman, iris and liver data sets. At the same time, DE-ESNN yielded the best structure on hepatitis data set as well as on average. Furthermore, DEPT-ESNN gives the best structure on the heart data set, while HSMODE-ESNN provides the best structure on the ionosphere data set. Investigation of these results, as given in Figure 6, reveals that MODE-ESNN is the best structure of ESNN network compared to other methods for almost

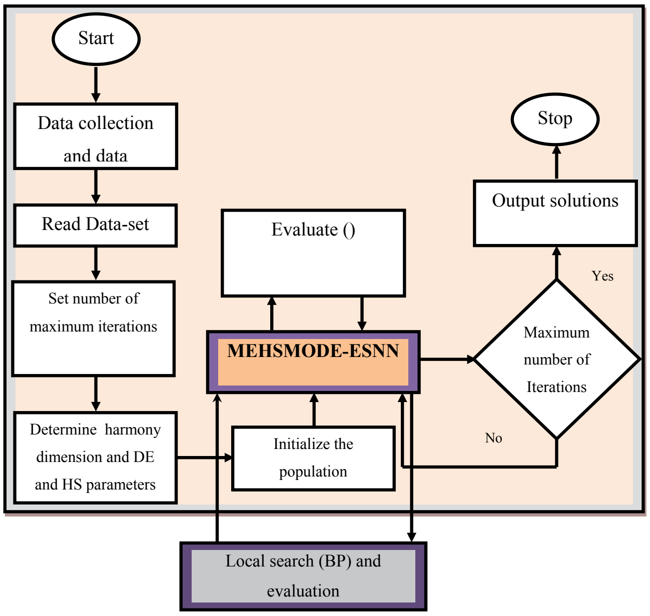

Start
Data collection and data
Read Data-set
Set number of maximum iterations
Determine harmony dimension and DE and HS parameters
Evaluate ()
MEHSMODE-ESNN
Initialize the population
Local search (BP) and evaluation
Stop
Output solutions
Yes
Maximum number of Iterations
No

Figure 5: Schematic representation of the proposed MEHSMODE-ESNN

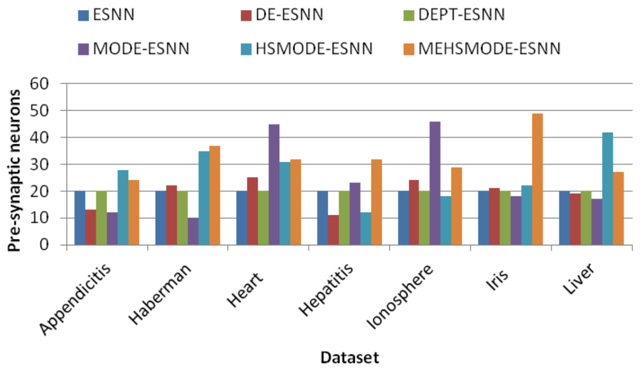


Figure 6: Comparison of the proposed methods for ten-fold cross-validation in terms of pre-synaptic neurons analysis.

all data sets. It is explicitly noted that the nature of data set has a big impact in determining the number of pre-synaptic neurons for each model. Besides, the algorithm itself plays a crucial role in deciding which number of pre-synaptic neurons is suitable for achieving the best performance. Knowledge discovery from these results of pre-synaptic neurons shows that the structure of model can be affected by the behavior of data set and combination of optimization. Moreover, corresponding to the final accuracy results, it demonstrates that the proposed methods give better results of pre-synaptic neurons than the original ESNN model.

Figs. 7-10 present the comparison between the proposed algorithm compared to other algorithms in terms of SEN, SPE, GM and ACC on the testing set for all data sets. The sensitivity analysis of the proposed methods is shown in Figure 7. The results show that HSMODE-ESNN generates the best values with 100% sensitivity for heart and hepatitis data sets, 75% for the ionosphere data set, 85.8% for the liver data set. For MEHSMODE-ESNN the best sensitivity values are obtained from Haberman data set with 99.5% and 100% for the hepatitis data set. Similarly to MODE-ESNN the best sensitivity values are achieved through the appendicitis data set with 83.5%. In addition, DE-ESNN yields the best sensitivity values for the iris data set with 100%.

It can be observed that HSMODE-ESNN has produced the highest results of sensitivity almost in all data sets compared to other methods (Figure 7). However, the proposed method is not good in obtaining better sensitivity for the appendicitis, Haberman and iris data sets. In medical diagnosis, high sensitivity rate means there are very few false negative which means not having the disease like in Haberman, heart and hepatitis. This is vital since test could change patients case because of the given negative results. On the other hand, it can be observed that Ionosphere data set obtains low sensitivity because of the nature of the data set. The distribution of Ionosphere data set makes the proposed methods produce much more false positives which leads to low sensitivity.

The proposed methods are further validated using the SPE analysis as given in Figure 8. As illustrated in Table 7 and Figure 8, DE-ESNN has proven the best SPE results with 62.12% for the hepatitis data set, 100% for the iris data set. Furthermore, MEHSMODE-ESNN produces the best SPE results with 71% for Haberman data set, 58.3% for the heart data set and 100% for the ionosphere data set. In addition, MODE-ESNN has the best SPE results with 55.1% for the liver data set, while ESNN gives the best SPE results for the appendicitis data set with 85% and 100% for the iris data set.

From observation (Table 7 and Figure 8), DE-ESNN and ESNN yield similar SPE values of 100% for the iris data set, whereas MEHSMODE-ESNN generates 100% SPE results for the ionosphere data set. In conclusion, MEHSMODE-ESNN and DE-ESNN are superior to other methods in almost all data sets and on average. It is argued that some of the proposed methods gain low specificity for imbalanced data set which reflects upon increased false positive case. Even though the proposed methods have the highest sensitivity , they lose the cost of specificity. These results reflect that although they have higher true positive rate at classifying the minority class, it performed poorly at classifying the majority class due to its lower specificity.

The results of all proposed algorithms based on GM measure are shown in Table 8 and Figure 9. It can be noticed that HSMODE-ESNN has the highest GM results with 75.63% for the heart data set, 82.75% for the ionosphere data set. As for MEHSMODE-ESNN, the best GM results are obtained from the Hagerman data set with 84.05% and 79.31% for the hepatitis data set. In addition, DEPT-ESNN

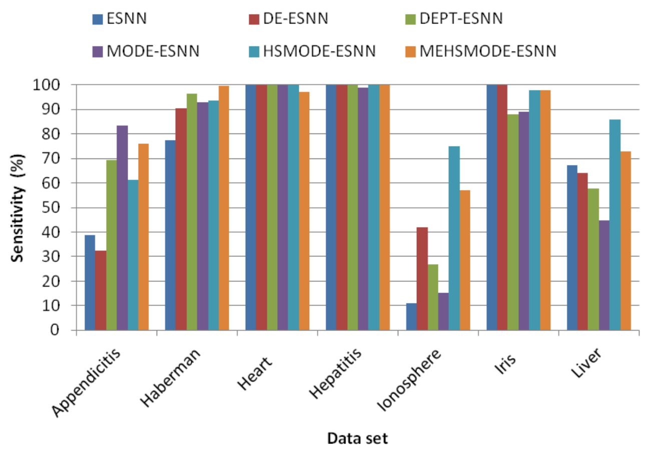


Figure 7: Comparison of the proposed methods for ten-fold cross-validation in terms of sensitivity analysis.

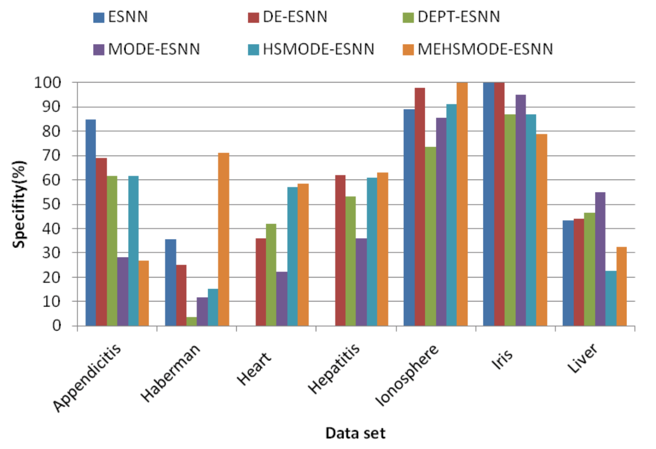


Figure 8: Comparison of the proposed methods for ten-fold cross-validation in terms of SPE analysis.

Data set	ESNN	DE-ESNN	DEPT-ESNN	MODE-ESNN	HSMODE-ESNN	MEHSMODE-ESNN
Appendicitis	**85**	69	61.7	28.3	61.6	26.7
Haberman	35.7	25.1	3.7	11.7	15.3	**71**
Heart	0	36.12	42.11	22.16	57.2	**58.3**
Hepatitis	0	62.12	53.21	36	61.1	**62.9**
Ionosphere	89.1	98	73.6	85.4	91.3	**100**
Iris	**100**	**100**	87	95	87	79
Liver	43.2	44.1	46.6	**55.1**	22.5	32.3

Table 7: SPE analysis for all proposed methods for ten-fold cross-validation.

demonstrates the best performance with 65.44% GM for the heart data set. For DE-ESNN, best performance is obtained with 100% GM for the iris data set. Finally, ESNN has achieved 100% GM values for the iris data set and 53.88% for the liver data set. GM is also smaller than that of MEHSMODE-ESNN due to its lower sensitivity value as in Ionosphere or specificity value as in appendicitis data set.

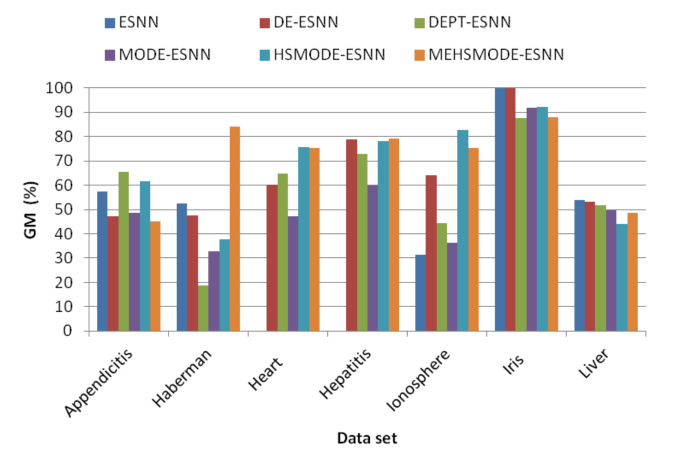


Figure 9: Comparison of the proposed methods for ten-fold cross-validation in terms of GM analysis.

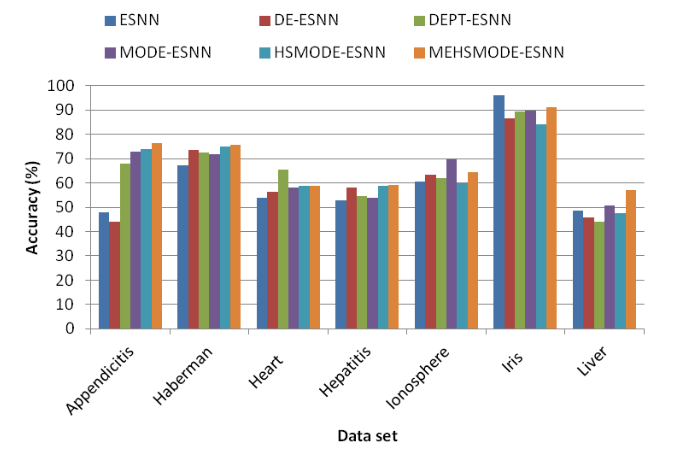


Figure 10: Comparison of the proposed methods for ten-fold cross-validation in terms of accuracy analysis.

The results of all proposed algorithms based on ACC measure, are shown in Table 9 and Figure 10. It can be noticed that MEHSMODE-ESNN gives the highest accuracy results with 76.31% for appendicitis, 75.57% for Haberman, 59.34% for hepatitis and 57.07% for liver data sets. As for MODE-ESNN, the best accuracy results are obtained from the ionosphere data set with 69.55%. In addition, ESNN demonstrates better performance with 95.99% accuracy for the iris data set. Finally, DEPT-ESNN has achieved 58.93% ACC value for the heart data set

These investigations show that MEHSMODE-ESNN produces the best accuracy in almost data sets, compared to other methods.

All the experiments are evaluated and analyzed based on the structure of the ESNN network (pre-synaptic neurons), the SEN, SPE, GM and ACC. The results of ten-fold cross-validation are summarized in Table 10.

For detailed analysis, evaluation of the performance of all proposed methods is compared to investigate the feasibility of the proposed methods in performing pattern classification tasks. The best method obtained from all proposed methods is selected in terms of all performance measures from the final analysis representing each test case.

For the appendicitis data set, MODE-ESNN produced the best results in terms of SEN and ESNN structure measures. As for MEHSMODE-ESNN, the best results are obtained in terms of ACC. Furthermore, DEPT-ESNN provided the best results for GM measure. For the Hagerman data set, MEHSMODE-ESNN produced the highest values in terms of SEN, SPE, GM and ACC measures; while MODE-ESNN has yielded the highest ESNN structure value measure. For the heart data set, MEHSMODE-ESNN demonstrated the best results in terms of SPE measure. Furthermore, HSMODE-ESNN provided the best results for SEN and GM measures. In addition, DEPT-ESNN generated the best results for ACC measure and ESNN structure value. For the hepatitis data set, MEHSMODE-ESNN presented the best results in terms of GM, SEN, SPE and ACC measures. Moreover, DE-ESNN has yielded the best results for ESNN structure measure. For the ionosphere data set, HSMODE-ESNN gave the best results in terms of

Data set	ESNN	DE-ESNN	DEPT-ESNN	MODE-ESNN	HSMODE-ESNN	MEHSMODE-ESNN
Appendicitis	57.5	47.36	**65.44**	48.61	61.5	45.02
Haberman	52.57	47.67	18.88	32.97	37.84	**84.05**
Heart	0	60.1	64.89	47.08	**75.63**	75.26
Hepatitis	0	78.82	72.95	59.7	78.17	**79.31**
Ionosphere	31.45	64.16	44.33	36.15	**82.75**	75.49
Iris	100	**100**	87.5	91.95	92.34	87.99
Liver	**53.88**	53.13	51.81	49.68	43.93	48.56

Table 8: GM analysis for all proposed methods for ten-fold cross-validation.

Data set	ESNN	DE-ESNN	DEPT-ESNN	MODE-ESNN	HSMODE-ESNN	MEHSMODE-ESNN
Appendicitis	48	44	68	73	74	**76.31**
Haberman	67.32	73.66	72.66	72	75	**75.57**
Heart	53.99	56.33	**65.57**	58.2	58.66	58.93
Hepatitis	52.67	58	54.67	54	58.67	**59.34**
Ionosphere	60.57	63.43	62.14	**69.55**	60	64.29
Iris	**95.99**	86.67	89.33	89.71	84	91.15
Liver	48.57	45.71	44	50.57	47.43	**57.07**

Table 9: Accuracy analysis for all proposed methods for ten-fold cross-validation.

Data set	Analysis Criteria	ESNN	DE-ESNN	DEPT-ESNN	MODE-ESNN	HSMODE-ESNN	MEHSMODE-ESNN
Appendicitis	STRUCTURE				•		
	SEN				•		
	SPE	•					
	GM			•			
	ACC						•
Haberman	STRUCTURE				•		
	SEN						•
	SPE						•
	GM						•
	ACC						•
Heart	STRUCTURE			•			
	SEN					•	
	SPE						•
	GM					•	
	ACC			•			
Hepatitis	STRUCTURE		•				
	SEN						•
	SPE						•
	GM						•
	ACC						•
Ionosphere	STRUCTURE					•	
	SEN					•	
	SPE						•
	GM					•	
	ACC				•		
Iris	STRUCTURE				•		
	SEN		•				
	SPE		•				
	GM		•				
	ACC	•					
Liver	STRUCTURE				•		
	SEN					•	
	SPE				•		
	GM	•					
	ACC						•
Number of wins	STRUCTURE	0	1	1	4	1	0
	SEN	0	1	0	1	3	2
	SPE	1	1	0	1	0	4
	GM	1	1	1	0	2	2
	ACC	1	0	1	1	0	4

Table 10: Summary analysis of all proposed methods.

ESNN structure, SEN and GM measures, while MEHSMODE-ESNN provided the best SPE results. In addition, MODE-ESNN has produced the best results in terms of the ACC measure. For the iris data set, MODE-ESNN provided the best results in terms of ESNN structure, while DE-ESNN produced the best results in terms of SEN, SPE and GM measures. In addition, ESNN provided the best results in terms of ACC measure. For the liver data set, MODE-ESNN demonstrated the best results in terms of SPE measure and ESNN structure. In addition, HSMODE-ESNN demonstrated the best results in terms of sensitivity measure, while MEHSMODE_ESNN has demonstrated the best ACC results.

On the other hand, detail investigation, as given in Table 10, reveals that MEHSMODE-ESNN produced the best results for SPE and ACC for four of the seven data sets, respectively. In addition, MODE-ESNN produced the best ESNN structure measure in four of the seven data sets. As for HSMODE-ESNN, the best SEN results are obtained from three of the seven data sets. Moreover, both MEHSMODE-ESNN and HSMODE-ESNN produced the best results for GM measures from two of the seven data sets.

A careful look into the results draws an important conclusion: the variance in performance of the proposed methods on a particular data set is significantly smaller compared to the variance in performance of the same proposed method on different data sets. The statement holds for seven data sets. Consequently, it can be said that performance is strongly dependent on the nature of the data set. The important factors that affect the classification performance of the proposed methods are high-dimensionality, samples, missing values, multiple classes, imbalanced classes and noise.

Statistical analysis of the proposed hybridization models with ESNN

The Friedman test is carried out to test whether k random samples drawn from a population k have the same mean. Consequently, n data sets of appendicitis, Haber man, hepatitis, heart, ionosphere, iris and liver have been used to obtain N samples from k random samples of accuracy performance measurements. Program IBM SPSS Statistics 20 was utilized to conduct these statistical experiments. The results are significantly different among all proposed methods with chi-square =12.469, df=5 with a significance level of p=0.029 which indicates that the results are true with a confidence interval of 95%. Thus, we reject the null hypothesis of the Friedman test which states that all algorithms are behaviourally same. The algorithms are significantly different in their behaviour.

Conclusion and Future Works

A hybrid MEHSMODE-ESNN method was proposed in this paper to determine the optimal number of pre-synaptic neurons as well as the parameters for a given dataset simultaneously. A comparative study has been conducted between MEHSMODE-ESNN and ESNN, DE-ESNN and DEPT-ESNN, MODE-ESNN and with other data mining methods to show the performance improvement of ESNN. Both MODE-ESNN and HSMODE-ESNN other methods have been used to perform the classification on standard data sets. The results show that MEHSMODE-ESNN is able to classify data set with mostly better accuracy than the other algorithms. Moreover, MEHSMODE-ESNN uncovered the optimum parameters of ESNN which is considered important for good accuracy. Additionally, MEHSMODE-ESNN mostly shows better results in SPE, GM and ACC measurements. These hybrid multi objective evolutionary algorithms with ESNN motivate researchers to investigate the usefulness of integration of other new Meta heuristic algorithms to enhance ESNN.

References

1. Ahmed FYH, Shamsuddin SM, Hashim SZM (2013) Improved spikeprop for Using Particle Swarm Optimization, Mathematical Problems in Engineering.
2. Gerstner W, Kistler WM (2002) Spiking neuron models: Single neurons, populations, plasticity. Cambridge University press, USA.
3. Schliebs S, Defoin-Platel M, Worner S, Kasabov N (2009) Integrated feature and parameter optimization for an evolving spiking neural network: Exploring heterogeneous probabilistic models, Neural Networks 22: 623-632.
4. Hamed (2012) Novel Integrated Methods of Evolving Spiking Neural Network and Particle Swarm Optimization in Auckland University of Technology.
5. Abbass HA (2003) Speeding up back propagation using multi objective evolutionary algorithms, Neural Computation 15: 2705-2726.
6. Igel C, Zitzler E, Coello C, Aguirre AH (2005) Multi-objective Model Selection for Support Vector Machines in Evolutionary Multi-Criterion Optimization, Springer, Berlin Heidelberg: 534-546.
7. Jin Y (2006) Multi-objective machine learning, Springer.
8. Jin Y, Okabe T, Sendoff B (2004) Neural network regularization and ensembling using multi-objective evolutionary algorithms in Evolutionary Computation. Congress on IEEE pp: 1-8.
9. Jin Y, Sendhoff B, Körner E (2005) Evolutionary Multi-objective Optimization for Simultaneous Generation of Signal-Type and Symbol-Type Representations: C. Coello, A. Hernández Aguirre, E. Zitzler (Eds.) Evolutionary Multi-Criterion Optimization, Springer, Berlin Heidelberg pp: 752-766.
10. Fieldsend JE, Singh S (2005) Pareto evolutionary neural networks, Neural Networks, IEEE Transactions on 16: 338-354.
11. Abbass HA (2003) Pareto neuro-evolution: Constructing ensemble of neural networks using multi-objective optimization: Evolutionary Computation, CEC'03. The 2003 Congress on IEEE pp: 2074-2080.
12. Chandra X, Yao (2004) DIVACE: Diverse and accurate ensemble learning algorithm: Intelligent Data Engineering and Automated Learning–IDEAL Springer pp: 619-625.
13. Tan KC, Lee TH, Khor EF (2001) Evolutionary algorithms with dynamic population size and local exploration for multi-objective optimization, Evolutionary Computation, IEEE Transactions 5: 565-588.
14. Jin Y, Wen R, Sendoff B (2007) Evolutionary multi-objective optimization of spiking neural networks, in: Artificial Neural Networks–ICANN Springer pp: 370-379.
15. Wang G, Guo L (2013) A novel hybrid bat algorithm with harmony search for global numerical optimization, Journal of Applied Mathematics.
16. Purnomo HD, Wee HM (2014) Maximizing production rate and workload balancing in a two-sided assembly line using Harmony Search, Computers & Industrial Engineering 76: 222-230.
17. Gao LQ, Li S, Kong X, Zou DX (2014) On the iterative convergence of harmony search algorithm and a proposed modification, Applied Mathematics and Computation 247: 1064-1095.
18. Saleh AY, Hameed HNBA, Najib M, Salleh M (2014) A Novel hybrid algorithm of Differential evolution with Evolving Spiking Neural Network for pre-synaptic neurons Optimization . Int J Advance Soft Compu Appl.
19. Saleh AY, Shamsuddin SM, Hamed HNBA (2014) Parameter Tuning of Evolving Spiking Neural Network with Differential Evolution Algorithm, International Conference of Recent Trends in Information and Communication Technologies.
20. Wysoski S, Benuskova L, Kasabov N (2006) On-line learning with structural adaptation in a network of spiking neurons for visual pattern recognition, Artificial Neural Networks–ICANN pp: 61-70.
21. Kasabov N, Dhoble K, Nuntalid N, Indiveri G (2013) Dynamic evolving spiking neural networks for on-line spatio-and spectro-temporal pattern recognition, Neural Networks 41: 188-201.
22. Schliebs S, Kasabov N (2013) Evolving spiking neural network- A survey, Evolving Systems 4: 87-98.
23. Kasabov NK (2014) neucube: A spiking neural network architecture for mapping, learning and understanding of spatio-temporal brain data, Neural Networks 52: 62-76.
24. Kasabov N (2012) Evolving spiking neural networks and neurogenetic systems for spatio-and spectro-temporal data modelling and pattern recognition: Advances in Computational Intelligence, Springer 234-260.
25. Kasabov N, Feigin V, Hou ZG, Chen Y, Liang L, et al. (2014) Evolving spiking neural networks for personalised modelling, classification and prediction of spatio-temporal patterns with a case study on stroke, Neurocomputing 134: 269-279.
26. Bohte SM, Kok JN, La Poutre H (2002) Error-back propagation in temporally encoded networks of spiking neurons, Neurocomputing 48: 17-37.
27. Hamed N, Kasabov Z, Michlovský S (2009) Shamsuddin, String Pattern Recognition Using Evolving Spiking Neural Networks and Quantum Inspired Particle Swarm Optimization, in Neural Information Processing, Springer, Berlin Heidelberg pp: 611-619.
28. Thorpe S (1997) How can the human visual system process a natural scene in under 150ms? Experiments and neural network models, D-Facto public, ISBN: 9600049.
29. Marler RT, Arora JS (2004) Survey of multi-objective optimization methods for engineering, Structural and multidisciplinary optimization 26: 369-395.
30. Yen GG, Lu H (2003) Dynamic multi-objective evolutionary algorithm: adaptive cell-based rank and density estimation, Evolutionary Computation, IEEE Transactions on 7: 253-274.
31. Cohon JL (2010) Multi-objective programming and planning, New York.
32. Steuer R (1986) Multiple criteria optimization: theory, computation, and application. New York.
33. Philipson R, Ravindran A (1978) Application of goal programming to machinability data optimization, Journal of Mechanical Design 100: 286-291.
34. Charnes WW (1957) Cooper Management models and industrial applications of linear programming, Management Science 4: 38-91.
35. Ijiri Y (1965) Management goals and accounting for control, North Holland Publication.

36. Abbass HA, Sarker R, Newton C (2001) PDE: A Pareto-frontier differential evolution approach for multi-objective optimization problems, in: Evolutionary Computation. Proceedings of the 2001 Congress on IEEE pp: 971-978.

37. Das S, Suganthan PN (2011) Differential evolution: A survey of the state-of-the-art, Evolutionary Computation, IEEE Transactions on 15: 4-31.

38. Schaffer JD (1985) Multiple Objective Optimizations with Vector Evaluated Genetic Algorithms, in: Proceedings of the 1st International Conference on Genetic Algorithms, L. Erlbaum Associates Inc pp: 93-100.

39. Zitzler E, Laumanns M, Bleuler S (2004) A tutorial on evolutionary multiobjective optimization, in: Met heuristics for multiobjective optimization, Springer pp: 3-37.

40. Deb K, Thiele L, Laumanns M, Zitzler E (2005) Scalable test problems for evolutionary multiobjective optimization, Springer.

41. Qasem SN, Shamsuddin SM (2011) Memetic elitist Pareto differential evolution algorithm based radial basis function networks for classification problems Applied Soft Computing 11: 5565-5581.

42. Mezura-Montes E, Reyes-Sierra M, Coello CAC (2008) Multi-objective optimization using differential evolution: a survey of the state-of-the-art: Advances in differential evolution, Springer pp: 173-196.

43. Geem ZW, Kim JH, Loganathan G (2001) A new heuristic optimization algorithm: harmony search, Simulation 76: 60-68.

44. Yang XS (2009) Harmony search as a met heuristic algorithm, in: Music-inspired harmony search algorithm, Springer pp: 1-14.

45. Abdolalipour A (2012) Alibabaee, Harmony Search algorithm, Int J Acad Res Appl Sci 1: 13-16.

46. Moh'd O Aliam, Mandava R (2011) The variants of the harmony search algorithm: An overview. Artificial Intelligence Review 36: 49-68.

47. Lee KS, Geem ZW (2005) A new meta-heuristic algorithm for continuous engineering optimization: harmony search theory and practice, Computer methods in applied mechanics and engineering 194: 3902-3933.

48. Yang XS (2009) Harmony search as a metaheuristic algorithm. Music-inspired harmony search algorithm, Springer pp: 1-14.

49. Soltani MAZ, Haghighat AT, Rashidi H, Chegini TG (2011) A Couple of Algorithms for K-Coverage Problem in Visual Sensor Networks.

50. Moslehpour S (2011) Transmission Network Expansion Planning Based on Hybridization Model of Probabilistic Neural Networks and Harmony Search Algorithm.

51. Razfar MR, Zinati RF, Haghshenas M (2011) Optimum surface roughness prediction in face milling by using neural network and harmony search algorithm, The International Journal of Advanced Manufacturing Technology 52: 487-495.

52. Moscato P (1989) On evolution, search, optimization, genetic algorithms and martial arts: Towards memetic algorithms, Caltech concurrent computation program, C3P Report p: 826.

53. Neri F, Cotta C (2012) Memetic algorithms and memetic computing optimization: A literature review, Swarm and Evolutionary Computation 2: 1-14.

54. Merz P, Freisleben B (1999) A comparison of memetic algorithms, tabu search, and ant colonies for the quadratic assignment problem in: Evolutionary Computation CEC. Proceedings of the 1999 Congress on IEEE.

55. Elbeltagi E, Hegazy T, Grierson D (2005) Comparison among five evolutionary-based optimization algorithms. Advanced Engineering Informatics 19: 43-53.

Citation: Saleh AY, Shamsuddin SM, Hamed HNA (2016) Memetic Harmony Search Algorithm Based on Multi-objective Differential Evolution of Evolving Spiking Neural Networks. Int J Swarm Intel Evol Comput 5: 130. doi: 10.4172/2090-4908.1000130