Evolving Systems (2021) 12:319–336
https://doi.org/10.1007/s12530-019-09280-x

ORIGINAL PAPER

An efficient method for classifying motor imagery using CPSO-trained ANFIS prediction

M. R. Mosavi[1] · A. Ayatollahi[1] · S. Afrakhteh[1]

Received: 23 December 2018 / Accepted: 8 March 2019 / Published online: 29 March 2019

Abstract

Adaptive neuro-fuzzy inference system (ANFIS) is a hybrid of two soft computing methods of the artificial neural network (ANN) and fuzzy logic. Fuzzy logic has the advantage to change the qualitative aspects of human knowledge and insights into the process of precise quantitative analysis. However, it does not have a defined method that can be used as a guide in the process of transformation and human thought into rule-based fuzzy inference system (FIS). The fuzzy system cannot learn or adapt itself to the new environment, while the ANN is ambiguous to the user. By combining these two methods, the ANN becomes more transparent, and the fuzzy system takes on the ability of learning. With this combination, a more effective model in the medical domain could be built. In this paper, the ANFIS trained with corrected particle swarm optimization (CPSO), is proposed to classify the brain–computer interface (BCI) motor imagery mental tasks based on electroencephalography signals. The dataset is used in this paper, is BCI competition IV dataset. For evaluating the proposed method in order to obtain more classification rate, it is compared to the conventional ANFIS and the ANFIS trained by other evolutionary algorithms (EAs) such as genetic algorithm, particle swarm optimization, differential evolution, and biogeography-based optimization (BBO) that are more closely to the CPSO algorithm than the other EAs. The results showed that the ANFIS prediction trained by CPSO algorithm has more performance compared to conventional ANFIS prediction and ANFIS trained by other EAs.

Keywords Brain–computer interface · Neural networks · Adaptive neuro-fuzzy inference system · Particle swarm optimization

1 Introduction

Although brain–computer interface (BCI) systems are not usable in comparison with other control devices, for people who are motorically disabled, BCI systems are the most important means of communicating with the external world (Schalk et al. 2004). Devices creating a bridge between people and the environment using brain signals are called BCI-based systems. After several decades of research and many of the improvements, particularly over the last few years, we are still far from having the daily use of the BCI systems. Most of these systems are used for medical purposes. BCI systems have an interesting feature: these systems are the only human-known tools which require no muscle movements; therefore, brain and computer systems are beneficial for those who have lost their ability to control muscles (Wolpaw et al. 2002). For patients suffering from locked-in syndrome, BCI systems are the most critical manner of communicating with the outside world, which enhances their autonomy about the environment (Moore 2003). Many studies have been conducted, considering the importance of the issue. In this area, researchers have been used the methods of machine learning and signal processing in order to enhance the efficiency of the BCI systems. Pfurtscheller et al. (1998) proposed an adaptive autoregressive (AAR) algorithm for classification of electroencephalography (EEG) signals. Lemm et al. (2004) conducted the probabilistic modeling of sensorimotor μ rhythms for hand movement imaginary classification. Zhou et al. (2008) classified the mental task

✉ M. R. Mosavi
m_mosavi@iust.ac.ir

A. Ayatollahi
Ayatollahi@iust.ac.ir

S. Afrakhteh
s_afrakhteh94@elec.iust.ac.ir

[1] School of Electrical Engineering, Iran University of Science and Technology, Narmak, Tehran 16846-13114, Iran

features using linear discriminant analysis (LDA) and support vector machines (SVM). Ma et al. (2016) used of particle swarm optimization (PSO) algorithm for optimizing the performance of SVM classifier. Subasi and Ercelebi (2005) applied neural networks for EEG signal classification and we classify EEG signal using neural networks trained by hybrid population physic based algorithm in (Afrakhteh et al. 2018). The concept of EFS is intended in the twentieth century (Angelov and Buswell 2001; Angelov and Plamen 2013; Kasabov and Song 2002) to address the needs of flexible, yet robust and interpretable systems for the advanced industry, independent systems, and intelligent systems. Technical systems that claim to be smart are far from real intelligence. One of the main reasons is that information cannot be fixed, but is evolving. As human beings, during their life, learn from the experience and shape of the new laws, and accordingly their actions are adapted, overlooked and replaced by new laws, systems are also being evolve and should adapt themselves to different circumstances. In general, evolving fuzzy systems (EFS) can be of different type, e.g. of the so called Zadeh–Mamdani type (Zadeh 1973; Mamdani and Assilian 1975). The original TS type fuzzy system as described above is multi-input–single-output (MISO). EFS can also use multi-input–multi-output (MIMO) TS fuzzy systems which can be described in (Angelov et al. 2004a, b). Fuzzy rule-based classifiers with rules that are evolved from streaming data are called evolving fuzzy classifiers (EFC) (Angelov and Zhou 2008). Edwin Lughofer (2013) discussed achievements and open issues in the interpretability of EFS. Angelov et al. (2011) proposed a new real-time approach based on three modern techniques for automatic detection, object identification, and tracking in video streams, respectively. The novelty detection and object identification were based on the newly proposed recursive density estimation (RDE) method. Then evolving Takagi–Sugeno (eTS)-type fuzzy system was proposed for tracking. Precup et al. (2018) suggested a set of evolving Takagi–Sugeno–Kang (TSK) fuzzy models that characterize the finger dynamics of the human hand in the framework of myoelectric (ME) control of prosthetic hands. A novel evolving fuzzy ensemble classifier, namely Parsimonious Ensemble (pENsemble), is proposed in Pratama et al. (2018). pENsemble differs from existing architectures in the fact that it is built upon an evolving classifier from data streams, termed Parsimonious Classifier (pClass). A new on-line evolving clustering approach for streaming data was proposed in Baruah and Angelov (2012). The approach was based on the concept that local mean of samples within a region has the highest density and the gradient of the density points towards the local mean. In Angelov and Kasabov (2005), a new computational model for intelligent systems is presented based on data integration. This approach was also suitable for the integration of new data and other existing models into models that can be incrementally adapted to future incoming data. The mechanism for rule-base evolution, one of the central points of the algorithm together with the recursive clustering and modified recursive least squares (RLS) estimation, was studied in Angelov et al. (2004). In Angelov (2014), the new typicality and eccentricity based data analytics (TEDA) was proposed which was based on the spatially-aware concepts of eccentricity and typicality which represent the density and proximity in the data space. A new online evolving clustering approach for streaming data was proposed in Baruah and Angelov (2014). This approach efficiently estimated cluster centers based on the evolution of Takagi–Sugeno models.

The adaptive neuro-fuzzy interface system (ANFIS) is included of two approaches of neural networks and fuzzy. If we combine these two intelligent approaches, it will be achieving good reasoning in quality and quantity. In other words, we have fuzzy reasoning and network calculation. Various techniques have been proposed for the learning process that some of the most important ones are mentioned, to develop the ANFIS model. Mascioli et al. (1997) proposed a method that combines min–max and the ANFIS model to obtain an optimal set of fuzzy rules. Jang and Mizutani (1996) used the non-linear least square (LS) to train and determine the parameters of the ANFIS model. Also, in Jang (1993), gradient descent, LS, and sequential LS were used to update the model parameters in order to train the ANFIS model. The disadvantages of these methods are their high complexity and stuck in local traps. In this work, the ANFIS classifier is proposed which is trained using evolutionary algorithms (EAs). In other words, in the process of ANFIS training, instead of using the back propagation (BP) process, EAs such as PSO (Eberhart and Kennedy 1995), GA (Holland 1992), differential evolution (DE) (Storn and Price 1997) and biogeography-based optimization (BBO) (Simon 2008) are used and we use an EA based approach to update the ANFIS model. In PSO algorithm, in order to improve the exploration power of each particle, to avoid premature convergence and to fall into local traps, we put much importance in low iteration for personal experience, and in higher repetitions, the importance of this experience is reduced, and we give more value to the experience of the global. So, we introduce new version of PSO algorithm is called CPSO. In order to investigate the efficiency of the algorithm, we apply this algorithm to a set of benchmark functions with known global optimum. The graphical results showed that the CPSO algorithm is superior to the rest of the algorithms in finding the global optimum. Then, we use CPSO to optimize the ANFIS parameters. Finally, this CPSO-trained ANFIS used as the proposed classifier of the identification problem.

The structure of the paper is as follows. In Sect. 2, the main structure of the BCI system is prepared. In Sect. 3, the typical common spatial pattern (CSP) for feature extraction is introduced. Section 4 discusses ANFIS networks and describes the theory that governs them. In Sect. 5, the proposed method is presented. In Sect. 6, the results are discussed in detail. Finally, the conclusion of this paper is prepared in Sect. 7.

2 Main structure of BCI system

The main structure of EEG-based BCI system is shown in Fig. 1 and includes four part as follows (Wessel 2006): input of the BCI system that includes brain measurements, preprocessing on obtained signals from the previous step, feature translation process that is decomposed to two parts; feature extraction and classification, and output of the system that is classified signal for controlling the external device. In the pre-processing stage, EEG signals of each channel are sampled at 100 Hz. Then, these signals are filtered by a band-pass finite impulse response (FIR) filter with zero phase and the passband range of 8–30 Hz (Ramoser et al. 2000). This frequency band has been selected because: firstly, it includes frequency bands of μ (8–13 Hz) and β (14–30 Hz); secondly, the frequency of artifacts caused by eye movements and muscle movements is outside of this range. These artifacts and the 50 Hz electricity noise power will be removed by this filter (Jasper and Penfield 1949). The type of filter is used in this paper, is the third order Butterworth one. The reason for using this type is the smoothness of its response compared to the other common filters such as Chebyshev or Bessel.

In the next step, the obtained signals are given to the feature extraction stage. In this stage, the CSP method is used for feature generation. Finally, the obtained features are given to the classification stage to control the external device. In this paper, the data set 1, and the calibration data part of BCI competition IV are used (http://www.bbci.de/competition/iv/). This is a two-class dataset. The recording was made using Brain Amp MR plus amplifiers and an Ag/AgCl electrode cap. Signals were measured from 59 EEG positions that were most densely distributed over sensorimotor areas. This dataset has been recorded from seven subjects (Blankertz et al. 2007). The locations of EEG electrodes are shown in Fig. 2 which is plotted using EEGLAB toolbox in MATLAB. In this figure 'F', 'P', 'O' and 'C' indicate the frontal, parietal, occipital, and central parts of the head, respectively. In the next section, the feature extraction algorithm is described in detail.

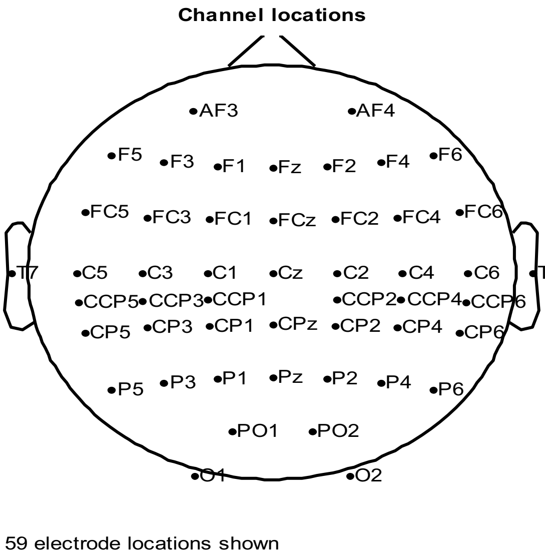


Fig. 2 Channel location of the EEG electrode for recording data (extracted from EEG-lab)

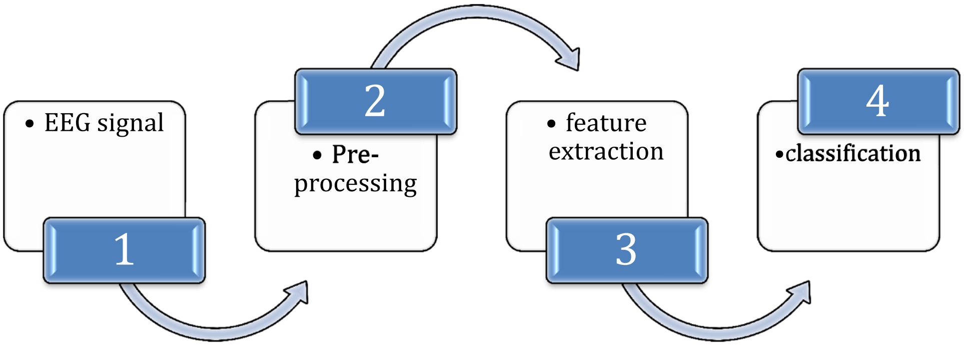
3 CSP method for feature extraction

The main goal of the CSP is to design a filter that maximizes the variance of the filtered signals of one class, while minimizing the variance of filtered signals from another class (Ramoser et al. 2000a, b; Lotte and Guan 2010; Arvaneh et al. 2011). Thus, the spatial filter V is obtained when the following function is maximized (Arvaneh et al. 2011):

$$J(v) = \frac{v^T C_1 v}{v^T C_2 v}. \tag{1}$$

In this case, T represents the transposed matrix and C_i represents covariance matrix of the i-th class data obtained from the following equation:

Fig. 1 Main structure of the BCI system



$$C_i = \frac{X_i.X_i^T}{trace\,(X_i.X_i^T)}, \tag{2}$$

where matrix X_i is the matrix of i-th class data. The data samples of each channel (electrode) are in one row per experiment. Finally, the average covariance matrix of different tests for class i is achieved (Simon 2008). This problem can be solved by generalized eigenvalue problem. However, it can be also solved by two times of standard eigenvalue problem. First, we decompose the total covariance matrix as follows:

$$C_1 + C_2 = UEU^T, \tag{3}$$

where U is a set of eigenvectors, and E is a diagonal matrix of eigenvalues. Next, we compute $P := \sqrt{E^{-1}U^T}$, then we have:

$$\hat{C}_1 = PC_1P^T, \tag{4}$$

$$\hat{C}_2 = PC_2P^T. \tag{5}$$

It should be noted that $\hat{C}_1 + \hat{C}_2 = I$. Thus, any orthogonal matrices F satisfies $F^T(\hat{C}_1 + \hat{C}_2)\,F = I$. Finally, it is decomposed as:

$$\hat{C}_1 = F\Lambda F^T, \tag{6}$$

where F is a set of eigenvectors and Λ is a diagonal matrix of eigenvalues. A set of CSP filters is obtained as:

$$V = P^T F. \tag{7}$$

So, we have:

$$V^T C_1\, V = \Lambda = \begin{bmatrix} \lambda_1 & & \\ & \ddots & \\ & & \lambda_{ch} \end{bmatrix}, \tag{8}$$

$$V^T C_2\, V = I - \Lambda = \begin{bmatrix} 1-\lambda_1 & & \\ & \ddots & \\ & & 1-\lambda_{ch} \end{bmatrix}, \tag{9}$$

where $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_{ch}$. Therefore, first CSP filter v_1 provides maximum variance of class 1, and last CSP filter v_{ch} provides maximum variance of class 2. We select first and last m filters to use as:

$$V_{csp} = \begin{pmatrix} v_1 & \cdots & v_m & v_{ch-m+1} & \cdots & v_{ch} \end{pmatrix} \in R^{2m\times ch}. \tag{10}$$

So, the filtered signal is given by:

$$y(t) = V_{csp}\, x(t) = (y_1(t) \quad \cdots \quad y_{2m}(t)\,)^T. \tag{11}$$

After the features are extracted, these features must be applied to the classifier. Because in the classification phase, ANFIS is used, we will state the ANFIS theory in the next section.

4 Adaptive neuro-fuzzy inference system

ANFIS systems are highly suggested for non-linear modeling systems, and their performance accuracy is related to the parameters of the initial structure of these systems. It includes the number of input variables, the number of membership functions, the type of these functions, the type of the membership functions, the rule number of the fuzzy system and the parameters related to the training of these systems, including the method of training and the initial conditions. The correct choice of these parameters is a matter which depends on the experience of the designer and the application of the ANFIS system because there are no general practical rules for this purpose.

Fuzzy logic can convert the qualitative aspects of human knowledge and insight into the process of detailed quantitative analysis. However, it is not a standard method that can be used as a guide in the process of personal conversion and be thinking to a fuzzy interface system (FIS), and also spent much time to adapt the membership functions (Cheng et al. 2005). ANN has a higher ability to learn and adapt to its environment. The primary goal of ANFIS is to optimize the parameters of the equivalent fuzzy logic system by employing a learning algorithm to the input–output dataset. An adaptive network is an example of a feed-forward neural networks with multiple layers. In the learning process, these networks often use a supervised learning algorithm. Also, the adaptive network has the architecture characteristics that consists of some adaptive nodes interconnected directly without any weight values between them. There are several types of FIS, namely Takagi–Sugeno, Mamdani, and Tsukamoto. A FIS of Takagi–Sugeno model was found to be widely used in the application of the ANFIS method (Eberhart and Kennedy 1995). As seen in Fig. 3, the ANFIS architecture has five layers that are discussed in detail. The name of *layer 1* is the *fuzzification* layer, where signals are obtained and transformed to the next layer.

In Fig. 3, for simplicity, it is assumed that there are two inputs x and y, and one output f. Two “If-Then” rules were used for Takagi–Sugeno model, as follows:

$$\begin{aligned} &Rule\,1: \;\; if\ x\ is\, A_1\ and\ y\ is\ B_1\ Then\ f_1 = m_1x + n_1y + r_1 \\ &Rule\,2: \;\; if\ x\ is\, A_2\ and\ y\ is\ B_2\ Then\ f_2 = m_2x + n_2y + r_2, \end{aligned} \tag{12}$$

where A1, A2, B1, and B2 are the membership functions of each input x and y in “if part,” while m1, n1, r1, m2, n2, and

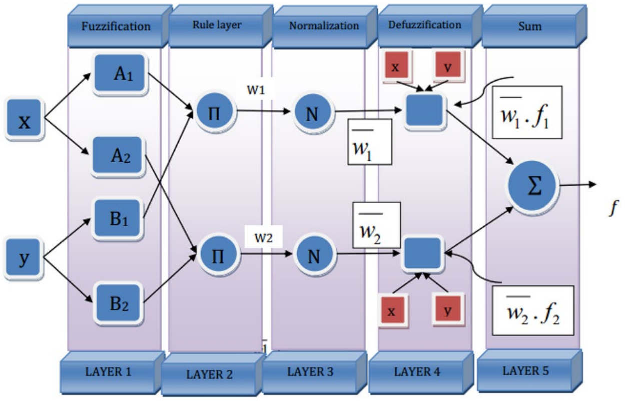


Fig. 3 A basic structure of ANFIS (Jang 1993)

r2 are linear parameters in "then part" of Takagi–Sugeno fuzzy inference model.

The outputs of layer1 are:

$$O_{1i} = \mu A_i(x), \quad i = 1, 2$$
$$O_{1i} = \mu B_{i-2}(x), \quad i = 3, 4, \tag{13}$$

where A_i, B_i are the membership functions of each input x and y. μA_i, μB_i are the membership degrees that are calculated for this function. For the Gaussian membership function μA_i is calculated as follows:

$$\mu A_i = \exp\left[-\left(\frac{x - c_i}{2\,a_i}\right)^2\right]. \tag{14}$$

In here, a_i and c_i are sigma and central parameters of the membership function, respectively. These parameters are the membership parameters that can change the membership function. The parameters in this layer are typically referred to as the premise parameters.

The name of layer 2 is the *rule layer*. In this layer, the circle nodes are labeled as Π. The output node is the result of multiplying of the signal coming into the node and delivered to the next node. Each node in this layer represents the firing strength for each rule. The output of this layer is calculated as follows:

$$O_{2i} = w_i = \mu A_i(x) \;*\; \mu B_i(y). \tag{15}$$

The name of the third layer is the *normalization* layer. The output of this layer is calculated as follows:

$$O_{3i} = \overline{w_i} = \frac{w_i}{\sum_i w_i}. \tag{16}$$

Layer 4 is named as *defuzzification* layer. Each node in this layer is an adaptive node to an output. The output of this layer is calculated by:

$$O_{4i} = \overline{w_i} f_i = \overline{w_i}\ (m_i x + n_i y + r_i), \tag{17}$$

where the $m_i x + n_i y + r$ is a parameter in the node. Finally, in layer 5, which is named as summation layer, computes the total output using the summation of its inputs from the previous layer:

$$O_{5i} = \sum_i \overline{w_i} f_i. \tag{18}$$

The ANFIS network has a set of parameters that we need to determine the best structure for this network to achieve optimal performance. Therefore, in the next section, a proposed method is presented for determining the optimal structure.

5 Proposed method

Try, and error methods and information categorization methods do not always guarantee the best structure. ANFIS has two parameter types that have to be updated. These are premise parameters and consequent parameters. Premise parameters belong to the gauss membership function that is given as {ai,ci} in Eq. (14). The total number of the premise parameters is equal to the sum of the parameters in all membership functions. Consequent parameters are the ones that are used in defuzzification layer, shown in Eq. (17) as {$\mathrm{m_i}$, $\mathrm{n_i}$, $\mathrm{r_i}$}. So, in this paper, an optimization method for these parameters based on EAs is used for this purpose, and its performance accuracy is examined. In neural networks, training is the process of calculating the weights of the neuron-connecting branches. In ANFIS systems, training and mathematical techniques are similar to neural networks, but the goal is to determine the parameters associated with membership functions. The form of the membership functions in the if-part and the parameters in the then-part, which are the same parameters of the output functions, are evaluated as the weights for identification. The accuracy of the trained ANFIS system depends on the structural parameters and parameters related to the training of these systems. In this paper, an ANFIS system trained by EAs such as CPSO is proposed in order to optimize the classification accuracy. The process of this method is shown in Fig. 4. As is evident, the proposed method is based on four steps:

Step 1: Initialization of ANFIS system parameters.

Step 2: The ANFIS system estimates the outputs based on features extracted from the feature extraction step.

Step 3: The outputs are compared with the target values, and the error is obtained. However, this error is not excellent. So, in order to optimize it, the learning process should be done.

Step 4: Using CPSO, some parameters are set to minimize this error or to arrive at an acceptable error.

The steps of 2–4 will be repeated until we reach the stop criteria and convergence condition.

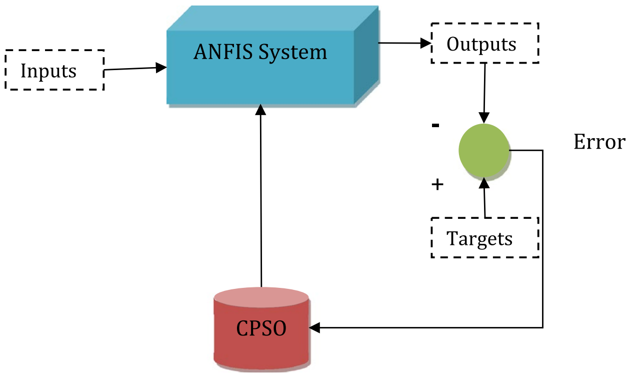


Fig. 4 Block diagram that shows the proposed method

In the following, we introduce an improved CPSO algorithm for ANFIS training.

5.1 Corrected particle swarm optimization (CPSO) algorithm for training the ANFIS system

5.1.1 Particle swarm optimization algorithm

PSO is a population-based stochastic optimization technique developed by Eberhart and Kennedy in 1995, inspired by the social behavior of birds (Holland 1992). This method uses the number of particles (candidate solutions) in the search space to find the best solution. All particles travel towards the best particle (best solution) that are on their way. PSO is initialized with a group of random particles (solutions) and then searches for optima by updating generations. In each iteration, each particle is updated by following two *best* values. The first one is the best solution (fitness) that has achieved so far (the fitness value is also stored), which is called *pbest*. Another best value that is tracked by the particle swarm optimizer is the best value in the population. This best value is a *global best* and called *gbest*. After finding the two best values, the particle updates its velocity and positions with the following Eqs. (19) and (20):

$$v_i^{t+1} = wv_i^t + c_1 \times rand() \times (pbest_i - \chi_i^t) + c_2 \times rand() \times (gbest - \chi_i^t), \tag{19}$$

$$\chi_i^{t+1} = \chi_i^t + v_i^{t+1}, \tag{20}$$

where v_i^t is the particle velocity, χ_i^t is the current position of a particle, *pbest*, and *gbest* are defined as stated before, rand () is a random number in (0, 1) and c1 and c2 are learning factors. The cognitive coefficients including *c1* and *c2* in standard PSO have a value of 2 (c1 = c2 = 2). *w* is the inertia component keeps the particle moving in the same direction it was originally heading. In other words, the inertia weight is used to control the effect of the previous velocities on the current velocity. This makes compromise between a global and local exploration abilities of the swarm. The size of inertia weight is closely related to searching ability of particle. The larger value of the inertia weight is the larger particle velocity is. *rand ()* is a uniformly distributed random number between 0 and 1 and it is used to introduce a stochastic element in the search process. In this algorithm, the particles are initialized with random positions and velocities in the problem space. The number of repetitions is 500, c1 = c2 = 2, and the inertia weight is also linearly reduced from 0.9 to 0.4.

In the next part, we present a CPSO algorithm to achieve global optimum.

5.1.2 Main idea of CPSO

In the PSO algorithm, the cognitive coefficients, c1, c2, with random vectors *rand* () control the statistical effect of cognitive components on the total velocity of a particle. The coefficient c1 determines how much a particle is assured to itself and its nearest neighbor, while the coefficient c2 indicates how much other neighbors can be trusted. If c1 = c2 = 0, then the particle moves at the same speed as before to reach the boundary of the search space, which will cause a lot of time and cost. If c1 > 0 and c2 = 0, all particles move independently, and the experiment of any particles does not affect the motion of another. In fact, each particle finds the best place. If the best location is found better than the previous one, it replaces the new place. In this case, we say that the particle is a local search. If c2 > 0 and c1 = 0, all particles are absorbed into a particle and do not care about their experience and trust the best particle. However, the whole community can collaborate and make the right answer by sharing their personal and global experiences with less time and cost. Therefore, this collaboration can be useful when c1 and c2 have a right balance between each other. In many applications, c1 = c2 is used to drag particles into average positions. If c1 >> c2, each particle goes further towards the best individual experience. If c2 >> c1, particles are more attracted to the best global experience. In this paper, in order to improve the exploration power of each particle, to avoid premature convergence and to fall into local traps, we put much importance in low iteration for personal experience, and in higher repetitions, the importance of this experience is reduced, and we give more value to the experience of the global. With these interpretations, we improve the cognitive coefficients c1 and c2 as follows:

$$c_1(t) = 3 - c_2(t)$$

$$c_2(t) = \min\{c_2\} \times \log\left\{ \left(\frac{\max\{c_2\}}{\min\{c_2\}} \right)^{\frac{2}{1+20\,(iter/\max\_iter)}} \right\}, \tag{21}$$

where *iter* and *max_iter* indicate the iteration and maximum iteration number, respectively. We named this version of

PSO as CPSO. The changes in values of c1 and c2 in various times are shown in Fig. 5.

In the next part, we present the optimization problem in this paper and the link of the CPSO algorithm to this problem is determined.

5.1.3 Optimization problem

For representing the problem, we have to identify the variables of the problem of training ANFIS. The main parameters of an ANFIS are the premise parameters and consequent parameters. As mentioned in Sect. 5, the premise parameters and consequent parameters are {ai, ci} and {$\mathrm{m_i}$, $\mathrm{n_i}$, $\mathrm{r_i}$}, respectively. Therefore, they have to be represented as a vector as follows:

$$\begin{aligned} x &= \{\text{premise parameters}\,,\ \text{consequent parameters}\} \\ &= \left\{\{ai,\ ci\},\ \left\{m_i,\ n_i,\ r_i\right\}\right\}. \end{aligned} \tag{22}$$

This vector includes all the parameters to be optimized by a training algorithm. The number of variables in this vector defines the dimension of the search agent (candidate solutions) in evolutionary algorithms such as CPSO.

The next step is to define the objective function. In this study, the measurement for quantifying the performance of ANFIS, is Mean Squared Error (MSE), which is defined as follows:

$$MSE = \sum_{i=1}^{m} (T_i^k - O_i^k)^2. \tag{23}$$

The MSE can be calculated for each of training samples. However, an ANFIS should be adapted to classify all training samples. So, we calculate the MSE for all the training samples and average them. This gives overall performance of ANFIS when classifying training samples. The equation for this purpose is as follows:

$$\overline{MSE} = \sum_{k=1}^{N} \frac{\sum_{i=1}^{m} (T_i^k - O_i^k)^2}{N}, \tag{24}$$

where $\mathrm{T_i^k}$ is the desired output of the *i*-th input, $\mathrm{O_i^k}$ is the actual output of the *i*-th input unit when the *k*-th training sample appears in training. m is the number of outputs and n is the number of training samples. So, the optimization problem can be formulated as follows:

$$Minimize\colon f(\vec{x}) = \overline{MSE}. \tag{25}$$

Finally, we use the CPSO algorithm to optimize it. In the next section, several analyzes are performed on some benchmark functions. Also, the analysis of the results from the classification of motor imagery is presented.

6 Simulation results

6.1 Applying CPSO on some mathematical benchmark functions

In the case of global optimization, an efficient algorithm should possess two abilities, namely exploration and exploitation. The exploration is the ability of an algorithm to search the whole feasible space of the problem. In contrast to exploration, the exploitation is the convergence ability to the best solution near a good solution. Therefore, in order to investigate the efficiency of the algorithm, we apply this algorithm to a set of benchmark functions with known global optimal. These test functions are divided into two groups: the first group of unimodal functions (F1–F7) is suitable for

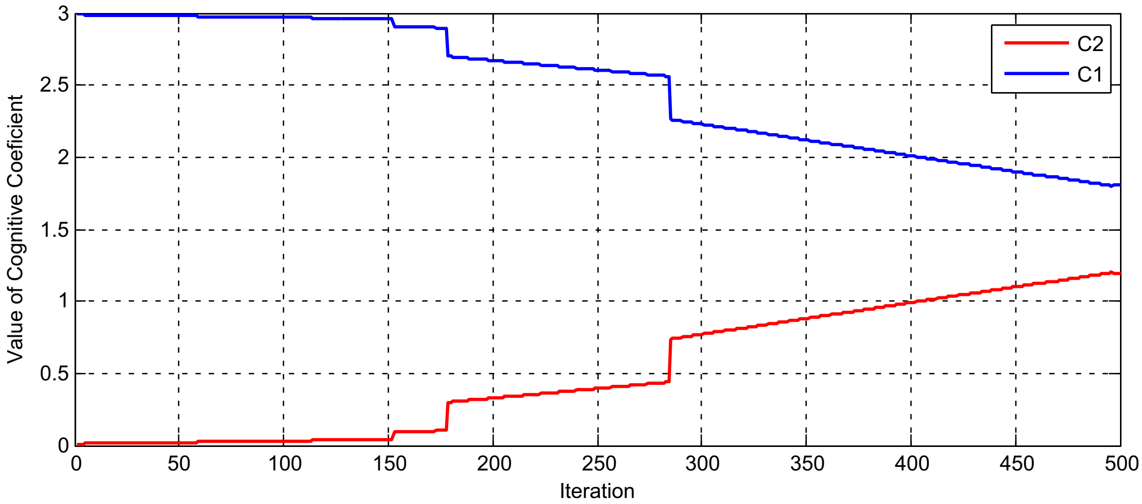
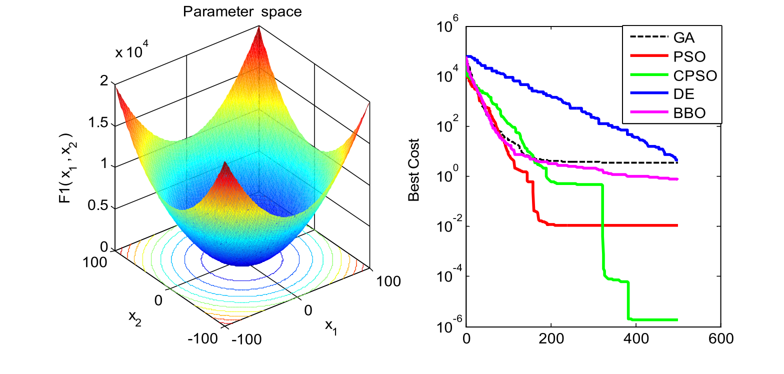
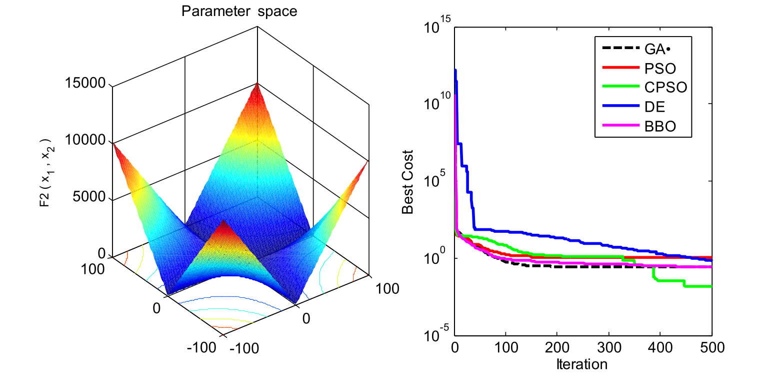
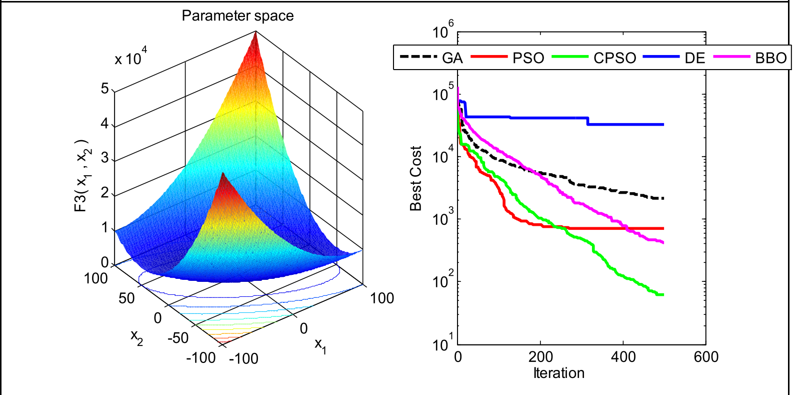
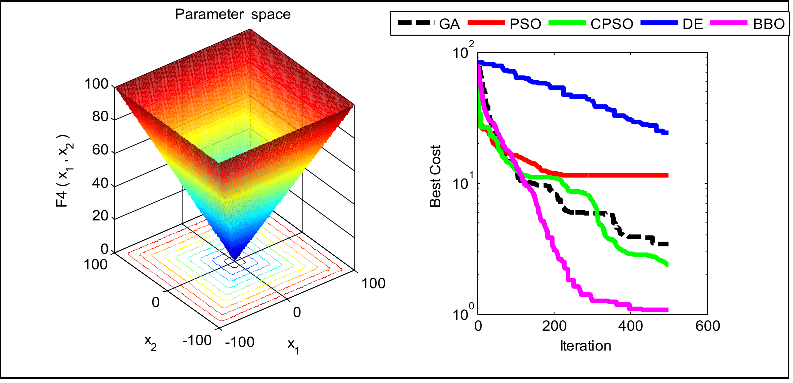


Fig. 5 The updating of cognitive coefficients in different repetitions

Table 1 Applying CPSO on some benchmark function and comparing with other algorithms (for all benchmark functions $f_{min} = 0$)

Function 1 Formula: $f_1(x) = \sum_{i=1}^{n} x_i^2$ $Dim = 100$ $Range = [-100, 100]$ $f_{\min} = 0$	
Function 2 Formula: $f_2(x) = \sum_{i=1}^{n} \lvert x_i \rvert + \prod_{i=1}^{n} \lvert x_i \rvert$ $Dim = 100$ $Range = [-100, 100]$ $f_{\min} = 0$	
Function 3 Formula: $f_3(x) = \sum_{i=1}^{n} \left(\sum_{j-1}^{i} x_j \right)^2$ $Dim = 100$ $Range = [-100, 100]$ $f_{\min} = 0$	
Function 4 Formula: $f_4(x) = \max\left\{ \lvert x_j \rvert ,\ 1 \le i \le n \right\}$ $Dim = 100$ $Range = [-100, 100]$ $f_{\min} = 0$	

exploitation testing because these functions have an optimal point and no other local optimal. However, the second group is multimodal functions (F8–F12) that have a large number of local optimums and can be helpful for examining the exploration and preventing falling into local traps. The mathematical formulation of these two groups of test functions, the graphical results of the proposed algorithm and its comparison with the other algorithms examined in this paper

Table 1 (continued)

<table>
<tr>
<td>

Function 5

Formula:

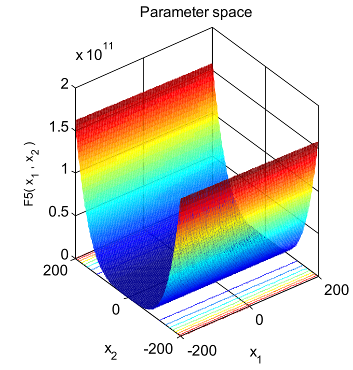
$$f_5(x) = \max\left\{\left|x_j\right|,\ 1 \le i \le n\right\}$$

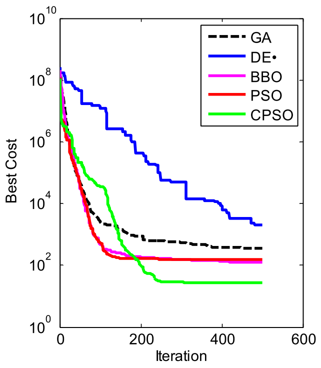
$$Dim = 100$$

$$Range = [-200, 200]$$

$$f_{\min} = 0$$

</td>
<td>





</td>
</tr>
<tr>
<td>

Function 6

Formula:

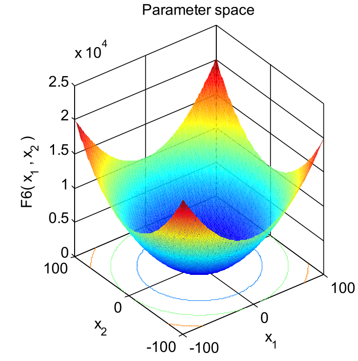
$$f_6(x) = \sum_{i=1}^{n} ([x_i + 0.5])^2$$

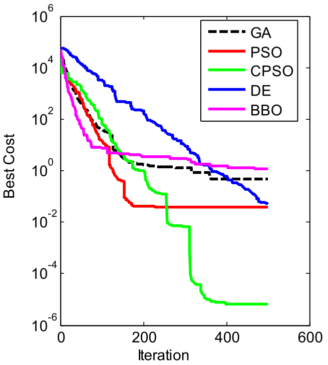
$$Dim = 100$$

$$Range = [-100, 100]$$

$$f_{\min} = 0$$

</td>
<td>





</td>
</tr>
<tr>
<td>

Function 7

Formula:

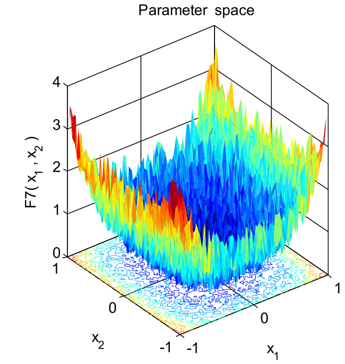
$$f_7(x) = \sum_{i=1}^{n} ix_i^{\ 4} + rand\,(0,1)$$

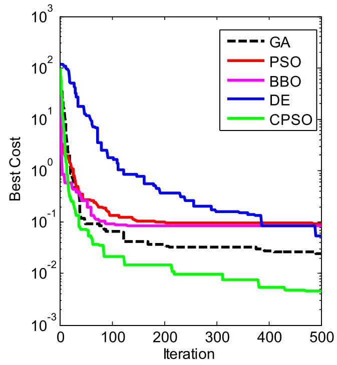
$$Dim = 100$$

$$Range = [-1, 1]$$

$$f_{\min} = 0$$

</td>
<td>





</td>
</tr>
<tr>
<td>

Function 8
Formula:

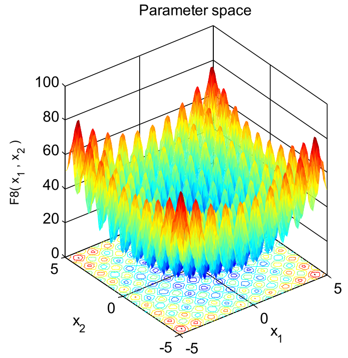
$$f_8(x) = \sum_{i=1}^{n} [x_i^{\ 2} + 10\cos(2\pi x_i) + 10]$$

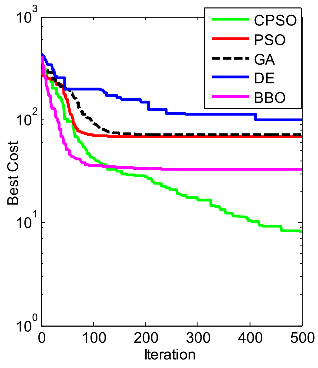
$$Dim = 100$$

$$Range = [-5, 5]$$

$$f_{\min} = 0$$

</td>
<td>

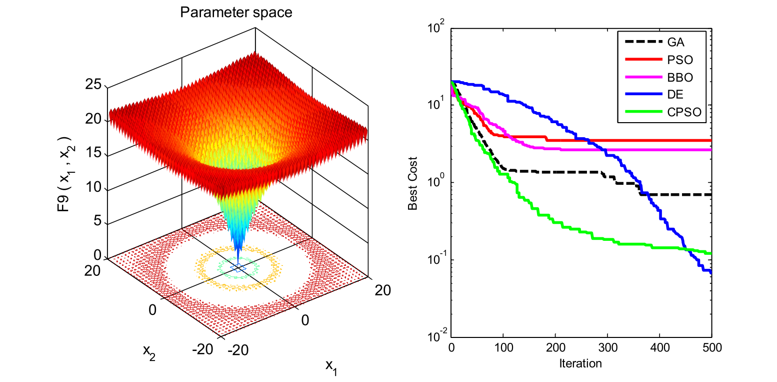
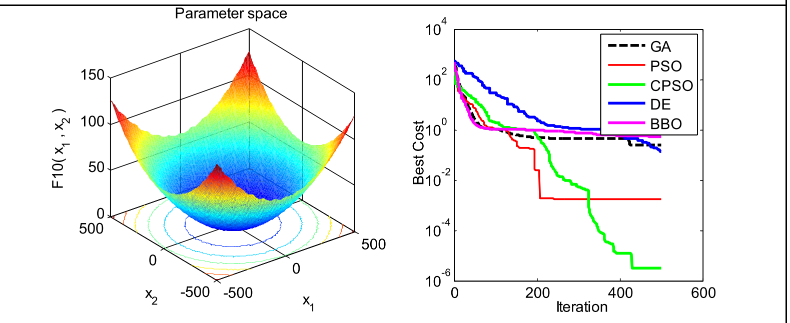
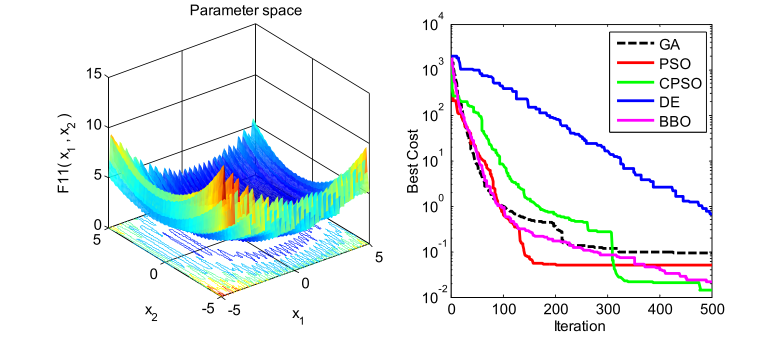
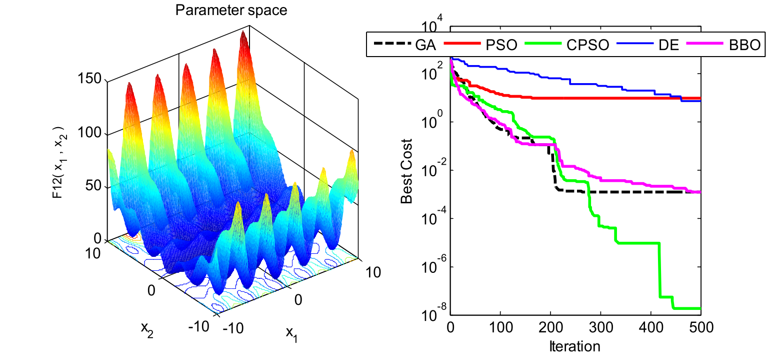




</td>
</tr>
</table>

are shown in Tables 1 and 2. This point should be noted that the convergence curves show the averages of 20 independent runs. Also, there are 30 search agent and 500 iterations for each algorithm. As shown in Tables 1 and 2, except for F4 in the set of unimodal functions, and F9 in the multimodal functions set, the CPSO algorithm is superior to the rest of

Table 1 (continued)

Function 9 Formula: $f_9(x) = -20\exp\left(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2}\right) - \exp\left(\frac{1}{n}\sum_{i=1}^{n}\cos(2\pi x_i)\right) + 20 + e$ $Dim = 100$ $Range = [-20, 20]$ $f_{\min} = 0$	
Function 10 Formula: $f_{10}(x) = \frac{1}{4000}\left(\sum_{i=1}^{n} x_i^2\right) - \prod_{i=1}^{n}\cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$ $Dim = 100$ $Range = [-500, 500]$ $f_{\min} = 0$	
Function 11 Formula: $f_{11}(x) = 0.1\{\sin^2(3\pi x_1) + \sum_{i=1}^{n-1}(x_i - 1)^2[1 + \sin^2(3\pi x_i + 1)] + (x_n - 1)^2[1 + \sin^2(2\pi x_i + 1)]\} + \sum_{i=1}^{n} u(x_i, 5, 100, 4)$ $Dim = 100$ $Range = [-5, 5]$ $f_{\min} = 0$	
Function 12 Formula: $f_{11}(x) = \frac{\pi}{n}\{10\sin(\pi y_1) + \sum_{i=1}^{n-1}(y_i - 1)^2[1 + 10\sin(\pi y_{i+1} + 1)] + (y_n - 1)^2\} + \sum_{i=1}^{n} u(x_i, 10, 100, 4)$ $Dim = 100$ $Range = [-10, 10]$ $f_{\min} = 0$	

the algorithms in finding the global optimum. Nevertheless, in F4 and F9, the CPSO algorithm is ranked second best regarding finding the optimal point. Therefore, the CPSO algorithm operates better than other algorithms both in the exploration and exploitation phase, since it proved to be more potent than the rest of the discussed algorithms in finding the global optimum for both unimodal and multimodal functions.

Table 2 Global optimum that obtained by each algorithm (for all benchmark functions $f_{min} = 0$)

Algorithm Function name	PSO	GA	DE	BBO	CPO
Function 1	0.010653	3.5904	4.5748	0.6919	**1.9208e−06**
Function 2	1.1305	0.28374	0.62687	0.26345	**0.015742**
Function 3	709.6118	2128.0712	32014.0023	413.5479	**61.6431**
Function 4	11.4886	3.4371	24.262	**1.0779**	2.3831
Function 5	160.3797	266.729	3356.092	151.5941	**27.7524**
Function 6	0.040064	0.47587	0.049303	1.2207	**6.3514e−06**
Function 7	0.093506	0.024613	0.05379	0.08636	**0.0046304**
Function 8	68.6525	71.6382	99.3118	32.6599	**8.1763**
Function 9	3.5184	0.68957	**0.059806**	2.6602	0.11906
Function 10	0.0017969	0.24685	0.13856	0.5149	**3.0952e−06**
Function 11	0.049474	0.095337	0.67223	0.02177	**0.014508**
Function 12	9.5955	0.0012145	7.312	0.0012708	**1.8828e−08**

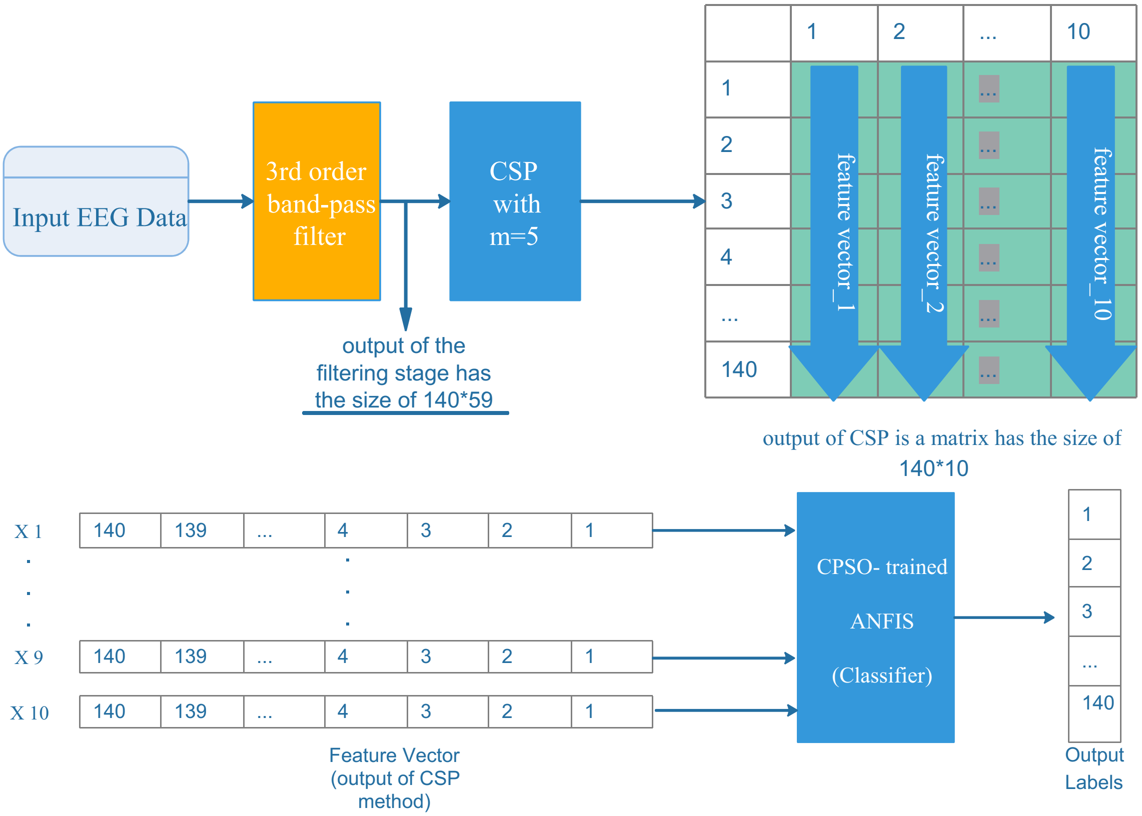

Input EEG Data
3rd order band-pass filter
CSP with m=5
output of the filtering stage has the size of 140*59
1
2
...
10
1
2
3
4
...
140
feature vector_1
feature vector_2
feature vector_10
...
output of CSP is a matrix has the size of 140*10
X 1
X 9
X 10
140
139
...
4
3
2
1
Feature Vector (output of CSP method)
CPSO- trained ANFIS (Classifier)
1
2
3
...
140
Output Labels

Fig. 6 Main procedure of the classification algorithms

6.2 Results of applying CPSO for ANFIS training to classify EEG signals

In this paper, our main goal is the classification of two motor imagery (right and foot) based on the EEG signal for the aforementioned dataset in Sect. 2. At first, in pre-processing, the EEG signal is filtered using a 3rd order band-pass filter. Then, in feature extraction, the CSP filter (with m = 5) is used. According to what was said in describing the CSP method in Sect. 3, the number of features is equal to 2 m (2m = 10). So, the output of the feature extraction stage is a 140 × 10 matrix with 140 labels. Now, this matrix must be applied to the CPSO-trained ANFIS classifier. The following figure shows this procedure (Fig. 6).

The main results include a comparison between desired outputs (targets) and calculated outputs (outputs). In this

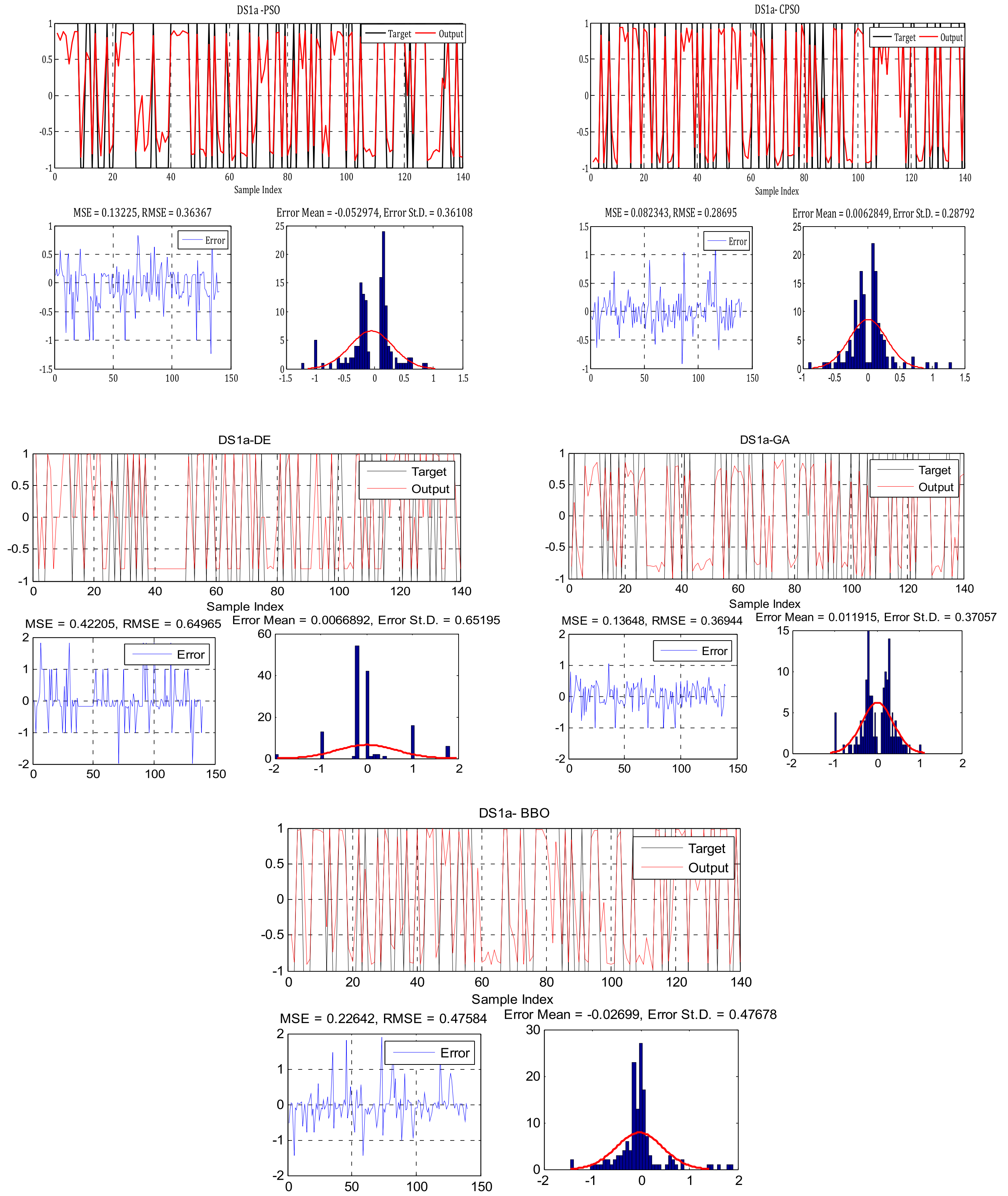


Fig. 7 Prediction results for DS1a

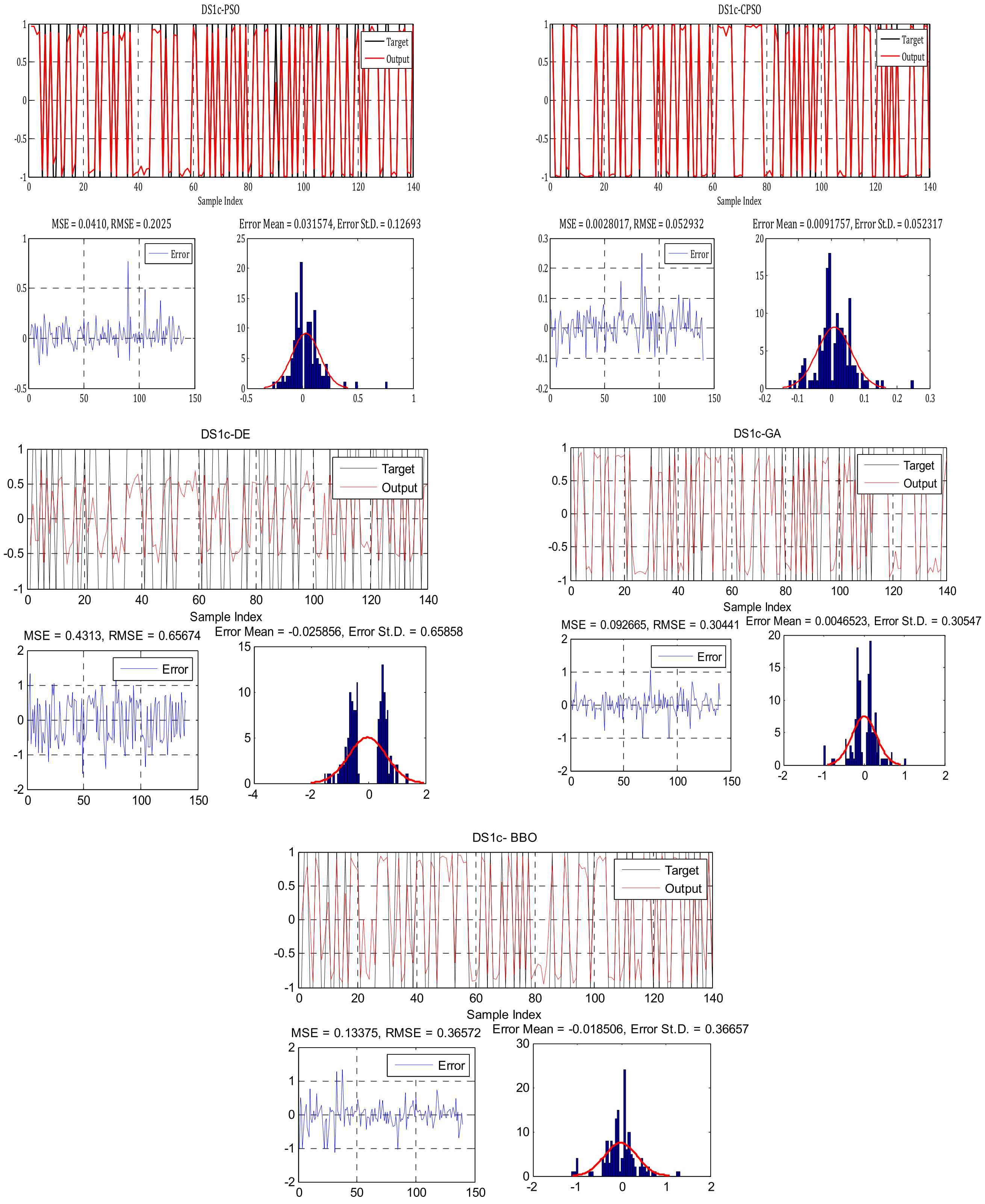


Fig. 8 Prediction results for DS1c

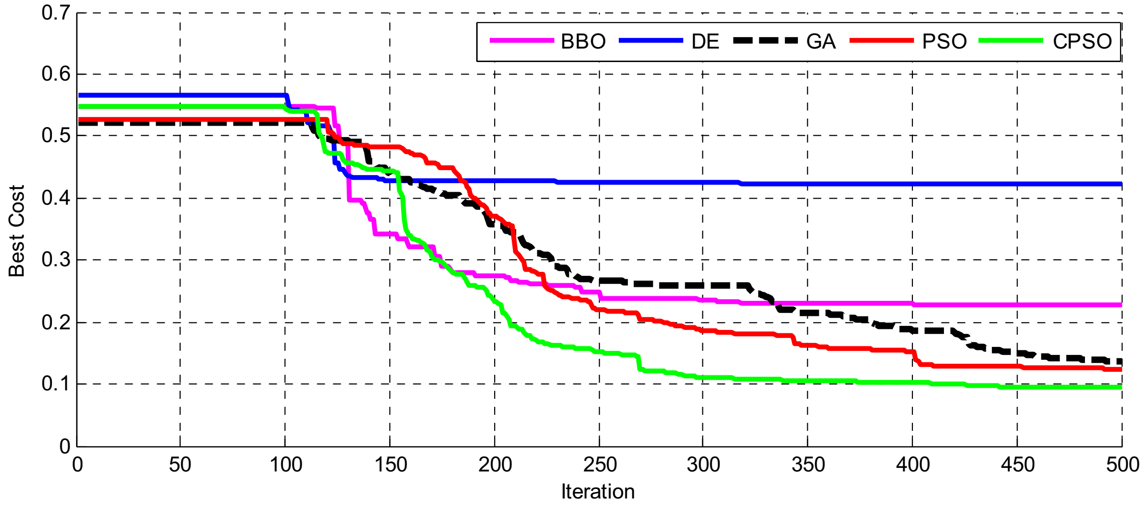


Fig. 9 The convergence curves for DS1a

part, the target and output curves are compared to each other, and the error curve is shown. Also, the error dispersion is shown around zero, and for each, the standard deviation is calculated for each one. For the reliability of the results, these were obtained from 20 independent runs, and the mean value of them are measured. These results are shown in Figs. 7 and 8 for two subjects DS1a and DS1c, respectively. The histogram displays the extent of the accumulation of error information.

- Convergence curves

Convergence curve defines the relationship between the grid interval and the analysis error. From these curves, it can be found that which algorithm has less error and more convergence speed. These curves are shown in Figs. 9 and 10 for DS1a and DS1c, respectively. Table 3 shows these results for all subjects. Also, Fig. 11 shows the bar plot of MSE values for all subjects.

Results reveal that the CPSO algorithm has minimum MSE in ANFIS training in order to classify motor imageries. It is clear that the CPSO-trained ANFIS predicts the target output very well, and in the next rank, the PSO algorithm obtains a precise result.

For a comprehensive comparison, in addition to accuracy, we also compare the convergence of algorithms. In this regard, we have recorded a repetition in which convergence occurs for each algorithm, which is shown in Table 4 for each subject. As shown in this table, on average, the DE algorithm is best in terms of the convergence rate and the CPSO algorithm is in the next rank. But the problem with the DE algorithm is that it does not approach the optimal response and encounters local traps and converges in these local traps. In other words, early convergence is occurred for this algorithm. But the proposed CPSO algorithm, as above results confirm, is closer to the global optimum and has an acceptable speed than the rest of the algorithms under discussion; therefore, by considering the compromise between speed and the accuracy of this algorithm, it is superior to the rest of discussed algorithms.

- Comparison with other works

For a good comparison, the proposed technique performance is compared with some benchmark methods that test this dataset in similar conditions. Table 5 shows the motor imagery classification accuracy obtained per subject. As seen, the ANFIS-CPSO approach reaches the best accuracy in comparison with the method given in (Higashi and Tanaka 2013) that includes common spatio-time-frequency patterns to design the time windows for the motor imagery task. The motor imagery classification procedure described in (He et al. 2012) also involves the EMD-based CSP preprocessing. The proposed adaptive frequency band selection together with the developed method of feature extraction is insufficient, causing in a low classification performance with a high standard deviation value. An approach given in (Zhang et al. 2012) is based on a robust learning method that extracts spatio-spectral features for discriminating multiple EEG tasks. The achieved motor imagery classification has the lowest performance among the comparative approaches. Another technique given in (Álvarez-Meza et al. 2015) is based on feature relevance analysis within the motor imagery classification framework. This method reached 92.86% classification accuracy that is a good performance, but the proposed method in this paper, as can be seen in Fig. 12 and Table 5, the ANFIS-CPSO, has the highest performance compared to the other techniques in terms of

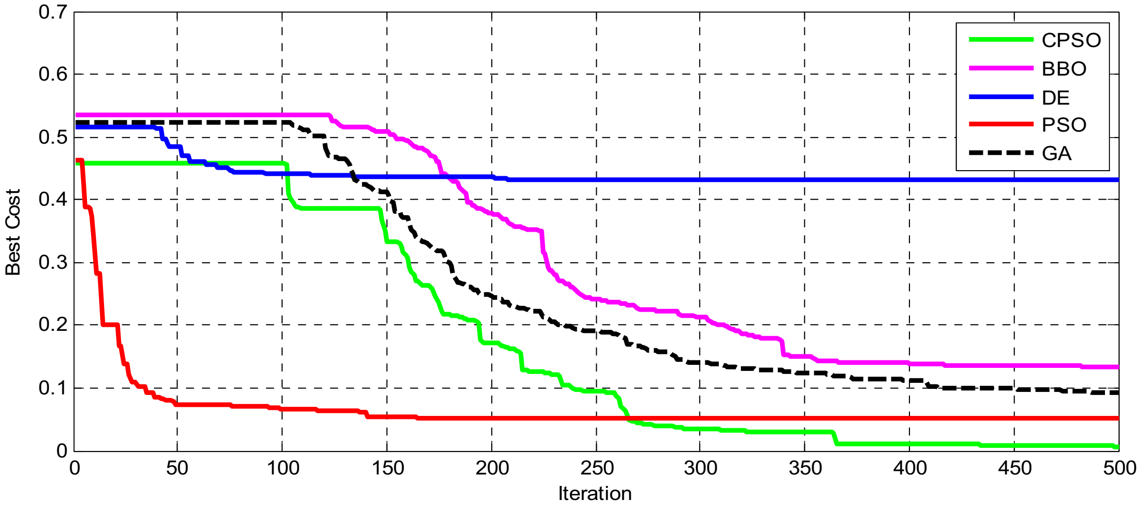


Fig. 10 The convergence curves for DS1c

Table 3 MSE and RMSE values for all subjects

Algorithm	ds1a		ds1b		ds1c		ds1d		ds1e		ds1f		ds1g	
	MSE	RMSE	MSE	RMSE	MSE	RMSE	MSE	RMSE	MSE	RMSE	MSE	RMSE	MSE	RMSE
ANFIS-GA	0.28628	0.53505	0.25943	0.50935	0.092665	0.3044	0.26795	0.51764	0.14351	0.37883	0.12478	0.3532	0.16213	0.4027
ANFIS-DE	0.3899	0.6244	0.38491	0.6204	0.4313	0.6567	0.30221	0.5497	0.26991	0.5195	0.4800	0.6204	0.30012	0.5478
ANFIS-PSO	0.13225	0.36367	0.10454	0.32332	0.0410	0.2025	0.13003	0.11403	0.03896	0.1974	0.03204	0.1790	0.05496	0.2344
ANFIS-BBO	0.22642	0.47580	0.37442	0.6119	0.13375	0.3657	0.33245	0.5766	0.18845	0.4341	0.25786	0.5078	0.21218	0.4606
ANFIS-CPSO	**0.082343**	**0.28695**	**0.07142**	**0.2672**	**0.00280**	**0.05293**	**0.09689**	**0.31127**	**0.01576**	**0.12555**	**0.00848**	**0.0921**	**0.01921**	**0.1386**

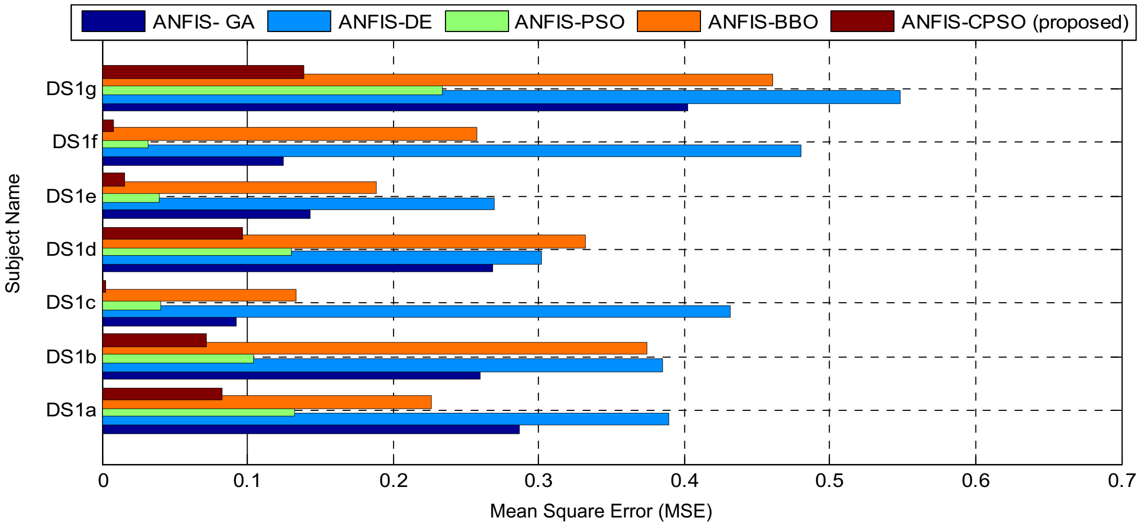


Fig. 11 The bar plot representation of MSE values for all subjects

classification accuracy. So, the superiority of the proposed method over another benchmark method in terms of classification accuracy is clear for all subjects.

Moreover, the classification accuracy of ANFIS-CPSO is compared with some popular machine learning classifiers such as support vector machine (SVM) (Cortes et al.

Table 4 The iterations that convergence of algorithms is occurred

Algorithms Subjects	DE	GA	PSO	CPSO	BBO
DS1a	**150**	500	420	270	300
DS1b	260	470	425	**230**	275
DS1c	**100**	450	170	370	380
DS1d	**190**	480	380	200	320
DS1e	**175**	400	370	210	350
DS1f	**130**	390	415	190	250
DS1g	**180**	420	400	220	300
Mean	**~169**	~444	~369	~241	~310

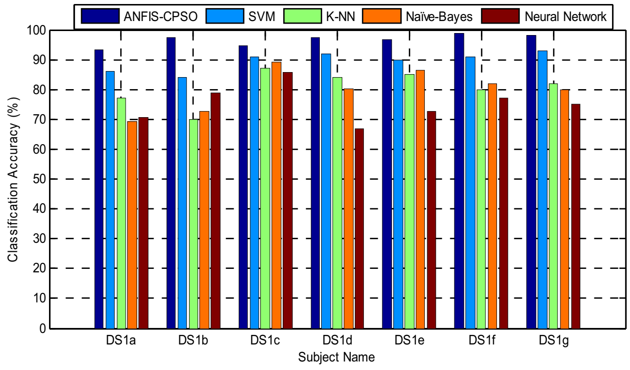


Fig. 13 Comparing the classification accuracy of the proposed algorithm to some popular machine learning algorithms for all datasets

Table 5 Comparison of the proposed method with other works in term of classification accuracy (average accuracy ± standard deviation)

Subject	Higashi and Tanaka (2013)	He et al. (2012)	Zhang et al. (2012)	Álvarez-Meza et al. (2015)	ANFIS-CPSO (proposed)
DS1a	92.30 ± 02.50	67.70 ± 02.20	77.20 ± 00.03	91.50 ± 01.20	93.3929 ± 01.1084
DS1b	90.60 ± 07.20	70.70 ± 01.20	70.80 ± 0.02	96.50 ± 03.37	97.3571 ± 0.9421
DS1c	–	83.90 ± 01.30	–	91.50 ± 04.74	94.6451 ± 1.0710
DS1d	–	93.00 ± 01.20	–	87.00 ± 06.32	97.5000 ± 01.7021
DS1e	–	93.20 ± 01.20	–	91.50 ± 07.47	96.8571 ± 01.0178
DS1f	93.30 ± 03.60	–	76.80 ± 0.03	98.50 ± 02.42	98.7512 ± 0.9987
DS1g	94.10 ± 04.10	–	80.00 ± 0.03	93.50 ± 07.09	98.0000 ± 1.1270
Mean ± STD	92.58 ± 01.51	81.70 ± 12.06	76.20 ± 03.87	92.86 ± 03.77	**96.6433 ± 1.9205**

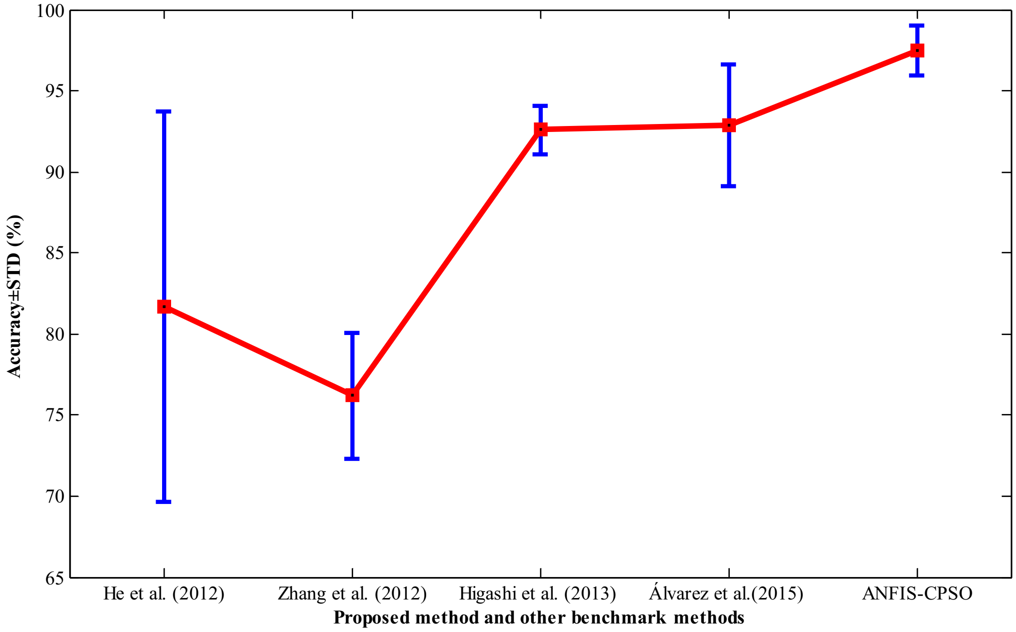


Fig. 12 Comparison of the proposed method with other works in terms of classification accuracy and standard deviation (Mean ± STD). The vertical lines show the variation of accuracy for each method

1995), k-nearest neighborhood (KNN) (Altman 1992), Naïve Bayes (Zhang et al. 2009) and neural networks (Hansen and Salamon 1990) in terms of classification rate. Figure 13 shows this comparison, and it can be seen that the classification accuracy of the proposed algorithm, ANFIS-CPSO, is much better than others.

7 Conclusion

This paper has proposed the CPSO to train ANFIS for two class motor imagery classification. An extensive study was conducted on 12 mathematical benchmark functions to analyze exploration, exploitation, local optima avoidance, and convergence behavior of the proposed algorithm. CPSO was found to be competitive enough with other state-of-the-art meta-heuristic methods. The CSP method was used to extract the features of an EEG signal. Using CSP, the data dimension has been reduced from 59 to 10. Then, these features are classified using an ANFIS that its parameters are trained by CPSO algorithm. The classification accuracy of this classifier was compared with ANFIS classifiers trained by other meta-heuristic algorithms such as PSO, GA, DE, and BBO. The criteria of MSE and RMSE were compared for different algorithms. These criteria were reported for various algorithms and seven relevant data. The graph containing the histogram indicates the concentration of the calculated error, and the closer the histogram around zero is, the better the efficiency. Also, the proposed method compared with other benchmark methods on this EEG dataset in terms of classification accuracy, the results showed that the classification accuracy of ANFIS trained by CPSO is better than the other discussed algorithms and has the acceptable convergence.

References

Afrakhteh S, Mosavi MR, Khishe M, Ayatollahi A (2018) Accurate classification of EEG signals using neural networks trained by hybrid population-physic-based algorithm. Int J Autom Comput. https://doi.org/10.1007/s11633-018-1158-3

Altman NS (1992) An introduction to the kernel and nearest neighbor nonparametric regression. Am Stat 46(3):175–185. https://doi.org/10.1080/00031305.1992.10475879

Álvarez-Meza AM, Velásquez-Martínez LF, Castellanos-Dominguez G (2015) Time-series discrimination using feature relevance analysis in motor imagery classification. Neurocomputing 151:122–129. https://doi.org/10.1016/j.neucom.2014.07.077

Angelov P (2014) Outside the box: an alternative data analytics framework. J Autom Mob Robot Intell Syst 8(2):29–35. https://doi.org/10.14313/JAMRIS_2-2014/16

Angelov P, Buswell R (2001) Evolving rule-based models: a tool for intelligent adaptation. In proc. 9th IFSA World Congress, Vancouver, BC, Canada, pp 1062–1067. https://dspace.lboro.ac.uk/2134/10190

Angelov P, Kasabov N (2005) Evolving computational intelligence systems. Proceedings of the international workshop on genetic fuzzy systems, pp 76–82 http://eprints.lancs.ac.uk/948/

Angelov P, Plamen P (2013) Evolving rule-based models: a tool for design of flexible adaptive systems. Springer, Berlin. https://www.springer.com/us/book/9783790814576

Angelov P, Zhou X (2008) Evolving fuzzy-rule-based classifiers from data streams. IEEE Trans Fuzzy Syst 16(6):1462–1475. https://doi.org/10.1109/TFUZZ.2008.925904

Angelov P, Xydeas C, Filev D (2004a) On-line identification of MIMO evolving Takagi-Sugeno Fuzzy Models. IEEE Joint Conference on Neural Networks and Fuzzy Systems http://eprints.lancs.ac.uk/951/

Angelov P, Victor J, Dourado A, Filev D (2004b) On-line evolution of Takagi-Sugeno fuzzy models. IFAC Proc 37(16):67–72 https://doi.org/10.1016/S1474-6670(17)30852-2

Angelov P, Sadeghi-Tehran P, Ramezani R (2011) An approach to automatic real-time novelty detection, object identification, and tracking in video streams based on recursive density estimation and evolving Takagi-Sugeno fuzzy systems. Int J Intell Syst 26(3):189–205. https://doi.org/10.1002/int.20462

Arvaneh M, Guan C, Ang KK, Quek C (2011) Optimizing the channel selection and classification accuracy in EEG-based BCI. IEEE Trans Biomed Eng 58(6):1865–1873. https://doi.org/10.1109/TBME.2011.2131142

Baruah RD, Angelov P (2012) Evolving local means method for clustering of streaming data. IEEE Conf Fuzzy Syst. https://doi.org/10.1109/FUZZ-IEEE.2012.6251366

Baruah RD, Angelov P (2014) DEC: Dynamically evolving clustering and its application to structure identification of evolving fuzzy models. IEEE Trans Cybern 44(9):1619–1631. https://doi.org/10.1109/TCYB.2013.2291234

Blankertz B, Dornhege G, Krauledat M, Müller KR, Curio G (2007) The non-invasive Berlin brain–computer interface: fast acquisition of effective performance in untrained subjects. Neuroimage 37(2):539–550

Cheng CT, Lin JY, Sun YG, Chau K (2005) Long-term prediction of discharges in manwan hydropower using adaptive-network-based fuzzy inference systems models. Int Conf Adv Nat Comput pp. 1152–1161 https://doi.org/10.1007/11539902_145

Cortes C, Vapnik V (1995) Support-vector networks. Mach Learn 20(3):273–297

Eberhart R, Kennedy J (1995) A new optimizer using particle swarm theory. Proceedings of the Sixth International Symposium on Micro Machine and Human Science, pp. 39–43 https://doi.org/10.1109/MHS.1995.494215

Hansen LK, Salamon P (1990) Neural network ensembles. IEEE Trans Pattern Anal Mach Intell 12(10):993–1001. https://doi.org/10.1109/34.58871

He W, Wei P, Wang L, Zou Y (2012) A novel EMD-based common spatial pattern for motor imagery brain–computer interface. IEEE-EMBS Int Conf Biomed Health Inform. https://doi.org/10.1109/BHI.2012.6211549

Higashi H, Tanaka T (2013) Common spatio–time–frequency patterns for motor imagery-based brain–machine interfaces. Comput Intell Neurosci 2013:1–13. https://doi.org/10.1155/2013/537218

Holland JH (1992) Genetic algorithms. Sci Am 267(1):66–72. https://www.jstor.org/stable/24939139?seq=1#page_scan_tab_contents

Jang JSR (1993) ANFIS: adaptive-network-based fuzzy inference system. IEEE Trans Syst Man Cybern 23(3):665–685. https://doi.org/10.1109/21.256541

Jang JSR, Mizutani E (1996) Levenberg-Marquardt method for ANFIS learning. Fuzzy Inf Process Soc. https://doi.org/10.1109/NAFIPS.1996.534709

Jasper H, Penfield W (1949) Electrocorticograms in man: effect of voluntary movement upon the electrical activity of the precentral gyrus. Eur Arch Psychiatry Clin Neurosci 183(1):163–174. https://doi.org/10.1007/BF01062488

Kasabov NK, Song Q (2002) DENFIS: dynamic evolving neural-fuzzy inference system and its application for time-series prediction. IEEE Trans Fuzzy Syst 10(2):144–154. https://doi.org/10.1109/91.995117

Lemm S, Schafer C, Curio G (2004) BCI competition 2003-data set III: probabilistic modeling of sensorimotor/spl mu/rhythms for classification of imaginary hand movements. IEEE Trans Biomed Eng 51(6):1077–1080. https://doi.org/10.1109/TBME.2004.827076

Lotte F, Guan C (2010) Regularizing common spatial patterns to improve BCI designs: theory and algorithms regularizing common spatial patterns to improve BCI designs: theory and algorithms. IEEE Trans Biomed Eng 58(2):355–362. https://doi.org/10.1109/TBME.2010.2082539

Lughofer E (2013) On-line assurance of interpretability criteria in evolving fuzzy systems-achievements, new concepts and open issues. Inf Sci 251:22–46. https://doi.org/10.1016/j.ins.2013.07.002

Ma Y, Ding X, She Q, Luo Z, Potter T, Zhang Y (2016) Classification of motor imagery EEG signals with support vector machines and particle swarm optimization. Comput Math Methods Med 2016:1–8. https://doi.org/10.1155/2016/4941235

Mamdani EH, Assilian S (1975) An Experiment in linguistic synthesis with a fuzzy logic controller. Int J Man Mach Stud 7(1):1–13. https://doi.org/10.1016/S0020-7373(75)80002-2

Mascioli FM, Varazi GM, Martinelli G (1997) Constructive algorithm for neuro-fuzzy networks. Proceedings of the Sixth IEEE International Conference on Fuzzy Systems 1:459–464. https://doi.org/10.1109/FUZZY.1997.616411

Moore MM (2003) Real-world applications for brain–computer interface technology. IEEE Trans Neural Syst Rehabil Eng 11(2):162–165. https://doi.org/10.1109/TNSRE.2003.814433

Pfurtscheller G, Neuper C, Schlogl A, Lugger K (1998) Separability of EEG signals recorded during right and left motor imagery using adaptive autoregressive parameters. IEEE Trans Rehabil Eng 6(3):316–325. https://doi.org/10.1109/86.712230

Pratama M, Pedrycz W, Lughofer E (2018) Evolving ensemble fuzzy classifier. IEEE Trans Fuzzy Syst 26(5):2552–2567. https://doi.org/10.1109/TFUZZ.2018.2796099

Precup RE, Teban TA, Albu A, Szedlak-Stinean AI, Bojan-Dragos CA (2018) Experiments in incremental online identification of fuzzy models of finger dynamics. Sci Technol 21(4):358–376. http://www.romjist.ro/abstract-607.html

Ramoser H, Muller-Gerking J, Pfurtscheller G (2000a) Optimal spatial filtering of single trial EEG during imagined hand movement. IEEE Trans Rehabil Eng 8(4):441–446. https://doi.org/10.1109/86.895946

Ramoser H, Muller-Gerking J, Pfurtscheller G (2000b) Optimal spatial filtering of single-trial EEG during imagined hand movement. IEEE Trans Rehabil Eng 8(4):441–446. https://doi.org/10.1109/86.895947

Schalk G, McFarland DJ, Hinterberger T, Birbaumer N, Wolpaw JR (2004) BCI2000: a general purpose brain–computer interface (BCI) system. IEEE Trans Biomed Eng 51(6):1034–1043. https://doi.org/10.1109/TBME.2004.827072

Simon D (2008) Biogeography-based optimization. IEEE Trans Evol Comput 12(6):702–713. https://doi.org/10.1109/TEVC.2008.919004

Storn R, Price K (1997) Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces. J Global Optim 11(4):341–359

Subasi A, Ercelebi E (2005) Classification of EEG signals using neural network and logistic regression. Comput Methods Progr Biomed 78(2):87–99. https://doi.org/10.1016/j.cmpb.2004.10.009

Wessel M (2006) Pioneering research into brain–computer interfaces. Thesis in Delft University of Technology http://www.kbs.twi.tudelft.nl/docs/MSc/2006/Wessel_Mark/thesis.pdf

Wolpaw JR, Birbaumer N, McFarland DJ, Pfurtscheller G, Vaughan TM (2002) Brain–computer interfaces for communication and control. Clin Neurophysiol 113(6):767–791. https://doi.org/10.1016/S1388-2457(02)00057-3

Zadeh LA (1973) Outline of a new approach to the analysis of complex systems and decision processes. IEEE Trans Syst Man Cybern 1:28–44. https://doi.org/10.1109/TSMC.1973.5408575

Zhang M-L, Peña JM, Robles V (2009) Feature selection for multi-label naive bayes classification. Inf Sci 179:3218–3229. https://doi.org/10.1016/j.ins.2009.06.010

Zhang H, Guan C, Ang KK, Wang C, Chin ZY (2012) BCI competition IV—data set I: learning discriminative patterns for self-paced EEG-based motor imagery detection. Front Neurosci 6:1–7. https://doi.org/10.3389/fnins.2012.00007

Zhou SM, Gan JQ, Sepulveda F (2008) Classifying mental tasks based on features of higher-order statistics from EEG signals in brain–computer interface. Inf Sci 178:1629–1640. https://doi.org/10.1016/j.ins.2007.11.012