Sparse learning of band power features with genetic channel selection for effective classification of EEG signals.

PADFIELD, N., REN, J., MURRAY, P. and ZHAO, H.

2021

Sparse Learning of Band Power Features with Genetic Channel Selection for Effective Classification of EEG Signals

Natasha Padfield [a] , Jinchang Ren [b,c] *, Paul Murray [a], Huimin Zhao [c]

[a] *Centre for Signal and Image Processing, University of Strathclyde, Glasgow, G1 1XW, UK*
[b] *National Subsea Centre, Robert Gordon University, Aberdeen, U.K.*
[c] *School of Computer Sciences, Guangdong Polytechnic Normal University, Guangzhou 510665, China*

Abstract

In this paper, we present a genetic algorithm (GA) based band power feature sparse learning (SL) approach for classification of electroencephalogram (EEG) (GABSLEEG) in motor imagery (MI) based brain-computer interfacing (BCI). The band power in the alpha and beta bands was extracted from the EEG segments and used as features to construct the SL dictionary, in which the GA was employed for channel selection. The GABSLEEG system was tested in three functional areas: i) classification of MI data and idle state data; ii) performance with decreased training data size; and iii) computational efficiency. The system was evaluated by dividing the data into training, validation, and testing sets. The proposed GABSLEEG model is found to significantly outperform conventional classifiers, including the support vector machine (SVM) classifier in (i-iii), and the random forest (RF) and the *k*-nearest neighbour (*k*-NN) classifiers in (i-ii). The GABSLEEG system consistently had a higher classification accuracy, sensitivity, and specificity. The average accuracy of the proposed system was 99.65%, on BCI Competition IV dataset 1 and 96.08% for BCI Competition III dataset IVa with the idle state included as a class, which was on a par with state-of-the-art SL and even deep learning approaches.

Keywords
brain-computer interface (BCI); motor imagery (MI) electroencephalography (EEG); sparse learning (SL); genetic algorithm (GA); channel selection.

1. Introduction

Electroencephalogram (EEG) based brain-computer interfaces (BCIs) have the potential to deeply impact healthcare and consumer industries due to their non-invasive and portable nature. BCIs based on imagined movement, known as motor imagery (MI) BCIs, hold potential in neurorehabilitation and prosthetics control systems. The onset of a MI task is characterized by a decrease in activity within the alpha and beta frequency bands, which are 7Hz-13Hz and 13Hz-30Hz, respectively [1].

Various learning systems have been developed for the classification of EEG using different features such as time [2], frequency [3], time-frequency [4], common spatial patterns (CSPs) [5] and angle-amplitude transformations [6]. These features can be used in conjunction with conventional machine learning classifiers such as support vector machine (SVM) [2][4][5], *k*-nearest neighbor (*k*-NN) [2][6], decision tree/random forest (RF) [2] and discriminative analysis [3]. Other classification pipelines include Riemannian manifolds [7] and deep learning [8][9], which have been shown to perform strongly in EEG classification.

Sparse learning (SL) approaches based on sparse representation have been used widely in signal processing [10][11] and have shown great potential for high accuracy classification of MI EEG data [12][13][14]. Using wavelet features and standard sparsity-based classification, Sreeja et al. [12] obtained an accuracy of 96.10% for a two-class MI classification problem and 91.92% for a four-class problem. The authors then added weights to the dictionary, with the weights being calculated based on dissimilarity information between the training data and the test data. This approach improved the accuracy to 97.98% for the two-class problem and 92.23% for the four-class problem. Instead of using sparsity-based classification, Taran and Bajaj [13] obtained a sparse signal representation using the tunable Q-factor wavelet transform (TQWT). Classification was carried out using a least-squares support vector machine (LS-SVM), and an accuracy of 96.89% was reported. In [14], an extreme learning machine (ELM) classifier was used for sparse classification based on CSP features and obtained an accuracy of 87.54%. Many recent SL systems do not incorporate channel selection, and either use all the EEG channels available in the evaluation dataset [14], or an arbitrarily chosen subset of the channels which may not be optimal [12][13].

[*] Corresponding author: jinchang.ren@ieee.org

Dictionary-based SL has been shown to excel at MI EEG classification [12][15][16] and due to this, a dictionary-based SL classifier was used in this study. In dictionary-based approaches, feature vectors extracted from the training dataset are used to construct a dictionary. A test feature vector is assigned a classification label by sparse encoding it over the dictionary and analyzing the reconstruction error. To construct the feature vectors, features are extracted from each of the EEG channels and are concatenated. In previous work, channels have been represented by wavelet features [12][15][16], for example the energy of the approximation and detail coefficients obtained using the discrete wavelet transform [12], wavelet energy [15][16] or frequency-domain bandpower after CSP filtering [15]. Since EEG data is closely related to changes in the alpha and beta bandpowers, but the exact frequency ranges where these changes occur can vary between subjects [1], in this paper time-domain bandpower features for the combined alpha and beta bands were used.

Channel selection is usually carried out to reduce computational processing demands and to improve classification accuracy [20][22][23][24]. In some applications both aims are achieved, whereas in others the aim of channel selection is to maintain the accuracy of a classifier whilst substantially reducing the number of EEG channels, leading to lower computational times on the test set [20]. Due to the strong classification performance of the dictionary-based SL systems, in this work, channel selection was applied with the main aim of reducing impact of the computational complexity of the encoding algorithm, whilst preserving or improving the classification performance. Both wrapper and filter techniques have been used for MI EEG channel selection [20][21], with filter techniques relying on analytical measures to select EEG channels, and wrapper methods are iterative techniques optimize the EEG channel subset based on the classification accuracy obtained for candidate subsets. Typically, wrapper techniques [22][24] obtain better performance than filter techniques, but at the cost of increased computational complexity [20]. Heuristic wrapper methods, such as genetic algorithms [24], typically produce solutions more efficiently and quicker than conventional wrapper techniques, but tend not to find the optimum solution [20]. However, they have been shown to be effective for MI EEG channel selection [24]. Since the SL classifier was already computationally complex, in this work, a genetic algorithm was chosen for channel selection since it provided a trade-off between the renouned performance of wrapper channel selection techniques and the reduced computational complexity of heuristic wrappers. Thus, the aim of this work was to merge the capabilities of metaheuristic search algorithms for channel selection with sparse learning-based MI EEG classification. Within this context, the effect of the tunable parameters of the metaheuristic algorithm on final classification accuracy was also investigated.

In this paper, we propose the genetic algorithm (GA)-based band power feature sparse learning and classification of EEG (GABSLEEG) system for MI BCI applications. The main contributions of the proposed approach can be highlighted as follows:

1) Band power features from each EEG channel are used as features for sparse learning and accurate classification of EEG in three classes, namely two MI states and the idle state. This is novel because the idle state has not been included in similar work, i.e. dictionary-based MI EEG sparse learning [12][15][16];
2) The GA is applied within the sparse learning framework for channel selection, where channel selection helps to improve the classification efficacy and efficiency on the test-set, an important indicator for real-time BCI applications;
3) The proposed GABSLEEG system has outperformed several state-of-the-art approaches, including conventional and deep learning methods. It also shows robustness with reduced training data and improved computational efficiency compared to support vector machine (SVM) classification.

The rest of this paper is organized as follows. Section 2 presents an overview of the proposed methodology. Section 3 describes the datasets and experimental settings. Section 4 summarises the results with discussion. Finally, Section 5 gives some concluding remarks and future perspectives.

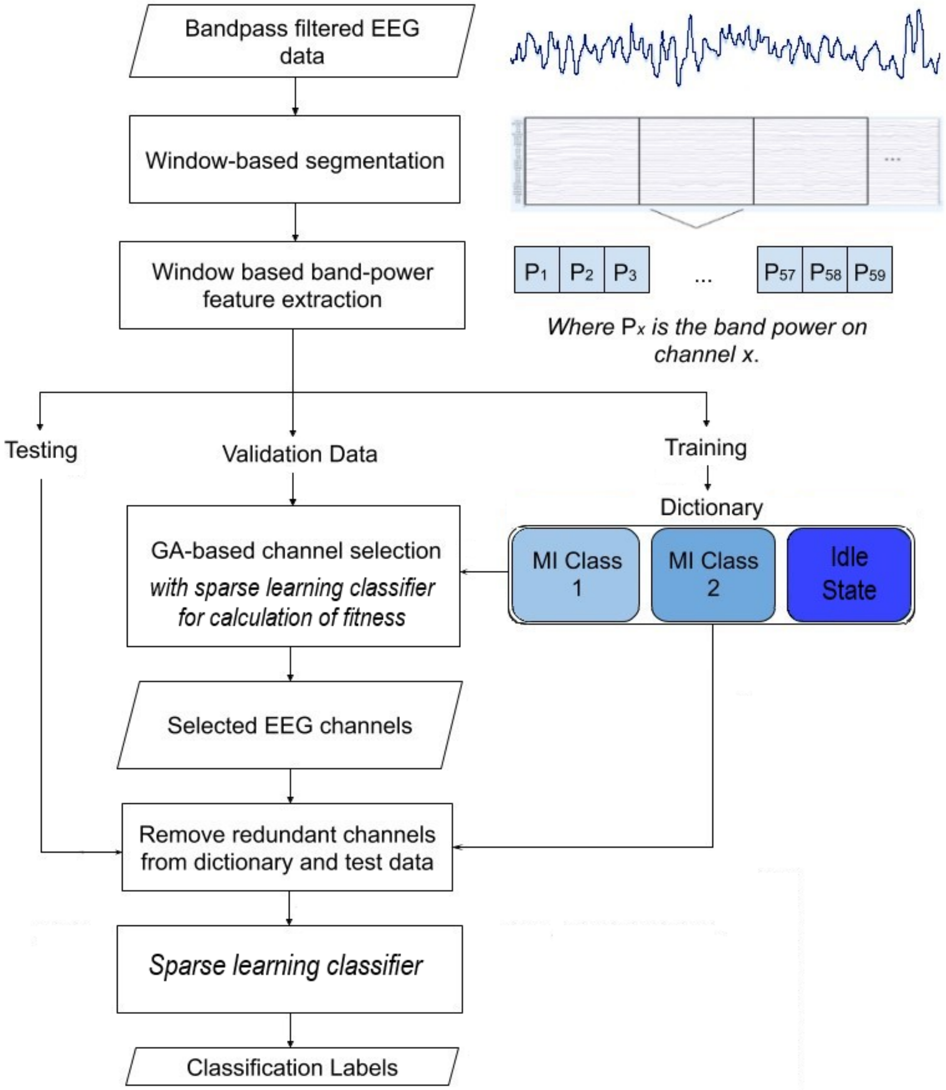
Bandpass filtered EEG data
Window-based segmentation
...
Window based band-power feature extraction
P1 P2 P3 ... P57 P58 P59
Where Px is the band power on channel x.
Testing
Validation Data
Training
Dictionary
GA-based channel selection with sparse learning classifier for calculation of fitness
MI Class 1
MI Class 2
Idle State
Selected EEG channels
Remove redundant channels from dictionary and test data
Sparse learning classifier
Classification Labels

Fig. 1. A diagram of the proposed GABSLEEG classification system. Bandpass filtered EEG signals are segmented, and feature vectors are extracted. The training data is used to construct a sparse dictionary, and the validation data is used for GA-based channel selection. Afterwards, a final sparse learning module assigns the test-set classification labels.

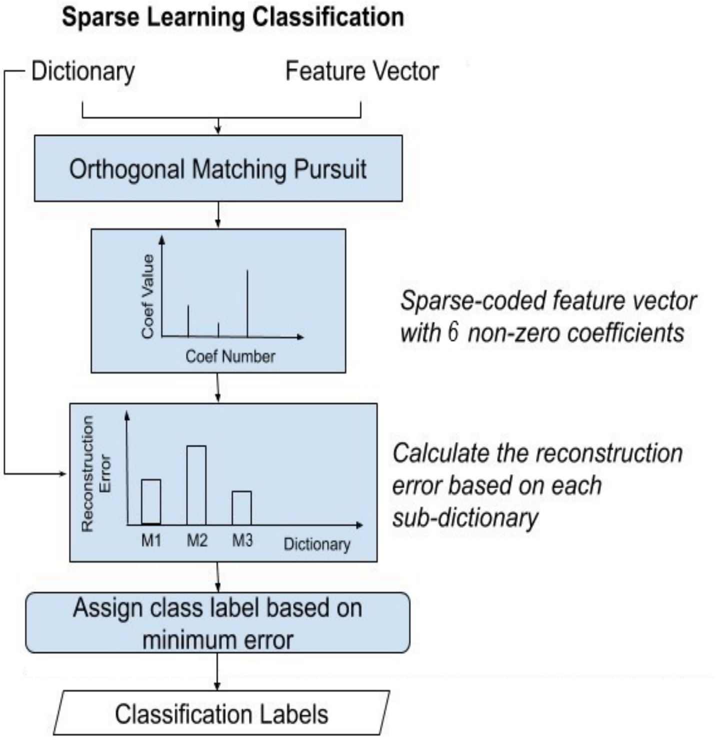
Sparse Learning Classification
Dictionary
Feature Vector
Orthogonal Matching Pursuit
Coef Value
Coef Number
Sparse-coded feature vector with 6 non-zero coefficients
Reconstruction Error
M1
M2
M3
Dictionary
Calculate the reconstruction error based on each sub-dictionary
Assign class label based on minimum error
Classification Labels

Fig. 2. The sparse learning classifier. Orthogonal matching pursuit is used to sparse encode the feature vector over the dictionary. The reconstruction error based on each sub-dictionary was then used to assign the final classification label.

2. The Proposed GABSLEEG Approach

Fig. 1 shows the flowchart of the proposed genetic algorithm-based band power feature sparse learning (GABSLEEG) approach. The EEG data is first bandpass filtered before extracting segment-based band power features from each window. The training data is used to construct the sparse dictionary, which is composed of three distinct sub-dictionaries, i.e. MI class 1, MI class 2 and the class of the idle state.

The dictionary and the validation data are fed into the GA-based channel selection module, so that a subset of EEG channels can be selected. The GA is a wrapper method and uses the classification accuracy obtained with a SL classifier, shown in Fig. 2, to select a channel subset. Once the subset is selected, the redundant data channels are removed from the dictionary and the test dataset. The final classification labels are obtained through a final sparse learning-based classifier block which takes as input the dictionary and test dataset, both containing only the selected EEG channels.

Consider the SL classification module in Fig. 2. In this block, the feature vector is sparsely encoded over the dictionary using orthogonal matching pursuit (OMP) to obtain the SR coefficients. The reconstruction errors from each sub-dictionary are calculated and the label is assigned to the class whose sub-dictionary has produced the lowest reconstruction error. This process is repeated for each of the feature vectors in the set. Subject-specific dictionaries were constructed for each subject using data from that subject, in accordance with a prevalent methodology in the literature for subject-specific modelling [9][12][13][19].

In this system, whole EEG trials are processed. This enables the different stages of a MI event to be captured within the dictionary [1] and has a precedent in the literature [9][12][13], although some studies choose to process only the central part of MI trials [14] because of the dynamic behavior at the start and end of MI trials [1].

The following three sub-sections 2.1-2.3 describe in detail the proposed GABSLEEG approach, including pre-processing, the SL module, and the GA-based channel selection, respectively.

2.1 Preprocessing and Feature Extraction

For the given EEG signals, the following preprocessing and feature extraction are applied before the SR was extracted. First, the raw EEG data was mean-centered and bandpass filtered between 7.5Hz and 30Hz using a 10th order Butterworth filter to extract the alpha and beta bands.

As illustrated in Fig. 1, filtered data was windowed, using windows of length t seconds. Within a window there are thus $T = t \times F_s$ samples per channel, where F_s is the sampling frequency (100Hz). Thus, each segment is of size $T \times M$, where M is the number of EEG channels. For experiments in this paper, t was set to 0.05.

For each window of data, the average power, p_j, in each of the EEG channels was calculated as follows: $p_j = \frac{1}{L} \sum_{i=1}^{T} x_{ji}^2$ where x_{ji} is the data on channel j at the i^{th} time sample. The power values are concatenated to form a $1 \times M$ feature vector for the windowed sample.

2.2 Sparse Learning (SL)

In the SL process, the training data was used to construct the dictionary, whose structure is shown in Fig. 1. The dictionary used consisted of three parts: one for MI class 1, one for MI class 2 and one for the idle state class. Each part of the dictionary was constructed by deriving the $1 \times M$ feature vectors for training samples in the respective class. It was ensured that the three sub-dictionaries were of the same length, L, which was determined by the class with the minimum number of training samples.

Classification was then carried out on the test set using the dictionary. During the testing, a feature vector from the test set was sparse encoded across the dictionary using OMP, with the representation being limited to 6 non-zero coefficients.

The sparse reconstruction of a feature vector is given by (1):

$$\hat{\boldsymbol{y}} = \boldsymbol{D}\boldsymbol{x} \tag{1}$$

where $\hat{\boldsymbol{y}}$ is the estimated feature vector, $\boldsymbol{D}$ is the dictionary and $\boldsymbol{x}$ is the coefficient vector. The OMP implementation works to build a support, S, of the signal. The support consists of the indices of those dictionary entries which, according to the OMP algorithm, best represent the signal. The size of the support is constrained by the level of sparsity chosen by the user, which in this work was set to 6.

In the sparse coefficient vector, only the dictionary entries named within the support set have non-zero coefficients. The support is built by reducing the residual error, e as follows [26]:

$$\boldsymbol{e} = \boldsymbol{y} - \sum_{i \in S} \boldsymbol{x}_i \boldsymbol{d}_i \tag{2}$$

where y is the original feature vector for the test sample, $\boldsymbol{d}_i$ is the dictionary atom and $\boldsymbol{x}_i$ is the coefficient associated with that atom. OMP is a greedy algorithm which builds the support by selecting the next support atom as the one which correlates the most with the residual error.

The actual values of the non-zero sparse coefficients, $\boldsymbol{x}$, are found by minimizing the sparse representation error in (3):

$$\boldsymbol{x} = \min_{\boldsymbol{x}_S} \|\boldsymbol{y} - \boldsymbol{D}_S \boldsymbol{x}_S\| \tag{3}$$

where $\boldsymbol{D}_S$ is the dictionary subset based on the support, and $\boldsymbol{x}_s$ is the set of non-zero coefficients.

The optimization problem in (3) can be solved using the least squares approach [26]:

$$\boldsymbol{x}_S = (\boldsymbol{D}_S^T \boldsymbol{D}_S)^{-1} \boldsymbol{D}_S^T \boldsymbol{y} \tag{4}$$

The sparse encoding occurred along the length of the dictionary, thus the coefficient vector had length $3L$ when including both the non-zero and zero valued coefficients. This approach, of encoding the test samples over a dictionary constructed from EEG training samples, with the length of the sparse code depending on the number of training samples, has previously been used in EEG SR [12][15][16]. The sparse encoding was separated into three parts to obtain the part associated with each sub-dictionary. Since the dictionary was constructed as in Fig. 1, this was done by dividing the coefficient vector into three parts, each of length L. Classification was carried out by calculating the reconstruction error based on each of the sub-dictionaries using (5):

$$e_j = \|\boldsymbol{y} - \hat{\boldsymbol{y}}\|_2^2 = \|\boldsymbol{y} - \boldsymbol{D}_j \boldsymbol{c}_j\|_2^2 \tag{5}$$

where e_j is the reconstruction error for the dictionary associated with class j, and $\hat{\boldsymbol{y}}$ is the estimated value of $\boldsymbol{y}$, based on the dictionary representation. $\hat{\boldsymbol{y}}$ is given by $\boldsymbol{D}_j\boldsymbol{c}_j$, where $\boldsymbol{D}_j$ is the sub-dictionary for class j and $\boldsymbol{c_j}$ are the coefficients associated with that sub-dictionary. $\|...\|_2$ denotes the Euclidean norm. The class of the sub-dictionary for which the lowest error was found was the class assigned to the test sample.

2.3 GA Channel Selection

In our proposed SR system, the GA was used for simplifying the dictionary. Essentially, it was used to perform channel selection from the 59 available EEG channels to form a smaller subset which retains similar accuracy. GAs aim to find the optimal local minimum but provide no guarantee of finding the global minimum of the optimization problem [27]. An exhaustive, sequential search of the solution space for EEG channel selection is not currently feasible due to the sheer number of channel combinations. For example, if one were to choose 30 channels from a selection of 59 EEG channels, there are 5.91 x 10^{16} possible combinations. Thus, GAs are suitable for the channel selection problem, and the combinatorial nature of the problem makes transcription into chromosomes straightforward.

A GA was used to find EEG channel subsets of length n. The channel selection process was carried out individually for each of the subjects. Pseudocode describing the whole GA process can be found in Algorithm 1. Prior to running the GA, the data for the subject was randomly divided into three sets: a training dataset, $\mathbf{X}_{\text{train}}$, comprising 80% of the data, a validation dataset, $\mathbf{X}_{\text{val}}$, comprising 10% of the data and a test dataset, $\mathbf{X}_{\text{test,}}$ comprising the remaining 10% of the dataset. The SR dictionary was generated using the training dataset, and the GA optimized with respect to the validation dataset. The accuracy of classification with the test dataset was used to evaluate the generalization capabilities of the final selected channel subset.

The chromosomes had the chosen channel length n, and consisted of the EEG channel numbers. An initial population of z individuals was generated using a random number generator in Python, and the population size remained constant. The fitness was the classification accuracy calculated on the validation dataset, using the SR classification approach previously detailed in part 2.2 of this section of the paper. Selection was carried out using the fitness-proportional roulette wheel approach [27], where fitter individuals were more likely to be selected. For a selected individual k, the associated probability p of selection is determined below, where $Fitness_j$ is the fitness of individual j:

$$p = \frac{Fitness_k}{\sum_{j=0}^{z-1} Fitness_j} \times 100\% \tag{6}$$

During each iteration, (z-5) parents were selected from the population for crossover. An alteration of two-point crossover [27] was used for the crossover process, where random locations were chosen within the parents, and

Algorithm 1: Genetic Algorithm Based Channel Selection

```
Inputs: n, the size of the EEG subset, z the size of the population, D the dictionary based on the training dataset $X_{train}$, $X_{val}$ and $X_{test}$
Initialization: stagnation = 0, tolerance = 0.000009, bestFitness = 0
population = a set of z randomly generated chromosomes of length n, generated from numbers in the range 0 to 59 without replacement
while stagnation < 3: # Monitor if the bestFitness has remained the same for three consecutive iterations
        for each chromosome in the population: # Calculate the population fitness
                carry out classification using (4) and (5) with D[population{current chromosome},:] and $X_{val}$ [population{current chromosome},:]
                calculate the accuracy and store in population_fitness
        end
        lastBestFitness = bestFitness # Update the best fitness and the best chromosome
        if max(population_fitness) > bestFitness:
                bestFitness = max(population_fitness)
                bestChromosome = population[argmax(population_fitness)]
        end
        if abs(lastBestFitness - bestFitness) < tolerance: # Monitor for convergence
                stagnation = stagnation +1
        cumulative_fitness = sum(population_fitness) # Carry out selection
    end
    Select (z-5) chromosomes to be used in crossover. Store in parents. Probability of selection, p is calculated as in (6)
    From the parent chromosomes generate the children as described in (7) and (8)
    In the population replace the (z-5) chromosomes with the lowest fitness with the children
    Mutation step: 20% probability that a random mutation in one chromosome will occur.
end
Calculate the test_set _accuracy for the final bestChromosome:
Carry out classification using (3) and (5) with D[population{bestChromosome},:] and $X_{test}$ [population{bestChromosome},:] and calculate the
accuracy
Output: bestChromosome and test_set_accuracy
END
```

the gene at that position and the next adjacent position were exchanged if the exchange did not result in a duplicate gene in either of the children. If duplication were to occur, crossover is not carried out, and another random location is chosen and crossover is again attempted, either until a suitable crossover was carried out, or 4 attempts had been made, at which point the parents were returned to the population. Only the 5 fittest individuals are guaranteed to not be replaced during the crossover process.

Given two parent chromosomes, $\boldsymbol{parent_1}$ and $\boldsymbol{parent_2}$, which produce two child chromosomes, $\boldsymbol{child_1}$ and $\boldsymbol{child_2}$, the crossover process is summarized as follows:

$$\boldsymbol{child}_1 = [\boldsymbol{parent}_1[0,\ldots,p_1], \boldsymbol{parent_2}[p_2,p_2+1],\ \boldsymbol{parent}_1[p_1{:}end]\,] \tag{7}$$

$$\boldsymbol{child_2} = [\boldsymbol{parent_2}[0,\ldots,p_2], \boldsymbol{parent_1}[p_1,p_1+1],\ \boldsymbol{parent_2}[p_2{:}end]\,] \tag{8}$$

where the parent and child variables are row vectors of length n; p_1 and p_2 ($p_1, p_2\ \neq\ n$) each represents a random location in $parent_1$ and $parent_2$, respectively.

During each iteration, there was a 20% chance that one of the chromosomes in the population would experience a random mutation within a gene, which involved one randomly selected individual having a randomly selected gene changed to a randomly selected value not already used in the chromosome.

When the best fitness remains the same for three consecutive iterations of the GA, the algorithm is exited, and the current best individual is the selected subset of EEG channels. The generalizability of the chosen subset could then be evaluated using the test dataset.

2.3.1 A comment on channel selection

SR encoding has often been used for band selection in hyperspectral imaging, where sparse coefficients are obtained for each band and then the locations of the non-zero coefficients are analyzed to select the most discriminative bands [21][22]. In this case, the hyperspectral bands could be analogous to the EEG channels. However, in the representation approach used here, the sparse encoding is carried out along the training samples stored in the dictionary, which has length $3L$ as previously mentioned. Thus, the non-zero coefficients are not associated with the EEG channels, but with the training samples. Thus, a GA was used to carry out channel selection, as opposed to the sparse representation coefficients.

2.4 Theoretical Time Complexity

This section presents the theoretical time complexity analysis of the SL classifier and GA channel selection modules. The SL classifier used the *sklearn* [25] implementation of the batch OMP algorithm [25]. The runtime complexity, T_{OMP} , of the batch OMP algorithm is [51]

$$T_{OMP} =\ O(n) +\ 3O(n^2) + O(n^3) \Rightarrow T_{OMP} = O(n^3)$$

For the time complexity analysis, the GA channel selection process can be broken into individual processes, namely: initialization, fitness calculation, selection, crossover and mutation, with the theoretical time complexities of T_I, T_F, T_S, T_C and T_M, respectively. The fitness calculation depends on the SL module, and thus $T_F = O(n) + O(n^3) =\ O(n^3)$. For the other processes, we have $T_I = O(n) + O(G)$, $T_S = O(1) + 2nlog(n) + 2O(n) \Rightarrow T_s = O(nlog(n))$, $T_C =\ O(3) + O(n) + O(n^2) \Rightarrow T_C =\ O(n^2)$, and $T_M = O(1) + O(n) + O(n^2) \Rightarrow T_M = O(n^2)$, where G is the total number of generations (iterations) of the GA. Therefore, the theoretical time complexity of the GA channel selection module, T_{GA}, becomes:

$$T_{GA} = T_I + G \times [T_F + T_S + T_C + T_M] \Rightarrow T_{GA} = O(n) + O(G) + G \times [O(nlog(n)) + 2O(n^2) + O(n^3))]$$

$$\therefore\ T_{GA} = G \times O(n^3)$$

3. Datasets, Experimental Settings and Hyperparameter Overview

This section begins with a brief overview of the datasets used and the classification systems used for benchmarking. Afterwards, the hyperparameter tuning process for the GABSLEEG system is discussed. A brief analysis of the impact of heuristic parameters in the GA on classification accuracy is conducted, followed by a discussion of how sparsity level impacts fitness within the GA. The tuning of the benchmarking classifiers is then summarized. The section closes with some comments on the evaluation methodology used.

The GABSLEEG algorithm, as well as the SVM, *k*-NN and RF classifiers were implemented in Python 3.7 [25]. The SVM, *k*-NN and RF classifiers were selected due to their established use in MI EEG classification [2][4][5][6] and were used to assess the effectiveness of the sparse learning classifier. Later in the paper, in section 4.3, the GABSLEEG system is compared to a deep learning system.

3.1 Datasets Description

The BCI competition IV, dataset 1, was used [28]. This dataset has two parts, the calibration dataset, and the evaluation dataset. The calibration dataset was used for hyperparameter tuning, whilst the evaluation dataset was used to obtain the final classification results, to assess the generalizability of the parameters derived from the calibration dataset. The datasets consist of data down-sampled from 1000Hz to 100Hz from four human subjects, labelled as 1a, 1b, 1f and 1g. The MI EEG data for two MI activities – from right hand, left hand, and foot - are available for each subject. For the foot class, subjects were free to decide which foot to imagine moving or could imagine both moving at the same time. The calibration dataset consists of 4s intervals of MI interspersed with 4s intervals of the idle state. The evaluation dataset aims to simulate an asynchronous BCI, consisting of intervals of MI between 1.5s and 8s long, interspersed by similar length intervals of the idle state. During some intervals, music or videos were playing to distract the subject [29]. In the case of the final evaluation results, the data is randomly divided into training (80%), validation (10%) and testing (10%) divisions. This approach of dividing data into sets has been used in the literature, particularly in systems which may have more involved processing times, such as CNNs [6][9]. The GA, being an iterative technique, also has an involved processing time and thus this testing approach was used.

Another dataset, BCI Competition III, dataset IVa [30] was used to validate the GABSLEEG approach presented, and to compare the results obtained with those in other papers. This dataset had data from five subjects labelled aa, al, av and aw, a down-sampled frequency of 100Hz and two MI EEG classes.

To assess how the GABSLEEG system and comparison systems operate with practical data, no effort was made to discard data which had artifacts.

3.2 Classification Approaches used for Benchmarking

Three standard MI classification techniques were used for benchmarking, namely: an SVM classifier with radial-basis function (RBF) kernel, a *k*-NN classifier, and a RF classifier, which all use nonlinear decision boundaries.

The benchmarking classifiers were trained using the same feature vectors derived from the training dataset that were used to construct the SL dictionary. Thus, the classifiers were provided with feature vectors consisting of M values, calculated over a 0.05s window of EEG data, with each value corresponding to the averaged power at an electrode within the window. The feature vectors in the training set were scaled to have zero mean and unit variance, and the same transformation was applied to the validation and testing sets.

For fair comparisons to the GABSLEEG system, GA channel selection was also carried out when using the benchmarking systems. The same algorithm and methodology described in Section 2.3 was used, with the SVM, k-NN or RF classifier replacing the SL classification module when the fitness was calculated.

3.3 Hyperparameter Tuning

This sub-section covers the hyperparameter tuning of the proposed GABSLEEG system and the benchmarking systems. Hyperparameter tuning was carried out on the calibration data in the BCI Competition IV dataset I. Hyperparameters providing the highest average accuracy calculated across all four subjects in the dataset were selected. For the GABSLEEG system, a brief analysis of the effect of different hyperparameters on performance

Table 1

Test-set accuracy of the proposed GABSLEEG system with different sizes of channel subsets obtained on the calibration data. The variance captures the variation in accuracy between different population sizes.

Channel Subset Size	5	10	15	20	25	30
Average Accuracy	61.42 ± (9.6 × 10^{-3})	95.29 ± (5.5 × 10^{-3})	97.28 ± (2.4 × 10^{-3})	97.93 ± (1.3 × 10^{-3})	98.28 ± (5.5 × 10^{-4})	98.44 ± (4.3 × 10^{-4})

is also carried out.

3.3.1 Hyperparameter Analysis for the GABSLEEG System

The GABSLEEG system has two modules: the SL module and the GA channel selection module. In the SL module, tunable hyperparameters were the window size and the number of non-zero coefficients in the sparse representation. In the GA module, the tunable hyperparameters considered were the number of EEG channels in the subset (size of the subset) and the initial population size of the GA.

The SL module was tuned first, using the classification module shown in Fig. 2. Performance for window sizes in the set {0.05,0.06,0.07,0.08,0.09,0.10,0.15,0.20} seconds and numbers of non-zero coefficients in the set {3,4,5,6,7} was analyzed. A peak 10-fold classification accuracy of 99.07% was obtained for a window size of 0.05 and 6 non-zero coefficients. We then repeated this analysis but extracted the separate alpha and beta band powers for each channel during feature vector construction, leading to feature vectors which were twice as long. A peak average 10-fold cross-validation accuracy of 98.97% was obtained using a window size of 0.05 and 5 non-zero coefficients. The proposed design, which uses the combined alpha-beta power to characterize each channel, gave a slightly better classification accuracy.

The GA module was then tuned. An analysis was carried out to select the best parameters. Channel subset sizes in the set {5,10,15,20,25,30} channels and initial population sizes in the set {10,15,20,25,30,35,40} chromosomes were considered. To evaluate each pairing, the calibration data was divided into 3 groups, with 80% being used for training, 10% being used as a validation set and 10% being used as a test set. The SL dictionary was generated using the training dataset, and the GA optimized the channel subsets with respect to the validation dataset. The classification accuracy obtained on the test subset was used to evaluate the generalization capabilities of the final selected channel subset and was used to rank the different parameter pairings.

The results for GA hyperparameter optimization are shown in Table 1, which records the average test set accuracy for each instance of channel subset size. The accuracy values were calculated by averaging the results across different sizes of the initial populations. The standard deviation was included with the accuracy to capture the variation across different initial population sizes.

The channel subset size had a substantial effect on the accuracy, although the results tend to stabilize for subsets of size 10 and greater. Regardless of the channel subset size, the standard deviation is of an order of 10^{-3}-10^{-4}, indicating relatively low variation between different initial population sizes. Increasing the channel subset size tended to decrease the standard deviation in the results. The results indicate that the size of the channel subset could have a substantially greater impact on accuracy performance than the initial population size, particularly when the size is below 10 channels.

The peak accuracy was obtained with a subset of 30 channels and an initial population of 35 chromosomes. This set-up obtained an accuracy on the test set of 98.49%, which was like the control test set accuracy of 98.62%, which was calculated using all 59 EEG channels. Thus, the effect of reducing the channels to 30 using the GA was of only 0.13%. In the rest of this analysis, 30 EEG channels were used for analysis. However, since there was only a discrepancy of approximately 3% in the average accuracy when using 10 EEG channels compared to 30 EEG channels, in the computational complexity analysis (section 4.1.3) the case when 10 EEG channels used is also analyzed, to observe how much computational time can be improved.

The final part of this hyperparameter analysis involves observing the effect of sparsity on the test set fitness of the GA. For the selected GA parameters, the number of non-zero coefficients in the OMP was varied between 6 and 500 and the test set fitness (i.e. accuracy) was recorded and shown in Fig. 3. Increasing sparsity tended to improve the classification accuracy, with the accuracy decreasing from over 98% accuracy when less than 25 non-zero coefficients were used, to stabilize around 95.75% when more than 200 coefficients were used. In

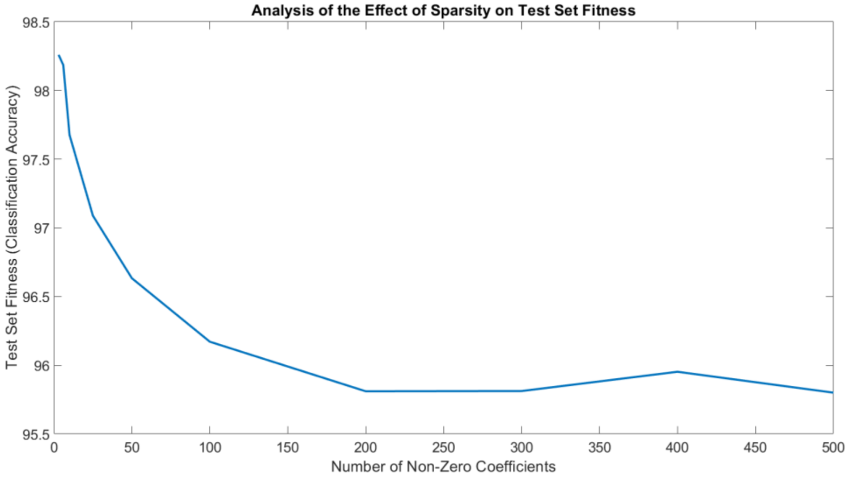

Fig. 3 Analyzing the changes in test set fitness for changes in the number of non-zero coefficients used in the OMP reconstruction. A smaller number of non-zero coefficients indicates increased sparsity.

further analysis in this paper, 6 non-zero coefficients were used.

3.3.2 Hyperparameter Tuning of the Benchmarking Classifiers

The benchmarking classifiers were tuned used Bayesian optimization, which enabled efficient exploration of the parameter spaces for each classifier. Parameters were chosen based on the 10-fold CV accuracy. For the k-NN classifier, values of k between 3 and 20 were considered, with a peak classification accuracy of 93.32% obtained for k=3. For the RF classifier, values for the maximum depth parameter, the minimum number of samples per split and the minimum leaf size in the ranges 10 to 500, 2 to 20 and 1 to 10, respectively, were considered. A peak accuracy of 86.27% was obtained for a maximum depth of 500 trees, a minimum number of samples per split of 2 and a minimum leaf size of 1. For the SVM classifier, C and ϒ were tuned for values of C from 0.1 to 100 and ϒ in the range 0.01 to 10. A peak accuracy of 84.99% was obtained for a C value of 100 and a ϒ value of 10.

3.4 Comments on Evaluation Methodology

Many MI EEG classification systems, including dictionary-based SL classifiers, tend to evaluate performance based only on the classification of MI EEG classes [12][14][15]. However, practical BCI systems should be able to distinguish not only between MI states, known as control states, but also the idle state, when the user is not performing any MI activity. Identification of the idle state is a widespread problem within EEG-based BCIs [19][21], and to assess the capabilities of the proposed algorithm, this paper carries out offline MI EEG classification using data from both control and idle states.

Another problem area for practical BCIs is the training phase. EEG data must be recorded from the user prior to use of the BCI and then the algorithm is trained [19], leading to a significant latency which grows with the amount of training data which needs to be recorded. This is replicated in offline systems by training models for individual subjects [9][12][13]. Research has shown that for other MI EEG classification pipelines, the training dataset can be reduced without a significant deterioration in performance, particularly using transfer learning [5], although this area requires further investigation since lengthy training phases detract from the practicality of EEG-based BCIs. Herein, the performance of the proposed system with reduced training data size is assessed.

4. Results and Discussion

This section begins with a comparison of the impact of GA channel selection on the performance of the GABSLEEG and other benchmarking systems. The performance of the GABSLEEG approach is then studied under the condition of a reduced training data size. This is followed by a complexity analysis comparing the training and testing latencies of the GABSLEEG system to those of the benchmarking systems. An analysis of the channels selected by the GA is then presented. Finally, the GABSLEEG approach is compared to related works in the literature.

Table 2

Comparison of Control Test Set Accuracy (%) for the proposed SLEEG System and three benchmarking systems.

Systems	1a	1b	1f	1g	Average
SLEEG	**99.89**	**99.70**	**99.86**	**99.92**	**99.84**
SVM	94.49	93.80	96.80	93.66	94.69
k-NN	98.05	94.89	99.54	97.55	97.51
RF	89.94	85.39	95.57	90.74	90.40

4.1 Performance Comparison

This section compares the performance of the GABSLEEG system with the benchmarking classification systems. The evaluation data from the BCI Competition IV dataset I was used for these tests.

4.1.1 Performance Comparison of the GABSLEEG System and Benchmarking Systems

The classification accuracies of the SL-based systems were compared to those of the benchmarking systems in Tables 2 and 3. In order to assess the impact of the channel selection module, two sets of accuracy results are presented: the ‘Control Test Set Accuracy’ results in Table 2 were obtained when using all 59 EEG channels, without GA channel selection, whilst the ‘Test Accuracy’ results in Table 3 were obtained when using the GA module for channel selection. Without the GA module, the GABSLEEG system is reduced to a sparse learning-EEG (SLEEG) system. The data was divided into training, validation, and test sets, as previously explained in Section 3 and the results in this section are for the test set. To ensure a fair comparison, the same data divisions were used for each of the classifiers. In both tables, the SL-based systems, SLEEG and GABSLEEG, had the highest classification accuracies for each individual subject, followed by the *k*-NN classifier, SVM and RF classifier.

Table 3

Comparison of Test Accuracy (%) for the proposed GABSLEEG system and three benchmarking systems.

Systems	1a	1b	1f	1g	Average
GABSLEEG	**99.52**	**99.62**	**99.77**	**99.69**	**99.65**
GA-SVM	96.89	94.73	98.54	94.13	96.07
GA-kNN	97.63	94.76	99.03	96.63	97.01
GA-RF	89.60	85.78	96.25	89.09	89.49

Table 4

Comparison of the Test-set Sensitivity and Specificity associated with each class (%), averaged across subjects, for the proposed GABSLEEG System and three benchmarking systems.

	Sensitivity			*Specificity*		
Systems	MI Class 1	MI Class 2	Idle Class	MI Class 1	MI Class 2	MI Class 3
GABSLEEG	99.92	99.86	99.44	99.81	99.77	99.96
GA-SVM	94.31	92.21	99.01	99.56	99.70	93.39
GA-kNN	99.13	98.49	95.14	98.13	98.39	99.23
GA-RF	93.58	91.83	86.20	94.32	94.32	96.00

Channel selection resulted in slightly decreased average classification accuracies for the SL, *k*-NN and RF classifiers, which were found to have decreased by 0.19%, 0.50% and 0.91%, respectively. For the SVM classifier, however, the classification accuracy was improved by 1.38% when adding the channel selection module. The mixed effect of channel selection agrees with results in the literature, where channel selection has been reported to improve the classification accuracy in some cases [23][31] but showing no effect or even a slight decrease of performance in other cases [32]. These results indicate that the impact of the channel selection is classifier dependent. Of the systems which experienced slightly decreased accuracy because of channel selection, the GABSLEEG system experienced the smallest decrease in performance, of only 0.19%. Despite this decrease, the GABSLEEG system remains the most accurate classifier compared to the benchmarking classifiers, including SVM. These results indicate the potential of the GABSLEEG system to benefit from the reduced data processing on the test-set associated with channel selection without a significant decrease in accuracy.

Table 4 compares the sensitivity and specificity associated with each class. The results for each class were obtained by averaging the results over the subjects. The GABSLEEG system had the best performance, followed by the GA-*k*NN classifier. The GA-RF classifier had the poorest performance, which was as expected from the accuracy results in Table 3. Furthermore, the GABSLEEG system had the most consistent performance across different classes.

In summary, the proposed GABSLEEG system has significantly outperformed the benchmarking systems in terms of accuracy, sensitivity, and specificity, exhibiting consistently better results across the four subjects for classification of MI EEG states and the idle state.

4.1.2 Analysis of Performance with Reduced Training Data

The training data was reduced from 100% to 20% in steps of 20% to check the effect of reduced training data on the classification accuracy. Fig. 4 shows how the performance in terms of accuracy, sensitivity, and specificity changes for the GABSLEEG and the benchmarking systems with the GA channel selection module.

The GABSLEEG system consistently outperformed the three benchmarking systems in terms of accuracy, sensitivity, and specificity for each of the reduced training data sizes. The RF classifier is consistently the poorest performing system. When just 20% of the training data was used, the accuracy of the GABSLEEG system was 90.18%, and it was the only system with an accuracy above 90% for this volume of training data. The GA-*k*NN and GA-SVM both had middling performance.

Since the time spent on the recording of training data from users detracts from the practicality of an EEG-based BCI, signal processing algorithms which can perform with high accuracy on relatively shorter training data are desirable. The consistently strong performance of the GABSLEEG system with reduced training data indicates that it could be a candidate for practical BCI implementations.

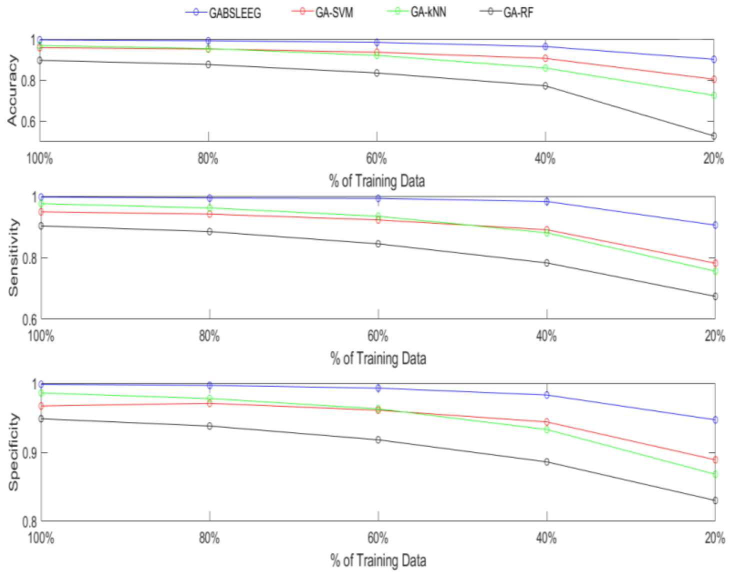

Fig. 4 Analyzing the changes in average accuracy, sensitivity, and specificity with decreased training data size. Results for the GABSLEEG system are plotted in blue, for the GA-SVM system are in red, for the GA-*k*NN system are plotted in green and for the GA-RF system are plotted in black. The x-axis labels denote the percentage of the training data used for training.

TABLE 5

Average and worst-case training and testing times per trial segment for the GABSLEEG and benchmarking systems.

Systems	Average Training Time/ms	Average Testing Time/ms	Worst case Training Time/ms	Worst Case Testing Time/ms
GABSLEEG	199.0 ± 10.7	13.1 ± 0.7	208.5	13.9
GA-SVM	557.0 ± 49.1	36.2 ± 2.0	601.8	39.2
GA-kNN	**4.0 ± 0.3**	**0.3 ± 0.0**	**4.0**	**0.3**
GA-RF	5.2 ± 1.8	**0.3 ± 0.0**	7.8	**0.3**

4.1.3 Computational Complexity Analysis

A computational complexity analysis was carried out on the GABLEEG, GA-SVM, GA-*k*NN and GA-RF systems. All tests were run on a Lenovo™ ideapad 330 laptop with 64-bit Windows 10 operating system, and an Intel® Core™ i5- 8300H, 2.30GHz CPU. Unnecessary background processes were suspended, and the laptop was fully charged during all tests.

In the case of each system, the mean training and testing times per trial segment were calculated for each subject by recording the total time taken for training or testing and dividing it by the number of segments processed. The mean training and testing times per trial segment, averaged across the subjects, were recorded in Table 5. The worst-case training and testing times, which were the maximum recorded times per segment, were also included in the table. The lowest training and testing times were highlighted in bold. Comparing the worst-case results to the average results, the GA-*k*NN classifier experienced no difference, whilst the GA-RF classifier only experienced a slight increase in the training time. The GA-SVM and GABSLEEG systems exhibited the largest differences between average and worst-case training, although the difference was larger for the SVM classifier. Furthermore, the worst-case testing time for the GABSLEEG system was less than 1ms longer than the average testing time.

Short training times are important to reduce the latency between training and use of a BCI. The GABSLEEG had the second-longest training time, with the GA-SVM system having the longest training time. The GA-*k*NN system had a much shorter training time.

Short test times are important as they determine the latency the user would experience during 'real-time' use of the BCI. The testing time results are like those for the training time, with the GABSLEEG system performing poorer than the GA- *k*NN and GA-RF systems. However, the testing latency of the GABSLEEG system may be suitable for a real-time prosthetics control system or to control a graphical user interface since humans can perceive only visual latencies above 13ms [33]. The GA-SVM system, however, had a latency of 36.2ms which may be unacceptable for a 'real-time' system.

These results indicate that the GABSLEEG system has potential for use in a real-time system but has room for improvement in terms of the computational complexity when compared to widely used classifiers such as *k*-NN and RF-based systems. These can be easily improved if the systems are implemented with a more powerful CPU or programmed onto an FPGA as opposed to using Python on a laptop.

The core aim of introducing the GA channel selection module was to improve the computational test times whilst preserving accuracy. When using the SLEEG classifier with all 59 EEG channels, the average test time per segment was 32.39ms, which was over twice the 13.1ms time for the GABSLEEG system. With a latency significantly above 13ms, the SLEEG system may not be suitable for visual interfaces. Thus, introducing the GA channel selection module resulted in a decrease of less than 0.5% in classification accuracy, but substantially improved the test-time latency. This improvement came at the cost, however, of the training interval of the GA, whereas the original SLEEG classifier had no classifier training latency beyond feature extraction and dictionary assembly.

Reducing the EEG channel subset size would reduce the training and testing times of the GABSLEEG system further. Recall from the initial analysis in Table 1 that the average accuracy on the calibration dataset when using 10 EEG channels was 95.29%, which was comparable to the 98.44% accuracy obtained when using 30 EEG channels. When using a channel subset size of 10 channels with the evaluation dataset, an average classification accuracy of 96.12% was obtained, which was 3.54% less than when 30 channels were used. However, the average training time when using 10 EEG channels was 70.3ms, whilst the average testing time was 6.6ms, which was a substantial improvement when compared to the results in Table 5, which used 30 EEG channels. These results illustrate the trade-off between accuracy and computational time which exists within the GABSLEEG classifier.

4.2 Channel Analysis

The aim of this supplementary analysis is to discuss the channels most frequently selected by the GA channel selection approach. The graphic Fig. 5 shows how frequently each of the 59 EEG channels were selected by the GA channel selection module for the results in Section 4.1.1, Table 3. The fraction of times a channel is selected was calculated across all 4 subjects and across all 4 classification approaches (GABSLEEG, GA-SVM, GA-*k*NN and GA-RF). This implies that it is calculated over 16 different optimized channel subsets. Since the GA introduces elements of stochasticity both during the random initialization of the algorithm and during the mutation step, channels which occurred less than 50% of the time (corresponding to 8/16 subsets) may have been included in a subset randomly, or possibly due to that channel being specifically important to a certain subject. Since this discussion is focused on analyzing the most selected channels across the subjects and classifiers, those occurring in more than 50% of subsets, only, were considered. The channel montage consisted of 11 different scalp regions, namely: central (C), central-central-parietal (CCP), central-frontal-central (CFC), central-parietal (CP), frontal-central (FC), parietal (P), frontal (F), parietal-occipital (PO), occipital (O), temporal (T) and anterior-frontal (AF).

The results in Fig. 5 indicate a strong presence of electrodes from central-associated EEG channel groups, namely C, CP, CCP and CFC groups. In fact, 58.33% of the most frequently selected channels were from central-associated groups. These results indicate that central-associated channels tended to be important in the selection process. This finding agrees with previous work in channel selection for MI EEG classification without the idle state being included suggest that central EEG channels are the most discriminative [1][22][23].

However, electrodes from outside the central regions also feature prominently within the selected channel subsets. Other works have reported some electrodes outside of the central region being included within the channel subsets for MI EEG classification for [22][23][24], although central channels generally tend to be dominant [22] [23]. The inclusion of a variety of channels outside the central region across the subjects observed in this experiment may suggest that for problems involving the idle state class, electrodes outside of the central regions may capture valuable information. The literature supports this concept [34] [35] [36] [37]. For example, significant activity within the alpha frequency band originating in the O region, known as the Posterior Dominant Rhythm, has been observed during the idle state [34], and the SR representation was built on band power features within the alpha and beta bands. Furthermore, the frontal region has been associated with planning or control of voluntary activity [35], whilst the parietal region has been associated with motivated attention [36], and thus these regions may be important in distinguishing between control and idle states. Finally, cues for whether to perform motor imagery or the idle state were provided using an audial stimulus [29], with audial evoked potentials occurring within the T lobe [37]. This may be related to the occurrence of T channels within in the subsets.

Although the GA is a prime candidate for EEG channel selection, the discussion of the channels selected via the GA must be carried out with caution. Since the candidate solutions within the initial population of the GA were generated randomly, and since this initialization step creates the starting point for the local search, the optimized result depends in part on this random initialization. Whilst this analysis suggests that for idle state classification electrodes outside of the central region may capture discriminative data, a more rigorous analysis would be required to establish whether electrodes or groups of electrodes contribute significantly to the MI BCI accuracy when distinguishing between MI and the idle state.

4.3 Comparison to other state-of-the-art approaches

The performance of the GABSLEEG classification system was then validated using on the BCI competition III dataset IVa. This dataset had 118-channel EEG data, however the GABSLEEG system parameters remained the same, with the genetic algorithm selecting an optimal subset of 30 EEG channels.

The results are shown in Table 6 and compared with other state-of-the-art classification approaches tested on the same dataset. To facilitate direct comparison, classification was carried out on the two MI classes within the dataset and the idle state was not included, as is the case for the other papers in Table 6. ‘AA’ stands for ‘Average Accuracy’. The comparison included other SL-based systems, as well as conventional and deep learning-based approaches. Systems which used channel selection were also included. As can be seen, the proposed approach has produced strong results, and has outperformed several SL models, conventional approaches and two deep learning methods. Although GABSLEEG is slightly outperformed by the deep learning model in [9], the 30 channels used in our model is significantly less than the 118 channels available in [9].

It should be noted that in this work the size of the EEG channel subset was fixed, and the optimal channels for each subject were identified via the GA. This contrasts with the channel selection methods within Table 6, in which the size of the EEG channel subset was optimized for each subject individually. Two of the channel selection systems [23][24] also had average subset sizes which were smaller than the 30-channel subset used within this paper. The strong performance of this the GABSLEEG system indicates that fixing the channel subset size across subjects is a promising alternative approach.

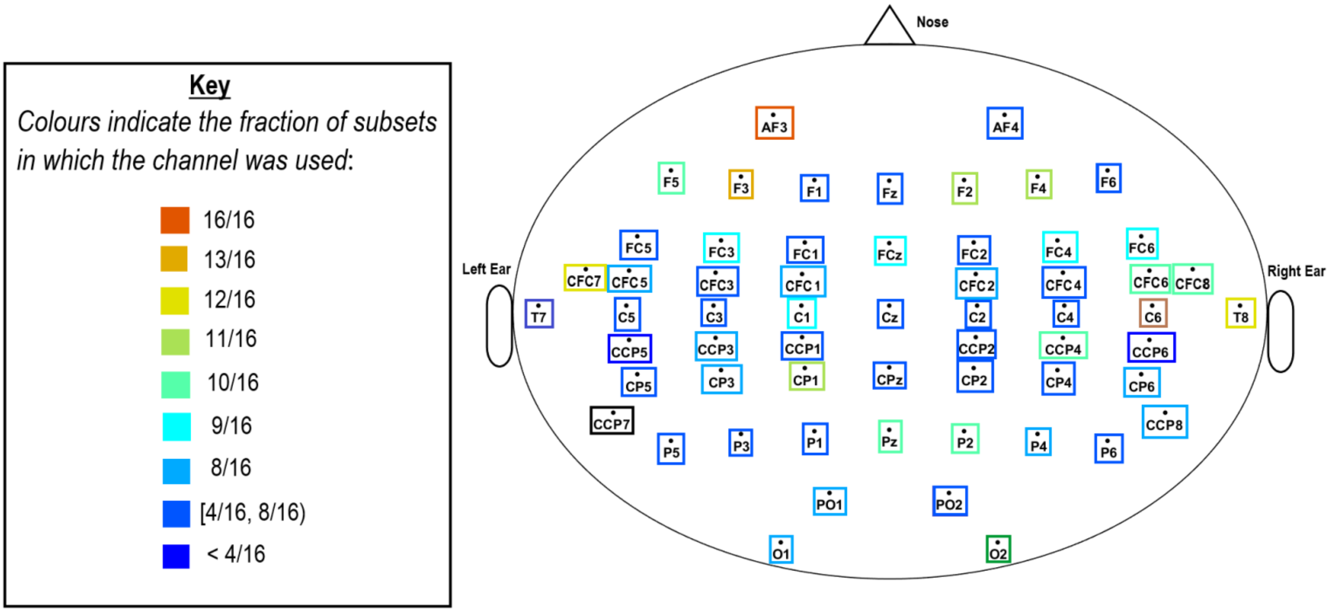

Fig. 5. The electrode map of the montage used in this paper, with the colours denoting the frequency of selection of each electrode. The fractions were calculated across all 4 subjects and across four the different classification approaches - GABSLEEG, GA-SVM, GA-*k*NN and GA-RF. Thus there were a total of 16 subset selections.

TABLE 7

Comparison of the classification accuracies of the GABSLEEG system to state-of-the-art implementations. Results were obtained for BCI Competition IV dataset I (columns 3 and 4) and BCI Competition III dataset IVa (columns 5 and 6). Results for different window segmentation sizes are shown; as well as recording results using a 50ms window as in Sections 4.1 and 4.2 of this paper, window sizes of 0.5s and 2s were also used when comparing to the implementation of Sreeja et al. [12] and EEGNet [46], since these were the window sizes used in the reported worked.

		Dataset 1		**Dataset IVa**	
Comparison System	Window Size	GABSLEEG	Comparison System	GABSLEEG	Comparison System
Sreeja et al. [12]	0.5s	**99.69%**	97.95%	**98.11%**	92.11%
Sreeja et al. [12]	50ms	**99.65%**	98.01%	**96.08%**	90.14%
Fisher Score Channel Selection + SL [45]	50ms	**99.65%**	99.54%	96.08%	**96.17%**
EEGNet [46]	2s	**78.33%**	77.81%	**87.43%**	83.75%

Comparison to the literature is carefully designed for consistency. For a direct comparison, three existing systems were compared, following the methodology used in this paper for training, validation, and testing. The three systems used here include a SL-based system [12], a channel selection approach [45], and a deep learning classifier [46]. The comparison was carried out for three class classification problems involving 2 MI states and the idle state. Two datasets, the BCI Competition IV dataset I [28] and the BCI competition III dataset IVa [30] were used and the average classification accuracies, calculated across subjects, were used for comparison.

The SL-based system [12], used 30 arbitrarily chosen EEG channels to generate the results, and a window size of 0.5s was used to build the dictionary. For consistency, results were generated using a window size of 0.5s as in the original implementation in [12] and a window size of 50ms as in the GABSLEEG system.

Channel selection based on the Fisher score is widely used in the literature, particularly for benchmarking novel channel selection approaches [44][45]. To assess the effectiveness of our proposed GA-based GABSLEEG system, GA channel selection was replaced with Fisher score channel selection, based on the approach in [47], which caters for multi-class classification. Since Fisher score channel selection does not require an explicit validation set, the training and validation sets were concatenated to obtain the Fisher score. The Fisher score, F_h, for channel h is determined by [47]:

TABLE 6

Comparing the results obtained within this paper and state-of-the-art SR, conventional and deep learning methods.

Papers	Approaches	Features	Classifiers	Channels	AA
GABSLEEG	GA-SL	Band power features	Dictionary-based	30	98.74
Sreeja et al. [12]	SL	Discrete wavelet transform features with weighted representation	Dictionary-based	30[1]	97.98
Taran & Bajaj [13]		TQWT features	LS-SVM	5[1]	96.89
She et al. [14]		CSP, Fisher discriminant structured dictionary	ELM	118	80.68
Meng et al. [42]	Conventional Approaches	Spatial and Spectral Features with maximized mutual information	SVM-RBF	118	90.7
Taran et al. [38]		Analytic intrinsic mode function features	LS-SVM	10[1]	97.56
Kevric & Subasi [39]		Multiscale principal component analysis de-noising, wavelet packet decomposition features	*k*-NN	3[1]	94.50
Ortiz-Echeverri et al. [40]	Deep Learning	Blind source separation, continuous wavelet features	CNN	18[1]	94.66
Kumar et al. [41]		CSP features and LDA feature scoring	Autoencoders & softmax	118	90.70
Chaudhary et al. [9]		Time-frequency representation	CNN+ transfer learning	118	**99.35**
He et al. [24]	Automated Channel Selection	CSP with Rayleigh coefficient maximization based GA channel selection	Fisher's LDA	15.5[2]	88.20
Qiu et al. [23]		CSP and selected floating forward channel selection	SVM-RBF	30.8[2]	83.30
Park and Chung [43]		Filter-bank CSP and correlation coefficient channel selection	SVM	8.2[2]	88.62

[1] Channels selected arbitrarily, not with channel selection.

[2] Subject-specific channel subset sizes were used, results show the average subset length.

$$F_h = \frac{\sum_{k=1}^{3}\left(m_{k,h} - m_{h_{total}}\right)^2}{\sum_{k=1}^{3} v_{k,h}} \tag{9}$$

where $m_{k,h}$ and $v_{k,h}$ denote respectively the mean and variance of the features extracted from channel h for class k $(k=1,2,3)$. $m_{h_{total}}$ is the mean of all the features associated with channel h. Once the Fisher score for each channel was obtained, the first 30 channels with the highest Fisher scores were selected for constructing the final dictionary and evaluating the performance.

As a deep learning-based system, EEGNet is a landmark CNN classifier used for MI EEG classification [46], where the input data is segmented into windows of 2s. Due to the structure of the CNN layers, 50ms windows were too small to be processed, thus the comparison between the GABSLEEG system and EEGNet was carried out using the window size of 2s. EEGNet uses all the available EEG channels within the dataset, with no channel selection carried out. The supplementary code provided by the authors was used.

The results for both the datasets are shown in Table 7. In general, the results for Dataset IVa were lower for those obtained with Dataset I, indicating the importance of testing systems on different datasets. In general, the GABSLEEG outperformed the sparse learning system presented by Sreeja et al. [12] regardless of the window size. The Fisher score channel selection and SL implementation also outperformed [12] for a window size of 50ms. These results indicate that channel selection can be an effective method of improving classification of sparse-representation systems. The GABSLEEG system performed on a par with the Fisher score channel selection and SL classification approach and outperformed the EEGNet classifier on both datasets. These results confirm that the GABSLEEG system can perform well when compared to the state-of-the-art.

5. Conclusion

This paper presented a novel GABSLEEG pipeline for MI-EEG classification which combines band power features sparse representation-based classification with genetic algorithm channel selection under some restraints which may be found in practical context. The GA module was effective in preserving the classification accuracy and using a subset of EEG channels substantially improved computational time. The GABSLEEG system consistently outperformed SVM, *k*-NN and RF classifiers in terms of classification accuracy, sensitivity, and specificity. It was also more robust to changes in training data size than the benchmarking systems. The main contributions can be summarized as follows: i) the GABSLEEG classification system had an accuracy of 99.67% when classifying MI EEG data and idle state data; ii) the GABSLEEG system was robust to changes in training data size, with classification accuracy remaining consistently above 90%, even as the training data size was reduced to just 20%; iii) the GA module was effective in preserving classification accuracy when using a reduced EEG channel subset. Decreasing the number of EEG channels from 59 channels to 30 channels, resulted in an average decrease in test-set accuracy of just 0.1%. This reduction in the number of EEG channels led to improved training times.

Future work could focus on testing GABSLEEG systems using a cross-validation approach which does not shuffle the data prior to division, to test robustness to changes within the multichannel EEG time series. Furthermore, since EEG can experience drift and other changes with time such as perturbations due to electrode movements, an investigation into how often the hyperparameters would need to be retuned, or channels reselected would be useful. Since tuning the GA channel selection module is computationally expensive, grouping Bayesian optimization [48] could be applied to speed up the hyperparameter tuning. Other metaheuristic algorithms such as the firefly algorithm or differential evolution could also be applied to assess whether they are more computationally efficient to tune than the GA for the channel selection problem. Furthermore, a key contributor to computational time was the OMP sparse learning module. In this work, a traditional CPU-based implementation of the OMP algorithm was used, however current research into GPU or FPGA-based implementations of OMP [49][50] have shown improved computational times. In future, these more efficient implementations could be applied to EEG classification systems based on sparse learning.

Within the wider sphere of offline analysis for practical BCIs, future work can focus on the role signal decomposition could play within the pre-processing stage to reduce the effects of noise and drift within practical systems, as well move on from the assessment of data from healthy subjects to subjects exhibiting pathologies, a necessary step if EEG-based BCIs are to be used for neurorehabilitation or diagnostics in the future. Learning systems will be an essential tool for future BCI research. The core of this would be adaptive systems capable of being trained on a user with minimal electrodes, and still perform strongly when the user returns on another day despite a different mental state and slightly different EEG cap position. Also, deep learning systems which depart from subject-specific training and tackle cross-subject training are also a promising frontier. The challenges of small datasets available for EEG analysis have opened avenues for research into EEG dataset augmentation for deep learning through generative adversarial networks and transfer learning.

Funding

This work was supported in part by the Dazhi Scholarship of the Guangdong Polytechnic Normal University (GPNU), Innovation Team Project of the Education Department of Guangdong Province (2017KCXTD021), the National Natural Science Foundation of China (62072122), and the University of Strathclyde Scholarship.

References

[1] B. Graimann, B. Allison, and G. Pfurtscheller, “Brain–Computer Interfaces: A Gentle Introduction,” in Brain-Computer Interfaces, Springer, 2009, https://doi.org/10.1007/978-3-642-02091-9_1.

[2] P. Batres-Mendoza, C.R. Montoro-Sanjose, E.I. Guerra-Hernandez, D.L. Almanza-Ojeda, H. Rostro-Gonzalez, R.J. Romero-Troncoso and M.A. Ibarra-Manzano, “Quaternion-based signal analysis for motor imagery classification from electroencephalographic signals,” in Sensors, vol. 16, no. 3, p. 336, 2016, https://doi.org/10.3390/s16030336.

[3] A. Liu, K. Chen, Q. Liu, Q. Ai, Y. Xie, and A. Chen, “Feature selection for motor imagery EEG classification based on firefly algorithm and learning automata,” in Sensors, vol. 17, no. 11, p.2576, https://doi.org/10.3390/s17112576.

[4] M. T. Sadiq, X. Yu, Z. Yuan, Z. Fan, A. U. Rehman, G. Li. and G. Xiao, “Motor imagery EEG signals classification based on mode amplitude and frequency components using empirical wavelet transform,” IEEE Access, vol. 7, pp.127678-127692, https://doi.org/10.1109/ACCESS.2019.2939623.

[5] A. M. Azab, H. Ahmadi, L. Mihaylova, and M. Arvaneh, “Dynamic time warping-based transfer learning for improving common spatial patterns in brain-computer interface,” J. Neural Eng., vol. 17, no. 1, 2020, https://doi.org/10.1088/1741-2552/ab64a0.

[6] B. H. Yilmaz, C. M. Yilmaz, and C. Kose, “Diversity in a signal-to-image transformation approach for EEG-based motor imagery task classification,” Med. Biol. Eng. Comput., vol. 58, no. 2, pp. 443–459, 2020, https://doi.org/10.1007/s11517-019-02075-x.

[7] F. Tang, M. Fan and P. Tiňo, "Generalized Learning Riemannian Space Quantization: A Case Study on Riemannian Manifold of SPD Matrices," in IEEE Transactions on Neural Networks and Learning Systems, https://doi.org/10.1109/TNNLS.2020.2978514.

[8] H. Zhao, Q. Zheng, K. Ma, H. Li and Y. Zheng, "Deep Representation-Based Domain Adaptation for Nonstationary EEG Classification," in IEEE Transactions on Neural Networks and Learning Systems, https://doi.org/10.1109/TNNLS.2020.3010780.

[9] S. Chaudhary, S. Taran, V. Bajaj and A. Sengur, "Convolutional Neural Network Based Approach Towards Motor Imagery Tasks EEG Signals Classification," in IEEE Sensors Journal, vol. 19, no. 12, pp. 4494-4500, 15 June15, 2019, https://doi.org/10.1109/JSEN.2019.2899645.

[10] H. Sun, J. Ren, H. Zhao, Y. Yan, J. Zabalza and S. Marshall, “Superpixel based feature specific sparse representation for spectral-spatial classification of hyperspectral images,” in Remote Sensing, vol. 11, no. 5, 536: 1-17, 2019, https://doi.org/10.3390/rs11050536.

[11] F. Qi, Y. Li and W. Wu, "RSTFC: A Novel Algorithm for Spatio-Temporal Filtering and Classification of Single-Trial EEG," in IEEE Transactions on Neural Networks and Learning Systems, vol. 26, no. 12, pp. 3070-3082, Dec. 2015, https://doi.org/10.1109/TNNLS.2015.2402694.

[12] S. R. Sreeja, Himanshu, and D. Samanta, “Distance-based weighted sparse representation to classify motor imagery EEG signals for BCI applications,” Multimed. Tools Appl., 2020, https://doi.org/10.1007/s11042-019-08602-0.

[13] S. Taran and V. Bajaj, “Motor imagery tasks-based EEG signals classification using tunable-Q wavelet transform,” Neural Comput. Appl., vol. 31, no. 11, pp. 6925–6932, 2019, https://doi.org/10.1007/s00521-018-3531-0.

[14] Q. She, K. Chen, Y. Ma, T. Nguyen, and Y. Zhang, “Sparse representation-based extreme learning machine for motor imagery EEG classification,” Comput. Intell. Neurosci., vol. 2018, 2018, https://doi.org/10.1155/2018/9593682.

[15] S.R. Sreeja, and D. Samanta, “Classification of multiclass motor imagery EEG signal using sparsity approach,” Neurocomputing, vol. 368, pp.133-145, 2019.

[16] S.R. Sreeja, J. Rabha, D. Samanta, P. Mitra and M. Sarma, “Classification of motor imagery based EEG signals using sparsity approach,” International Conference on Intelligent Human Computer Interaction, pp. 47-59, 2017, Springer, Cham.

[17] X. Mao et al., “Progress in EEG-based brain robot interaction systems,” Comput. Intell. Neurosci., vol. 2017, 2017, https://doi.org/10.1155/2017/1742862.

[18] R. Zerafa, T. Camilleri, O. Falzon and K.P. Camilleri, “A real-time ssvep-based brain-computer interface music player application,” presented at the XIV Mediterranean Conf. on Medical and Biological Engineering and Computing 2016, pp. 173-178, Springer, Cham.

[19] F. Lotte et al., “A review of classification algorithms for EEG-based brain-computer interfaces: A 10-year update,” J. Neural Eng., vol. 15, no. 3, 2018.

[20] M. Z. Baig, N. Aslam, and H. P. H. Shum, “Filtering techniques for channel selection in motor imagery EEG applications: a survey,” Artif. Intell. Rev., 2019, https://doi.org/10.1007/s10462-019-09694-8.

[21] N. Padfield, et al, “EEG-based brain-computer interfaces using motor-imagery: Techniques and challenges,” Sensors (Switzerland), vol. 19, no. 6, pp. 1–34, 2019, https://doi.org/10.3390/s19061423.

[22] M. Li, J. Ma, and S. Jia, “Optimal combination of channels selection based on common spatial pattern algorithm,” 2011 IEEE Int. Conf. Mechatronics Autom. ICMA 2011, pp. 295–300, 2011, https://doi.org/10.1109/ICMA.2011.5985673.

[23] Z. Qiu, et al, “Improved SFFS method for channel selection in motor imagery based BCI,” Neurocomputing, vol. 207, pp. 519–527, 2016, https://doi.org/10.1016/j.neucom.2016.05.035.

[24] L. He, Y. Hu, Y. Li, and D. Li, “Channel selection by Rayleigh coefficient maximization based genetic algorithm for classifying single-trial motor imagery EEG,” Neurocomputing, vol. 121, pp. 423–433, 2013, https://doi.org/10.1016/j.neucom.2013.05.005.

[25] Scikit-learn: Machine Learning in Python, Pedregosa et al., JMLR 12, pp. 2825-2830, 2011.

[26] B. Dumitrescu and P. Irofti, “Sparse Representations,” in Dictionary Learning Algorithms and Applications, Cham: Springer, 2018.

[27] O. Kramer, “Genetic Algorithms,” in Genetic Algorithm Essentials, 2017, pp. 11–19, https://doi.org/10.1007/978-3-319-52156-5.

[28] B. Blankertz, G. Dornhege, M. Krauledat, K. R. Müller, and G. Curio, “The non-invasive Berlin Brain-Computer Interface: Fast acquisition of effective performance in untrained subjects,” Neuroimage, vol. 37, no. 2, pp. 539–550, 2007, https://doi.org/10.1016/j.neuroimage.2007.01.051.

[29] M. Tangermann et al., “Review of the BCI competition IV,” Front. Neurosci., vol. 6, no. JULY, pp. 1–31, 2012, https://doi.org/10.3389/fnins.2012.00055.

[30] G. Dornhege, B. Blankertz, G. Curio, and K. R. Müller, “Boosting bit rates in noninvasive EEG single-trial classifications by feature combination and multiclass paradigms,” IEEE Trans. Biomed. Eng., vol. 51, no. 6, pp. 993–1002, 2004, https://doi.org/10.1109/TBME.2004.827088.

[31] W-K Tam, Z. Ke, K-Y Tong, “Performance of common spatial pattern under a smaller set of EEG electrodes in brain–computer interface on chronic stroke patients: a multi-session dataset study,” presented at the 2011 Ann. Int. Conf. IEEE Engineering in Medicine and Biology Society, IEEE, pp 6344–6347.

[32] K. Ansari-Asl, et al, "A channel selection method for EEG classification in emotion assessment based on synchronization likelihood," 2007 15th Euro. Signal Proc. Conf., Poznan, 2007, pp. 1241-1245.

[33] M. C. Potter, et al, “Detecting meaning in RSVP at 13 ms per picture,” Attention, Perception, Psychophys., vol. 76, no. 2, pp. 270–279, 2014, https://doi.org/10.3758/s13414-013-0605-z

[34] J. W. Britton et al., “The Posterior Dominant Rhythm,” in Electroencephalography (EEG): An Introductory Text and Atlas of Normal and Abnormal Findings in Adults, Children, and Infants [Internet], Chicago: American Epilepsy Society, 2016.

[35] S. D. Armstrong, M. V. Sale, and R. Cunnington, “Neural oscillations and the initiation of voluntary movement,” Front. Psychol., vol. 9, no. DEC, pp. 1–16, 2018, https://doi.org/10.3389/fpsyg.2018.02509.

[36] D. J. L. G. Schutter, P. Putman, E. Hermans, and J. Van Honk, “Parietal electroencephalogram beta asymmetry and selective attention to angry facial expressions in healthy human subjects,” Neurosci. Lett., vol. 314, no. 1–2, pp. 13–16, 2001, https://doi.org/10.1016/s0304-3940(01)02246-7.

[37] P. Kidmose, D. Looney, M. Ungstrup, M. L. Rank and D. P. Mandic, “A Study of Evoked Potentials from Ear-EEG,” IEEE Trans. Biomed. Eng., vol. 60, no. 10, 2013, https://doi.org/10.1109/TBME.2013.2264956.

[38] S. Taran, V. Bajaj, D. Sharma, S. Siuly, and A. Sengur, “Features based on analytic IMF for classifying motor imagery EEG signals in BCI applications,” Meas. J. Int. Meas. Confed., vol. 116, no. October 2017, pp. 68–76, 2018, https://doi.org/10.1016/j.measurement.2017.10.067.

[39] J. Kevric and A. Subasi, “Comparison of signal decomposition methods in classification of EEG signals for motor-imagery BCI system,” Biomed. Signal Process. Control, vol. 31, pp. 398–406, 2017, https://doi.org/10.1016/j.bspc.2016.09.007.

[40] C. J. Ortiz-Echeverri, et al, “A New Approach for Motor Imagery Classification Based on Sorted Blind Source Separation, Continuous Wavelet Transform, and Convolutional Neural Network,” Sensors (Switzerland), vol. 19, no. 20, p. 4541, 2019, https://doi.org/10.3390/s19204541.

[41] S. Kumar, et al, “A Deep Learning Approach for Motor Imagery EEG Signal Classification,” Proc. - Asia-Pacific World Congr. Comput. Sci. Eng. 2016 Asia-Pacific World Congr. Eng. 2016, APWC CSE/APWCE 2016, pp. 34–39, 2017, https://doi.org/10.1109/APWC-on-CSE.2016.017.

[42] J. Meng, et al, "Simultaneously Optimizing Spatial Spectral Features Based on Mutual Information for EEG Classification," in IEEE Transactions on Biomedical Engineering, vol. 62, no. 1, pp. 227-240, Jan. 2015, https://doi.org/10.1109/TBME.2014.2345458.

[43] Y.Park and W. Chung, "Optimal Channel Selection Using Correlation Coefficient for CSP Based EEG Classification," in IEEE Access, vol. 8, pp. 111514-111521, 2020, https://doi.org/10.1109/ACCESS.2020.3003056.

[44] H.V. Shenoy and V. A. Prasad, "Optimized bi-objective eeg channel selection and cross-subject generalization with brain-computer interfaces," in IEEE Transactions on Human-Machine Systems, vol. 46, no. 6, pp. 777-786, 2016, https://doi.org/10.1109/THMS.2016.2573827.

[45] S.M. Tariq, et al, "Motor Imagery EEG Signals Decoding by Multivariate Empirical Wavelet Transform-Based Framework for Robust Brain-Computer Interfaces," in IEEE Access, vol. 7, pp. 171431-141451, 2019, https://doi.org/10.1109/ACCESS.2019.2956018.

[46] V. J. Lawhern, A. J. Solon, N. R. Waytowich, S. M. Gordon, C. P. Hung and B. J. Lance, "EEGNet: a compact convolutional neural network for EEG-based brain–computer interfaces," in Journal of Neural Engineering, vol. 15, no. 5, 2018.

[47] C. Li and J. Xu, "Feature selection with Fisher score followed by the Maximal Clique Centrality algorithm can accurately identify the hub genes of hepatocellular carcinoma," in Scientific Reports, vol. 9, article no. 17283, 2019, https://doi.org/10.1038/s4 1598-019-53471-0

[48] H. Ke, D. Chen, B. Shi, J. Zhang, X. Liu, X. Zhang, and X. Li, “Improving brain E-health services via high-performance EEG classification with grouping Bayesian optimization,” in IEEE Transactions on Services Computing, vol. 13, no. 4, 2019, doi: 10.1109/tsc.2019.2962673

[49] C.O. Quero, D. Durini, R. Ramos-Garcia, J. Rangel-Magdaleno and J. Martinez-Carranza, “Hardware parallel architecture proposed to accelerate the orthogonal matching pursuit compressive sensing reconstruction”, in Computational Imaging V, vol. 11396, pp.113960N, 2020, https://doi.org/10.1117/12.2558937

[50] A. Kulkarni and T. Mohsenin. “Accelerating compressive sensing reconstruction OMP algorithm with CPU, GPU, FPGA and domain specific many-core,” 2015 IEEE International Symposium on Circuits and Systems (ISCAS), pp. 970-973, 2015, IEEE

[51] R. Rubinstein, M. Zibulevsky and M. Elad, “Efficient Implementation of the K-SVD Algorithm using Batch Orthogonal Matching Pursuit Technical Report “, CS Technion, 2008.