Neurocomputing 249 (2017) 345–356

Contents lists available at ScienceDirect

Neurocomputing

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

A double-elimination-tournament-based competitive co-evolutionary artificial neural network classifier

Bee Yan Hiew[a], Shing Chiang Tan[a,*], Way Soong Lim[b]

[a] Faculty of Information Science and Technology, Multimedia University, Jalan Ayer Keroh Lama, Bukit Beruang, 75450 Melaka, Malaysia
[b] Faculty of Engineering and Technology, Multimedia University, Jalan Ayer Keroh Lama, Bukit Beruang, 75450 Melaka, Malaysia

ARTICLE INFO

Article history:
Received 3 January 2016
Revised 6 August 2016
Accepted 25 November 2016
Available online 28 March 2017

Keywords:
Competitive co-evolutionary algorithm
Double elimination tournament
Radial basis function artificial neural network
Multilayer perceptron artificial neural network
Classification

ABSTRACT

This paper presents a competitive co-evolutionary (ComCoE) that engages a *double elimination tournament* (DET) to evolve artificial neural networks (ANNs) for undertaking data classification problems. The proposed model performs a global search by a ComCoE approach to find near optimal solutions. During the global search process, two populations of different ANNs compete and fitness evaluation of each ANN is made in a subjective manner based on their participations throughout a DET which promotes competitive interactions among individual ANNs. The adaptation and fitness evaluation processes drive the global search for a more competent ANN classifier. A winning ANN is identified from the global search. Then, the Scaled Conjugate Backpropagation algorithm, which is a local search, is performed to further train the winning ANN to obtain a precise solution. The performance of the proposed classification model is evaluated rigorously; its performance is compared with the baseline ANNs of the proposed model as well as other classifiers. The results indicate that the proposed model could construct an ANN which could produce high classification accuracy rates with a compact network structure.

1. Introduction

Many computational intelligence (CI) techniques are inspired from nature. A branch of CI is evolutionary computation [1], which is a computing paradigm biologically inspired from nature according to Darwin's notions. A variant of evolutionary algorithm (EA) is co-evolutionary algorithm (CoEA), which is based on the concept of co-evolution in biology defined as “the change of a biological object triggered by the change of a related object” [2]. CoEA has a unique feature that distinguishes itself from other EAs. The fitness evaluation of an individual in CoEA is subjective [3], which is based on an idea of measuring and computing interactions between the individual and other individuals [4]. CoEA is available in two different modes: cooperative or competitive. From the literature, a cooperative CoEA is formulated by adopting a “divide and conquer” strategy [5], which is a popular approach of algorithm design in computer science; the strategy is used to break a problem into sub-problems of the same type until these sub-problems can be solved in an easy manner. In general, a cooperative CoEA works in the following steps: (1) a problem is split into sub-problems and hence forming several populations; (2) these populations are evolved concurrently; (3) the selected individuals (based on a fitness criterion) are combined to form a solution [5]. Individuals interact in the combination phase of the cooperative CoEA. The fitness of each individual is computed by measuring how well it collaborates/cooperates with the other individuals [5]. On the other hand, in competitive CoEA, individuals interact in an adversary mode. The fitness of an individual is determined from a simple direct competition with other individuals.

The role of CoEA in problem solving is versatile. For example, CoEA has been employed to integrate with artificial neural networks (ANNs) to handle pattern classification problems. From the literature, several versions of co-evolutionary artificial neural networks (CoE-ANNs) have been proposed to optimise classification performances (e.g. [6-12]). CoEA finds a collection of neurons, sub-ANNs or ANNs to constitute an optimised single or ensemble ANN. In general, these CoE-ANNs could be composed by (1) a competitive [6–12]; (2) a cooperative [13–18] and (3) a combination of both competitive and cooperative approaches [19]. Majority of them are cooperative CoE-ANNs, which are constructed by invoking a “divide-and-conquer” strategy. Under this cooperative CoEA scheme, several sub-groups (sub-populations) of an ANN ensemble or an ANN component (e.g. neurons, sub-ANNs or ANNs) go on an evolutionary process in parallel, and a subset of them, which represent the best candidates from each sub-group, are combined to

* Corresponding author.
E-mail addresses: vivianhby@yahoo.com (B.Y. Hiew), sctan@mmu.edu.my (S.C. Tan), wslim@mmu.edu.my (W.S. Lim).

http://dx.doi.org/10.1016/j.neucom.2016.11.082
0925-2312/

form an optimised single or ensemble ANN. Notably, the cooperative interaction only occurs during this combination process. Prior to this combination process, individual solution candidates from all sub-populations do not evolve in an interactive manner. Besides, the fitness of the individual in each sub-population is entirely evaluated from an *objective* perspective (e.g. [13,14,16,18]). No *subjective* fitness evaluation has been considered. In other words, the idea of CoEA is not fully explored in the existing CoE-ANNs. The contribution of the research work in this paper is introduction of a CoE-ANN classification model of which its adaptation is reinforced through rapid interactions among ANNs by a game approach during the co-evolutionary (CoE) process. In our proposed model, two types of ANNs, i.e. multi-layer perceptron artificial neural networks (MLPANNs) and radial basis function artificial neural networks (RBFANNs) take part in the CoE process wherein these ANNs compete, and their interactions are quantified and evaluated using a tournament strategy. For simplicity, the proposed competitive CoE model is called as ComCoE(DET) ANN.

The rest sessions of this paper are set as follows: Section 2 provides a brief review of related work. The proposed ComCoE(DET) ANN model is described in Section 3. Section 4 presents results and analysis from an experimental study. Concluding remarks are stated in Section 5.

2. Related work

Among many versions of CoEAs, research in applying a competitive co-evolutionary (ComCoE) approach for designing a classification model is relatively less active in the machine-learning community. This is evidenced by a small number of publications [6–12] that are available in the literature since last two decades. These methods [6–12] have a common setup:

1. A population of classifiers is created to co-evolve competitively with a group of data samples. Examples of this type of research works include [6,7,10–12].
2. The entire CoE process involves adaptation of a population of classifiers through crossover and mutation. However, the data samples do not evolve [6,7,10,11]).
3. The methods search for an optimal solution based on an objective function only that is usually set is to achieve good classification accuracy. Examples of this kind of research study are [6–12].
4. During the fitness evaluation in each generation of CoE, the fitness of an individual solution (i.e. a classifier) is determined after going through several pairwise competitions with its opponents (i.e. data samples) (e.g. [6,7,10,11]). The number of pairwise competitions conducted in a generation of CoE is predefined. In each pairwise competition, a classifier (that is selected from a classifier population) is paired with a data sample (that is selected from a data sample population). If a classifier can classify the data sample correctly then the classifier earns a point from the competition; otherwise, the data sample will earn a point from the competition as a result of incorrect classification. Before proceeding to the next generation of CoE, all individuals in the classifier population and the data samples population are respectively sorted based on their fitness values. The CoE process focuses on classifying difficult data samples. A classifier that is more competent in classifying data samples (with a higher fitness value) is frequently selected to compete with a data sample which is (1) more difficult to be classified (with a higher fitness value); or (2) not yet being classified. The fitness value of each classifier and data sample is updated with the point(s) they have won before their fitness values are finalised in the last generation.
5. The model consists of one type of ANN classifier. For example, in [6,7], a type of ANN, i.e. multilayer perceptron ANN, was adopted.

On the other hand, a more popular research is focused on optimising the architecture and weights of a single ANN classifier [13,16,17], a modular ANN classifier [14,15] or an ANN ensemble [18] using cooperative CoEA. These methods build classification models by initially defining solutions to a problem into a few sub-populations where each sub-population consist of a number of neurons (e.g. CO-RBFNN [16], XLCC [13]), sub-networks (e.g. COVNET [14], GEPNET [15]) or the entire ANNs [18]. The entity in each sub-population evolves and its fitness is measured from one generation to another. Upon the end of a search process of cooperative CoEA, the best combination of ANNs, sub-networks or neurons are chosen from different sub-populations to construct an optimal ANN ensemble [18], a modular ANN [14,15] or an optimal ANN [13,16,17]. As mentioned in the Section 1, each sub-population goes through a search and adaptation process independently under a divide and conquer strategy; the cooperative interaction happens only when the selected neurons/sub-networks or ANNs are combined (e.g. [13–18]). Overall, the level of interactions among individuals from different sub-groups is low in the context of co-evolution.

In this paper, the interactions among different computing entities in the proposed ComCoE(DET) ANN are modelled by a game approach. Indeed, games are important elements of every culture in the past and present. Games focus on strategic interactions among participants and help develop competing individuals [20,21]. Games are defined by rules and could be conducted in tournaments. Many tournaments have been adopted to decide the winner of a game. Some of the prominent tournaments are [22]:

1. Round robin tournament (RRT): Each contestant plays against all other contestant exactly once. This tournament is commonly applied to sports games such as football league.
2. Single elimination tournament (SET): Contestants proceed to round $r+1$ upon winning in round r of a tournament. The loser from each match is instantly eliminated. The matches continue until only one contestant wins in the final match. In this case, a single "champion" is identified. This tournament format is generally used in sports games such as basketball, bowling, tennis etc.
3. Double elimination tournament (DET): A competition where a contestant is only eliminated after losing two competitions/matches. Commonly, this type of tournament is used in baseball and softball tournaments.

The above mentioned tournament strategies have been applied to provide solutions to problems in many disciplinary studies such as economics [23], finance [24] and artificial intelligence (AI) [25–27]. CoEAs have been applied to design AI games such as Tic-Tac-Toe [25] (SET was adopted), Othello [26] (RRT was applied), Blondie24 [27] (RRT was used) etc. In [28], RRT and SET were utilised to assess each individual's fitness of a CoE model for function optimisation problems and the Nim game. In [29], DET was adopted to formulate an ensemble of performance metrics for comparing six multi-objective evolutionary algorithms (MOEAs). In [30], text classifiers were designed using either RRT or SET to determine the language of origin of an English speaker as either native or non-native. For clarity, DET has yet been integrated with machine-learning techniques to perform data classification tasks.

The main weakness of RRT is the fairly large number of games required to assess each contestant. As compared with RRT and DET, SET involves the fewest matches. But, in SET, after each game, 50% of the contestants are eliminated [22]. The motivation of adopting DET is that it grants each contestant at least two opportunities to participate in the matches in a tournament. In SET, a good contestant is lost forever if he/she loses in a match. Nevertheless, in DET, this contestant still has a chance to compete and win. Hence, it helps in preserving good contestants. By choosing DET instead of RRT, a relatively large number of matches could be avoided [22]. By looking at the perspective of number of *interactions*, individuals interact more in DET as compared to the individuals in SET. In our work, DET is selected to enhance interaction among ANNs for search and adaptation for a better solution.

3. The proposed ComCoE(DET) ANN model

In this research, both MLPANN and RBFANN are utilised to form a ComCoE(DET) ANN model where their operations are described separately in Sections 3.1 and 3.2. The idea of the proposed ComCoE(DET) ANN model is then explained in Section 3.3.

3.1. Multilayer perceptron artificial neural network (MLPANN)

The MLPANN[1] model [31] is composed by an input layer, two hidden layers and an output layer. Mathematically, the MLPANN can be expressed as below:

$$y_n = \Phi_o\left(\sum_{d=1}^{D}\omega_{nd}^{(2)}\Phi_i\left(\sum_{t=1}^{T}\omega_{dt}^{(1)}x_t + b_d^{(1)}\right) + b_n^{(2)}\right) n = 1\ldots,N \quad (1)$$

where $\mathbf{x}_t$ is the $t-$ input vector; $\omega_{dt}^{(1)}$ and $\omega_{nd}^{(2)}$ denote the weight matrices of the first and second hidden layer respectively; $b_d^{(1)}$ and $b_n^{(2)}$ are the respective biases in the two hidden layers; T is the total number of input units; D is the total number of the first hidden layer's hidden units; N is the total number of outputs. Φ_i and $\Phi_0(\cdot)$ are the activation functions given in Eq. (2) and Eq. (3) respectively.

$$\Phi_i(\nu) = tanh(\nu) \quad (2)$$

$$\Phi_o(\nu) = \nu \quad (3)$$

The weights of MLPANN are randomly initialised and drawn from a Gaussian distribution having a zero mean and a unit of variance. The MLPANN is trained by the Scaled Conjugate (SCG) Backpropagation algorithm [32].

3.2. Radial basis function artificial neural network (RBFANN)

The RBFANN[1] model [31] is composed by an input layer, a hidden layer and an output layer. The number of neurons in the input layer is equal to the number of input attributes of the data. The data are first passed from the input layer to the hidden layer. Each hidden neuron is a radial basis function and it excites by computing a value of activation. Next, this activation value is transmitted to the output layer. In the output layer, all activation values of the hidden neurons to an output neuron are linearly combined. Hence, the uth output neuron from the output layer is modelled by a weighted-sum formula, as follows:

$$y_u(\boldsymbol{x}_t) = \boldsymbol{\omega}_{u0} + \sum_{j=1}^{J}\boldsymbol{\omega}_{uj}\varphi_j(\boldsymbol{x}_t) \quad \text{for } u = 1,\ldots,U \text{ and } t = 1,\ldots,T \quad (4)$$

where $\mathbf{x}_t$ is the tth input vector, $\boldsymbol{\omega}_{u0}$ denotes bias term, $\boldsymbol{\omega}_{uj}$ is the weights of the output layer, the activation function is represented by φ_j. In every hidden unit, the radial basis function φ_j is the Gaussian function, as follows.

$$\varphi_j(\boldsymbol{x}_t) = \exp\left\{-\frac{\left\|\boldsymbol{x}_t - \boldsymbol{c}_j\right\|^2}{2\sigma_j^2}\right\} j = 1,\ldots,J \quad (5)$$

where $\|\cdot\|$ is the Euclidean norm.

Given that σ_j is the width, which is set to one; $\boldsymbol{c}_j$ is the centre of the jth neuron in the hidden layer; ω_{u0}, $\boldsymbol{\omega}_{uj}$, $\boldsymbol{c}_j$ are obtained through the random initialisation from a normal distribution with a zero mean and a unit of variance. RBFANN is trained by the SCG algorithm [32].

3.3. The ComCoE(DET) ANN model

In the proposed ComCoE(DET) ANN model, two populations of ANNs, namely, an MLPANN population and an RBFANN population co-evolve competitively by implementing a DET during fitness evaluation phase. These ANNs have different number of hidden neurons. In the fitness evaluation phase of the ComCoE(DET) ANN model, a population of MLPANNs and a population of RBFANNs are randomly mixed and evaluated through a DET. In this tournament, several pairwise competitions between two ANNs of either the same or different types are conducted in several rounds by considering two performance indicators. The idea of the fitness evaluation is explained in details in Section 3.3.2. Before this, the procedure of the ComCoE process that includes the fitness evaluation phase is explained in Section 3.3.1.

3.3.1. The ComCoE process

The ComCoE algorithm is designed to adapt the dynamics of (1) an MLPANN for its neurons' weights $\omega_{dt}^{(1)}$, $\boldsymbol{\omega}_{nd}^{(2)}$, and biases $b_d^{(1)}$, $b_n^{(2)}$; and, (2) an RBFANN for its neurons' width σ_j, centre $\boldsymbol{c}_j$, weight $\boldsymbol{\omega}_{uj}$ and bias, ω_{u0}. The ComCoE is essentially a global-local search algorithm that is executed in two conservative phases to find the fittest ANN. In phase-1, a competitive co-evolution starts off to conduct a global search for identifying solutions that are in the surrounding area of optimality in the solution space. In phase-2, the selected solution (i.e. ANN) from phase-1 goes through a local optimisation process wherein the ANN is trained with SCG Backpropagation algorithm [32]. Such a global-local search strategy is deployed with a condition that the solution obtained from the global search is refined during the local search and becomes more accurate. The construction of the proposed ComCoE(DET) ANN model is based on the following steps:

1. *Chromosome representation, population initialisation and initial fitness evaluation*. Two populations of L (in this work, $L = 128$) chromosomes, i.e. a population of L chromosomes of MLPANNs and a population of L chromosomes of RBFANNs, are randomly initialised. Each ANN has different number of hidden neurons, h. The hidden neurons number, h of each ANN is a pseudorandom integer value drawn from a discrete uniform distribution within a certain range. The initial fitness, f, of each ANN is calculated. How the fitness of an ANN is calculated through DET will be described in details in Section 3.3.2. After the initial fitness calculation, every ANN is encoded as a chromosome z_i where z_i is the ith chromosome representing the ith ANN. To construct a chromosome z_i, the weights, $\omega_{dt}^{(1)}$ and $\omega_{nd}^{(2)}$, biases, $b_d^{(1)}$ and $b_n^{(2)}$ of an MLPANN (Eq. (6)) or the centre, $\boldsymbol{c}_j$, width, σ_j, weight, $\boldsymbol{\omega}_{uj}$ and bias, ω_{u0} of an RBFANN (Eq. (7)) are concatenated into a single row vector, as follows:

$$z_i = \left[\omega_{nd}^{(2)}\ b_d^{(1)}\ b_n^{(2)}\right] \quad (6)$$

or

$$z_i = \left[\boldsymbol{c}_j\sigma_j\boldsymbol{\omega}_{uj}\ \omega_{u0}\right] \quad (7)$$

Besides, the following parameters are set as 0:

[1] http://www.aston.ac.uk/eas/research/groups/ncrg/resources/netlab/downloads/.

(a) the total score of the fitness values of all RBFANNs in a generation, rg
(b) the total score of the fitness values of all MLPANNs in a generation, mg
(c) the accumulated score of the fitness values of all RBFANNs throughout all generations, rG
(d) the accumulated scores of the fitness values of all MLPANNs throughout all generations, mG

2 *Selection and variation*. The main search or variation operator utilised in this model is the Gaussian mutation [33]. Selection and mutation happen under one of these conditions:
(a) if $mg > rg$, the whole population of MLPANN chromosomes is selected to go through a mutation process;
(b) if $rg > mg$, the whole population of RBFANN chromosomes is selected to go through a mutation process;
(c) if $mg == rg$, both populations of MLPANN and RBFANN chromosomes go through a mutation process.

The above rules are set based on this assumption: if $mg(rg)$ is higher than $rg(mg)$, it shows that MLPANNs (RBFANNs) could perform better than RBFANNs (MLPANNs) in general and hence the possibility of getting the best performing ANN in the MLPANN's (RBFANN's) search space is greater than RBFANN's (MLPANN's) search space. So, MLPANN (RBFANN) chromosomes are mutated. A user-defined mutation rate MR is introduced to govern the mutation operation on the selected population of chromosomes. The Gaussian mutation is executed when a probability value is smaller than MR. Subsequently, as given in Eq. (4) and Eq. (5), an offspring z_i' is produced.

$$\eta_i' = \eta_i \exp\left(\tau' N(0,1) + \tau N(0,1)\right) \tag{4}$$

$$z_i' = z_i + \eta_i' N(0,1) \tag{5}$$

where $N(0, 1)$ denotes a random number drawn from a normally distribution with a mean of zero and a variance of one; η_i indicates the variance vector. In the experiments, τ is set to $(\sqrt{2\sqrt{n}})^{-1}$ whereas τ' is set to $(\sqrt{2n})^{-1}$ [34].

3. *Fitness evaluation*. Each chromosome z_i' is decoded to either (1) an MLPANN represented by a list of the weights, $\omega_{dt}^{(1)}$ and $\boldsymbol{\omega}_{nd}^{(2)}$ biases, $b_d^{(1)}$ and $b_n^{(2)}$ of an MLPANN or; (2) an RBFANN represented by a list of the centre, $\mathbf{c}_j$, width, σ_j, weight, $\boldsymbol{\omega}_{uj}$ and bias, ω_{u0}. The fitness score of each ANN is computed from a procedure explained in Section 3.3.2. After the fitness evaluation, all ANNs are encoded back to chromosomes.
4. *The creation of new populations*. Two new populations, which are a population of MLPANN chromosomes and a population of RBFANN chromosomes, are formed. After step 4, the process goes back to step 2. The processes of mutation operation and fitness evaluation are repeated until a predefined number of generations (i.e. G) has been reached to indicate the end of phase-1.
5. *Phase-2 local search*. The values of rG and mG are compared to determine which elite chromosome from MLPANNs or RBFANNs obtained from phase-1 (covered by steps 1 - 4) will proceed to the phase-2 local search:
(a) if $mG > rG$, the elite chromosome is selected from MLPANNs;
(b) if $rG > mG$, the elite chromosome is selected from RBFANNs;
(c) if $mG == rG$, both of the elite members of MLPANN and RBFANN are paired to compete. During the competition, the fitness scores of these ANNs are obtained by comparing their $eAcc$ as well as h (as explained in Fig. 4). Then, the winning chromosome from this competition is selected. If their fitness scores are the same, one of these chromosomes is randomly picked.

In the phase-2 search, the winning chromosome could be decoded as either an MLPANN or an RBFANN. Then, this ANN is trained with the SCG Backpropagation algorithm. The Phase-2 local search ends.

6. *Performance evaluation*. The trained ANN is evaluated using data samples from a test set.

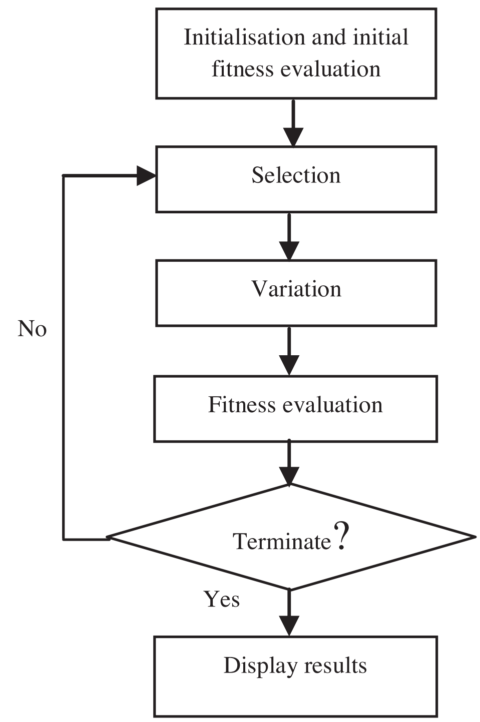


Fig. 1. A typical EA.

3.3.2. Fitness evaluation

A generic cycle of EA (Fig. 1) comprises a few stages of operation: (1) population initialisation and initial fitness evaluation; (2) selection for parent chromosomes; (3) variation by applying the genetic operators (e.g. crossover and mutation) and (4) fitness evaluation on the chromosomes from a new population. A population of chromosomes is generated and initialised randomly. The initial degree of goodness of each chromosome is computed with an objective function. This degree of goodness indicates a measure of fitness of a chromosome. A selection strategy is applied to choose parent chromosomes based on their fitness values. Genetic operators are then applied to the parent chromosomes to create offspring. The fitness of each offspring is computed. These offspring replace the parents to form a population of chromosomes in the new generation before another round of chromosome selection, variation and fitness evaluation processes begins. EA operates in a number of cycles until a terminating condition has been met [35]. A CoEA is conceptually different from an EA from the approach to evaluating the fitness of a chromosome. In the EAs, the degree of goodness of a chromosome is measured by using an objective fitness function. On the contrary, the fitness of a chromosome in the CoE framework is measured in a subjective manner, which is determined by assessing its interaction with other chromosomes [36].

In this work, the fitness evaluation process of the proposed model is designed by an interactive tournament-based approach wherein the concept of DET is implemented. By referring to Fig. 2, the process of fitness evaluation can be explained as follows:

1. All MLPANNs and RBFANNs are randomly mixed and then formed as a pool of players in the tournament. Before the tour-

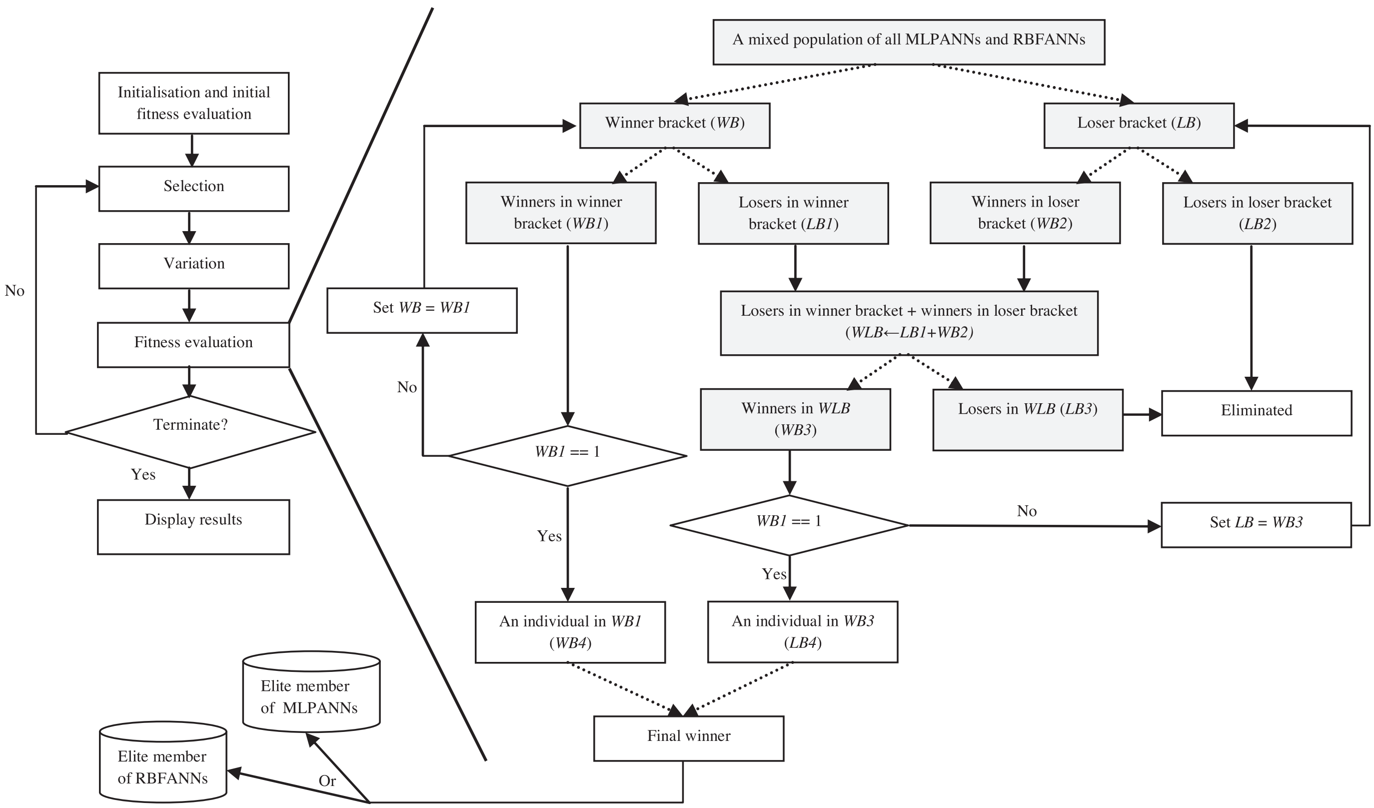


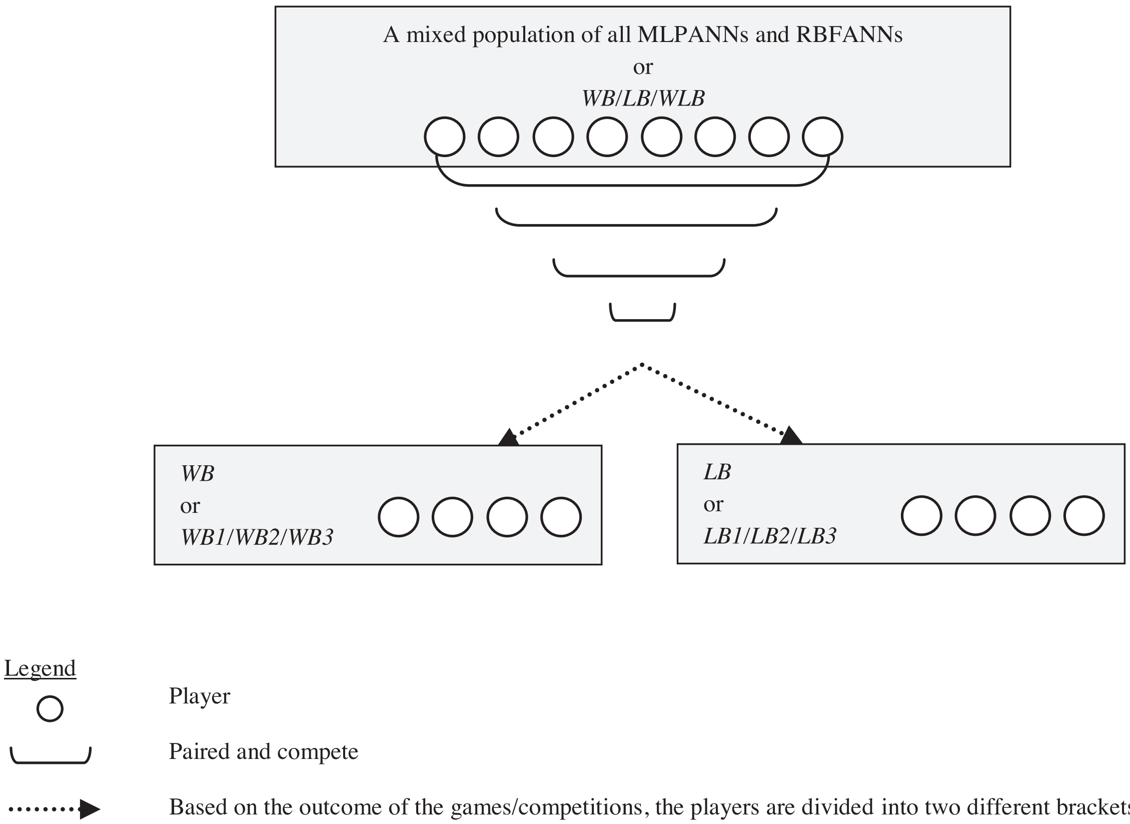
Fig. 2. Fitness evaluation that applies the concept of DET in a ComCoE algorithm.

nament starts, the classification accuracy of every player is calculated with all data samples from a training set.

2. The whole pool of the mixed population of chromosomes (or players) participate in pairwise competitions. The process of the pairwise competitions is illustrated in Fig. 3 and can be described as follows:
   2.1. Before the pairwise competitions begin, the fitness score of every player in the pool is initially set to 0.
   2.2. During the pairwise competitions, every player merely plays a game. Let say M players are available in the pool, the mth player and the $M+1-m$th player are paired to play a game where $m = 1, 2, \ldots, M/2$. For example, the first player meets the last player to compete in a game. The second player is assigned to play with the second last player and so on. The fitness of a player and its opponent are determined by considering both metrics i.e. the number of hidden neurons, h and training accuracy rate, *eAcc*. In this case, a player's fitness is not directly assigned with its *eAcc* neither h, which is the usual practice in objective fitness evaluation. Indeed, the fitness of a player is determined in a subjective manner by referring to its encounters with other players. For any pairwise competition between two players in a game, the fitness score of a player is determined by comparing their results in *eAcc* and h. Fig. 4 presents the assignment of fitness score based on *eAcc* and h of a pair of ANNs, which can be described as follows:
   2.2.1 two points are awarded to an ANN if its *eAcc* is higher than its opponent
   2.2.2 one point is awarded to an ANN if its h is lower than its opponent.

   The reason why the award of point of *eAcc* is set greater than h can be explained as follows. The function of a classifier is to map an instance of a set of input features to an output category where the instance could be in [37]. In this regards, given a classification task, the first goal of the classifier is to complete the task with high accuracy performance [38,39]. If the classifier could classify the data in high accuracy rates, the next goal is focused on designing a classifier in low complexity. For the case of an ANN, its low complexity could be indicated by a small number of hidden neurons residing in its structure after a training session. *eAcc* has a higher award point than h; these settings indicate classification accuracy has a priority over number of hidden neurons when building a ComCoE(DET) ANN model.

   Whenever the number of players is even in each round of the tournament, no player will get a bye. Hence, to avoid any player from getting a bye during the pairwise competitions, the total number of players L in a population must be set to a value of a power of two. Upon completion of the pairwise competitions (illustrated in Figs. 2 and 3), the mixed population of players are divided equally into two *brackets*: (1) the winners from the competitions are placed in the winner bracket (*WB*); (2) the losers from the competitions are put in the loser bracket (*LB*).
3. The players in *WB* perform pairwise competitions as described in Steps 2.1 and 2.2 (and also illustrated in Fig. 3). The players in *WB* are further divided equally into two subgroups, i.e. (1) the winners in winner bracket (*WB1*); (2) the losers in winner bracket (*LB1*).
4. Next, the players in *LB* participate in pairwise competitions that are again based on the procedure in Steps 2.1 and 2.2. The players in *LB* are then split equally into two subgroups (illustrated in Fig. 2): (1) the winners in loser bracket (*WB2*); (2) the losers in loser bracket (*LB2*). The players in *LB2* have lost the games for two times and hence, they are eliminated.
5. The individuals in *LB1* and *WB2* (denoted as *WLB* in Fig. 2) who have only lost a game are paired (illustrated in Fig. 3) and compete for one more time (repeat Steps 2.1 and 2.2). The losers



— Set *M= pool size of a mixed population of all ANNs (MLPANNs and RBFANNs) or pool size of WB/LB/WLB*

for $1 \leq m < M/2$

— Select a pair of ANNs from the pool, i.e. m^{th} ANN and $M+1-m^{th}$ ANN

— Evaluate the selected pair of ANNs (point assignment is entailed in Fig. 4)

end for

Fig. 3. Pairwise competitions among individuals in a mixed population of all ANNs (MLPANNs and RBFANNs) or *WB/LB/WLB*.

— The selected pair of ANNs are evaluated by comparing their training accuracy, *eAcc* and the number of hidden neurons, *h*

- 2 points are awarded to an ANN if its *eAcc* is higher than its opponent
- 1 point is awarded to an ANN if its *h* is lower than its opponent.

Fig. 4. Fitness score assignment during the fitness evaluation on the selected pair of ANNs.

from *WLB* (denoted as *LB3* in Fig. 2) who have lost the games twice are now eliminated.

6. If the number of individual in *WB1* is equal to one, proceed to Step 7. Otherwise, repeat the process from Step 3 to Step 5. Before executing Step 3, set *WB* = *WB1*, *LB* = *WB3*. In this case, the individuals of *WB1* are reserved in the winner bracket for the next round of competition. The players of *WB3* will be assigned to the loser bracket for the next round of competition.
7. An individual from *WB1* (denoted as *WB4* in Fig. 2) and an individual from *WB3* (denoted as *LB4* in Fig. 2) compete and their fitness scores are determined by comparing their *eAcc* and *h* (as explained in Fig. 4). If the one from *WB4* wins, it will be declared as the final winner. If the one from *LB4* wins, another round of competition will be held to decide the ultimate winner. If their fitness scores are the same after the first match, the final winner is randomly chosen. The fitness evaluation process is ended after a final winner has been decided.

4. Experiments

In this section, the performance of the proposed ComCoE(DET) ANN is compared with the performances of the baseline ANNs (i.e.

Table 1
The properties of the benchmark datasets.

No.	Datasets	Classes	Attributes	Number of instances	Missing attribute values
1	Breast Cancer W-D	2	30	569	No
2	Breast Cancer W-P	2	33	198	Yes
3	New Thyroid Gland	3	5	215	No
4	Parkinsons	2	22	195	No
5	SPECTF Heart	2	44	267	No
6	Vehicle	4	18	846	No
7	Credit Approval	2	15	690	Yes
8	Ionosphere	2	34	351	No
9	Breast	2	9	699	Yes
10	Pima	2	8	768	No
11	Wine	3	13	178	No
12	Sonar	2	60	208	No
13	Heart	2	13	270	No
14	Iris	3	4	150	No
15	Heart-Cleveland	2	13	303	Yes

RBFANN and MLPANN) and other classification models. The experiments were performed by the use of a laptop with 8 GB RAM, Intel Core i5 2.53 GHz CPU and Windows 7 operating system. We adopted the same experimental setup as in [40] to repeat the experiments for 10 times using a 10-fold-cross-validation strategy. The results were average. The 10-times-of-10-fold cross-validation strategy was employed because it is one of the popular and rigorous classification evaluation methods, and is less bias from random sampling [41,42].

In this work, a total number of 10 algorithms were referred for making a performance comparison with the proposed model. These algorithms include EAPSONN [43], XLCC [13] and CC [13], CO-RBFNN [16], CO^2RBFN [19] and MPSON [44], ComCoE RBFANN [45], ELM [40], C4.5 [40] and ELM-Tree [40]. These 10 algorithms were evaluated with various benchmark datasets from the UCI Machine Learning Repository [46]. For the experimental study in this research, we only considered the benchmark datasets of which the number of output classes were not more than four. We also considered the usage rate of each dataset whereby it must be used by at least four out of these 10 algorithms. Subsequently, 15 benchmark datasets were identified. These datasets are listed in Table 1. All datasets were normalised in a range [0, 1]. Some of the datasets contained missing values. Every missing attribute value was allocated with the average of the attribute value. The datasets were utilised in different experimental studies presented from Sections 4.1 to 4.3.

The number of samples in all 15 datasets is basically small. On the other hand, we intended to construct a ComCoE(DET) ANN consisting of a small number of hidden neurons h without jeopardising its generalisation ability. A simple rule was applied to determine the h setting of ANNs (i.e. MLPANNs and RBFANNs) in these 15 classification tasks by referring to a ratio of h to the number of all records from a dataset. The h setting was identified from a small ratio that was not greater than 0.05. Hence, the h setting of each ANN was any discrete value ranged within [1–5]. Such h setting was applicable to ANNs in all classification tasks but excluding New Thyroid Gland and Vehicle; the h setting of an ANN in the two mentioned classification tasks could be any discrete value ranged within [5,10] and [20–25] respectively. The local training (the SCG Backpropagation algorithm), l was set as 100 for all classification tasks except for New Thyroid Gland and Vehicle. The proposed ComCoE was trained by $l = 550$ to classify the New Thyroid Gland and the Vehicle datasets.

4.1. Comparison with the baseline MLPANN and baseline RBFANN

In a performance comparison with the proposed ComCoE(DET) ANN model, both the baseline MLPANN and the baseline RBFANN that were developed from the Netlab toolbox[1] were also trained and evaluated by repeating the 10-fold cross validation for 10 times. The classification results were averaged. Both MLPANN and RBFANN were trained with the SCG algorithm with $l = 550$ using the New Thyroid Gland and the Vehicle datasets and $l = 100$ using the rest 13 datasets. For clarity, both MLPANN and RBFANN were trained and tested with the same partition of datasets that have been used in the experiment with the proposed model.

The classification results of the proposed model and the two baseline ANNs were expressed in terms of average number of hidden neurons, *#h*, and average test accuracy rate, *aAcc* The performance comparisons between the proposed model and the baseline ANNs were presented in Tables 2 and 3. A 2-tailed Wilcoxon's signed-ranks test [47] was used to compare (1) the average test accuracy rate, *aAcc* between a baseline ANN (i.e. either MLPANN or RBFANN) and the proposed model (2) the average number of hidden neurons, *#h*, between a baseline ANN and the proposed model. The alternative hypothesis was that the performance between the baseline ANN and the proposed ComCoE(DET) ANN was not the same. The hypothesis test was made at a level of significance $\alpha = 0.05$. By referring to Tables 2 and 3, all p-values for *aAcc* are lower than 0.05. Hence, these statistical test results indicate the proposed model could classify unseen data samples significantly more accurate than the baseline MLPANN and the baseline RBFANN.

By referring to the results of p-values for *#h* from Table 2, the complexity of the proposed model and the baseline MLPANN in terms of average number of hidden neurons is statistically indifferent in 11 classification tasks because the p-values are below 0.05. On the other hand, the results of p-values for *#h* from Table 3 show that the average number of neurons of the proposed ComCoE(DET) ANN is not significant different from that of the baseline RBFANN in 10 classification tasks. Hence, the proposed ComCoE(DET) ANN is a more accurate classifier of which its complexity in terms of number of hidden neurons is, in general, not significantly different from a statistical viewpoint with the complexity of the baseline MLPANN and the baseline RBFANN.

4.2. Performance comparison with other CoE-ANNs

The performances of the proposed model and other CoE-ANNs were compared. The performance comparison is presented separately in two subsections, i.e. between the proposed model and (1) other CoE-MLPANN models in Section 4.2.1; and (2) other CoE-RBFANN models in Section 4.2.2.

4.2.1. Performance comparison with other CoE-MLPANNs

Other CoE-MLPANNs, which were EAPSONN [43], XLCC [13] and CC [13], were selected for performance comparison with the proposed model in terms of average training accuracy, *Train Acc.*, av-

Table 2
The classification results of the proposed ComCoE(DET) ANN and the baseline MLPANN (*aAcc*—average test accuracy rate; *#h*—average number of hidden neurons).

Dataset	Proposed model		Baseline MLPANN		*p*-value (*aAcc*)	*p*-value (*#h*)
	aAcc (%)	*#h*	*aAcc* (%)	*#h*		
Breast Cancer W-D	98.0	3.3	96.7	3.0	0.0020	0.2500
Breast Cancer W-P	83.0	3.4	76.1	3.0	0.0020	0.0664
New Thyroid Gland	95.2	7.8	93.9	7.4	0.0098	0.1445
Parkinsons	92.1	3.1	89.6	3.0	0.0020	0.4316
SPECTF Heart	80.6	1.4	77.6	3.0	0.0020	0.0020
Vehicle	77.9	22.5	76.1	22.5	0.0059	0.8906
Credit Approval	86.0	2.8	84.7	2.9	0.0020	0.4961
Ionosphere	92.5	3.8	89.5	3.1	0.0020	0.0078
Breast	96.5	2.8	95.3	3.0	0.0020	0.1016
Pima	78.2	3.7	76.5	2.9	0.0020	0.0059
Wine	99.7	3.4	96.0	3.0	0.0039	0.0195
Sonar	85.1	3.6	78.4	3.1	0.0020	0.0703
Heart	83.4	3.3	81.4	3.3	0.0020	1.0000
Iris	97.8	3.1	96.6	3.2	0.0020	0.6406
Heart-Cleveland	83.6	3.2	77.8	2.9	0.0020	0.1523
Average	88.6	4.7	85.7	4.6		

Table 3
The classification results of the proposed ComCoE(DET) ANN and the baseline RBFANN.

Dataset	Proposed model		Baseline RBFANN		*p*-value (*aAcc*)	*p*-value (*#h*)
	aAcc(%)	*#h*	*aAcc*(%)	*#h*		
Breast Cancer W-D	98.0	3.3	85.0	3.1	0.0020	0.3457
Breast Cancer W-P	83.0	3.4	76.4	2.8	0.0020	0.0293
New Thyroid Gland	95.2	7.8	86.0	7.6	0.0020	0.6016
Parkinsons	92.1	3.1	80.9	2.9	0.0020	0.2344
SPECTF Heart	80.6	1.4	79.4	3.2	0.0020	0.0020
Vehicle	77.9	22.5	53.2	22.5	0.0020	0.8906
Credit Approval	86.0	2.8	81.8	3.1	0.0020	0.2852
Ionosphere	92.5	3.8	67.5	3.2	0.0020	0.0020
Breast	96.5	2.8	94.2	3.1	0.0039	0.1094
Pima	78.2	3.7	75.4	3.1	0.0020	0.0039
Wine	99.7	3.4	47.8	2.9	0.0020	0.0176
Sonar	85.1	3.6	53.7	3.3	0.0020	0.0566
Heart	83.4	3.3	80.5	3.0	0.0234	0.2871
Iris	97.8	3.1	89.9	3.0	0.0039	0.7441
Heart-Cleveland	83.6	3.2	79.4	3.0	0.0020	0.3066
Average	88.6	4.7	75.4	4.7		

Table 4
The classification results of the proposed model and other CoE-MLPANNs (*Train Acc.*—average training accuracy; *aAcc*—average test accuracy; and, *#h*—average number of hidden neurons).

Dataset	Proposed model			[a]EAPSONN [43]			[b]XLCC [13]			[b]CC [13]		
	Train Acc. (%)	*AAcc* (%)	*#h*	*Train Acc.* (%)	*aAcc* (%)	*#h*	*Train Acc..* (%)	*aAcc* (%)	*#h*	*Train Acc.* (%)	*aAcc* (%)	*#h*
Heart-Cleveland	87.4	83.6	**3.2**	86.0	**84.0**	7.6	88.0	79.2	7.0	88.0	79.7	7.0
Iris	97.7	**97.8**	**3.1**	–	–	–	95.0	94.9	4.0	95.0	93.7	4.0
Breast	97.4	96.5	**2.8**	96.7	96.6	8.9	95.0	**96.8**	5.0	95.0	96.3	5.0
Pima	78.4	78.2	**3.7**	85.6	**82.1**	5.6	–	–	–	–	–	–
Wine	99.6	**99.7**	**3.5**	–	–	–	95.0	93.4	4.0	95.0	93.0	4.0
Credit Approval	87.6	**86.0**	**2.8**	94.5	85.8	7.7	–	–	–	–	–	–
Ionosphere	96.2	92.5	**3.8**	–	–	–	98.0	94.7	6.0	98.0	**95.1**	6.0

Note: unavailable results are denoted as "–".
[a] 50% training & validation (65% training; 35% validation), 50% testing, 30 independent runs.
[b] 70% training, 30% testing, 50 independent runs.

erage test accuracy, *aAcc* and average number of hidden neurons, *#h*. EAPSONN is an evolutionary MLPANN; XLCC and CC are cooperative CoE-MLPANNs. They were considered in this comparative study because according to a literature survey, publications related to the ComCoE version of CoE-MLPANN are not available in the database. Notably, only 7 out of 15 datasets (i.e. the datasets numbered from 7 to 11, 14 and 15) from Table 1 were used by EAPSONN [43], XLCC [13] and CC [13]. Therefore, the classification results of the CoE-ANN models in these seven classification tasks in terms of *Train Acc.*, *aAcc* and *#h* are listed in Table 4. The results of EAPSONN are taken from [43] whereas the results of XLCC and CC are referred from [13]. In addition, *Train Acc.* of XLCC and CC are the predefined values, which are the targets of training accuracy for their ANNs to be achieved. The lowest *#h* as well as the highest *aAcc* of a classification model in a classification task are bolded. Findings from the results in Table 4 are two-fold; first, the training and test accuracy rates of the proposed model are close, and these results indicate the proposed model is not overfit from the adaptation processes. Second, when comparing with the results of EAPSONN, XLCC and CC, the proposed model is observed as a more

Table 5
The classification results of the proposed model and other CoE-RBFANNs.

Dataset	Proposed model			[a]CO-RBFNN [16]			[b]CO^2RBFN [19]			[b]MPSON [44]		
	Train Acc. (%)	*AAcc* (%)	*#h*	*Train Acc.* (%)	*aAcc* (%)	*#h*	*Train Acc.* (%)	*aAcc* (%)	*#h*	*Train Acc.* (%)	*aAcc* (%)	*#h*
Heart-Cleveland	88.7	83.4	**3.3**	85.1	83.0	12.7	–	–	–	85.1	**85.2**	5.3
Iris	97.7	**97.8**	**3.1**	–	–	–	99.0	96.3	6.0	87.6	89.1	10.0
Breast	97.4	96.5	**2.8**	96.6	96.9	11.6	97.8	97.1	5.0	97.4	**97.7**	5.8
Pima	78.4	**78.2**	**3.7**	78.8	77.0	23.1	77.9	76.0	4.0	76.3	77.3	7.9
Wine	98.4	**85.1**	**3.8**	91.5	76.0	22.2	79.9	75.1	8.0	–	–	–
Credit Approval	99.6	**99.7**	**3.5**	96.9	96.9	6.6	99.9	96.7	7.0	77.3	75.7	8.3
Ionosphere	96.2	92.5	**3.8**	94.3	**93.8**	24.4	93.5	91.4	8.0	–	–	–

Note: unavailable results are denoted as "–".
[a] 50% training, 25% validation, 25% testing.
[b] Single run of 10-fold cross validation.

Table 6
Classification results of the proposed ComCoE(DET) ANN and the ComCoE RBFANN [45].

Dataset	Proposed model		ComCoE RBFANN [45]		*p*-value (*aAcc*)	*p*-value (*#h*)
	aAcc(%)	*#h*	*aAcc*(%)	*#h*		
Breast Cancer W-D	**98.0**	**3.3**	97.0	5.4	0.0020	0.0020
Breast Cancer W-P	**83.0**	3.4	81.2	**2.6**	0.0020	0.0195
New Thyroid Gland	**95.2**	**7.8**	90.4	10.9	0.0020	0.0020
Parkinsons	**92.1**	**3.1**	87.3	4.9	0.0020	0.0020
SPECTF Heart	80.6	1.4	**80.7**	**1.1**	0.7520	0.0977
Vehicle	**77.9**	22.5	69.4	**22.1**	0.0020	0.3730
Credit Approval	**86.0**	**2.8**	85.8	4.3	0.3223	0.0020
Ionosphere	**92.5**	**3.8**	92.3	31.9	0.0840	0.0020
Breast	96.5	2.8	**96.6**	**2.1**	0.2891	0.0020
Pima	**78.2**	3.7	77.5	**2.3**	0.0020	0.0020
Wine	**99.7**	**3.4**	95.0	3.9	0.0020	0.0059
Sonar	**85.1**	3.6	71.8	**1.1**	0.0020	0.0020
Average	88.7	5.1	85.4	7.7		

compact ANN that could perform with comparable, if not higher, test accuracy rates.

4.2.2. Performance comparison with other CoE-RBFANNs

In addition, the proposed model was compared with other CoE-RBFANNs. These CoE-RBFANNs are non-ComCoE versions, namely CO-RBFNN [16], CO^2RBFN [19] and MPSON [44]. Seven datasets (numbered from 8 to 14 in Table 1) were employed by CO-RBFNN [16], CO^2RBFN [19] and MPSON [44]. CO-RBFNN is a cooperative CoE RBFANN; CO^2RBFN is an evolutionary cooperative-competitive RBFANN; and MPSON is a generic evolutionary based algorithm. Their classification results are referred in this comparative study because the ComCoE versions of CoE-RBFANNs proposed by other researchers are not available in the literature. The results are listed in Table 5 in terms of *Train Acc.*, *aAcc* and *#h* where the lowest *#h* as well as the highest *aAcc* of a classification model in a given task are indicated in bold face. The findings from the performance comparison between the proposed model and other CoE-RBFANNs in Table 5 are similar to the findings from the performance comparison between the proposed model and other CoE-MLPANNs in Table 4. The results show that the proposed model has a more compact ANN architecture and it could perform with comparable, if not higher test accuracy rates, than CO-RBFNN, CO^2RBFN and MPSON.

Besides, we also compared the classification performance of the proposed model with a ComCoE RBFANN [45] from our recent work. In [45], a compact and accurate RBFANN is identified from a single-population of RBFANNs that incorporate a single elimination tournament (SET) for fitness evaluation in the CoE process. In this fitness evaluation, each RBFANN participates in a SET, and the degree of goodness of each RBFANN is quantised based on their interactions/encounters with other RBFANNs in an intra-specific competition environment.

For performance comparison between the proposed ComCoE(DET) ANN and the ComCoE RBFANN [45], a total number of 12 datasets (i.e. based on the first 12 datasets from Table 1) were used. The results are listed in Table 6 in terms of average test accuracy rates, *aAcc* and average of the number of hidden neurons, *#h*. Again, a 2-tailed Wilcoxon's signed-ranks test [47] was utilised to check the performance difference between the two models. The significant level was set at $\alpha = 0.05$. As can be seen in Table 6, the *p*-values for *aAcc* of both models are lower than 0.05 in most of the classification problems except for SPECTF Heart, Credit Approval, Ionosphere and Breast. Overall, the average test accuracy rate of ComCoE(DET) ANN (i.e. 88.7%) is higher than ComCoE RBFANN (i.e. 85.4%). These results indicate the improvement of the proposed model from the ComCoE RBFANN [45] in the aspect of generalisation ability. The proposed model adopts a DET whereas the ComCoE RBFANN [45] imposes SET in the fitness evaluation process. In SET, a player is eliminated once losing in a match. On the other hand, a player in DET is only disqualified from participating in the next matches after losing two matches. In other words, in DET, after losing a match, instead of being eliminated immediately, each contestant has an additional opportunity to compete and thus, it helps in retaining good players for other matches. If a good player competes with another good player in DET, this may prevent one of the two good players being removed forever in early rounds of match. The individual ANNs in DET interact more rapidly than the individual ANNs in SET, and it is this rapid interaction among individuals that assures fitter individual ANNs for adaptation and results in better generalisation performance.

The *#h* of the proposed ComCoE(DET) ANN is significantly different from ComCoE RBFANN in all 10 classification tasks except for SPECTF Heart and Vehicle. This is because the *p*-values for *#h* are below 0.05 in these 10 classification tasks. Overall, the proposed ComCoE(DET) (with $\#h = 5.1$) generates a smaller network

Table 7
Comparing the results of the proposed model with other classification models.

Dataset	Proposed model		ELM-Tree [40]		C4.5 [40]		ELM [40]	
	aAcc (%)	*#h*	*aAcc* (%)	*#nodes*	*aAcc* (%)	*#nodes*	*AAcc* (%)	*#h*
Breast Cancer W-D	**98.0**	**3.3**	96.5	12.0	92.1	47.0	96.3	20.0
Breast Cancer W-P	**83.0**	**3.4**	78.8	5.0	63.1	65.0	77.8	20.0
New Thyroid Gland	**95.2**	**7.8**	91.7	17.0	91.7	27.0	90.2	20.0
Parkinsons	**92.1**	**3.1**	86.2	23.0	81.6	48.0	85.1	20.0
SPECTF Heart	**80.6**	**1.4**	79.8	8.0	74.1	64.0	79.1	20.0
Vehicle	**77.9**	22.5	64.3	68.0	66.6	423.0	62.7	**20.0**
Credit Approval	**86.0**	**2.8**	75.9	17.0	69.3	315.0	74.6	20.0
Ionosphere	**92.5**	**3.8**	85.8	41.0	88.9	60.0	86.6	20.0
Breast	96.5	**2.8**	**96.9**	13.0	93.7	77.0	96.4	20.0
Pima	**78.2**	**3.7**	77.6	10.0	70.2	325.0	77.2	20.0
Wine	**99.7**	**3.4**	98.3	18.0	92.1	23.0	97.8	20.0
Sonar	**85.1**	**3.7**	75.0	45.0	71.7	47.0	75.6	20.0
Average	88.7	5.1	83.9	23.1	79.6	126.8	83.3	20.0

Table 8
The ranking of the classification models based on *aAcc*.

Algorithm	Ranking
Proposed Model	**1.083**
ELM-Tree	2.292
ELM	3.083
C4.5	3.542

Table 9
The ranking of the classification models based on *#nodes*.

Algorithm	Ranking
Proposed Model	**1.083**
ELM-Tree	2.333
ELM	2.583
C4.5	4.000

size than ComCoE RBFANN (with $\#h = 7.7$). In other words, the proposed ComCoE(DET) could provide a compact network architecture and perform with a higher average test accuracy rate than ComCoE RBFANN.

4.3. Performance comparison with other classification models

The classification results of the proposed model are also compared against other classification models, which are extreme learning machine (ELM), the decision tree algorithm C4.5, and their hybrid extreme learning machine tree (ELM-Tree) [40]. C4.5 is a well-known algorithm introduced by Quinlan [48]. C4.5 builds decision trees to classify data [48]. An ELM-Tree is an improved version of a model tree [49]. Instead of having linear regression function in each leaf node (as practiced by a model tree), each leaf node of ELM-Tree is an ELM [40]. ELM is applied to train a feedforward ANN consisting of a single-hidden layer. The input weights of an ELM are randomly assigned. The weights of an ELM's output are analytically decided by imposing pseudoinverse of the output matrix of the hidden layer [50]. In this comparative study, we performed 10 runs of 10-fold cross validation on ComCoE(DET) ANN to comply to the experimental setup as described in [40]. Table 7 compares the classification results of ComCoE(DET), ELM, C4.5 and ELM-Tree. The results of ELM, C4.5 and ELM-Tree in 12 classification tasks are taken from [40], where these classification tasks utilise the first 12 datasets listed in Table 1.

A rank-based nonparametric Iman–Davenport test [47] was applied to conduct multiple comparisons among the proposed model, ELM-Tree, C4.5 and ELM [40] by referring to their results in test accuracy rates *aAcc* and the quantity of the extracted information (either *#nodes* of a tree or *#h* of an ANN, or simply *#nodes*). The *p*-values from the Iman–Davenport test for *aAcc* and *#nodes* are around 0; they are smaller than 0.05. Therefore, the classification performances of these models are significantly different in both aspects of *aAcc* and *#nodes*. Tables 8 and 9 list the average ranks of these models in terms of the classification accuracy and the number of nodes. The proposed ComCoE(DET) ANN is positioned at the top in both rankings of *aAcc* and *#nodes*. As compared to ELM-Tree, C4.5 and ELM, ComCoE(DET) ANN has the most compact

Table 10
The statistical results from the Holm test for *aAcc* in which ComCoE(DET) ANN is the control method.

i	Algorithm	Z	p-value	$\alpha \backslash i$	Hypothesis ($\alpha = 0.05$)
3	C4.5	4.6644	0.0000	0.0167	Rejected
2	ELM	3.7947	0.0001	0.0250	Rejected
1	ELM-Tree	2.2927	0.0219	0.0500	Rejected

Table 11
The statistical results from the Holm test for *#nodes* in which ComCoE(DET) ANN is the control method.

i	Algorithm	Z	p-value	$\alpha \backslash i$	Hypothesis ($\alpha = 0.05$)
3	C4.5	5.5340	0.0000	0.0167	Rejected
2	ELM	2.8460	0.0044	0.0250	Rejected
1	ELM-Tree	2.3717	0.0177	0.0500	Rejected

architecture and it has performed with the highest test accuracy. A Holm test [51] was also conducted wherein ComCoE(DET) ANN was chosen as the control method. The intention of applying the Holm test was to assure which classification model was inferior to ComCoE(DET) ANN after the Iman–Davenport test. Tables 10 and 11 summarise statistical results from the Holm tests based on *aAcc* and *#nodes*. In these Tables 10 and 11, the classification models are ordered by sorting the *z*-values in a descending direction. For a classification model arranged at the *i*th order, its *p*-value (i.e. p_i) that is computed from the Holm test is compared with the corresponding $\alpha \backslash i$ ($\alpha = 0.05$ in this study). If p_i is smaller than $\alpha \backslash i$, we can conclude that the performances between the ComCoE(DET) ANN and the *i*-ordered classification model are significantly different. By referring to the results in Tables 10 and 11, the test accuracy rates achieved by and the sizes of the ComCoE(DET) ANN, C4.5, ELM and ELM-Tree are significantly different. In a nutshell, the proposed ComCoE(DET) ANN model is more compact and is more accurate in classifying data when compared with these classification models.

5. Conclusions

In this paper, a competitive CoE-ANN model is proposed of which its formation is based on a double elimination tournament (DET) between two populations of different ANNs. A benchmark study which involves a series of performance comparisons between the proposed ComCoE ANN model and other classification models was reported. The experimental results reveal the benefit of imposing a DET during the fitness evaluation of CoE-ANN. DET promotes a more rapid interaction among ANNs from two populations and assures fitter ANNs to adapt more effectively during the CoE process. The empirical results show that the proposed ComCoE(DET) ANN is a compact network structure and can classify data correctly in high accuracy rates. The applicability of the proposed model in coping with real-world problems has yet been fully investigated. For this, the proposed model should be evaluated rigorously in the future using as many real datasets as possible from various industries.

Acknowledgements

The final stage of the research work in this paper is supported financially by the Ministry of Education Malaysia under a Fundamental Research Grant Scheme (grant number: MMUE.140074).

References

[1] R. Kicinger, T. Arciszewski, K.D. Jong, Evolutionary computation and structural design: a survey of the state-of-the-art, Comput. Struct. 83 (2005) 1943–1978.
[2] P.H. Raven, e-Study Guide for Environment, Content Technologies Inc., 2014.
[3] E. Popovici, A. Bucci, R.P. Wiegand, E.D. De Jong, Coevolutionary Principles, in: T.B. Grzegorz Rozenberg, J.N. Kok (Eds.), Handbook of Natural Computing, Springer, Berlin, Heidelberg, 2010.
[4] B. Ilhem, L. Julien, S. Patrick, A survey on optimization metaheuristics, Inf. Sci. 237 (2013) 82–117.
[5] R.P. Wiegand, An Analysis of Cooperative Coevolutionary Algorithms, Department of Computer Science, George Mason University, 2004.
[6] J. Paredis, Steps towards coevolutionary classification neural networks, in: R. Brooks, P. Maes (Eds.), Proceedings Artificial Life IV, MIT Press/Bradford Books, 1994, pp. 102–108.
[7] J. Paredis, Coevolutionary life-time learning, in: Proceedings of the Fourth International Conference on Parallel Problem Solving from Nature, Springer-Verlag, 1996, pp. 72–80.
[8] H. Juillé, J.B. Pollack, Co-evolving intertwined spirals, in: Proceedings of the Fifth Annual Conference on Evolutionary Programming, MIT Press, Cambridge, Mass, 1996, pp. 461–468.
[9] H. Juillé, J.B. Pollack, Dynamics of co-evolutionary learning, in: Proceedings of the Fourth International Conference on Simulation of Adaptive Behavior, MIT Press, Cambridge, Mass, 1996, pp. 526–534.
[10] A.A. Douglas, J.C.B. Helio, F.F.E. Nelson, Coevolution of data samples and classifiers integrated with grammatically-based genetic programming for data classification, in: Proceedings of the 10th Annual Conference on Genetic and Evolutionary Computation, ACM, Atlanta, GA, USA, 2008.
[11] C. Stoean, R. Stoean, M. Preuss, D. Dumitrescu, Coevolution for classification, Technical Report CI-239/08, Collaborative Research Center on Computational Intelligence, University of Dortmund, 2008.
[12] A.R. McLntyre, M.I. Heywood, Toward co-evolutionary training of a multi-class classifier, in: Proceedings of the 2005 IEEE Congress on Evolutionary Computation, IEEE Press, Edinburgh, UK, 2005, pp. 2130–2137.
[13] C. Rohitash, F. Marcus, Z. Mengjie, Crossover-based local search in cooperative co-evolutionary feedforward neural networks, Appl. Soft Comput. 12 (2012) 2924–2932.
[14] N. García-Pedrajas, C. Hervás-Martínez, J. Munoz-Perez, COVNET: a cooperative coevolutionary model for evolving artificial neural networks, IEEE Trans. Neural Netw 14 (2003) 575–596.
[15] N. García-Pedrajas, D. Ortiz-Boyer, C. Hervás-Martínez, Cooperative coevolution of generalized multi-layer perceptrons, Neurocomputing 56 (2004) 257–283.
[16] M.Q. Li, J. Tian, F.Z. Chen, Improving multiclass pattern recognition with a co-evolutionary RBFNN, Pattern Recognit. Lett. 29 (2008) 392–406.
[17] C. Rohitash, F. Marcus, Z. Mengjie, Modularity adaptation in cooperative coevolution of feedforward neural networks, in: Proceedings of the International Joint Conference on Neural Networks (IJCNN 2011), IEEE Press, San Jose, California, 2011, pp. 681–688.
[18] J. Tian, M. Li, F. Chen, J. Kou, Coevolutionary learning of neural network ensemble for complex classification tasks, Pattern Recognit. 45 (2012) 1373–1385.
[19] M. Perez-Godoy, A. Rivera, F. Berlanga, M. Del Jesus, CO^2RBFN: an evolutionary cooperative-competitive RBFN design algorithm for classification problems, Soft Comput. 14 (2010) 953–971.
[20] A.A. Ladjici, A. Tiguercha, M. Boudour, Nash equilibrium in a two-settlement electricity market using competitive coevolutionary algorithms, Int. J. Electr. Power Energy Syst. 57 (2014) 148–155.
[21] J. Wang, Z. Zhou, A. Botterud, An evolutionary game approach to analyzing bidding strategies in electricity markets with elastic demand, Energy 36 (2011) 3459–3467.
[22] J. Byl, Organizing Successful Tournaments, fourth ed., Human Kinetics, USA, 2013.
[23] C. Deck, E.O. Kimbrough, Single- and double-elimination all-pay tournaments, J. Econ. Behav. Organ. 116 (2015) 416–429.
[24] S. Ozbeklik, J.K. Smith, Risk taking in competition: evidence from match play golf tournaments, J. Corporate Finance (2014) In Press.
[25] E.S. Forrest, P.J. Angeline, J.B. Pollack, Competitive Environments Evolve Better Solutions for Complex Tasks, 1997, p. GA93.
[26] M. Szubert, W. Jaskowski, K. Krawiec, On scalability, generalization, and hybridization of coevolutionary learning: a case study for Othello, IEEE Trans. Comput. Intell. AI Games 5 (2013) 214–226.
[27] B. Al-Khateeb, G. Kendall, Introducing individual and social learning into evolutionary checkers, IEEE Trans. Comput. Intell. AI Games 4 (2012) 258–269.
[28] L. Panait, S. Luke, A comparison of two competitive fitness functions, in: Proceedings of the Genetic and Evolutionary Computation Conference, Morgan Kaufmann Publishers Inc, 2002, pp. 503–511.
[29] Z. He, G.G. Yen, Comparison of many-objective evolutionary algorithms using performance metrics ensemble, Adv. Eng. Softw. 76 (2014) 1–8.
[30] L. Guthrie, W. Liu, Y. Xia, Text classification with tournament methods, Lecture Notes Comput. Sci. 3658 (2005) 77–84.
[31] I.T. Nabney, NETLAB: Algorithms for Pattern Recognition, Springer-Verlag, New York, 2002.
[32] M.F. Moller, A scaled conjugate gradient algorithm for fast supervised learning, Neural Networks 6 (1993) 525–533.
[33] S. Baskar, P. Subbaraj, M.V.C. Rao, Performance of hybrid real coded genetic algorithms, Int. J. Comput. Eng. Sci. 02 (2001) 583–601.
[34] X. Yao, Y. Liu, A new evolutionary system for evolving artificial neural networks, IEEE Trans. Neural Networks 8 (1997) 694–713.
[35] A.E. Eiben, M. Schoenauer, Evolutionary computing, Inf. Process. Lett. 82 (2002) 1–6.
[36] R. Watson, J. Pollack, Coevolutionary dynamics in a minimal substrate, in: L. Spector, E. Goodman, A. Wu, W.B. Langdon, H. Voigt, M. Gen, S. Sen, M. Dorigo, S. Pezeshk, M. Garzon, E. Burke (Eds.), Proceedings of the Genetic and Evolutionary Computation Conference, Morgan Kaufmann, 2001, pp. 702–709.
[37] P. Francisco, M. Tom, B. Matthew, Machine learning classifiers and fMRI: a tutorial overview, Neuroimage 45 (2009) S199–S209.
[38] R.O. Duda, P.E. Hart, D.G. Stork, Pattern Classification, John Wiley & Sons, Inc, New York, 2001.
[39] T.M. Mitchell, Machine Learning, McGraw-Hill, 1997.
[40] R. Wang, Y.L. He, C.Y. Chow, F.F. Ou, J. Zhang, Learning ELM-tree from big data based on uncertainty reduction, Fuzzy Sets Syst. 258 (2014) 79–100.
[41] R. Kohavi, A study of cross-validation and bootstrap for accuracy estimation and model selection, in: International Joint Conference on Artificial Intelligence, 1995, pp. 1137–1143.
[42] R.R. Bouckaert, Choosing between two learning algorithms based on calibrated tests, machine learning, in: Proceedings of the Twentieth International Conference, 2003, pp. 51–58.
[43] L.M. Almeida, T.B. Ludermir, A multi-objective memetic and hybrid methodology for optimizing the parameters and performance of artificial neural networks, Neurocomputing 73 (2010) 1438–1450.
[44] Q. Sultan Noman, S. Siti Mariyam, H. Siti Zaiton Mohd, D. Maslina, A.-S. Eiman, Memetic multiobjective particle swarm optimization-based radial basis function network for classification problems, Inf. Sci. 239 (2013) 165–190.
[45] B.Y. Hiew, S.C. Tan, W.S. Lim, Intra-specific competitive co-evolutionary artificial neural network for data classification, Neurocomputing 185 (2016) 220–230.
[46] K. Bache, M. Lichman, UCI machine learning repository. http://archive.ics.uci.edu/ml, 2013.
[47] D.J. Sheskin, Handbook of Parametric and Nonparametric Statistical Procedures, fourth ed., CRC Press, Boca Raton, FL, USA, 2007.
[48] J.R. Quinlan, C4.5: Programs for Machine Learning, Morgan Kaufmann Publishers Inc., San Mateo, California, 1992.
[49] E. Frank, Y. Wang, S. Inglis, G. Holmes, I.H. Witten, Using model trees for classification, Mach. Learn. 32 (1998) 63–76.
[50] G.B. Huang, Q.Y. Zhu, C.K. Siew, Extreme learning machine: theory and applications, Neurocomputing 70 (2006) 489–501.
[51] S. Holm, A simple sequentially rejective multiple test procedure, Scand. J. Stat. 6 (1979) 65–70.

Bee Yan Hiew received her B.IT. (Hons) and her Master M.Sc. (IT) from University of Malaya, Malaysia and Multimedia University, Malaysia respectively. She was a lecturer of Faculty of Information Science and Technology, Multimedia University, Malaysia. She is presently a Ph.D. student of Faculty of Information Science and Technology, Multimedia University, Malaysia. Her research interests include fingerprint recognition, image processing, data classification, co-evolutionary algorithm and artificial neural network.

Shing Chiang Tan received his B.Tech. (Hons.) and M.Sc. (Eng.) degrees from Universiti Sains Malaysia. He obtained his Ph.D. degree from Multimedia University, Malaysia. He is currently an associate professor with the Faculty of Information Science and Technology, Multimedia University. His current research interests include computational intelligence (artificial neural networks, evolutionary algorithms, fuzzy logic, and decision trees) and its applications, which include but are not limited to, data classification, pattern recognition, industrial process condition monitoring, fault detection and diagnosis, optimisation and biomedical disease data analysis and classification. He was a recipient of the Matsumae International Foundation Fellowship, Japan, in 2010.

Way Soong Lim received his Ph.D. degree in Electronic Engineering from Multimedia University of Malaysia, in 2007. He joined the university in 1999 as a lecturer in the Faculty of Engineering and Technology. Since 2008, he is serving the faculty as a Deputy Dean and later promoted to associate professor in 2010. His research interest covers areas in artificial intelligence, neural network, pattern recognition, condition monitoring and robotics navigation. Along his career path, he has published numerous number of international journals and conference papers, book chapters and engineering text book. He is also a member of IET.