

A fuzzy-based classification strategy (FBCS) based on brain–computer interface

Ahmed I. Saleh¹ · Sahar A. Shehata¹ · Labeeb M. Labeeb¹

Published online: 1 December 2017
© Springer-Verlag GmbH Germany, part of Springer Nature 2017

Abstract Brain–computer interface systems help paralyzed people to control devices such as a computer cursor, robotic limbs, wheelchairs, or spellers by only using their thoughts. Nowadays, electroencephalogram (EEG) signals are mostly used to detect activity of various actions within the brain as they provide rich information about brain’s electrical activity. However, EEG signal generates large amount of data which is usually difficult to interpret and classify. This paper introduces a new classification strategy based on EEG signals, which is called fuzzy-based classification strategy (FBCS). FBCS minimizes the classification time by perfectly extracting the effective features of the produced EEG signals based on a set of elected electrodes using semantic analysis, then taking the classification decision accordingly. FBCS uses feature reduction and electrode selection techniques to reduce the dimensionality of data to be classified, which also improves the classification accuracy. Experimental results have shown that FBCS outperforms recent classification strategies in terms of accuracy and classification time.

Keywords Brain–computer interface · Electroencephalogram · Feature reduction · Electrode selection · Classification · Fuzzy logic

Communicated by V. Loia.

✉ Ahmed I. Saleh
aisaleh@yahoo.com

¹ Department of Computer Engineering and Systems, Faculty of Engineering, Mansoura University, Mansoura, Egypt

1 Introduction

Brain–computer interface (BCI) researches have grown over the past few decades since the 1970s. One goal of BCI research is to develop systems capable of classifying neural representations of natural movement planning and execution. Researchers aim to develop systems that help disable persons to communicate with external world and provide them a non-muscular pathway between their brains and prosthetic devices such as robotic limbs, wheelchairs, or spellers. Such applications are not only limited the use of prosthetic devices but also extended virtual gaming, tele-operation, communication, and robotics (McFarland and Wolpaw 2008; Daly and Wolpaw 2008).

However, BCI suffers from several challenges such as: (i) the training sets are sometimes relatively small, since the training process is affected by usability matter. Even though heavily training sessions are considered time-consuming and demanding for the subjects, in the training phase, trained subject’s signal has been used to learn the used classifier. Therefore, a significant challenge in designing a BCI is to balance the trade-off between the technological complexity of classifying the user’s brain signals and the amount of training needed for successful operation of the interface. (ii) Nonlinearity; the brain is a highly complex nonlinear system in which disordered behavior of neural ensembles can be detected. Thus, EEG signals can be better characterized by nonlinear dynamic methods than linear methods. (iii) Non-stationarity and noise; the used signals continuously changed over time either between or within the recording sessions. The mental, emotional state background through different sessions, fatigue, and concentration levels can contribute in EEG signals variability. Noise is also a big contributor in the BCI non-stationarity issue as it includes unwanted signals caused by alterations in electrode placement and environ-

mental noise. (iv) High dimensionality; signals are recorded from multiple channels to preserve high spatial accuracy. As the amount of data needed to properly describe different signals increases exponentially with the dimensionality of the vectors, various feature extraction methods have been proposed. They play an important role in identifying distinguishing characteristics. Thus, the classifier performance will be affected only by the small number of distinctive features instead of the whole recorded signals that may contain redundancy (Abdulkader et al. 2015).

The basic steps of BCI include acquisition of brain signals, preprocessing, feature extraction, and classification. The decisions generated by the employed classifier can be used to control an external device. The electroencephalogram (EEG) is a popular signal acquisition noninvasive technique that allows BCI systems to measure electrical potentials of the brain at a temporal resolution on the order of milliseconds through electrodes placed on the surface of the scalp. Typically EEG caps with 6 to 64 electrodes are the mostly used (some cases use a much greater number of electrodes, e.g., 256), so the dimension of the feature space is often very large having redundant features which not only creates additional overhead of managing the space complexity but also might include outliers, thereby reducing classification accuracy (Rakotomamonjy et al. 2005).

Feature selection is a subarea of dimensionality reduction aims to identify the best subset of features out of original feature space. In BCI applications, principal component analysis (PCA) (Yu et al. 2014), independent component analysis (ICA) (Guo et al. 2013), sequential forward search (SFS) (Pal et al. 2014), and particle swarm optimization (PSO) (Hsu 2013) have been used for feature selection to reduce the dimensionality of data. After feature extraction and reduction, classification algorithms are used having two functions in training and practical applications of BCI. During training, the task is to infer a mapping between signals and classes using the labeled feature vector produced by the feature extraction module. During the application of BCI, the task is to discriminate different types of neurophysiologic signals translating them into commands therefore to allow for control of a BCI.

Recently, a number of widely used classifiers such as linear discriminant analysis (LDA), K nearest neighbor (KNN) algorithms, support vector machine (SVM), decision trees, Naive Bayes (NB) classifier, and neural networks (NN) (Lotte et al. 2007) have been used as BCI classifiers. Linearity is the main limitation of LDA, which can cause poor outcomes (Lotte et al. 2007). SVM has a low-speed execution but has good generalization properties (Lotte et al. 2007). On the other hand, KNN assigns an unseen data sample to the dominant class among its K nearest neighbors formed using the training set. KNN may fail in some BCI experiments due to its sensitivity to the curse of dimensionality. However, it

performs efficiently with low-dimensional feature sets (Lotte et al. 2007). NB classifier is based on Bayes' theorem, with a strong assumption of independence of the features, and it is more suitable in BCI applications with small number of trials.

This paper introduces a new fuzzy-based classification strategy (FBCS) based on brain–computer interface. FBCS includes novel techniques for feature reduction and electrode selection to reduce the dimensionality of data. Accordingly, both training time and response time have been minimized that makes FBCS suitable for real-time applications which need a quick response. On the other hand, FBCS is based on a fuzzy inference system which is employed for the classification task. To accomplish such task, a new instance of KNN is introduced, which is called fuzzified KNN (FKNN). In addition to the high classification accuracy, FKNN has a salient property that traditional KNN does not have, which is overfitting resistant. The cause is that FKNN adds several classification heuristics besides the K nearest neighbors, which are the distance among items in the feature space as well as the belonging degree of the item to the class. Those heuristics have been merged via a fuzzy inference system. Accordingly, FKNN provides accurate classification decisions. FKNN has been compared against recent classification techniques that had been applied to BCI. Experimental results have depicted that FKNN outperforms recent techniques as it gives not only the maximum classification accuracy and sensitivity but also the minimum response time. This paper is organized as follows: In Sect. 2, an overview about the EEG-based BCI systems as well as their main parts have been introduced. Section 3 shows the previous efforts about the dimensional reduction and BCI classification techniques. Section 4 introduces the proposed fuzzy-based classification strategy (FBCS). Section 5 discusses the experimental results. And finally, conclusion of our work is presented in Sect. 6.

2 General scheme of EEG-based BCI

Figure 1 illustrates the basic principle of EEG-based BCI. Initially, signals from the brain are acquired. Generally, there are three methods to acquire (capture) signals that represent the human brain electrical activities, which are (i) invasive, (ii) partially invasive, and (iii) noninvasive. Invasive capture provides high-quality signal reading, but causes great inconvenience and risks to human health. Partially invasive capture provides lower-quality signals and lower risk to health. On the other hand, noninvasive capture is fully external to the body, more convenient, and easy to use and provides good quality signal capture and not present risk to users.

Although there are many methods to detect brain signals, EEG acquisition system has relatively short time constants,

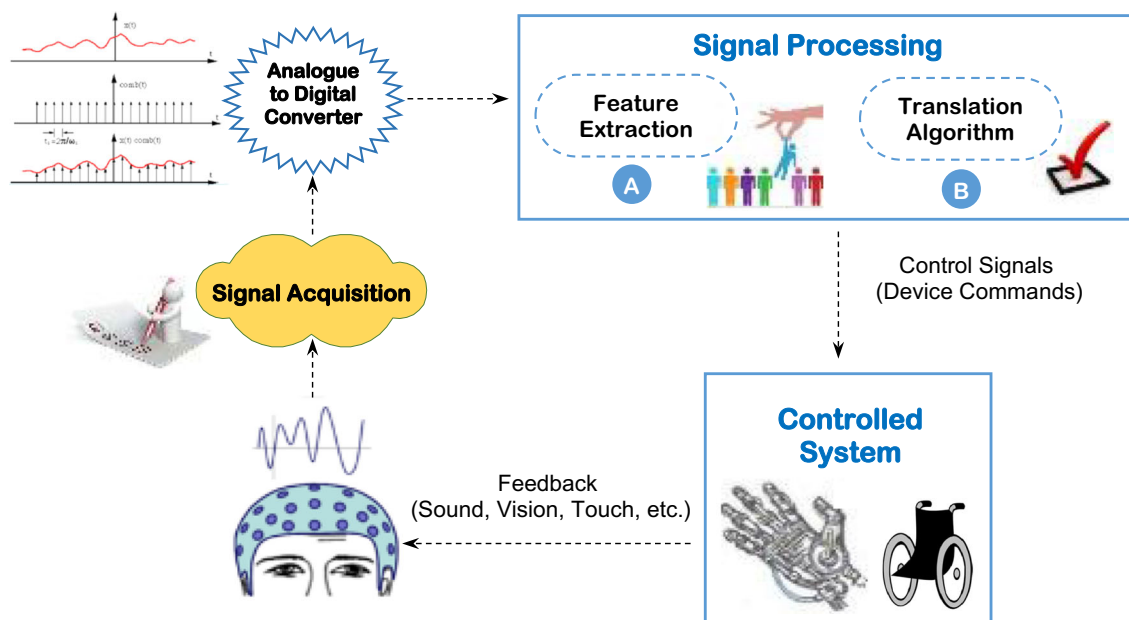


Fig. 1 General scheme of EEG-based BCI

can function in most environments, and require relatively simple and inexpensive equipment, offering the possibility of a new non-muscular communication and control channel. EEG signal is acquired with the help of a multi-channel headset having a certain sampling rate.

EEG (electroencephalogram) is the most popular noninvasive brain signal acquisition tool; thus, it is the cheapest and the simplest recording technique. However, it has low signal-to-noise ratio (SNR) due to the environmental noise and artifacts caused by muscle and eye movements. The EEG system contains electrodes, amplifiers, A/D converter, and a recording device, which may be a personal computer or similar. The electrodes acquire the signal from the scalp; the amplifiers process the analog signal to enlarge the amplitude of the EEG signals (signals on scalp are very small micro-volt range (1/1,000,000 V)), so that the A/D converter can digitalize the signal accurately. Then, the recording device stores and displays the data. The produced signal which is digitized and analyzed can be used to extract commands that can control a computer or a device.

Applications include spelling, computer mouse control, and prosthesis or robot control. Generally, BCI can be applied in several applications; mainly, it allows paralyzed people to control prosthetic limbs with their mind; visual images can be transmitted to the mind of a blind person, allowing them to see; auditory data can be transmitted to the mind of a deaf person, allowing them to hear. From another point of view, BCI allows gamers to control video games with their minds; it can also allow a mute person to have their thoughts displayed and spoken by a computer. Finally, a feedback is provided to the user for further interaction. An improvement in just

one of these steps can improve the performance of a BCI system.

3 Related work

The main target of this paper is to introduce a new classification strategy to enhance the classification performance of BCI systems using the concept of dimensional reduction. Hence, some of the recent efforts in employing some dimensional reduction and classification techniques in BCI applications will be represented in this section.

Principle component analysis (PCA) is a widely used dimensionality reduction linear transformation technique. However, the projections it finds are to maximize variances, which are not necessarily related to classification performance (the class labels); it is not particularly useful in classification and pattern recognition applications. Linear discriminant analysis (LDA) attempts to overcome this limitation of PCA by finding linear projections that maximize class separability under the Gaussian distribution assumption (Fukunaga 1990). The LDA projections are optimized based on the means and the covariance matrices of classes, which are not descriptive of an arbitrary probability density function (pdf). Independent component analysis (ICA) has also been used as a tool to find linear transformations that maximize the statistical independence of random variables. However, it has similar drawbacks as PCA. CSP (common spatial patterns) can be used instead of PCA and ICA (Naem et al. 2009).

Atyabi et al. (2012) introduced electrode reduction (ER) and feature reduction (FR) methods based on genetic

algorithms (GA) and particle swarm optimization (PSO). Evolution-based methods are used to generate a set of indexes presenting either electrode seats or feature points that maximize the output of a weak classifier representing a comparison between genetic algorithms (GA), particle swarm optimization (PSO), and random search algorithm as electrode and feature reduction methods. The results indicate that on average across all subjects, in GA-based ER, GA-based FR, random-based ER, random-based FR, and PSO-based FR, the electrode reduction (ER) had a greater impact on classification performance compared to feature reduction (FR), and the combination of polynomial SVM- and GA-based ER performed better than all other methods except the combination of the use of the full-set with polynomial SVM.

Sparse common spatial pattern (SCSP) algorithm was proposed in Arvaneh (2011), to select the least number of channels within a constraint of classification accuracy. To select channels using the SCSP method, first two sparse common spatial filters corresponding to two motor imagery tasks are obtained. After obtaining the sparse filters, channels corresponding to the zero elements in both of the spatial filters are discarded, and the rest are defined as the selected channels. To compare and consider the importance of each selected channel, a ranking method was proposed as follows: first, the top ranked channels for each motor imagery task are determined from the maximum of the absolute value of the corresponding sparse spatial filter. SCSP algorithm yielded an average improvement in 10% in classification accuracy compared to the use of three channels.

Multi-objective particle swarm optimization (MOPSO) method proposed in Hasan et al. (2009) solves the problem of effective channel selection for brain–computer interface (BCI) systems. The proposed method was tested and compared to another search-based method, sequential floating forward search (SFFS). The results demonstrate the effectiveness of MOPSO in selecting a fewer number of channels with insignificant sacrifice in accuracy, which is very important to build robust online BCI systems.

Muhammad et al. (2015) presented a comparison of mostly used classification algorithm with a new unsupervised learning technique for classification, i.e., self-organizing maps (SOM) based on neural network. SOM and other algorithms have been used to categorize the feature vector acquired from the EEG dataset into their corresponding classes. Both original and reduced feature sets have been used for classification of motor imagery-based EEG signals. The reduction is performed by applying principal component analysis (PCA). It has been depicted from measured data that SOM shows a maximum classification accuracy of 84.17% on PCA implemented reduce feature set.

Nanayakkara and Sakkaff (2012) presented a new classification method, which is closely related to K nearest neighbor

(KNN) classification method, named fixed distance neighbor (FDN) classifier. For comparison purposes, performance of KNN classification method and performance of the FDN method are tested with the same feature vectors derived from EEG datasets recorded for imagery motor movement mental tasks. It was found that FDN performed slightly better than KNN for most of the datasets used in this study, indicating that FDN is a viable classification method, which can be used in place of KNN in BCI systems.

Authors in [17] used a combination of bacterial foraging optimization and learning automata to determine the best subset of features from a given motor imagery electroencephalography (EEG)-based BCI dataset. They employed discrete wavelet transform to obtain a high-dimensional feature set and classified it by distance likelihood ratio test. This proposed feature selector produced an accuracy of 80.291% in 216 s. On the other hand, Zanchettin et al. (2012) presented a hybrid KNN-SVM method for cursive character recognition. The main idea was to increase the K nearest neighbor recognition rate, sensible to different classes with similar attributes, using the SVM as a decision classifier. The adaptation was to get the two most frequently classes in the KNN and use the SVM to decide between these two classes. The main disadvantage is the processing time.

The advantages of self-organizing maps (SOM) artificial networks and KNN were explored in Silva and Del-Moral-Hernandez (2011), so the KNN performs the classification process and the SOM work as preprocessing to the KNN classifier, applied to digits recognition in car plates. The main advantage of this method is that the time consumed by SOM-KNN is shorter than time consumed by KNN. Finally, a review of BCI several techniques for signal acquisition, preprocessing or signal enhancement, feature extraction, classification, and the control interface was discussed in Nicolas-Alonso and Gomez-Gil (2012), representing their advantages, drawbacks, and latest advances.

4 The proposed fuzzy-based classification strategy (FBCS)

This section illustrates the proposed fuzzy-based classification strategy (FBCS) in details. The different steps of FBCS are depicted in Fig. 2. As illustrated in Fig. 2, the proposed FBCS consists of seven sequential steps, namely (i) data acquisition, (ii) preprocessing, (iii) feature extraction, (iv) feature selection, (v) dimensionality reduction, (vi) classification, and (vii) decision making for choosing a certain action. However, FBCS mainly focuses on: (iv) feature selection to acquire a set of compact and informative features, (v) dimensionality reduction to minimize processing time, and (vi) classification to take the corresponding precise decisions. We claim that giving more attention to those steps will not

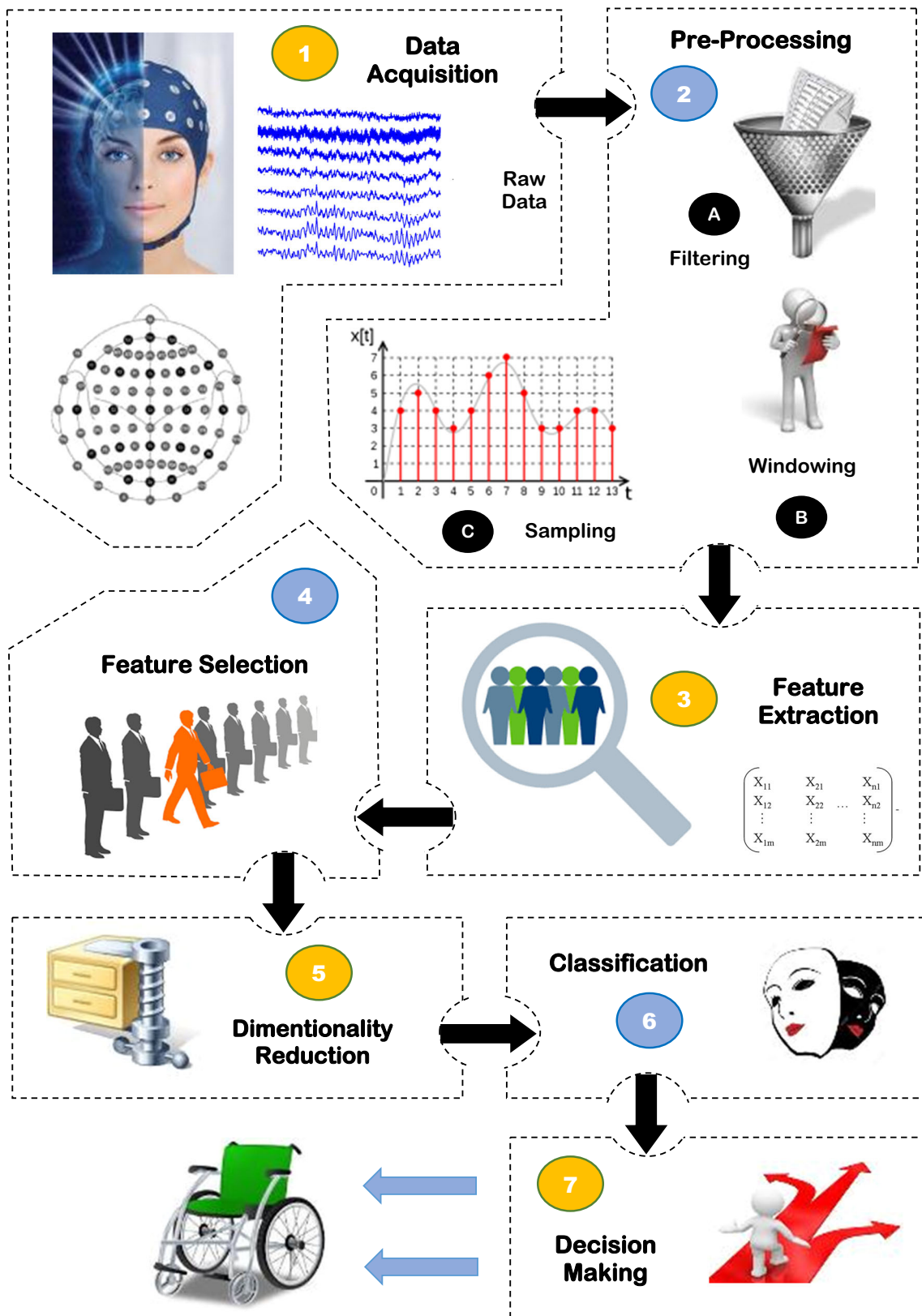


Fig. 2 Block diagram of feature reduction and EEG electrode selection BCI model

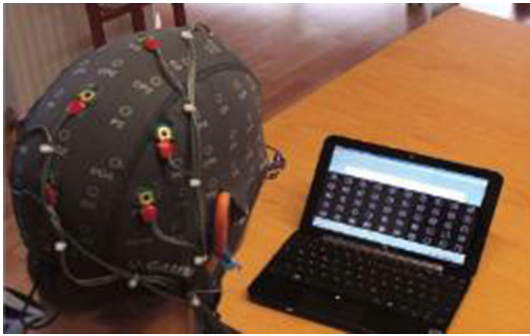


Fig. 3 An EEG cap filled with electrodes used in BCI data acquisition

only improve the performance of the BCI system, but also greatly reduce the computational load of the system. The next subsections explain those steps in more details.

4.1 Data acquisition

Data acquisition can be accomplished through an EEG cap. Figure 3 shows an example of EEG caps, representing a general view of EEG cap filled with electrodes. These electrodes are set up according to the standard 10/20 system of electrode placement method. EEG cap with 22 electrodes with 250 Hz sampling rate from Dataset 2a of BCI competition IV provided by BCI research group at Graz University (Brunner et al. 2008) and EEG cap with 118 electrodes with 1000 Hz sampling rate from Dataset IVa of BCI competition III (xxx yyy) are used as data acquisition components to evaluate the proposed FBCS.

4.2 Preprocessing

Generally, the purpose of signal preprocessing is to enhance the signal produced by EEG. Unfortunately, EEG-recorded data are highly challenging to evaluate due to the noise recorded together with EEG signal, non-stationary, and diverse artifacts. Artifacts are irrelevant unwanted signals present in BCI system. They have various origins, which include: utility frequencies such as noise, body movements, or eye blinks. As noise amplitude is usually larger than the signal of interest, the goal of preprocessing is to increase signal-to-noise ratio (SNR) for the signal acquired from EEG headset (Mallick and Kapgate 2015). Figure 4 represents the original signal before and after filtration to show the effect of noise.

The digital EEG signal is stored electronically and can be filtered. Filtering can be applied either in the frequency domain by selecting different pass bands, or in the spatial domain. Frequency filtering removes noises, such as filtering out direct current and high frequency noise (1–45 Hz). Frequency filtering also has the ability to select relevant

frequency components, such as sensorimotor rhythm 8–12 Hz (μ). The goal of spatial domain filtering is to create a subset of EEG channels, which are related to certain brain activity, as well as to enhance the separability of the data. The choice of spatial filter can affect the SNR greatly. Bipolar derivation, Laplacian derivation, principal components analysis (PCA), independent components analysis (ICA), and common spatial patterns analysis (CSP) are alternative methods for deriving weights for a linear combination of EEG channels (Jung et al. 2000). In FBCS, EEG signals were band-pass filtered from 8 to 30 Hz including μ (8–13 Hz) and beta (13–30 Hz) rhythms, which are used for classifying motor imagery data.

4.3 Feature extraction

The goal of feature extraction is to represent the characteristics of original signal without unwanted redundancy. The features can be extracted from the EEG signal in two different domains, which are: time domain features (TDF) and frequency domain features (FDF).

Unlike Fourier transform, which provides frequency domain analysis at a constant resolution on the frequency scale, discrete wavelet transform (DWT) provides frequency domain as well as time domain analysis at multiple resolutions. Frequency domain analysis is mainly based on the power and coherence of each frequency band in the EEG signals. Spectral power estimation is the primary means of frequency domain analysis. While time domain analysis method mainly analyzes the geometric property of EEG waveforms, such as amplitude, mean, variance. It is widely used by EEG researchers for its intuition and clear physical meaning (Zhao et al. 2015).

In this paper, DWT is used. Signals are passed through filters with different cutoff frequencies and different scales. The number of filter stages (levels) to be used depends on the resolution required. So the feature vector is prepared using the detail coefficients of third and fourth level (D3 and D4) for each electrode because these levels contain information in the frequency range of 8–12 and 16–24 Hz. Considering a headset of 14 electrodes as an example, the dimension of a feature matrix is $M = 1176$; thus, the number of samples is $S = 84$ samples.

4.4 Feature reduction

Feature reduction methods aim to identify a subset of ‘meaningful’ features out of the original set of features. Feature reduction has several advantages such as (i) overfitting avoidance, this is because the classification model is trained with the most precise and informative features, (ii) performance promotion, and (iii) processing time minimization, which makes the model more suitable for real-time applications

Fig. 4 EEG produced signals

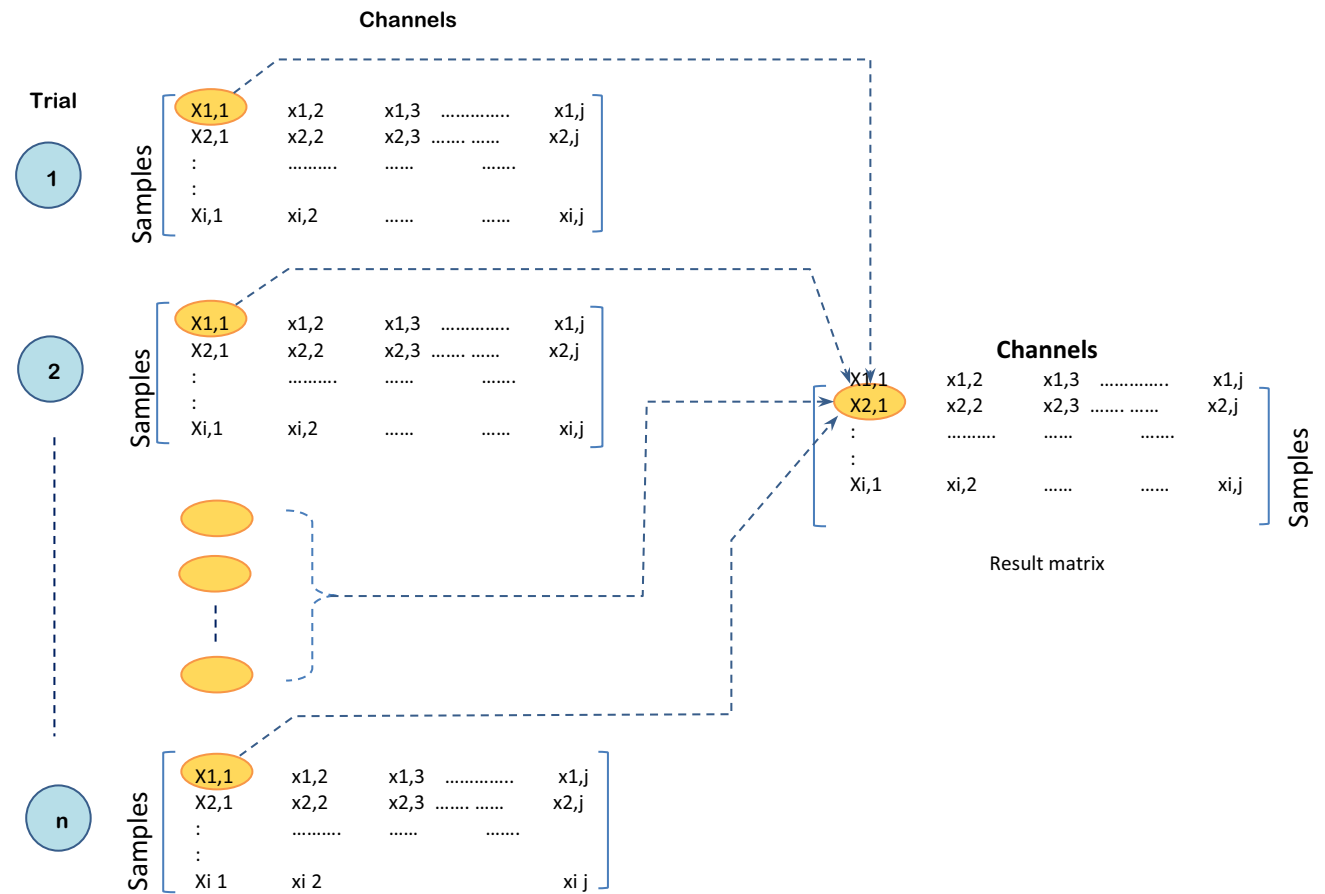
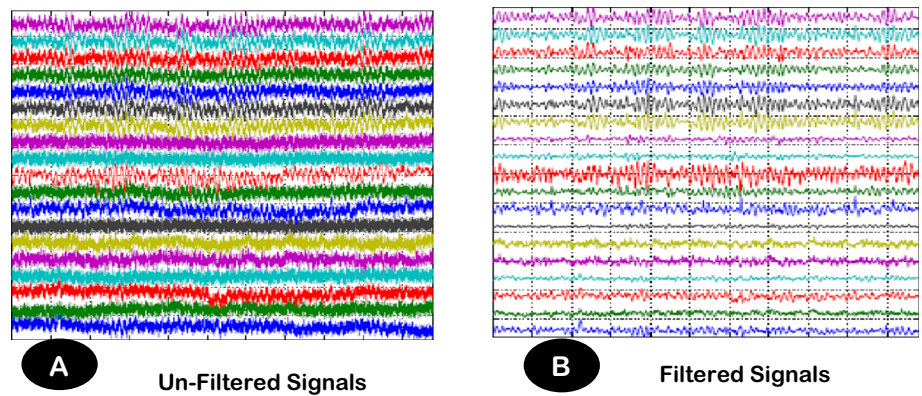


Fig. 5 Proposed feature reduction methodology

(Saleh et al. 2016; Saleh and Abulwafa 2017). Generally, feature reduction methods can be subdivided into filter, wrapper, and embedded methods (Saleh and Abulwafa 2017). Filter methods compute a score for each feature by their information content, and then select only the features that have the best scores. On the other hand, wrapper methods train a predictive model on subsets of features; then, the subset which gives the best accuracy is selected. Finally, embedded methods determine the optimal subset of features directly by the trained weights of the classification method.

In this paper, we propose a new filter approach for feature reduction; the goal is to determine one feature value for the repeated feature values of the same electrode. For number of n trails, there will be n feature matrices, i.e., n value for each element in the feature matrix of dimension $M(E \times S)$, for the above-mentioned example the matrix M of dimension (14×84) is repeated n times for the same action. So, we propose a feature reduction phase to have only one nearly equivalent value for each element.

As shown in Fig. 5, theoretically the value of the feature $x_{i,j}$ remains constant for n trails for the same action. But due

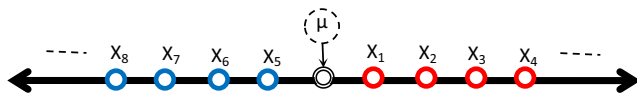


Fig. 6 Projecting the considered values on the numbering axis

to the effect of many factors like the personal feeling, fatigue, happiness, sadness, the value of any element $x_{i,j}$ may differ in each trail. By following the next steps, there will be one value for each element constructing one feature vector for n trails of one action.

For each element in the matrix $(x_{i,j})$, the following steps should be followed:

- *Step 1* represent the n values of the element $(x_{i,j})$ the linear axis shown in Fig. 6, as x_1, x_2, \dots, x_n , which are assumed normally to be nearly identical.
- *Step 2* calculate the average value μ of the n values of the element $x_{i,j}$, as:

$$\mu = \frac{\sum_{i=1}^n x_i}{n} \quad (1)$$

- *Step 3* find the set of values of x in the neighborhood of μ after determining the neighborhood width as:

$$NW = \frac{X_{\max} - X_{\min}}{2} \quad (2)$$

Then select the set

$$S_1 = \{x(i) \forall \mu - NW < x(i) < \mu + NW\} \quad (3)$$

- *Step 4* repeat step 2 and step 3 for a pre-defined (ξ) times till find the approximately value of the element $x_{i,j}$ as the average of the items $\in S_\xi$, or till we reached to one value of element $x_{i,j}$ before the complete of (ξ) times.

The pre-mentioned steps will be followed for all the elements; the data after n trails will be only one matrix of M elements. For illustration, if the system simply has 20 trails using 14 EEG electrodes and each electrode's signal has been sampled to 84 samples, so there will be 20 feature matrices each of dimension 14×84 . Thus, within each matrix each element $x_{i,j}$ will have 20 theoretically identical values which is equal 1176 elements. Table 1 shows an example of $x_{i,j}$ values from the 20th trails. The goal is to have only one feature matrix with a certain value for each element $x_{i,j}$. First, the average of the given 20 values should be determined, which will be equal to 0.537. Then, the range of neighborhood width as depicted in Eq. (2) is calculated, which is equal to 0.395. From Eq. (3), a selected list S_1 from the given feature values, which lies inside the elected range, is picked. Doing the same action for the selected set of values S_1 , a new set of feature

values S_2 , which has a number of items less than that of S_1 , can be selected. Assuming $\xi = 10$, the value of $x_{i,j}$ can be represented by only 0.44 from S_8 after 8 times of calculations. Repeating this procedure for the remaining elements of the feature matrix, the result is one matrix representing the 20 trails after removing the outlier values.

4.5 Electrode selection

Electrode selection mainly focuses on the use of electrodes and scalp locations that best represent the subject's intention and have a high contribution to classification accuracy. Different subjects may have different reactions toward the tasks, and the optimal electrode set for specific tasks may also vary among subjects. In this paper, a new approach for electrode selection will be introduced. Semantic analysis (Ogiela and Ogiela 2012) is used to determine the most informative set of electrodes for better classification accuracy and eliminates the electrodes that may badly affect the average accuracy of BCI several action system.

To reduce the dimensions of the feature matrix $F_{E \times S}$, in which E is the number of electrodes (equal 14 in the above example) and S is the number of samples (equal 84 in the above example), the proposed strategy aims to choose the most informative set of electrodes IS, regarding their effect on the classification accuracy of the underlying action.

Initially, using a simple classifier such as ELM (extreme learning machine) classifier (Geetha and Geethalakshmi 2011), determine the accuracy according to a certain action A_1 , i.e., determine the classification accuracy for the input matrix F_{ExS} (e.g., $F_{14 \times 84}$), considering the features from all electrodes, denoted as; Acc, which can be calculated by Eq. (4).

$$ACC = \frac{\sum_{i=1}^M n_{ii}}{N} \quad (4)$$

where the numerator represents the correctly classified samples and the denominator (e.g., N) represents the total number of samples. Then, the classification accuracy when using only one electrode E_i is calculated, if the accuracy decreased, then add it to the bad effect set of electrodes denoted as B , which reduces the accuracy. If using the features of E_i increases the accuracy, then add it to the informative set of electrodes IS. After that, the classification accuracy is determined when using the two electrodes E_i and E_{i+1} , if the accuracy decreased, then add the electrode E_{i+1} to the bad effect set of electrodes (e.g., B). Afterward, classification accuracy when using another electrode E_{i+2} with E_i is determined, else add E_{i+1} to the informative set of electrodes IS, and repeat again for all the remaining electrodes. Finally, there will be two sets of electrodes representing the informative electrode IS and the bad effect electrodes B for

Table 1 An example of the proposed feature reduction method

Normalized row values (R)	Selected values								(1) Computed average	(2) NW	(3) Selected features range	
	S1	S2	S3	S4	S5	S6	S7	S8			Greater than	Less than
X1	0,11											
X2	0,20											
X3	0,90											
X4	0,30	0,30	0,30	0,30	0,30	0,30			Average R	0,537	0,142	0,932
X5	0,55	0,55	0,55	0,55	0,55	0,55			Average S1	0,559	0,209	0,909
X6	0,44	0,44	0,44	0,44	0,44	0,44	0,44		Average S2	0,509	0,234	0,784
X7	0,50	0,50	0,50	0,50	0,50	0,50	0,50	0,44	Average S3	0,485	0,260	0,710
X8	0,44	0,44	0,44	0,44	0,44	0,44	0,44	0,50	Average S4	0,442	0,267	0,617
X9	0,90	0,90	0,90	0,90	0,90	0,90	0,90	0,44	Average S5	0,403	0,253	0,553
X10	0,90	0,90	0,90	0,90	0,90	0,90	0,90	0,44	Average S6	0,424	0,299	0,549
X11	0,70	0,70	0,70	0,70	0,70	0,70	0,70	0,60	Average S7	0,455	0,425	0,485
X12	0,60	0,60	0,60	0,60	0,60	0,60	0,60		The final value selected from the given 20 trail values, after repeating the proposed procedure 8 times, is : $X_{i,j} = 0,44$			
X13	0,20											
X14	0,25	0,25	0,25	0,25	0,25	0,25	0,25	0,44				
X15	0,44	0,44	0,44	0,44	0,44	0,44	0,44	0,44	0,44	0,44	0,44	0,44
X16	0,30	0,30	0,30	0,30	0,30	0,30	0,30	0,30	0,30	0,30		
X17	0,90	0,90	0,90	0,90	0,90	0,90	0,90	0,90				
X18	0,80	0,80	0,80	0,80	0,80	0,80	0,80	0,80				
X19	0,70	0,70	0,70	0,70	0,70	0,70	0,70	0,70				
X20	0,60	0,60	0,60	0,60	0,60	0,60	0,60	0,60				

a certain action. Hence, each action has a set of electrodes E (e.g., $E = \{E_1, E_2, E_3, \dots, E_n\}$) and a set of most informative selected set of electrodes $IS \subset E$. The remaining electrodes are considered as the bad effect set of electrodes for that action $B = E - IS$. The bad effect (BE) on the accuracy can be determined, as it is the difference between the best accuracy and the accuracy using every electrode in the bad effect set of electrodes. Also, for every informative electrode its goodness effect (G) can be determined based on the accuracy, which is the difference between the accuracy of the system without using this electrode and after using it.

will be IS and B sets for each action which may differ according to the underlying action.

To solve this issue, we select the common set of electrode, which guarantee the maximum classification accuracy of the whole classification system (e.g., for all considered actions). Electing the suitable set of electrodes is a true challenge. However, this can be accomplished through rule inference methodology and semantic analysis algorithm. The importance of each electrode to the whole system can be determined by analyzing the contents of all the resulted sets semantically and answering the following questions:

Electrode Selection Algorithm for 1 action

- **Input:**
 - Feature Matrix F $E \times S$.
 - E (number of electrodes).
 - S (number of samples).
 - IS (the most informative set of electrodes)=0.
 - B (the bad effect set of electrodes)=0.
 - Acc (the accuracy using a simple classifier)= Acc (total input matrix)
 - $G(E)$ (the goodness effect of a certain electrode on accuracy)
 - $BE(E)$ (the bad effect of a certain electrode on accuracy)

- **Output:**
 - A set of the most informative electrodes IS for the certain action .
 - A set of bad effect electrodes B for the certain action.
 - The effect of using each electrode(goodness G / bad effect BE).

• **Steps:**

```

1:  For each electrode  $E_i \in E$  do
2:      calculate the accuracy  $Acc(IS+E_i)$  from eq.(4).
3:      If  $Acc(IS+E_i) > Acc$  Then
4:           $G(E_i) = Acc(IS+E_i) - Acc$ 
5:           $Acc = Acc(IS+E_i)$ 
6:           $IS = IS + E_i$ 
7:      Else  $B = B + E_j$ 
8:      End if
9:  Next
10: For each electrode  $E_i \in B$  do
11:     calculate the accuracy  $Acc(IS+E_i)$  from eq.(4).
12:      $BE(E_i) = Acc(IS+E_i) - Acc$ 
13: Next
14: For each electrode  $E_i \in IS$  do
15:     calculate the accuracy  $Acc(IS-E_i)$  from eq.(4).
16:      $G(E_i) = Acc - Acc(IS-E_i)$ 
17: Next
    
```

Algorithm Parameters	
F	Feature Matrix has dimension $E \times S$
E	Number of electrodes
S	Number of samples
IS	The most informative set of electrodes
B	The bad effect set of electrodes
ACC	The accuracy using a simple classifier
G(E)	The goodness effect of a certain electrode on accuracy
BE(E)	The bad effect of a certain electrode on accuracy

Algorithm 1: Electrode Selection Algorithm

The above-mentioned procedure is used to determine the mutual effect of the system electrodes on the accuracy of recognizing each action and should be repeated for the remaining actions (e.g., if the system designed to classify four actions, repeat the same procedure for A_2, A_3, A_4). Considering four actions, the result will be the sets of electrodes that improved the accuracy for each action IS_1, IS_2, IS_3, IS_4 and sets of bad effect electrodes B_1, B_2, B_3, B_4 . Furthermore, since the system is designed to classify several actions, there

- If this electrode exists in all informative sets for all actions:
 - ✓ So it can be considered as an informative electrode for all the system.
- If this electrode exists in some pre-defined number of informative sets for some actions and has a little bad effect on the accuracy of the other actions:

- ✓ So it can be considered as an informative electrode for all the system.
- If this electrode exists in all bad effect sets for all actions:
 - ✓ So it can be considered as a bad effect electrode for all the system.
- If this electrode exists in some pre-defined number of informative sets for some actions and has a great not neglected bad effect on the accuracy of the other actions:
 - ✓ So it can be considered as a bad effect electrode for all the system.
- Otherwise a comparison between the goodness effect of this electrode on some actions and its bad effect degree on the other action could be established to determine its effect on all the system actions and its importance for the system.

Hence, a set of election rules could be concluded, then applied whenever it is needed to elect the most suitable set of electrodes to express the data for the whole classification system. Accordingly, this reduces the dimension of the employed dataset as well as the response time for the classification system. Election rules are illustrated in Table 2. On the other hand, electrode election methodology is depicted in Algorithm 2.

Table 2 Rules of electrode election methodology

Rule	Description	Function
R_1	If $E_i \in IS_A$ for all actions $\in AS$ Then (Accept E_i)	Acceptance rule
R_2	If $E_i \in IS_A$ for a pre-defined number of actions ξ , AND has little bad effect BE on the other actions Then (Accept E_i)	
R_3	If $E_i \in IS_A$ for a pre-defined number of actions ξ with a little goodness effect value, AND has a not neglected bad effect BE on the other actions Then (Reject E_i)	Rejection rule
R_4	If $E_i \in B_A$ for all actions $\in AS$ Then (Reject E_i)	
R_5	If E_i has acceptable goodness values for some actions and has not neglected bad effect on other actions Then (Compare goodness and bad effect then take the decision either to accept or reject)	Acceptance/ rejection rule

Illustrative example

For illustration, consider a system that uses 6-EEG electrodes to classify four actions with trial duration of 6 s and the sig-

Electrode Selection for all actions

• **Input:**

- AS (set of all actions).
- N of Feature matrices F EXS for all actions, E (number of electrodes), S (number of samples).
- E (set of all electrodes)
- ISA (set of informative electrodes for a certain action).
- BA (set of informative electrodes for a certain action).
- G(E) , BE(E) the effect of each electrode for every action
- CIS (the common informative set of electrodes)=0.
- CB (the common bad effect set of electrodes)=0.
- ξ (a predefined percentage of actions >50%)

• **Output:**

- A set of the common informative electrodes CIS for all actions.

• **Steps:**

```

1: For each electrode  $E_i \in E$  do
2:   If  $E_i \in IS_A$  for all actions  $\in AS$  Then  $CIS=CIS+E_i$ 
3:   else If  $E_i \in B_A$  for all actions  $\in AS$  Then discard  $E_i$ 
4:   else If  $E_i \in IS_A$  for  $\xi$  of actions Then
5:     If  $E_i$  has little bad effect BE on the remaining actions  $(100-\xi)\%$  Then  $CIS=CIS+E_i$ 
6:     else If  $E_i$  promotes slightly performance of  $\xi\%$  actions Then discard  $E_i$ 
7:     else Compare  $E_i$ 's bad effect BE against its goodness G Then take the decision to add or discard  $E_i$ 
8:     End if
9:   else Discard  $E_i$ 
10:  End if
11: Next
    
```

Algorithm Parameters	
AS	set of all actions
N	Number of feature matrices F
F	Feature Matrix of dimension ExS
E	number of electrodes
S	number of samples
ISA	set of informative electrodes for a certain action
BA	set of informative electrodes for a certain action
G(E) , BE(E)	the effect of each electrode for every action (Good/Bad effect)
CIS	the common informative set of electrodes
CB	the common bad effect set of electrodes
ξ	a predefined percentage of number of actions

Algorithm 2: Electrode Selection for all actions Algorithm

Table 3 Selection for the most informative set of electrodes for the first action

Step	Input attribute (information of)						Accuracy for action A1
	E1	E2	E3	E4	E5	E6	
*	√	√	√	√	√	√	89.1
1	√	X	X	X	X	X	89.5
2	X	√	X	X	X	X	88.1
3	X	X	√	X	X	X	87.5
4	X	X	X	√	X	X	85.3
5	X	X	X	X	√	X	83.4
6	X	X	X	X	X	√	68.9
7	√	√	X	X	X	X	90
:	:	:	:	:	:	:	:
50	√	√	√	X	X	X	87.7
51	√	√	X	√	X	X	90.2
52	√	√	X	√	√	X	91.5
53	√	√	X	√	√	√	90.4
Step	Input attribute (information of)						Accuracy for action A1
	E1	E2	E3	E4	E5	E6	
*	√	√	√	√	√	√	89.1
:	:	:	:	:	:	:	:
63	:	:	:	:	:	:	:

Table 4 Effect (G—goodness/BE—bad effect) of each electrodes for the first action

Step	Input attribute (information of)						Accuracy for action A1	G	BE
	E1	E2	E3	E4	E5	E6			
0	√	√	X	√	√	X	91.5	-	-
1	√	√	√	√	√	X	91	-	-0.5
2	√	√	X	√	√	√	90.4	-	-1.1
3	X	√	X	√	√	X	90.5	1	-
4	√	X	X	√	√	X	90.2	1.3	-
5	√	√	X	X	√	X	91.3	0.2	-
6	√	√	X	√	X	X	90	1.5	-

nal of each electrode has been sampled to 5 samples/s. So, there will be four feature matrices, each of dimension 6×30 , and the number of each matrix's elements is equal to 180 elements. The goal is to distinguish only a set of the most informative electrodes for the 4 actions. Use the extreme learning machine (ELM) classification (Geetha and Geethalakshmi 2011), as a simple learning algorithm on the training set to select the set of electrodes that achieve the best accuracy.

As illustrated in Table 3, the classification accuracy using all electrodes for the first action is 89.1%. However, it is lightly improved when using only features from E_1 . Determining the classification accuracy using features from E_2 and the features from E_1 gives another little improvement. Repeat the calculation of the classification accuracy after adding new electrodes to the previous set, showing that the system has the max classification accuracy equals to 91.5% using the set of electrodes $IS_1 = \{E_1, E_2, E_4, E_5\}$, while

adding electrodes E_3 and E_6 reduces the classification accuracy. Accordingly, they are considered as the items of the bad effect set, e.g., $B_1 = \{E_3, E_6\}$.

From Table 3, it will be easy to determine the goodness (G) or bad effect (BE) of each electrode according to the promotion or demotion in the system's performance (e.g., classification accuracy), as depicted in Table 4. Repeat the same to determine the most informative set of electrodes as well as the bad effect set of electrodes for the other three actions. Results are presented in Table 4.

It is clear from Table 5 that E_1 and E_5 are good electrodes for all actions. Hence, they should be used on the classification of the four actions. On the other hand, E_3 is a common bad effect electrode for all actions, so it should be discarded with no effect on the classification accuracy. E_2 is most informative for three actions and has only bad effect on A_2 . So, it can also be discarded. On the other hand, each of the electrodes E_4 and E_6 has a bad effect on two actions out of

Table 5 Selected electrodes on the four actions

Action	Input attribute (information of)						Accuracy
	<i>E1</i>	<i>E2</i>	<i>E3</i>	<i>E4</i>	<i>E5</i>	<i>E6</i>	
A1	$G = 1$	$G = 1.3$	$B = (-0.5)$	$G = 0.2$	$G = 1.5$	$B = (-1.1)$	91.5
A2	$G = 0.5$	$B = (-0.2)$	$B = (-1)$	$G = 0.1$	$G = 1.3$	$G = 0.2$	88.9
A3	$G = 1.1$	$G = 0.9$	$B = (-0.4)$	$B = (-0.9)$	$G = 1.4$	$B = (-0.8)$	90.4
A4	$G = 0.9$	$G = 1.5$	$B = (-0.2)$	$B = (-1.1)$	$G = 1.1$	$G = 0.1$	91.1

the four actions. So, the goodness and badness effect of them will be compared. From Table 4, it can be concluded that they have slightly goodness effect and their bad effect cannot be neglected; hence, the decision is to discard them. Finally, the common informative set of electrodes for the given classification system consists of only {*E1*, *E2*, *E5*} instead of the six electrodes. Accordingly, the input feature matrix for this classification system will have the dimension of 3 × 30 instead of 6 × 30, which in turn minimizes the number of the input elements to be the half of the original data.

4.6 Classification and decision making

BCI performance is measured by its classification accuracy (Aydemir and Kayikcioglu 2013). In order to guarantee online classification, the classifier must be quick enough to do real-time classification of the EEG signals. Accordingly, several issues must be carefully considered, which are: (i) classification should be robust with respect to outliers, since the neurophysiologic signals may contain several outliers as well as artifacts. (ii) The employed classification technique should have as low computational complexity as possible since data in BCI system should be processed in real time. (iii) Classification should provide confidence prediction levels as the nature basis to combine information obtained from different sources.

K nearest neighbors (KNN) classifier is a classical supervised method in the field of machine learning. It is based on statistical data and is widely used in many areas such as text classification, pattern recognition, and image processing. The decision rule of KNN algorithm is to find the *K* nearest or most similar training samples in the feature space, and then, the test sample is assigned to a majority vote of its *K* nearest neighbors (Zhao and Chen 2016). KNN performance depends on two factors, which are: (i) the assigned value of *K* that represents the number of considered neighbors, and (ii) the employed distance metric. The mostly used method to compute distance between a test sample and the specified training samples is Euclidean distance (Zhao and Chen 2016), which can be computed by Eq. (5).

$$\text{Dist}(X, Y) = \sqrt{\sum_{i=1}^n (x_i - y_i)^2} \tag{5}$$

where $\text{Dist}(X, Y)$ is the Euclidean distance between a test sample *X* and a specified training sample *Y* of features (1,2,...,n), x_i represents the features of the test sample *X*, y_i represents the features of the specified training sample *Y*, and *n* is the total number of features.

The main advantage of KNN is that it can easily deal with problems in which the number of classes is more than two. In addition, KNN allows adding examples to training dataset without retraining the classifier. The work in this paper extends the main concept of KNN to make the choice of the appropriate action. Accordingly, a new instance of KNN is produced with enhanced characteristics via a set of additional parameters, which are illustrated through the following definitions.

Definition 1 Distance To Center (DTC) DTC_i is defined as the distance from the testing item to the center of the class corresponding to the *i*th action.

Definition 2 Inverse Belonging Degree (IBD) IBD_i is defined as the average distance from the *K* nearest neighbors of the testing item to the center of the class corresponding to the *i*th action.

Definition 3 Number of the nearest neighbors (NNN) NNN_i is defined as the number of nearest neighbors of the class corresponding to the *i*th action for the testing item.

As depicted in the above definitions, one of these additional parameters is the distance between the input test signal and the center point of each action’s training data DTC_i (where *i* represents the corresponding action’s identifier). Moreover, the average distance from each of selected nearest neighbor points in the training dataset related to a certain action and the center point of this action is considered as a new parameter called inverse belonging degree (IBD_i). The inverse belonging degree affects the choice of the appropriated action that it helps to remove the outliers with highest IBD_i . A fuzzy inference system is used to combine those parameters for formulating the suitable decision. Hence, the new classification strategy is called fuzzy-based classification strategy (FBCS). Initially, during the training phase the centers of the underlying classes are calculated, which are denoted as; c_1, c_2, c_3, c_4 considering four actions (*A1*, *A2*, *A3*, and *A4*) as depicted in Fig. 7.

During the testing phase, initially, the distances from the unknown (unclassified) input item, which is expressed in a feature matrix and the pre-classified training items, are determined. Then, the K nearest training items are selected. Then the pre-mentioned parameters should be calculated to perfectly classify the input unclassified item, namely number of nearest neighbor (NNN), distance to center (DTC), and inverse belonging degree (IBD). Generally, the more the number of the nearest items related to the i th action (e.g., NNN_i) the more probability that the unknown item would be associated with that action. On the other hand, the less the distance from the tested item to the center of the i th class (e.g., DTC_i) the more the probability that the unclassified item related to that action. Moreover, the inverse belonging degree of the input item to i th class, which is the (average) distance from the selected nearest neighboring points of the training dataset related to each action and input unclassified

item can be calculated by Eq. (6). Actually, the less the inverse belonging degree to the i th class (e.g., IBD_i), the more the probability that the input item is related to that class (action).

$$IBDi = \frac{\sum_{j=1}^M ds_j}{NNN} \tag{6}$$

where NNN is the number of selected nearest points associated to an action A_i , ds_j is the distance between each selected nearest training point associated to the action A_i and the center point of the training set for that action. The above-mentioned three parameters NNN_i , DC_i , and IBD_i for each action i are considered as three different fuzzy sets. Then, a proposed fuzzy inference system is employed to predict the weight of each action (W_i) to be the appropriate action for the unknown input features. The fuzzy inference system is realized through three steps, namely (i) fuzzification of inputs, (ii) fuzzy rule induction, and finally (iii) defuzzification.

Fuzzy Based Classification Strategy (FBCS)

Input:

- X is the test feature matrix
- Y is the trained reduced feature matrix
- N is the number of all the training points in the data set
- K is the number of selected nearest neighbor training points to the test point
- $Dist$ is the distance between the test point and each one of the training points
- KNN is a list of the nearest neighbor training points to the test point
- C_i represents the center point of the training data for each action
- DTC is the distance to the center
- IBD is the inverse belonging degree
- NNN is the number of nearest neighbors associated to a certain action
- ds_{ij} is the distance between each nearest training point j of an action i to the center of the same action i
- W_i is the calculated weight of each action
- α, β are the parameters of the fuzzification membership function

Output:

- A represents the selected action after classification

Training:

- EEG data acquisition
- Signal filtering
- Feature extraction
- Feature reduction steps
- Electrode selection (algorithm 1, algorithm 2)
- Reduced training dataset Y
- Determine the C_i center point according to the data set of each action i

Testing:

- EEG data acquisition from the selected common electrodes
- For each $j=1$ to N do**
- Calculate $dist(x_j, y_j)$ from eq. (5)
- Next**
- Select the KNN list of K elements represents the min $dist(x_i, y_i)$
- For each action i do**
- Determine NNN_i from KNN list
- $DTC_i = dist(X, C_i)$ from eq. (5)
- For each point j in KNN corresponding to action i do**
- $ds_{ij} = dist(j, C_i)$ from eq. (5)
- Next**
- Calculate IBD_i from eq. (6)
- $W_i = Call\ Fuzzy_Cal(NNN_i, DTC_i, IBD_i)$
- Next**
- $A = A(max\ W_i)$

Algorithm Parameters	
X	The Test feature matrix
Y	The Trained reduced feature matrix
N	number of all the training points in the data set
K	number of selected nearest neighbor training points to the test point
Dist	distance between the test point and each one of the training points
KNN	list of the nearest neighbor training points to the test point
C_i	the center point of the training data for each action
DTC $_i$	Distance to center
IBD $_i$	inverse Belonging Degree
NNN $_i$	number of nearest neighbors associated to a certain action
ds_{ij}	distance between each nearest training point j of an action i to the center of the same action i
W_i	the calculated weight of each action i
α, β	parameters of the fuzzification membership function
A	the selected action after classification

Procedure Fuzzy_Cal(NNN,DTC,BD)

Input:

- μ is the membership value (Low/High) of each input parameter
- OL is the Fuzzy logic output of each rule
- OV is the Fuzzy membership value output of each rule
- OR is the output membership value of Fuzzy Rule Induction Step

Output:

- W is the weight value of the associated action

Steps:

- Determine $\mu_{Low}(NNN) \& \mu_{High}(NNN)$, where $\alpha=1, \beta=10$ from figure (9)
- Determine $\mu_{Low}(DTC) \& \mu_{High}(DTC)$, where $\alpha=0.169, \beta=0.2$ from figure (9)
- Determine $\mu_{Low}(IBD) \& \mu_{High}(IBD)$, where $\alpha=0.1, \beta=0.15$ from figure (9)
- For each Fuzzy rule i listed in table (7) do**
- Determine OL (i) associated to rule I from table (7)
- Calculate $OV(i) = \min(\mu_{Low/High}(NNN), \mu_{Low/High}(DTC), \mu_{Low/High}(IBD))$, where $\mu_{Low/High}$ selected according to the rule

Next

- Determine $OR_{Low/medim/High} = \max$ (associated value OV of $OL_{(Low/medim/High)}$)
- Plot three values of $OR_{Low/medim/High}$ on the output membership
- $W =$ the area under the resulting curve from eq.(9)

Return (W)
End Procedure

Algorithm 3: Fuzzy Based Classification Strategy (FBCS)

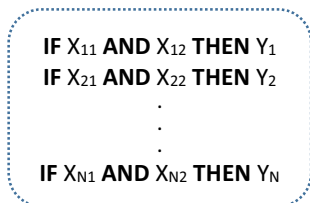
(i) *Fuzzification*

Generally, NNN_i , DTC_i , and IBD_i would be considered as three different fuzzy sets. During the fuzzification step, the inputs are transformed into degree of membership for linguistic terms ‘low’ and ‘high’ of the considered fuzzy set. Then, a membership function is employed to provide the similarity degree of the considered input to the corresponding fuzzy set. The result is a value between 0.0 (for non-membership) and 1.0 (for full-membership). The used membership functions for the considered fuzzy sets are depicted in Fig. 8. On the other hand, the used values of α and β are illustrated in Table 6, assuming $K = 10$.

(ii) *Fuzzy rule induction*

After fuzzification, the result is introduced as the input for the fuzzy rule base. The considered rules are in the form; if (A is x) AND (B is y) AND (C is z)... THEN (O is m), where A , B , and C represent the input variables (e.g., NNN_i , DTC_i , and IBD_i), while x , y , and z represent the corresponding linguistic terms (e.g., ‘Low’ or ‘High’), O represents the rule output, and m represents the corresponding linguistic terms (‘Low’, ‘Medium’, or ‘High’). Actually, there are 8 rules, which are illustrated in Table 7 (assuming L refers to ‘Low,’ H refers to ‘High,’ and M refers to ‘Medium’). For clarification, the first rule in Table 6 indicates that if $NNN(\text{action } i)$ is Low AND $DTC(\text{action } i)$ is Low AND $IBD(\text{action } i)$ is Low THEN Output is Medium.

As illustrated in Giarratano and Riley (2004), four methods for fuzzy rules inference are available. These are max–min, max-product inference, sum-dot method, and drastic product. We choose the max–min method to be used in this paper, which is based on choosing a min operator for the conjunction in the rule premise and for the implication function while the max operator is used for the aggregation. Consider the case of using two items of evidence per rule, the resultant rules will be;



Thus, the max–min composition inference rule would be:

$$\mu_Y = \overbrace{\max}^{\text{aggregation}} \left[\underbrace{\min}_{\text{implication}} (\mu_{X_{j1}}, \mu_{X_{j2}}) \forall j \in \{1, 2, 3, \dots, N\} \right] \tag{7}$$

This produces

$$\mu_Y = \max \left[\min (\mu_{X_{11}}, \mu_{X_{12}}), \min (\mu_{X_{21}}, \mu_{X_{22}}), \dots, \min (\mu_{X_{N1}}, \mu_{X_{N2}}) \right] \tag{8}$$

where μ_x is the value of membership function associated with each fuzzy parameter, N is the number of Fuzzy rules, and μ_Y represents the output membership value of fuzzy rule induction step.

(iii) *Defuzzification*

The output of the fuzzy rules after applying it on the fuzzified inputs is then defuzzified. Defuzzification process is a transformation from a space of fuzzy actions into a space of non-fuzzy ones. As depicted in Saleh et al. (2015), the most commonly used defuzzification techniques are: max-criterion, the mean of maxima, and center of gravity (COG). In COG, the weighted average of the area bounded by the membership function curve is computed to be the crisp value of the fuzzy quantity as illustrated in Eq. (9). Defuzzification would be accomplished by using the output membership function illustrated in Fig. 9.

$$COG = \sum \mu (W_i) * W_i / \mu (W_i) \tag{9}$$

So, the fuzzified KNN classifier algorithm can be written as the following steps:

- Determine the values of the three fuzzy input parameters NNN_i , DTC_i , and IBD_i for each action i .
- Apply the fuzzy rules according to them, then determine the output membership values (μ_{Low} , μ_{High} , μ_{Medium}) according to Eq. (8).
- Plot the output membership values (μ_{Low} , μ_{High} , μ_{Medium}) on the output membership function graph (Fig. 10),
- Finally, determine the area under the resulting curve to have the weight of the action to be the correspondent action to the unknown input feature according to Eq. (9), and select the action with the highest weight.

5 Results and discussion

As the proposed fuzzy-based classification strategy (FBCS) mainly contributes on; feature reduction to acquire a set of compact and informative features, electrode selection to minimize processing time, and classification to take the corresponding precise decisions, in this section, the proposed FBCS will be evaluated against some other approaches previously published in feature reduction, electrode selection and classification of BCI. Informedness (Powers 2003) is used to

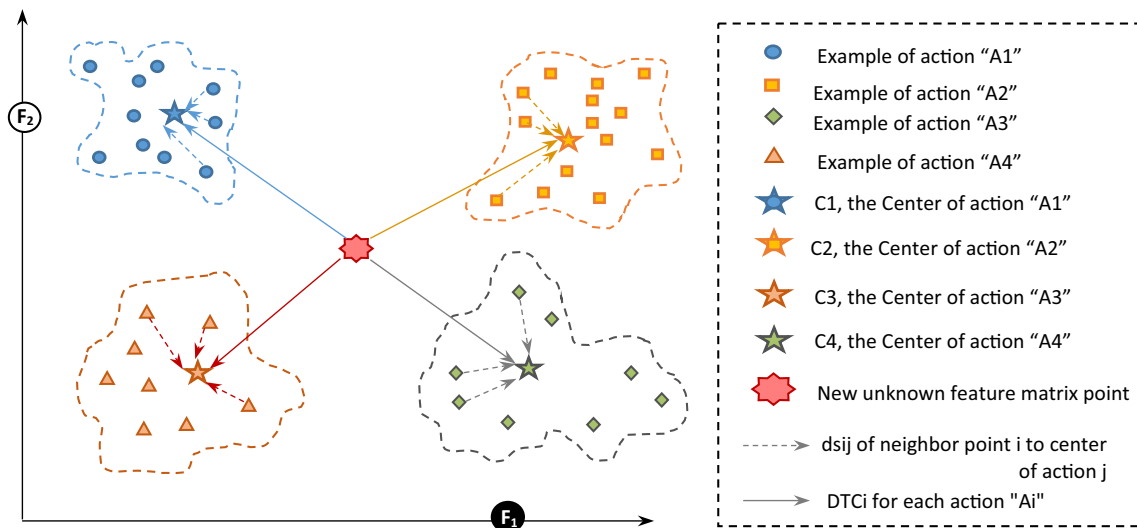


Fig. 7 An example for new unknown feature vector to be classified in the system which has training features for four actions and the proposed fuzzy parameters

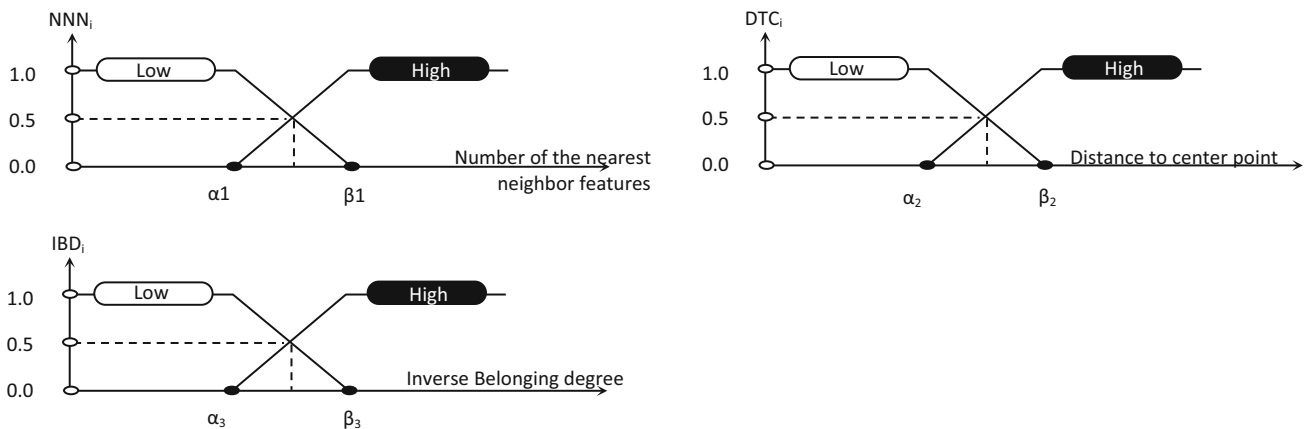


Fig. 8 Membership functions for the considered fuzzy sets

Table 6 Assigned values of α and β

Parameter	Assigned value
α_1	1
β_1	10
α_2	0.169
β_2	0.2
α_3	0.1
β_3	0.15

Table 7 Used fuzzy rules for the fuzzified KNN classifier

ID	NNN_i	DTC_i	IBD_i	Rule output
1	L	L	L	M
2	L	L	H	M
3	L	H	L	L
4	L	H	H	L
5	H	L	L	H
6	H	L	H	M
7	H	H	L	L
8	H	H	H	L

evaluate the performance, as used in the evaluation of feature selection and electrode reduction techniques in Atyabi et al. (2012), as it is more informative in comparison with accuracy taking specificity and sensitivity into account, as in Eq. 10.

$$\text{Informedness} = \frac{TP}{TP + FN} + \frac{TN}{TN + FP} - 1 \tag{10}$$

where TP is the true positive, which represents the number of positive samples correctly predicted, TN is the true negative, which represents the number of negative samples correctly predicted, FP is the false negative, which represents the num-

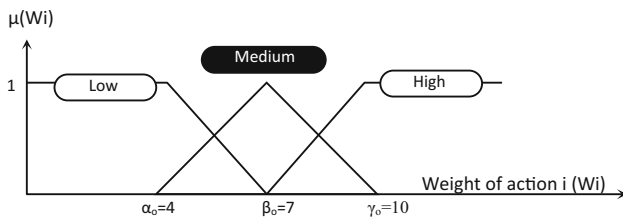


Fig. 9 Output membership function for defuzzification

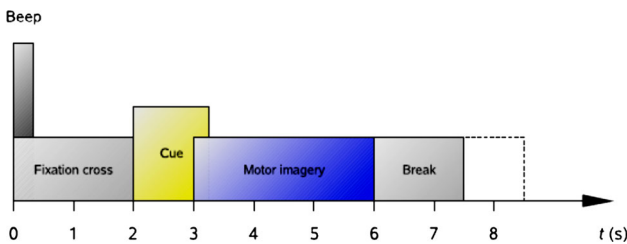


Fig. 10 Time line of EEG signal acquisition

ber of positive samples incorrectly predicted, and FN is the false negative, which represents the number of negative samples incorrectly predicted.

Moreover, the performance metrics: (i) classification accuracy (CA), (ii) sensitivity (SE), (iii) specificity (SP), and (iv) computational time (CT) are also used with the comparison of the classifiers, as applied in Geetha and Geethalakshmi (2011), and their calculating equations are depicted in Eqs. (11–13).

$$CA = \frac{TP + TN}{TP + TN + FP + FN} \times 100 \tag{11}$$

$$SE = \frac{TP}{TP + FN} \times 100 \tag{12}$$

$$SP = \frac{TN}{TN + FP} \times 100 \tag{13}$$

Higher accuracies reflect better decoding (prediction) of class information from the EEG data features. To obtain statistically significant conclusions, a Wilcoxon rank-sum test (Sheskin 2003), one nonparametric statistical test to determine whether two independent random samples are from the same distribution, was performed to compare the accuracies of the classification system achieved with the proposed algorithm with the accuracies achieved by the other competitors, due to the simplicity of these statistics. In statistical hypothesis testing, the *p* value or probability value is the probability for a given statistical model that, when the null hypothesis is true, the statistical summary would be the same as or more extreme than the actual observed results. The computation is based on a rank for each *p* value, a threshold value of 0.05 was applied which means that if the *p* value is very low (*p* < 0.05), we reject the null hypothesis and the result is considered significant. On the other hand, if the *p* value is greater

than 0.05, we accept the null hypothesis. The null hypothesis of this statistical test assumes the equivalent performance of all the competitor algorithms.

5.1 Employed datasets

In order to evaluate the proposed FBCS strategy, we carried out the experimental testing using two well-known BCI datasets. The first is labeled as Dataset D1, which is Dataset 2a of BCI competition IV provided by BCI research group at Graz University (Brunner et al. 2008). Recoded from 9 healthy subjects, the dataset consists of four different tasks known as motor imagery (MI) tasks, namely left hand, right hand, both feet, and tongue recorded on different days from each subject. The dataset is comprised of two sessions, each with six runs separated by short breaks. Every run includes 48 trials (12 trials for each task). Therefore, there are 288 trails in total per session. For a single trial paradigm, the subjects were sitting in a comfortable armchair in front of a computer screen. As shown in Fig. 11, at the beginning of a trial (*t* = 0s), a fixation cross appeared on the black screen. In addition, a short warning tone was presented. After 2s (*t* = 2s), a cue in the form of an arrow pointing either to the left, right, down, or up (corresponding to one of the four classes left hand, right hand, foot, or tongue) appeared and stayed on the screen for 1.25s. This prompted the subjects to perform the desired MI task. No feedback was provided. The subjects were asked to carry out the motor imagery task until the fixation cross disappeared from the screen at *t* = 6s. A short break followed where the screen was black again. Table 8 gives a short description of the essential parameters of dataset D1.

On the other hand, the second employed dataset is labeled as Dataset D2, which is the Dataset IVa of BCI competition III (http://www.bbc.de/competition/iii/desc_IVa.html). It contains 3.5 s of the following 3 motor imageries in which the subject should perform, namely (L) left hand, (R) right hand, (F) right foot. Motor imagery tasks performed by five healthy subjects over 280 trials. Even though the visual cue was presented to subjects for the total amount of 3.5 s, the beginning and end 0.5 s can be ignored since they represent the transition time during which subject is changing its state from non-task to task and vice versa. Therefore, it is reasonable to omit the first and last 0.5 s and only use the middle 2.5 s for the classification task. As shown in Table 9, the used datasets contain EEG recordings from 118 channels for 2.5 s over 280 trials with 1000Hz sample rate. The experiments have been done using MATLAB R2015b. Simulations have been done using a computer with 4 GB memory and processor core i3 2.53 GHz.

The two datasets (i.e., D1 and D2) were used in testing proposed feature reduction (FR), the proposed electrode selection (ES), and the proposed fuzzified KNN classifier

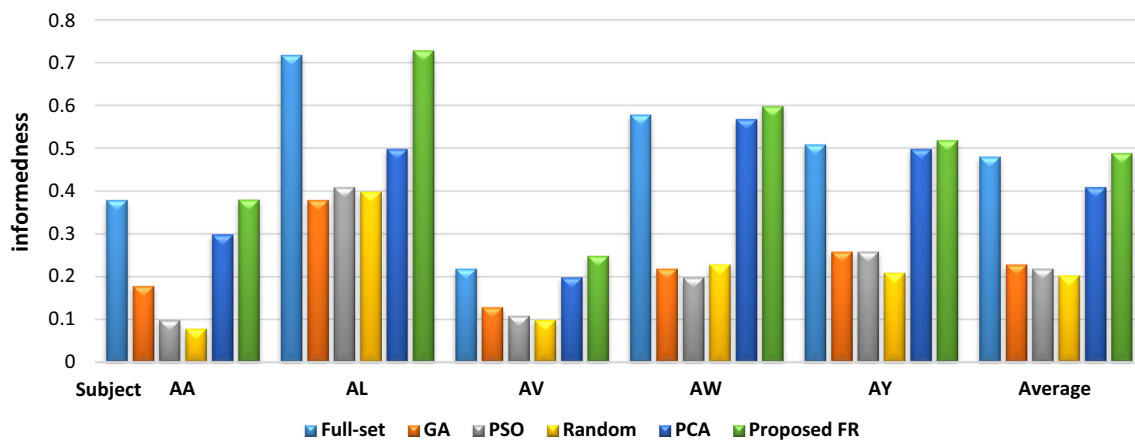


Fig. 11 Average informedness result achieved on testing set with SVM using feature reduction techniques

Table 8 Table of parameters of dataset D1

Parameter	Value
No. of electrodes	22
No. of classes	4 actions; left hand, right hand, feet, tongue
No. of subjects	9
Sampling rate	250 Hz
Filtering	0.5–100 Hz; notch filtered
No. of trials	576

Table 9 Table of parameters of dataset D2

Parameter	Value
No. of electrodes	118
No. of classes	3 actions; left hand, right hand, right foot
No. of subjects	5 (named AA, AL, AV, AW, AY)
Sampling rate	1000 Hz
Filtering	band-pass filtered between 0.05 and 200 Hz
No. of trials	280
Actual time of single trail signal	2.5 s

with some other approaches previously published, as will be discussed in the next subsections.

5.2 Experimental results

This section evaluates in details the contributions introduced in this paper. Three different experiments will be introduced through this section. The first experiment is employed using the dataset D2, for simplicity, to assess the impact of the proposed feature reduction (FR); a comparison is made with feature reduction methods based on genetic algorithms (GA),

particle swarm optimization (PSO), random search algorithm introduced in Atyabi et al. (2012) and principal component analysis (PCA) applied in Muhammad et al. (2015). The comparison shows the impact of the use of the full-set and the use of reduced set, after applying several feature reduction methods, on performance using SVM.

The second experiment is employed on the dataset D2 also to assess the impact of the proposed electrode selection (ES); a comparison is made with electrode selection methods based on genetic algorithms (GA) and random search algorithm introduced in Atyabi et al. (2012). The comparison shows the impact of the use of the full-set and the use of reduced set, after applying the electrode selection methods, on performance using SVM.

Finally, the third experiment is employed to assess the impact of the proposed fuzzified KNN classifier, a comparison is made on the two datasets D1 and D2 with six classifiers namely K nearest neighbor (KNN) (Lotte et al. 2007), support vector machines (SVM) (Lotte et al. 2007), linear discriminate analysis (LDA) (Lotte et al. 2007), Naive Bayes (NB) (Lotte et al. 2007), decision tree (DT) (Lotte et al. 2007), and self-organized map (SOM) (Arvaneh 2011) using their built-in MATLAB implementations with default competitors' parameters. As in Geetha and Geethalakshmi (2011), we used leave-one-out cross-validation (LOOCV) technique to estimate the most appropriate KNN's and SVM's parameters to avoid the problems of random selection as it selects parameters that provide the highest average performance matrices. Table 10 illustrates the values of the control parameters specific to each of the employed competitor.

5.2.1 Evaluating the proposed feature reduction methodology

Uses a 10*20 cross-validation (CV) to assess the impact of the proposed feature reduction (FR). A comparison is made

Table 10 Values of the control parameters specific to each of the employed competitor

Technique	Parameter	Employed value
GA	Population size	100
	Max. iteration	100
	Member of the population	<ul style="list-style-type: none"> • A binary mask formed as a vector containing indices representing features or electrodes (0 and 1 indicating the omission and selection of the feature or electrode, respectively) • For feature reduction: mask size = 250 features • For electrode selection: mask size = 118 electrodes
	No. of repetitions	10
	CV (cross-validation)	20-fold
	M (mutation no. of iterations after it the mask is to be replaced by a random one if it didn't improve)	$M = 10\%$ of Max. iterations (100)
PSO	Acceleration coefficients (C_1, C_2)	0.5 (fixed acceleration coefficient that control the effectiveness of cognitive component)
	Inertia weights (W_1, W_2)	Linearly decreasing inertia weight <ul style="list-style-type: none"> • $W_1 = 0.2$ • $W_2 = 1$
Random search	Population size	100
	Max. iteration	100
	Member of the population	<ul style="list-style-type: none"> • A binary mask formed as a vector containing indices representing features or electrodes (0 and 1 indicating the omission and selection of the feature or electrode, respectively) • For feature reduction: mask size = 30 features • For electrode selection: mask size = 10 electrodes
KNN	K value	The used value of K has been experimentally obtained. Hence, starting with ($K = 1$) as an initial value, a test set is used to estimate the error rate of the classifier. This process is repeated by continuously changing K 's value by gradually adding one more neighbor. Then, The K 's value that introduces the minimum error was selected, which was $K = 10$
SVM	The scaling factor of kernel function (σ)	Leave-one-out cross-validation (LOOCV) (Arlot 2010) technique is used to determine the best value of σ , searched between [0.1,1.5] step size of 0.1

with feature reduction methods based on genetic algorithms (GA), particle swarm optimization (PSO), random search algorithm represented in Atiyabi et al. (2012) and principal component analysis (PCA) applied in Muhammad et al. (2015) on each subject of the 5 subjects according to the D2 dataset . The comparison shows the impact of the use of the full-set and the use of reduced set, after applying several feature reduction methods, on performance using SVM.

Table 11 and Fig. 11 represent the impact of using feature reduction's above-mentioned techniques by the averaged informedness result achieved on testing set with SVM, which has superior results in most cases. The p values obtained through the Wilcoxon rank-sum statistical test between the best algorithm and each of the competitors are listed in Table 12. Values with p value < 0.05 indicate that the differences are statistically significant. The results listed in Tables 11 and 12 clearly indicate the superiority of the proposed FR in statistically significant fashion over the most

Table 11 Averaged informedness result achieved on testing set with SVM using feature reduction techniques

Subject	Full-set	GA	PSO	Random	PCA	Proposed FR
AA	0.38	0.18	0.1	0.08	0.3	0.381
AL	0.72	0.38	0.41	0.4	0.5	0.72
AV	0.22	0.13	0.11	0.10	0.2	0.25
AW	0.58	0.22	0.2	0.23	0.57	0.6
AY	0.51	0.26	0.26	0.21	0.5	0.52
Average	0.482	0.23	0.22	0.204	0.41	0.49

of the competitors. However, the statistical test for subjects named AV and AW indicates no significant difference of the proposed FR over PCA as their corresponding p values are 0.052 and 0.057, respectively.

Generally, the results indicate that on average across all subjects (named AA, AL, AV, AW, AY), the informedness

Table 12 p values obtained through the Wilcoxon rank-sum statistical test for the best algorithm—proposed FR—versus each of the competitors for each subject within the dataset

Subject	FR versus GA	FR versus PSO	FR versus Random	FR versus PCA
AA	0.0007	0.004	0.0014	0.0141
AL	0.0048	0.0074	0.0031	0.005
AV	0.0013	0.011	0.0101	0.052
AW	0.0022	0.002	0.023	0.057
AY	0.026	0.0206	0.0002	0.0005

Table 13 Averaged informedness result achieved on testing set with SVM using electrode selection techniques

Subject	Full-set	GA	Random	Proposed ES
AA	0.38	0.22	0.19	0.40
AL	0.72	0.60	0.51	0.75
AV	0.22	0.20	0.15	0.27
AW	0.58	0.40	0.30	0.65
AY	0.51	0.41	0.38	0.54
Average	0.482	0.366	0.306	0.522

has slightly improved from 0.482 to 0.49 using the proposed FR technique. The other techniques need to be improved to achieve high performance, so the next experiment shows the impact of electrode selection.

5.2.2 Evaluating the proposed electrode selection methodology

The same as experiment 1, experiment 2 evaluates the impact of the proposed electrode selection (ES) using D2 dataset. A comparison is made with electrode selection methods based on genetic algorithms (GA) and random search algorithm introduced in Atyabi et al. (2012). The comparison shows the impact of the use of the full-set and the use of reduced set, after applying the electrode selection methods, on performance using SVM.

Table 13 and Fig. 12 represent the impact of using the electrode selection's above-mentioned techniques by the

Table 14 p values obtained through the Wilcoxon rank-sum statistical test for the best algorithm—proposed ES—versus each of the competitors for each subject within the dataset

Subject	ES versus GA	ES versus random
AA	0.0022	0.019
AL	0.0506	0.002
AV	0.0202	0.0015
AW	0.0408	0.0003
AY	0.0041	0.0602

averaged informedness result achieved on testing set with SVM. The results indicate that on average across all subjects, the informedness has improved from 0.482 to 0.522 using the proposed ES technique. According to Table 14, where the p values obtained through the Wilcoxon rank-sum statistical test between the best algorithm and each of the competitors are displayed, the proposed ES is statistically significant better than the rest competitors. However, the statistical test for subject named AL indicates that no significant difference of the proposed ES over GA and for subject AY indicates insignificant dominance of ES over random as their corresponding p values are 0.0506 and 0.0602, respectively.

Comparing the results with the results of experiment 1, it is clearly that electrode selection techniques had a greater impact on classification performance compared to feature reduction. In the next experiment, evaluate the impact of the

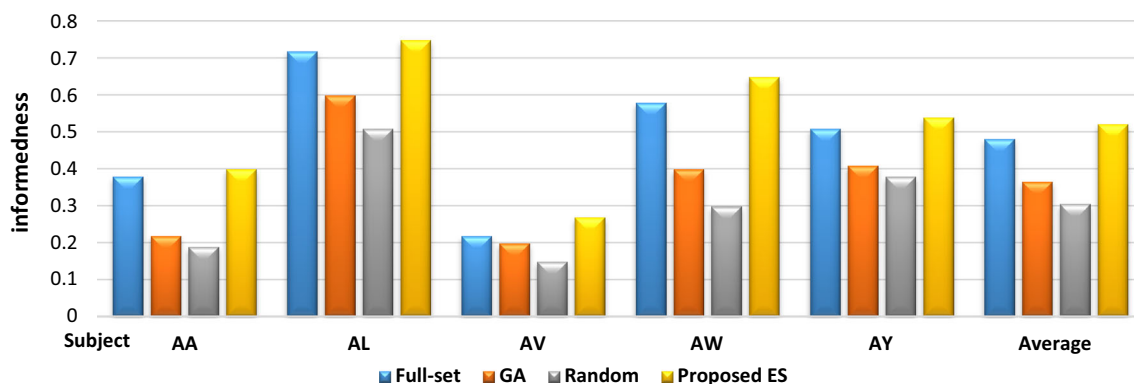


Fig. 12 Average informedness result achieved on testing set with SVM using electrode selection techniques

combination of proposed feature reduction (FR) and electrode selection (ES) with the proposed FKNN and other classifiers.

5.2.3 Evaluating the proposed fuzzified KNN classifier

Finally, to assess the impact of the proposed fuzzified KNN classifier ($K = 10$) using the two datasets (i.e., D1 and D2), a comparison is made against K nearest neighbor (KNN) (Lotte et al. 2007), support vector machines (SVM) (Lotte et al. 2007), linear discriminate analysis (LDA) (Lotte et al. 2007), naive Bayes (NB) (Lotte et al. 2007), decision tree (DT) (Lotte et al. 2007), and self-organized map (SOM) (Muhammad et al. 2015) classifiers using their built-in MATLAB implementations with default techniques’ parameters and values listed in Table 10. The performance metrics illustrated with Eqs. (11–13) are evaluated, giving the following results.

Figure 13 illustrates the average classification accuracy (CA), which represents the percentage of the number of trails classified correctly in the test set over the total trails, for the competing classifiers as well as FKNN. Both datasets (i.e., D1 and D2) are employed on the original datasets without using dimensionality reduction, noting that D2 (118 electrode) is higher dimensional than D1 (22 electrode). As illustrated in such figure, FKNN has the maximum average classification accuracy (78% for D1 and 74% for D2), while KNN showed the worst average classification accuracy (70.1% for D1 and 60.8% for D2) as it is very sensitive to the curse of dimensionality and this explains why KNN algorithms are not very popular in the BCI community. On the other hand, Fig. 14 represents the average classification accuracy for the same set of classifiers using the proposed feature reduction and electrode selection methodologies. It is clearly that using feature reduction and electrode selection improves the average accuracy for all classifiers, due to the removal of outlier features and bad effect electrodes. Moreover, it is found that FBCS still has the highest accuracy (85.7% for D1 and 87.8% for D2).

Again, D1 and D2 are employed for measuring the average classification sensitivity (called the true positive rate) for the competing classifiers as well as FKNN in two scenarios. In the first scenario, SE is measured for all classifiers with no dimensionality reduction, while in the second scenario, both the proposed feature selection and electrode reduction methodologies are employed. As illustrated in Fig. 15, SE for all classifiers is shown using the first scenario. It is shown that FKNN presents the maximum average classification sensitivity for both datasets, precisely 76.5% for D1 and 74% for D2, and as proven in average classification accuracy, KNN represents the worst sensitivity with average sensitivity 70.1% for D1 and 60.8% for D2 which has higher dimensions. On the other hand, when the second scenario is followed, as

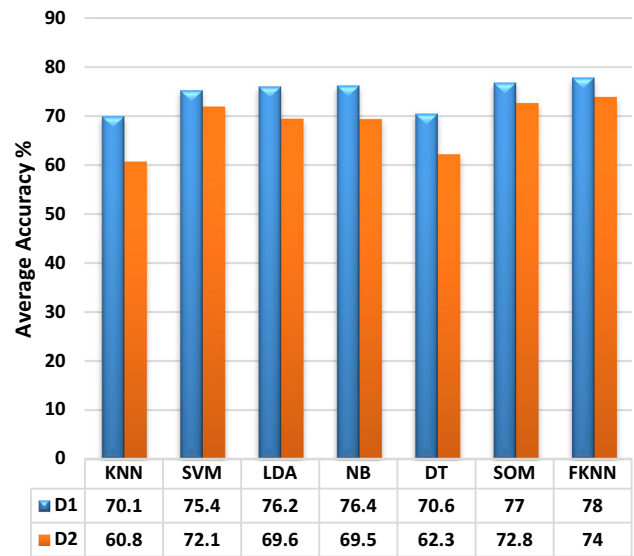


Fig. 13 Average classification accuracy without using dimensionality reduction

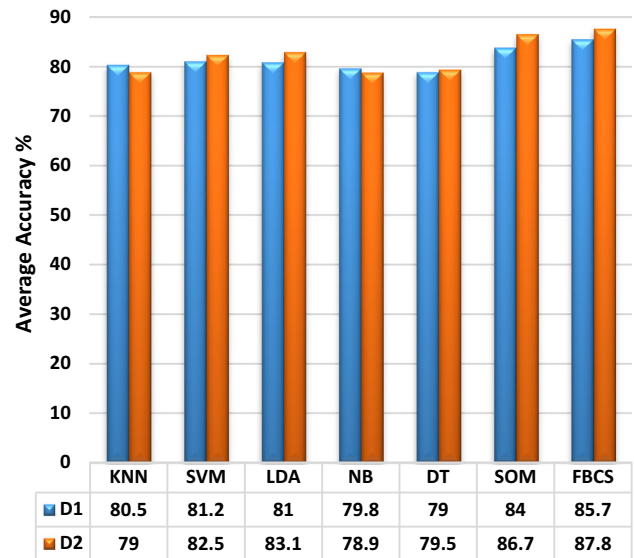


Fig. 14 Average classification accuracy with the proposed feature reduction and electrode selection

illustrated in Fig. 16, SE is promoted for all classifiers. The maximum SE is given by the proposed FKNN, which was 85.1% for D1 and 84.2% for D2. Note that, KNN classifier gives acceptable sensitivity results with feature reduction and electrode selection techniques 82% for D1 and 80.5% for D2 as the dimensions of the datasets were reduced.

In this experiment, the target is to measure the average classification specificity (SP) (also called the true negative rate) for the employed classifiers against FKNN. As illustrated in Fig. 17, with no dimensionality reduction, the proposed FKNN outperforms all other classifiers in terms of SP. It is found that FKNN introduces 79% for D1 and 88%

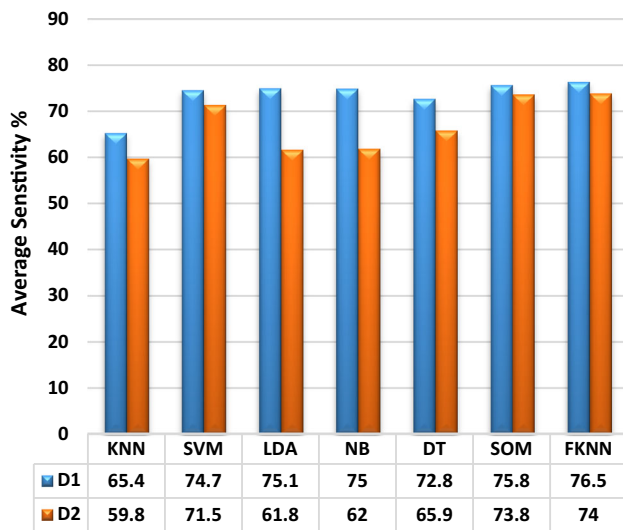


Fig. 15 An average classification sensitivity without using dimensionality reduction

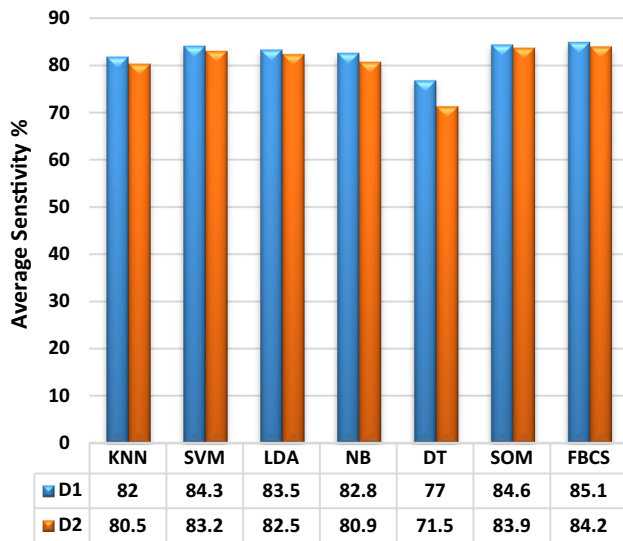


Fig. 16 Average classification sensitivity with the proposed feature reduction and electrode selection

for D2. From another point of view, when applying both the proposed feature selection and electrode reduction methodologies, it is found that SP is promoted for all classifiers as illustrated in Fig. 18. For FKNN, which introduces the maximum SP, the calculated values of SP were 89.9% for D1 and 90.2% for D2.

To insure that the proposed FKNN is suitable for real-time operation, it is essential to measure the testing time. For the training and testing of the two BCI competition datasets without dimensionality reduction using the different classifiers, the average computational time is depicted in Fig. 19. In such figure, it can be seen that KNN and FKNN had the shortest computational time in the training phase with average value

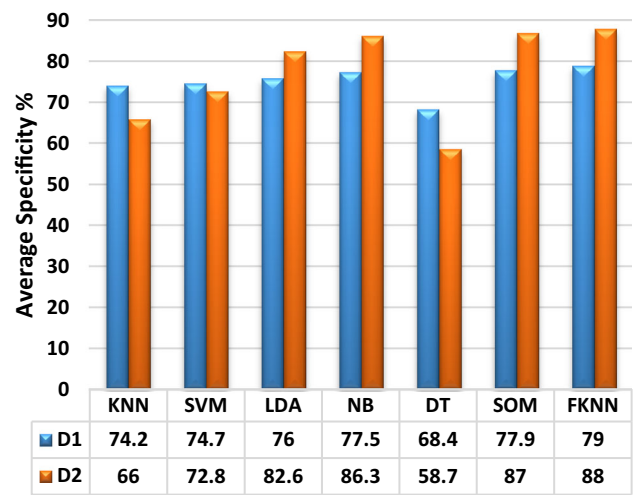


Fig. 17 Average classification specificity without using dimensionality reduction

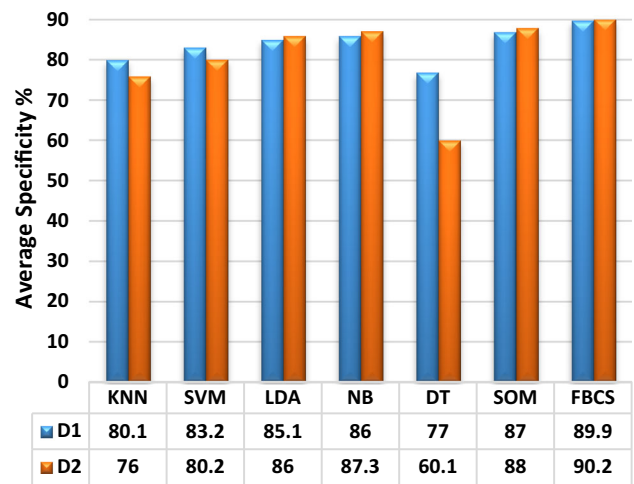


Fig. 18 Average classification specificity with the proposed feature reduction and electrode selection

of the two datasets equal 0.04, 0.05 s, respectively, this is logically true as KNN is a lazy learner. Hence, more time is spent during the testing phase as almost all calculations are done during the testing phase. SVM had the longest time as it performs classification tasks by constructing the best hyperplane in a multidimensional space by maximizing the margin as possible. For the testing phase, SVM and FKNN nearly had the same average computational times (CT) with values of 0.44, 0.5 s, respectively. As seen FKNN computational time is greater than the computational time of KNN in testing phase as it has more computations than simple KNN.

On the other hand, using the dimensional reduction techniques, the average computational times (CT) in the training phase increase for all classification algorithms applied, as the computations of feature reduction (FR) and electrode selec-

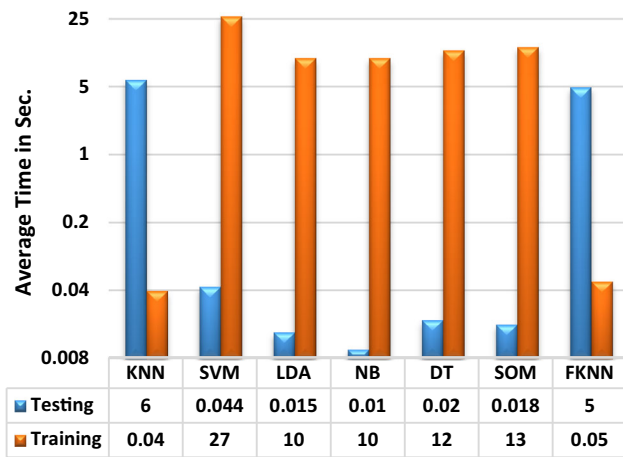


Fig. 19 Average classification computational time without using dimensionality reduction

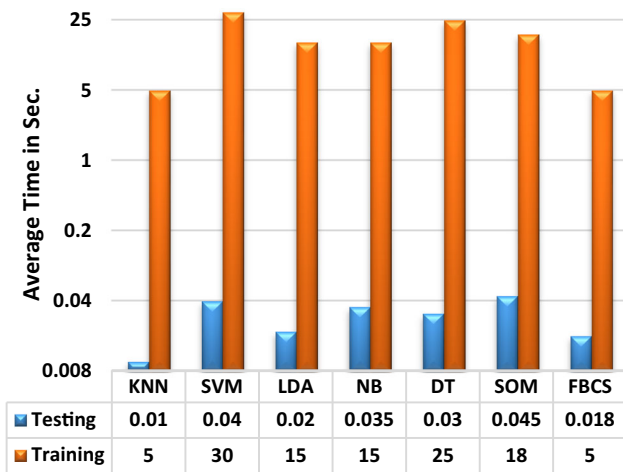


Fig. 20 Average classification computational time with the proposed feature reduction and electrode selection

tion (ES) techniques take more time in the training phase, even the value of CT for training the simple KNN and FBCS classifiers increases to be 5 s. and SVM classifier still having the highest average CT nearly about 30 s, as illustrated in Fig. 20. In the testing phase, the average CT for the classification decreases due to the reduction in the dataset dimensions. The main objective of the proposed strategy is to reduce the computational classification time specially in testing phase to meet the needs of real-time BCI applications. As discussed early, FR and ES need more time to select the best subset of

features and electrode, but there is no need to significantly reduce computational time in training step. For the data captured to be tested, the selected features from the selected electrodes are directly classified referring to the reduced trained set of data, with no need for any feature reduction and electrode selection calculations. KNN has the shortest testing time with CT value 0.01 s. but according to the accuracy we can say that FBCS is still the fastest strategy in testing phase using both datasets with CT value about 0.018 s to take the decision of what expressed by the data captured from the EEG headset.

Table 15 reports the *p* values obtained through the Wilcoxon rank-sum statistical test for the proposed FBCS algorithm versus each of the other competitors applied on the datasets D1 and D2. The results of the statistical test prove that the FBCS with dimension reduction is among the best performing models for the BCI recognition system. Unlike most publications in BCI field which recommended SVM and LDA as the highest performing classifiers, our three experiments show that for each feature reduction and electrode selection method at many classifiers should be tested including the purposed classifier, and generally there is not a best classifier or feature reduction or electrode selection method that outperforms all others as they differs according to the mentality actions of the subject and the combination of all model’s parameters (signal processing, feature extraction, dimensionality reduction and classifier).

6 Conclusion

Dimensional reduction in features is an open problem in brain–computer interfacing (BCI) research. Therefore, features extracted from brain signals are high-dimensional which affects the accuracy of the classifier. Selection of the most relevant features and electrodes improves the performance of the classifier and reduces the computational cost of the system. In this study, a new strategy called fuzzy-based classification strategy (FBCS) is proposed to determine the best subset of features from a selected number of electrodes of electroencephalography (EEG)-based BCI dataset for different actions. Applying the proposed strategy for feature reduction and electrode selection reducing the dimensions of the feature vector is tested and gains the best metrics of

Table 15 *p* values obtained through the Wilcoxon rank-sum statistical test for the best algorithm—proposed FBCS—versus each of the competitors for each dataset

Dataset	FBCS vs. KNN	FBCS vs. SVM	FBCS vs. LDA	FBCS vs. NB	FBCS vs. SOM
D1	0.0022	0.0104	0.052	0.00007	0.00859
D2	0.0052	0.0508	0.0068	0.0098	0.0012

classification performance (specially the computational time CT as a vital parameter) for Dataset 2a of BCI competition IV (Brunner et al. 2008) and Dataset IVa of BCI competition III (http://www.bbc.de/competition/iii/desc_IVa.html). Thus, our algorithm can be employed for further real-time processing of multi-class problem. Our future aim is to design a real system that has the ability to online classify brain tasks in a real environment with less computational time. Further study in this direction will aim to optimize the feature reduction, electrode selection, and classification techniques to be implemented in real-time applications of BCI.

Compliance with ethical standards

Conflict of interest We admit that our paper is well formed due to the ethical standards of the declared author policy. We also declare that they have no conflict of interest. Moreover, informed consent was obtained from all individual participants included in the study.

References

- Abdulkader Sarah N, Atia Ayman, Mostafa Mostafa-Sami M (2015) Brain computer interfacing: applications and challenges. *Egypt Inform J* 16:213–230
- Arlot Sylvain (2010) A survey of cross validation procedures for model selection. *Stat Surv* 4(2):40–79
- Arvaneh M (2011) Optimizing the channel selection and classification accuracy in EEG-based BCI. *IEEE Trans Biomed Eng* 58(6):1865–1873
- Atyabi A, Luerssen M, Fitzgibbon S, Powers DMW (2012) Evolutionary feature selection and electrode reduction for EEG classification. *WCCI 2012 IEEE world congress on computational intelligence*, 10–15 June 2012, Brisbane, Australia
- Aydemir Onder, Kayikcioglu Temel (2013) Comparing common machine learning classifiers in low-dimensional feature vectors for brain computer interface applications. *Int J Innov Comput Inf Control ICIC Int* 9(3):1145–1157
- Brunner C et. al., (2008) BCI competition 2008—Graz data set A. <http://www.bbc.de/competition/iv/#dataset2a>
- Daly JJ, Wolpaw JR (2008) Brain computer interfaces in neurological rehabilitation. *Lancet Neurol* 7:103243
- Fukunaga K (1990) *Introduction to statistical pattern recognition*, 2nd edn. Academic Press, New York
- Geetha G, Geethalakshmi SN (2011) Detecting epileptic seizure using electroencephalogram: a new and optimized method for seizure classification using hybrid extreme learning machine? In: *International conference on Process automation, control and computing (PACC)*, pp 1–6
- Giarratano J, Riley G (2004) *Expert systems: principles and programming*. Course Technology Inc., Cambridge
- Guo X, Wu X, Gong X, Zhang L (2013) Envelope detection based on online ICA algorithm and its application to motor imagery classification. In: *6th International IEEE/EMBS conference on neural engineering (NER)*, pp 1058–1061
- Hasan BAS, Gan JQ (2009) Multi-objective particle swarm optimization for channel selection in brain-computer interfaces. In: *Proceedings of the UK workshop on computational intelligence (UKCI 2009)*, Nottingham. <http://repository.essex.ac.uk/id/eprint/4148>
- Hsu WY (2013) Application of quantum-behaved particle swarm optimization to motor imagery EEG classification. *Int J Neural Syst* 23(6):1350026
http://www.bbc.de/competition/iii/desc_IVa.html
- Jung TP et al (2000) Removing electroencephalographic artifacts by blind source separation. *Psychophysiology* 37:163–178
- Lotte F, Congedo M, Lécuyer A, Lamarche F, Arnaldi B (2007) A review of classification algorithms for EEG-based brain-computer interfaces. *J Neural Eng* 4:R1–R13
- Mallick A, Kappate D (2015) A review on signal pre-processing techniques in brain computer interface. *Int J Comput Technol* 2(4):130–134
- McFarland DJ, Wolpaw JR (2008) Brain-computer interface operation of robotic and prosthetic devices. *Computer* 41(10):52–56
- Muhammad ZB et al (2015) Motor imagery based EEG signal classification using self organizing maps. *Sci Int (Lahore)* 27(2):1165–1170
- Naeem M, Brunner C, Pfurtscheller G (2009) Dimensionality reduction and channel selection of motor imagery electroencephalographic data. *Comput Intell Neurosci* 2009, Article ID 537504. <https://doi.org/10.1155/2009/537504>
- Nanayakkara Asiri, Sakkaff Zahmeeth (2012) Fixed distance neighbour classifiers in brain computer interface systems. *J Natl Sci Found Sri Lanka* 40(3):195–200
- Nicolas-Alonso LF, Gomez-Gil J (2012) Brain computer interfaces, a review. *Sensors* 12:1211–1279. <https://doi.org/10.3390/s120201211>. ISSN 1424-8220. www.mdpi.com/journal/sensors
- Ogiela L, Ogiela MR (2012) Beginnings of cognitive science. *Adv Cognit Inf Syst Cognit Syst Monogr* 17:1–18
- Pal M, Bhattacharyya S, Konar A, Tibarewala DN, Janarthanan R (2014) Decoding of wrist and finger movement from electroencephalography signal. In: *IEEE international conference on electronics, computing and communication technologies*, Bangalore
- Pal M, et. al. (2014) A bacterial foraging optimization and learning automata based feature selection for motor imagery EEG classification. *Conference paper*, July 2015. <https://doi.org/10.1109/SPCOM.2014.6983926>, 978-1-4799-4665-5/14/\$31.00 ©2014 IEEE
- Powers DMW (2003) Recall and precision versus the bookmaker. In: *International conference on cognitive science (ICSC-2003)*, pp 529–534
- Rakotomamonjy A, Guigue V, Mallet G, Alvarado V (2005) Ensemble of SVMs for improving brain computer interface P300 speller performances. *Int Conf Artif Neural Netw* 2005:45–50
- Saleh Ahmed I, Abulwafa Arwa E (2017) A web page distillation strategy for efficient focused crawling based on optimized Naïve bayes (ONB) classifier. *Appl Soft Comput* 53:181–204
- Saleh Ahmed I, El Desouky Ali I, Ali Shereen H (2015) Promoting the performance of vertical recommendation systems by applying new classification techniques. *Knowl Based Syst* 75:192–223
- Saleh Ahmed I, Rabie Asmaa H, Abo-Al-Ez Khaled M (2016) A data mining based load forecasting strategy for smart electrical grids. *Adv Eng Inform* 30:422–448
- Sheskin David J (2003) *Handbook of parametric and nonparametric statistical procedures*. CRC Press, Boca Raton
- Silva LA, Del-Moral-Hernandez E (2011) A SOM combined with KNN for classification task. In: *Proceeding of international joint conference on neural networks*, 31 July–5 Aug, 2011 San Jose, CA, pp 2368–2373
- Yu X, Chum P, Sim KB (2014) Analysis the effect of PCA for feature reduction in non-stationary EEG based motor imagery of BCI system. *Opt Int J Light Electron Opt* 125(3):1498–1502
- Zanchettin C, Bezerra BLD, Wazevedo W (2012) A KNN-SVM hybrid model for cursive hand writing recognition. In: *Proceeding of the IEEE international joint conference on neural networks*, 10–15 June, Brisbane, Australia, pp 1–8

Zhao H et al (2015) Analyze EEG signals with extreme learning machine based on PMIS feature selection. *Int J Mach Learn Cyber*. <https://doi.org/10.1007/s13042-015-0378-x>

Zhao Ming, Chen Jingchao (2016) Improvement and comparison of weighted K nearest neighbors classifiers for model selection. *J Softw Eng* 10(1):109–118