

Smart Bagged Tree-based Classifier optimized by Random Forests (SBT-RF) to Classify Brain-Machine Interface Data

Original Scientific Paper

Omar A. Sesa

Mansoura University,
Computers and Control Systems Engineering Dept.,
Faculty of Engineering,
Mansoura, Dakahlia, Egypt
omarsesa.os@std.mans.edu.eg

Amira Y. Haikal

Mansoura University,
Computers and Control Systems Engineering Dept.,
Faculty of Engineering,
Mansoura, Dakahlia, Egypt
amirayh@mans.edu.eg

Mostafa A. Elhosseini

Mansoura University,
Computers and Control Systems Engineering Dept.,
Faculty of Engineering, Mansoura, Dakahlia, Egypt
melhosseini@mans.edu.eg

Taibah University,
College of Computer Science and Engineering in
Yanbu, Yanbu, Madinah, Saudi Arabia
melhosseini@ieee.org

Hesham H. Gad

Mansoura University,
Computers and Control Systems Engineering Dept.,
Faculty of Engineering, Mansoura, Dakahlia, Egypt
hmgad2004@mans.edu.eg

Abstract – Brain-Computer Interface (BCI) is a new technology that uses electrodes and sensors to connect machines and computers with the human brain to improve a person's mental performance. Also, human intentions and thoughts are analyzed and recognized using BCI, which is then translated into Electroencephalogram (EEG) signals. However, certain brain signals may contain redundant information, making classification ineffective. Therefore, relevant characteristics are essential for enhancing classification performance. Thus, feature selection has been employed to eliminate redundant data before sorting to reduce computation time. BCI Competition III Dataset Iva was used to investigate the efficacy of the proposed system. A Smart Bagged Tree-based Classifier (SBT-RF) technique is presented to determine the importance of the features for selecting and classifying the data. As a result, SBT-RF is better at improving the mean accuracy of the dataset. It also decreases computation cost and training time and increases prediction speed. Furthermore, fewer features mean fewer electrodes, thus lowering the risk of damage to the brain. The proposed algorithm has the greatest average accuracy of ~98% compared to other relevant algorithms in the literature. SBT-RF is compared to state-of-the-art algorithms based on the following performance metrics: Confusion Matrix, ROC-AUC, F1-Score, Training Time, Prediction speed, and Accuracy.

Keywords: Brain-Machine Interface; Bagged Trees; Classification; Feature Selection; Optimization; Random Forests

1. INTRODUCTION

A BCI uses electrodes or sensors to interface machines with the human brain [1] based on neurosciences [2]. BCI receives and transmits electrical signals, which can help doctors to discover more information about brain issues and diseases like stroke to use in rehabilitation [3], [4]. It can also simulate a human brain to enhance machine learning and control objects as natural parts of its body representation[5]–[7]. BCI helps the medical field & health care and plays a crucial role in several areas, such as entertainment, education, marketing, and automated control [8]–[10]. Freely available datasets relevant to BCI can be used to test the suggested technique.

One of these datasets is BCI competitions. The Brain-Computer Interface (BCI) Competition was established to evaluate signal processing and classification methods for BCIs. BCI Competition III dataset IVa, five healthy subjects' brains activated for some MI tasks, namely, "aa," "al," "av," "aw," and "ay," were recorded [11]. This recording was taken using Brain Amp amplifiers and a 128-channel Ag/AgCl electrode cap from Electro-cerebral Inactivity (ECI) and 118 channels in those five healthy subjects by putting 118 electrodes international 10/20 system [12].

The feature selection algorithm is an important pre-processing mathematical stage to decrease data size by

reducing the dimension of the data set by removing irrelevant and unnecessary data and selecting the most suitable properties for the data categories. In addition, feature selection increases speed and accuracy, thus obtaining high performance [13], [14]. Feature selection methods can be widely categorized into filter, wrapper, and embedded methods [15], [16]. For example, RF is classified as embedded feature selection because it combines wrapper and filters' positive aspects such as high speed and high accuracies [17] proposed by Breiman [18].

Data is typically classified into a class or category via classification algorithms. There are three types of classification: binary classification, multiclass classification, and multilabel classification. Binary classification techniques are used to classify datasets with only two classes: normal and abnormal states, commonly referred to as "class 0" and "class 1," respectively. Datasets with more than two classes are classified using multiclass classification techniques. Some binary classification techniques can also be utilized for multiclass classification. Finally, multilabel classification techniques classify datasets with two or more classes, with one or more class labels predicted for each input [19], [20].

The selected features are classified using the Bagged Trees algorithm (BT). Researchers bolstered weak classification accuracies with a combination of classifiers, like bagging or kernel additions. A bagging algorithm combines multiple classifiers (Ensemble) proposed by Breiman. Bagging is a way to make the poor classifier better than the first by sampling the dataset into bootstrap samples to prevent the classifier from getting overfitted [21], [22] by learning every classifier individually and collecting the votes of the classifiers. The class with the highest number of votes is the winner. Bagging the decision tree will produce the BT algorithm [23]–[25].

Based on the details listed above, SBT–RF algorithm has been built from the Random Forests (RF) algorithm to choose dataset features and then apply the BT algorithm to classify the selected features. As a result, the proposed algorithm outperforms all different algorithms in accuracy testing.

The main contributions of this paper are:

- Delete superfluous channels by using RF to select features.
- Bagging the decision trees produces the BT classifier that will classify the dataset.
- Reducing the computational cost and time required to train and predict BCI datasets classes.

This paper is structured as follows: Section 2 covers the relevant work and emphasizes the pros and cons of each given input, Section 3 presents the main intended achievement of solving the previously mentioned problems through SBT–RF, Section 4 consists of dataset description, the used performance metrics and the machine results obtained. Finally, section 5 discusses the conclusion.

2. RELATED WORK

Literature on machine learning and feature selection application to BCI models has increased in recent years. For example, Md.A.M. Joadder et al. (2019) [26] developed a method to classify mental states for an SI-based BCI system. First, they applied several feature extraction techniques; Katz Fractal Dimension (KFD), Sub-band Energy, Log Variance, and Root Mean Square (RMS). After that, they used the obtained features as input to Linear Discriminant Analysis (LDA) classifier, and the best average accuracy value is 84.35% by KFD with LDA. However, this algorithm needs a large number of channels to classify well.

Yongkoo Park et al. (2019) [27] proposed a new method; Local Region Frequency Optimized Common Spatial Pattern (LRFCSPP). The features are extracted from the best local regions by applying Variance Ratio Dispersion Score (VRDS) and Interclass Feature Distance (ICFD) techniques to optimize CSPs. Unfortunately, with a mean classification accuracy of 92.93%, The approach is ineffective when classifying tiny samples.

Amin Hekmatmanesh et al. (2020) [28] proposed a technique to enhance a common spatial pattern algorithm to recognize and classify BCI Competition III dataset IVa by combining four different algorithms. Those were Kernel Linear Discriminant Analysis (KLDA), the Kernel Principal Component Analysis (KPCA), the Soft Margin Support Vector Machine (SSVM) classifier, and the Generalized Radial Bases Functions (GRBF) to create methods called DFBCSP DSLVQ SSVM GRBF with an average accuracy of 92.70%. Still, this method increases the error ratio for multiclass.

Sahar Selim et al. (2020) [29] compare feature recognition techniques; RMS, Renyi entropy, Shannon entropy, Katz fractal dimension, and CSP to minimize the number of features used as input to the classifier. The classifiers they used are Support Vector Machine (SVM) and Linear Discriminant Analysis (LDA) as a classifier, and the best average accuracy among all those algorithms is 79.46%. Furthermore, Renyi entropy with LDA and SVM achieved reasonable accuracy with high computational speed, which executes faster than CSP in computational time but with lower precision.

Yao Guo et al. (2020) [4] proposed two methods. The first is the Filtered Band Component Regularized Common Spatial pattern (FCCSP) to increase the robustness of CSP against the small samples by reducing estimations and adding priorities to the spatial filter. The second is Minimum Redundancy Maximum Relevance (mRMR) to discard unnecessary features and use the rest as input to LDA Classifier, and the highest average accuracy obtained was by FCCSP (82.01%). However, this approach may not perform well with multiclass motor imagery tasks.

Wenlong Hang et al. (2020) [30] proposed a new framework that includes a Support Matrix Machine

(SMM) as a core part of a deep stacked network to build a Deep Stack Support Matrix Machine (DSSMM). The framework has been applied to BCI competition III dataset IVa and performs effectively with feed-forward with an average accuracy of 83.68%. On the other hand, executing this method takes a high computational time.

Research gap: In light of the studies mentioned above, it appears that most of these methods suffer from the high amount of time spent versus poor quality in large or small sample settings. Furthermore, they have relatively high error margins when attempting to classify datasets. Table 1 summarizes the pros and cons of these methods.

Table 1. Summaries previously discussed algorithms focusing on various pros and cons

Author	Method	Mean Accuracy	No. of Channels	Pros	Cons
Md.A.M. Joadder et al. [26]	Katz + LDA	84.35%	118	-	This algorithm needs a large number of channels to classify well.
Yongkoo Park et al. [27]	LRFCSP	(92.93±3.99)%	18	Performs well with large sets	This method obtains poor performance when classifying a small sample set.
Amin Hekmatmanesh et al. [28]	DFBCSP DSLVQ SSVN GRBF	92.70%	118	Performs well with binary classes	This method increases the error ratio for multiclass.
Sahar Selim et al. [29]	CSP+LDA	79.77%	18	Executes fast	Low on accuracy.
Yao Guo et al. [4]	FCCSP	82.01%	118	Performs well with binary classes	This approach may not perform well with multiclass motor imagery tasks.
Wenlong Hang et al. [30]	DSSMM	83.68%	118	-	High computational time

3. MATERIALS AND METHODS

There are four components to a BCI: Data collection, Preprocessing, Feature selection, and classification.

3.1. DATA COLLECTION

The BCI Competition III Dataset Iva and other biomedical datasets have been used to evaluate the efficiency of the proposed approach.

3.1.1. BCI Dataset

This BCI Competition III Dataset IVa contains Motor imagery (MI) tasks of five healthy subjects, namely, "aa," "al," "av," "aw," and "ay" [31], which are recorded. The EEG data consists of three classes (right hand(R), Left hand (L), and foot (F)) [32]. Only cues for the classes "right hand" and "foot" were provided. There were two types of visual stimulation. The first type, where targets were indicated by letters appearing behind a fixation cross (which might motivate a little bit of target-correlated eye movements). The second and (2) where a randomly moving object indicated targets (inducing target-uncorrelated eye movements). This dataset was acquired from the participants seated on a chair well relaxed. Visual cues were displayed for 3.5 s, during which the subject had to perform the MI tasks; left or right hand and foot [33], as shown in (Fig. 1).

Brain Amp amplifiers, an ECI 128-channel Ag/AgCl electrode cap, and 118 electrodes were used [34] as defined by the international 10/20 system, as shown in (Fig. 2). The BCI Competition III Dataset Iva and other biomedical datasets have been used to evaluate the efficiency of the proposed approach.

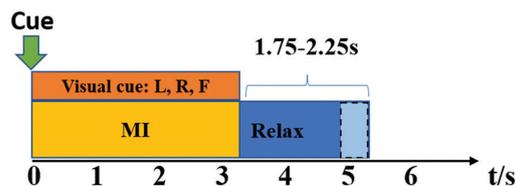


Fig 1. Procedure timeline.

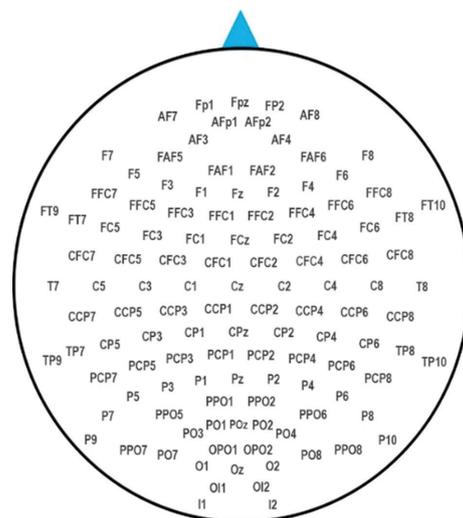


Fig. 2. 118-channel positions

The EEG segments representing the MI part were separated from the dataset. For that reason, the marker's position that indicated the start of 280 cues and the actuality that each movement was 3.5 s long had been utilized. Thus, 280 EEG segments were obtained for each subject toward the finish of this procedure, as described in Table 2.

Table 2. Dataset trials.

Subject	Among 280 trials				
	aa	al	av	aw	ay
Size of data with two classes (RH and RF)	298458 × 118	283574 × 118	283042 × 118	282838 × 118	283562 × 118
Number of trials considered as a training trial with the class label	168	224	84	56	28
Number of trials considered as a testing trial without a class label	112	56	196	224	252

The EEG segments representing the MI part were separated from the dataset. For that reason, the marker's position that indicated the start of 280 cues and the actuality that each movement was 3.5 s long had been utilized. Thus, 280 EEG segments were obtained for each subject toward the finish of this procedure, as described in Table 2.

3.2. SIGNAL PREPROCESSING

Raw EEG data are cluttered with noise and artifacts. Signals are preprocessed to remove artifacts. Among the functions are artifact rejection, channel selection, and baseline filtering. A Butterworth bandpass filter of the fifth order can be used in this case. Preprocessing of the signals is not our primary objective in this paper.

3.3. FEATURE SELECTION

Choosing features is a critical part of designing a machine-learning model. As well as helping to distinguish relevant from irrelevant attributes, it decreases the dimensionality of the original dataset, which helps improve performance. Furthermore, the selection of relevant attributes helps increase learning performance. The application of feature selection allows models to be interpreted after a short training period, which is essential for improving brain-computer interfaces based on motor imagery [35]. The RF explains the significance of each feature [18].

Feature selection using the RF algorithm has two targets [36], [37]–[41] first to discover highly related feature variables and find features with comparatively little data and better capacity to communicate anticipating results. An evaluation feature is usually valued according to two measures: the Gini index, also known as the Gini coefficient or the Gini impurity. It indicates the likelihood that a variable will be wrongly classified when selected at random. The second measure is the error rates of OOB, as shown in Algorithm 1. If z is the total number of features $\{x_1, x_2, \dots, x_z\}$ then the particular procedures to calculate feature importance for every x_j as follows:

- Applying the bootstrap method to the original training dataset creates stochastic sample sets K by fetching them for classification or regression. Then, send them back to the original dataset to create another stochastic sample set, and every time this procedure happens, it makes an OOB sample for the unsampled data.

- m_{try} features ($m_{try} \leq n$) are haphazardly extricated at every node of every tree in the RF algorithm as a randomly produced feature subset by ascertaining the data contained in each feature and calculating it. Feature with the best possible classification capability is chosen among the m_{try} features to split the node, which makes decision trees more diverse.
- Feature importance can be calculated by the Gini index as follows: For each feature x_j The Gini index value can be calculated by getting the average of the Gini index change amount before and after node impurity gets split in whole decision trees of the RF algorithm as shown in (Fig. 3). The Gini index can be calculated as follows [42],[43], [38]:

$$GI_m = \sum_{k=1}^{|k|} \sum_{k' \neq 1} P_{mk} P_{mk'} = 1 - \sum_{k=1}^{|k|} P_{mk}^2 \quad (1)$$

Where GI_m refers to the Gini index of node m , and K refers to K -categories, and P_{mk} refers to the category k proportion in node m . x_j the amount of Gini index change calculates the feature importance value at node m before and after node m gets split.

$$MVI_{jm}^{(Gini)} = GI_m - GI_l - GI_r \quad (2)$$

MVI represents the variable importance. GI_l represents the Gini index of the left branch and GI_r represent the Gini index also but for the right branch.

Feature x_j in decision tree I and belongs to set M , the importance of this feature [18],[44]:

$$MVI_{ij}^{(Gini)} = \sum_{m \in M} MVI_{jm}^{(Gini)} \quad (3)$$

Let N be the number of trees that are created in RF, then:

$$MVI_j^{(Gini)} = \sum_{i=1}^N MVI_{ij}^{(Gini)} \quad (4)$$

Normalize the whole scores of feature importance by:

$$MVI_j = \frac{MVI_j^{(Gini)}}{\sum_{j=1}^C MVI_j^{(Gini)}} \quad (5)$$

Algorithm 1: Random Forests

- 1 **for** features $x_j, j = 1$ to z do
- 2 **for each** tree learner in sample $k = 1$ to $|k|$ do
- 3 Find all nodes m that use x_j .

```

4 | Obtain feature importance by calculating
  | impurity before and after splitting of  $x_j$  using
  | the Gini index.
5 |   Aggregate the enhancements
6 | end
7 | Aggregate the improvements of all the decision
  | trees to get feature importance of  $x_j$ .
8 | end

```

```

1 | while developing tree
2 |   Calculate accuracy for OOB observations and
  | record the prediction
3 | end
4 | Do permutation to  $x_j$  OOB observations to
  | breakdown the relations between the target
  | and the permuted feature  $x_j$ .
5 | for each OOB observation
6 |   Calculate the accuracy and record the prediction
  | again
7 | end
8 | Get the average of the performance reduction that
  | had been happened due to permute  $x_j$  and utilize
  | it to measure the  $x_j$  feature importance.

```

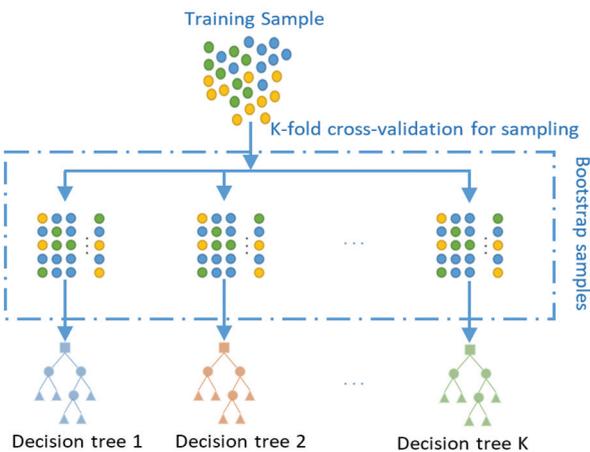


Fig. 3. RF algorithm process

3.4. CLASSIFICATION

A detailed explanation of the main algorithms is provided in this section.

3.4.1. Cart Decision Tree

The CART algorithm has been proposed by Breiman [42] to create generations of decision trees. The CART is a decision tree depending on binary methodology to generate a decision tree from the dataset [45]–[47], as shown in (Fig. 4). Using the CART as a classifier depends on the Gini index. The Gini index is applied on each node of the tree to detect the effectiveness of the data attribute to choose where the effective split will

be in the set to provide new subsets [21], [47], as presented in Algorithm 2. The smaller the Gini value of the attribute, the greater the purity of the node, and it will be chosen as the best one to do the splitting. The algorithm is represented mathematically [43] as follows:

D refers to a dataset, k th proportion sample of D

$$P_k (k=1,2,\dots,|k|)$$

Gini(D) shows the probability between 2 classes that have been randomly chosen from D

$$\text{Gini}(D) \propto \frac{1}{\text{the purity of } D} \quad (6)$$

Smaller the Gini value, the higher the purity. The Gini index of an attribute x can be obtained as follow:

$$\text{Gini_index}(D, x) = \sum_{v=1}^{|D^v|} \frac{|D^v|}{|D|} \text{Gini}(D^v) \quad (7)$$

Where D^v is a subset of D with tuples having value (v) For example, $x_* = \arg \min_{x \in A} \text{Gini_index}(D, x)$. Where A : the candidate attribute set.

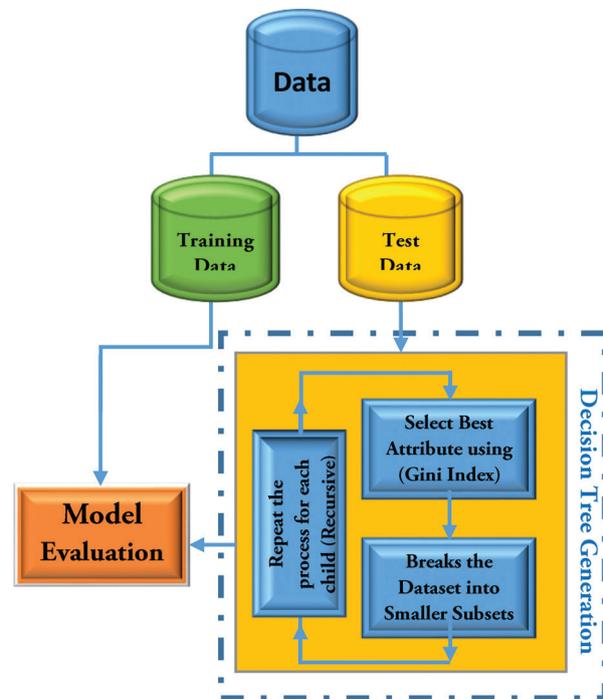


Fig. 4. CART decision trees

Algorithm 2: CART

```

1 | if Number of Feature values= $k$  Then
2 |   Possible splits = $k-1$ 
3 | end
4 | for each feature( $F$ ) in Dataset( $D$ )
5 |   while Split No. ( $SN$ ) $<k$ 
6 |     Find the best split ( $BS$ ) using Gini Index
7 |   end
8 |   Split the feature at  $BS$ 

```

```

9  | while the Stopping criterion is not satisfied
10 |   | for each node(N)
11 |   |   | Find the node's best split (NBS)
   |   |   | using Gini Index
12 |   |   | end
13 |   |   | Split the node at NBS
14 |   | end
15 | end

```

3.4.2. Classification Algorithm

As Breiman [22] proposed, bagging is an algorithm that produces multiple versions of a predictor that can be aggregated. For example, aggregation averaging the outcome by bagging a regressor algorithm to predict a numerical value [48], [49].

Bagging a classifier algorithm is done by collecting the number of votes for each classifier version to predict the output class. Algorithm 3 shows how the bagging algorithm works in steps on the dataset. First, the multiple versions are created by making frequent bootstraps of the learning set, which will be used as new learning sets [50], [51], as shown in (Fig. 5).

In classification [22], [52] a predictor $\phi(x, L)$ predicts class label $j \in \{1, \dots, j\}$

$$Q(j|x) = P(\phi(x, L) = j) \quad (8)$$

The meaning of $Q(j|x)$ is this: over many independent replicates of the learning set L , ϕ predicts class label j at input x with relative frequency $Q(j|x)$. Let $P(j|x)$ be the probability that input x generates class j . Then the likelihood that the predictor classifies the developed state at x correctly is

$$\sum_j Q(j|x)P(j|x) \quad (9)$$

The total probability of the right classification is

$$r = \int [\sum_j Q(j|x)P(j|x)]Px(dx) \quad (10)$$

where $Px(dx)$ is the probability distribution of x

Please note this for any $Q(j|x)$

$$\sum_j Q(j|x)P(j|x) \leq \max_j P(j|x) \quad (11)$$

with equality only if

$$Q(j|x) \begin{cases} 1 & \text{if } P(j|x) = \max_i P(i|x) \\ 0 & \text{else} \end{cases} \quad (12)$$

The predictor $\phi^*(x) = \arg \max_i P(i|x)$ (defined as the Bayes predictor) conduces to the previous representation for $Q(j|x)$ and achieves the highest possible correct classification rate:

$$r^* = \int \max_j P(j|x)Px(x) \quad (13)$$

Call ϕ order correct at the input x if

$$\arg \max_j Q(j|x) = \arg \max_j P(j|x) \quad (14)$$

It implies that if input x occurs more frequently than any other variable in class j , then ϕ also predicts class j at x more regularly than the others. An order-correct predictor is not always an accurate predictor. So, the aggregated predictor is:

$$\phi_A(x) = \arg \max_j Q(j|x) \quad (15)$$

The correct classification probability at x for the aggregated predictor is:

$$\sum_j I(\arg \max_i Q(i|x) = j) P(j|x) \quad (16)$$

$I(\cdot)$ refer to the indicator function. If ϕ is ordered correct at x , then the previous equation equals $\max_j P(j|x)$.

Let us assume that C is the set of all inputs x at which ϕ is correctly ordered, the expression for the proper classification probability of ϕ_A will be:

$$r_A = \int_{x \in C} \max_j P(j|x)Px(dx) + \int_{x \in C^*} [\sum_j I(\phi_A(x) = j)P(j|x)]Px(x) \quad (17)$$

According to the previous equations, if the predictor has a good sensation to predict order correct for most inputs of x , then aggregation can convert it into an almost optimum predictor. In contrast to the numerical prediction situation, weak predictors can be converted into bad ones. Bagging unstable classifiers improve them [22].

Algorithm 3: Bagging

```

1  Variables: OD: Original Dataset
   N: Number of bootstrap samples
   L: Learning Algorithm
   C*: Bagging ensemble classifier
2  for j=1 to N
3  |   BSSj ← bootstrap sample from OD
4  |   Create Classifier Cj ←L(BSSj)
5  end
6  for each new instance, predict the class label
7  |    $C^*(x) = \arg \max_y \sum_{j=1}^N [C_j(x) = y]$ 
8  end

```

In machine learning, Cross-validation is a blind technique used frequently to enhance model prediction and reduce bias. The BCI dataset has been divided randomly into k sets for each subject individually (k -fold cross-validation) [53]. In this study, 5-fold cross-validation has been used. One of these sets was utilized as a testing set, while the other four were used as training sets. This method is repeated five times, each time with a different set.

Previous work has suffered from weaknesses since it was designed as an independent algorithm. However, such defects can be overcome by combining different algorithms. Therefore, this work combines BT and RF algorithms to perform better than operating individually.

The BT algorithm reduces the pre-steps needed to obtain high-dimensional feature data, including dimension reduction and feature selections. Thus, from the speed evaluation perspective, the BT algorithm can speed up the process impressively while maintaining the easiness of creating parallel methods and the simplicity of implementation. Furthermore, if things go wrong, it can fix the error, as balancing errors occur by unbalanced data sets or trying to reduce the impact of losing a huge part of features by maintaining accuracy despite the current condition.

On the other hand, the RF algorithm focus on getting the best feature selection possible by calculating the Gini index and OOB error, which leads to judging the importance of features and interaction among different ones.

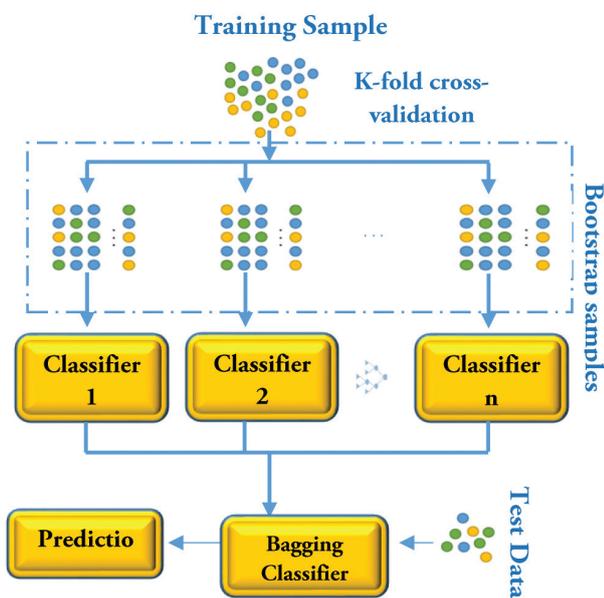


Fig. 5. Bagging process

In machine learning, Cross-validation is a blind technique used frequently to enhance model prediction and reduce bias. The BCI dataset has been divided randomly into k sets for each subject individually (k-fold cross-validation) [53]. In this study, 5-fold cross-validation has been used. One of these sets was utilized as a testing set, while the other four were used as training sets. This method is repeated five times, each time with a different set.

Previous work has suffered from weaknesses since it was designed as an independent algorithm. However, such defects can be overcome by combining different algorithms. Therefore, this work combines BT and RF algorithms to perform better than operating individually.

The BT algorithm reduces the pre-steps needed to obtain high-dimensional feature data, including dimension reduction and feature selections. Thus, from the speed evaluation perspective, the BT algorithm can speed up the process impressively while maintaining the easiness of creating parallel methods and the simplicity of implementation. Furthermore, if things go wrong, it can fix the error, as balancing errors occur by unbalanced data sets or trying to reduce the impact of losing a huge part of features by maintaining accuracy despite the current condition.

On the other hand, the RF algorithm focus on getting the best feature selection possible by calculating the Gini index and OOB error, which leads to judging the importance of features and interaction among different ones.

3.4.3. Smart Hybrid Algorithm

SBT-RF is implemented by combining the feature selection algorithm RF and classification algorithm BT to work with the BCI Data set. (Fig. 6) and Algorithm 4 shows the SBT-RF mechanism, starting with preparing each subject's dataset until it gets classified with 5-fold cross-validation.

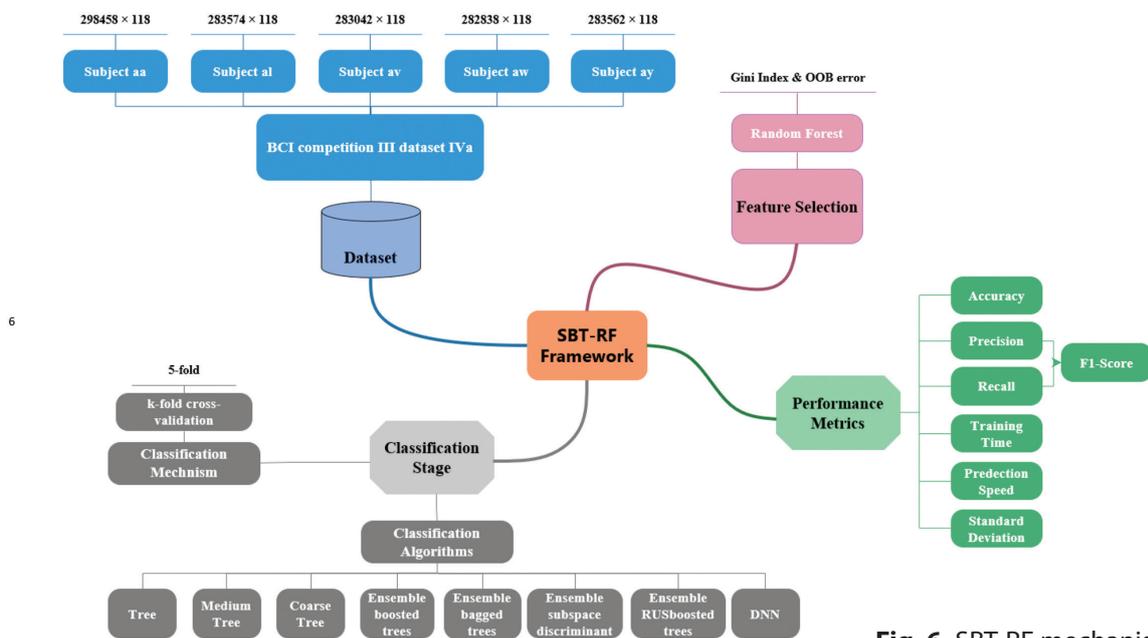


Fig. 6. SBT-RF mechanism

Algorithm 4: SBT-RF

```

1 Variables: OOB: Out of bag
   OD: Original Dataset
   N: Number of bootstrap samples
   L: Learning Algorithm
   C*: Bagging ensemble classifier
2 for each Subject
3   for features  $x_j$ ,  $j = 1$  to  $z$  do
4     for each tree learner in sample  $k = 1$  to  $|k|$  do
5       Find all nodes  $m$  that use  $x_j$ .
6       Obtain feature importance by calculating
       impurity before and after splitting of  $x_j$  using
       the Gini index.
7       Aggregate the enhancements
8     end
9     Aggregate the improvements of all the
       decision trees to the feature importance of  $x_j$ .
10    end
11    while developing tree
12      Calculate accuracy for OOB observations and
       record the prediction
13    end
14    Do premutation to  $x_j$  OOB observations to
       breakdown the relations between the target &
       the permuted feature  $x_j$ .
15    for each OOB observation
16      calculate the accuracy and record the
       prediction again
17    end
18    Get the average of the performance
       reduction that had been happened due to
       permute  $x_j$  and utilize it to measure the  $x_j$ 
       feature importance.
19    Train BT algorithm using the selected feature
       with 5k-fold
20    for  $j=1$  to  $N$ 
21      BSS $_j$   $\leftarrow$  bootstrap sample from OD
22      Create Classifier  $C_j \leftarrow L(BSS_j)$ 
23    end
24    for each new instance, predict the class label
25       $C^*(x) = \mathop{\text{arg max}}_y \sum_{j=1}^N [C_j(x) = y]$ 
26    end
       Evaluate the model performance
27 end

```

4. SIMULATION AND COMPUTER RESULTS

This section has three parts. The first section describes the dataset and how it was prepared. The second section discusses the performance metrics used. The final section displays the results of the proposed SBT-RF algorithm and other algorithms.

4.1. PERFORMANCE METRICS

Performance metrics indicate how the proposed method compares to each state-of-the-art algorithm, as shown in (Fig. 7).

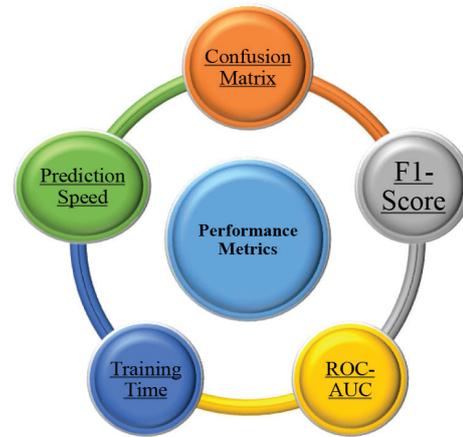


Fig. 7. Performance metrics.

Confusion Matrix: A Confusion Matrix is a table frequently used to specify the output of a classification model on a set of test data whose real values are known [54], [55]. It allows the performance of an algorithm to be visualized [56].

Accuracy and Misclassification rate equations are given as shown:

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN} \quad (18)$$

$$\text{Misclassification rate} = \frac{FP+FN}{TP+TN+FP+FN} \quad (19)$$

Where TP: True Positive, TN: True Negative, FP: False Positive, FN: False Negative.

F1-Score: F1-Score is a technique to measure the performance depending on the harmonic way between Precision and Recall to grant a balanced measure of the misclassified cases than the confusion matrix [56].

$$\text{Precision} = \frac{(\text{True Positive})}{(\text{True Positive} + \text{False Positive})} \quad (20)$$

$$\text{Recall} = \frac{(\text{True Positive})}{(\text{True Positive} + \text{False Negative})} \quad (21)$$

From equations (20), (21); F1-Score can be calculated as shown:

$$F1 = \frac{2}{(1/\text{Precision} + 1/\text{Recall})} \quad (22)$$

ROC-AUC: The Receiver Operating Characteristic Curve (ROC) is the plot that can show binary classifier performance as a function of the cut-off threshold between True positive rate (Sensitivity) and false positive rate (1-specificity) [57], [58]. Area Under Curve (AUC) supplies an overall performance measurement among every potential classification threshold [59].

Training Time: The overall time required for the model to learn.

Prediction Speed: The number of observations the machine learning model can produce per second.

4.2. COMPUTER SIMULATIONS AND RESULTS

The following section will describe the experiment that was conducted. First, putting several algorithms to use on BCI Competition III dataset IVa., then performing SBT-RF 50 times on each subject.

This section will present and apply several algorithms to BCI Competition III dataset IVa. All these algorithms were tested with 5-fold cross-validations to ensure the data's variance and reduce the computation cost. However, the algorithm does not need more K-folds because the BT algorithm uses bootstrap sampling to prevent the machine learning model from overfitting by sampling the dataset into random sets and training every group alone [18], [22].

Table 3 contains the average results of 50 runs on the BCI-competition III dataset IVa using eight different algorithms. The test was carried out using the paired device's resources. As the key performance indicators in our study, classification accuracy and F1-score in each run are calculated. Bolded values indicate the best values for each algorithm. The standard deviation (SD) was also calculated to evaluate the algorithm's stability.

Comparing several algorithms and evaluating performance metrics on the models for each subject on the dataset are shown in Table 3. Regarding training time, some algorithms perform better, such as the Coarse Tree with all subjects at 38.8s. Others overcome in predicting observation per second like medium in (aa and av subjects), Coarse Tree in (al, aw and ay subjects), and Fine Tree in (av and ay subjects). The best average prediction Speed value belongs to the Coarse Tree algorithm with 180000 obs/sec, but the accuracy and F1-score for these algorithms were extremely low. Therefore, they are not reliable for the machine learning model. Although the BT algorithm did not achieve a high value in prediction speed and training time, this algorithm dominates the others in accuracy and F1-score for all subjects with a mean accuracy of 97.64% farther than Deep Neural Network (DNN) by 15.31% and a mean F1-score 97.58% farther than DNN by 14.1%. While dealing with BCI datasets, the most important factor is reliability, so according to the results, the BT algorithm has been chosen to be a classification algorithm for this research. Now BT algorithm has good accuracy and F1 score. However, with average training time and prediction speed, a dimension reduction is made using feature selection to enhance computation cost, considering the same accuracy and F1-score or better values.

Table 3. Reported performance metrics.

Algorithm	Performance Metrics	Subjects					Mean
		aa	al	av	aw	ay	
Tree	Accuracy (%)	64.002	68.019	82.9	74.801	91.701	76.2846
	F1 score (%)	58.4	61.286	71.831	74.504	94.179	72.04
	Training Time (sec)	139	200	73	47	14	94.6
	Prediction Speed (obs/sec)	180000	180000	170000	140000	180000	170000
	Standard deviation (±%)	0.227	0.178	0.159	0.193	0.143	0.18
Medium Tree	Accuracy (%)	60.602	60.408	70.904	66.018	85.409	68.6682
	F1 score (%)	61.025	59.406	68.702	68.475	90.175	69.5566
	Training Time (sec)	105	120	44	27	10	61.2
	Prediction Speed (obs/sec)	190000	200000	170000	140000	170000	174000
	Standard deviation (±%)	0.157	0.108	0.112	0.15	0.127	0.1308
Coarse Tree	Accuracy (%)	57.702	56.706	67.304	60.009	76.802	63.7046
	F1 score (%)	59.921	42.751	61.206	66.775	84.794	63.0894
	Training Time (sec)	65	73	31	18	7	38.8
	Prediction Speed (obs/sec)	180000	240000	150000	150000	180000	180000
	Standard deviation (±%)	0.124	0.096	0.099	0.117	0.13	0.1132
Ensemble boosted trees	Accuracy (%)	66.015	65.306	79.305	72.4	93.411	75.2874
	F1 score (%)	60.233	64.53	62.165	73.789	95.401	71.2236
	Training Time (sec)	2604	3551	893	1385	194	1725.4
	Prediction Speed (obs/sec)	78000	70000	69000	140000	99000	91200
	Standard deviation (±%)	0.111	0.057	0.084	0.111	0.126	0.0978
Ensemble bagged trees	Accuracy (%)	96.615	99.109	98.902	95.352	98.261	97.6478
	F1 score (%)	96.397	99.475	98.714	96.614	99.018	98.0436
	Training Time (sec)	1652	2208	368	282	72	916.4
	Prediction Speed (obs/sec)	17000	21000	37000	27000	42000	28800
	Standard deviation (±%)	0.128	0.06	0.113	0.146	0.122	0.1138
Ensemble subspace Discriminant	Accuracy (%)	76.706	72.608	81.302	89.008	91.802	82.2852
	F1 score (%)	70.123	72.65	68.803	89.357	94.322	79.051
	Training Time (sec)	929	1207	163	202	47	509.6
	Prediction Speed (obs/sec)	4900	4300	7000	6500	7800	6100
	Standard deviation (±%)	0.208	0.167	0.173	0.162	0.223	0.1866

Algorithm	Performance Metrics	Subjects					Mean
		aa	al	av	aw	ay	
Ensemble RUSboosted trees	Accuracy (%)	60.709	60.576	72.905	70.808	91.206	71.2408
	F1 score (%)	65.163	59.512	67.086	68.598	93.783	70.8284
	Training Time (sec)	3131	4922	771	719	167	1942
	Prediction Speed (obs/sec)	89000	83000	66000	100000	110000	89600
	Standard deviation ($\pm\%$)	0.126	0.112	0.118	0.116	0.146	0.1236
DNN	Accuracy (%)	76.24	92.36	64.57	87.68	90.83	82.336
	F1 score (%)	78.24	92.42	67.57	89.62	91.83	83.936
	Training Time (sec)	1199	990	115	244	212	552
	Prediction Speed (obs/sec)	115400	125200	139300	118500	145100	128700
	Standard deviation ($\pm\%$)	0.058	0.06	0.086	0.071	0.065	0.0682

Boxplots in (Fig. 8) and (Fig. 9) reveal that the dataset's median is too near to its median after RF. The RF algorithm chose the right characteristics to describe the data well and rejected redundant ones [60], [61].

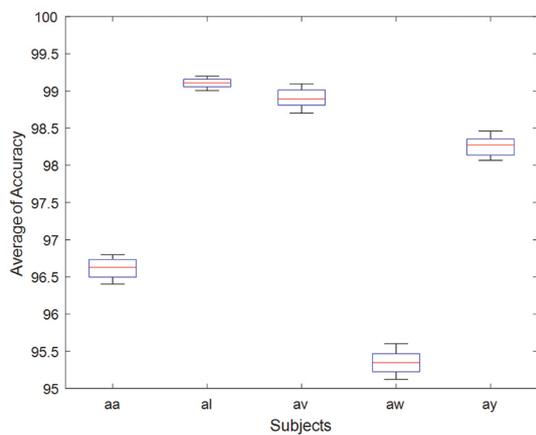


Fig. 8. 118-channel boxplot for BCI III Dataset IVa

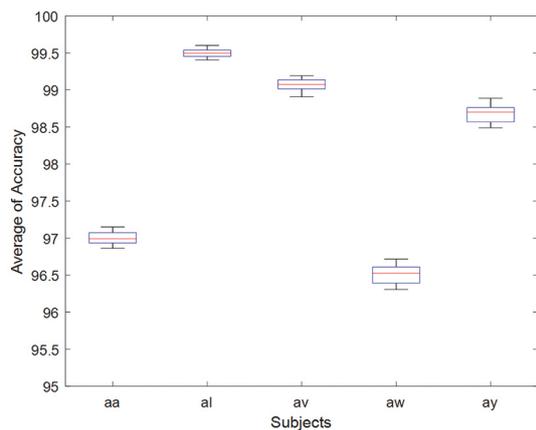


Fig. 9. RF algorithm selected channels boxplot for BCI III Dataset IVa

The second part of the experiment is executing SBT-RF on each subject 50 times. Again, two primary performance metrics (classification accuracy and F1-Score) were deployed to each run. (Fig. 10) and (Fig. 11) show that the SBT-RF algorithm is better than the BT algorithm's accuracy and the high F1 score, as shown in Table

4. Furthermore, (Fig. 12) delivers a decrease in training time (4582 to 1209) sec, while (Fig. 13) shows an increase in average prediction speed (28,800 to 70,600) obs/sec. Furthermore, the rate was increased by 12 times, the risk was limited (by reducing the electrodes) [62], and the computational and financial costs were decreased.

Table 4. SBT-RF vs. BT

Subject	Precision (%)	Recall (%)	F1-Score (%)	Accuracy (%)	Training Time (sec)	Prediction Speed (obs/sec)	AUC
BT							
aa	95.48	96.54	96.001	96.615	1652	17000	0.9
al	99.24	99.00	99.1	99.109	2208	21000	1
av	98.16	98.77	98.461	98.902	368	37000	1
aw	96.07	95.14	95.611	95.352	282	27000	0.9
ay	99.66	97.85	98.748	98.261	72	42000	1
SBT-RF							
aa	95.91	97.04	96.397	97.003	475	46000	1
al	99.52	99.41	99.475	99.498	470	82000	1
av	98.48	98.96	98.714	99.073	133	92000	1
aw	97.07	96.18	96.614	96.512	98	59000	0.9
ay	99.69	98.39	99.018	98.682	33	74000	1

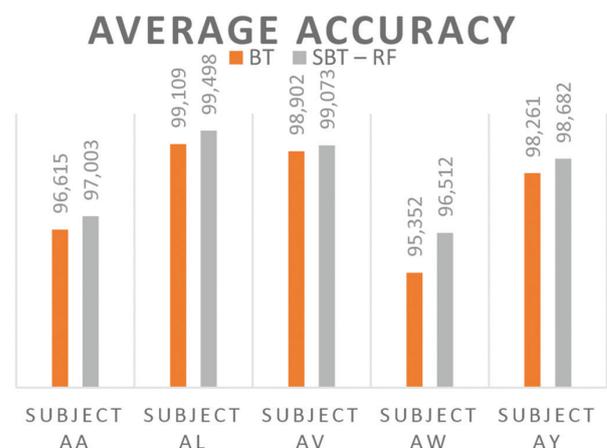


Fig. 10. Average accuracy for BT vs SBT-RF

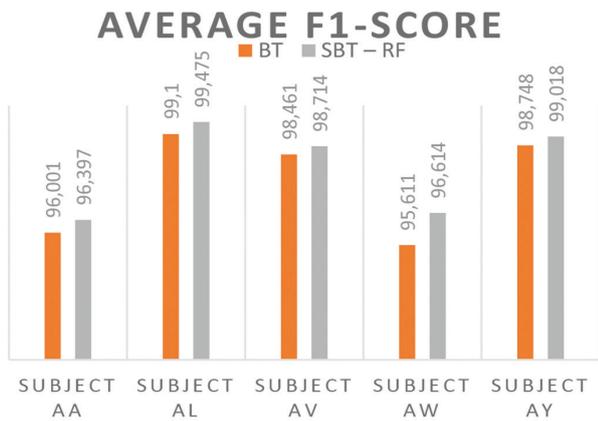


Fig. 11. Average F1-Score for BT vs SBT-RF



Fig. 12. Training time for BT vs SBT-RF



Fig. 13. Average prediction speed for BT vs SBT-RF

Table 5 compares the proposed method SBT-RF with related studies that use different algorithms on the same data set. SBT-RF outperforms all related work for av, aw, and ay subjects by 99.1%, 96.5%, and 98.7%, respectively. However, the enhanced CSP + LS-SVM method shows better results only in aa with 3%. While CSP+LDA and DSSMM methods achieve better results only in al, SBT-RF achieves the highest accuracy with 98.16% on average.

5. CONCLUSIONS

BCI is an advanced approach that helps analyze and recognize human intentions and thoughts that are further transformed into EEG signals. However, it is believed that some signals detected from certain brain channels may contain redundant data that decreases the classification efficiency. Accordingly, feature selection methods have been applied to evacuate the redundant data before the classification process to reduce computation costs. Furthermore, scientists usually combine different classifier combinations to overcome weak ones to make the classification algorithm more accurate. A smart hybrid algorithm (SBT-RF) for classifying BCI datasets (SBT-RF) is proposed in this work. The proposed algorithm is implemented in two stages; firstly, the RF algorithm evaluates the features' importance to select the most useful features. Then, this algorithm (i.e., RF) measures the extent of each feature individually by calculating the lowest impurity using the Gini index and obtaining OOB error. The selected features from the previous stage are inputs to the Ensemble BT classifier. Next, the BT classifier samples the processed "BCI Competition III Dataset IVa" into bootstrap samples, then classifies every sample individually using a decision tree. As a result, it prevents overfitting and collects the votes of the decision tree classifier to predict the class. It is revealed that the proposed algorithm has the highest average accuracy of ~98 % compared to other relevant algorithms reported in the literature.

Table 5. SBT-RF vs related works

Author	Method	Subject Results					Mean
		aa	al	av	aw	ay	
Amin Hekmatmanesh et al. [28]	DFBCSP DSLVQ SSVM GRBF	93.5%	98.57%	81.78%	93.57%	96.07%	92.70%
Sahar Selim et al. [29]	CSP+LDA	79.46%	100%	57.14%	92.41%	69.84%	79.77%
Yao Guo et al. [4]	FCCSP	72.32%	98.21%	68.87%	78.57%	92.06%	82.01%
Wenlong Hang et al. [30]	DSSMM	75.89%	100%	76.53%	89.73%	76.19%	83.68%
Md.A.M. Joadder et al. [26]	Katz + LDA	86.78%	90.35%	68.92%	92.14%	83.57%	84.35%
Yongkoo Park et al. [27]	LRFCSF	98.93%	93.21%	81.79%	93.21%	97.5%	92.93%
Proposed Method	SBT-RF	97%	99.49%	99.07%	96.51%	98.68%	98.15%

6. REFERENCES:

- [1] Y. You, W. Chen, T. Zhang, "Motor imagery EEG classification based on flexible analytic wavelet transform", *Biomedical Signal Processing and Control*, Vol. 62, 2020, p. 102069.
- [2] J. Wang, Z. Feng, X. Ren, N. Lu, J. Luo, L. Sun, "Feature subset and time segment selection for the classification of EEG data based motor imagery", *Biomedical Signal Processing and Control*, Vol. 61, 2020, p. 102026.
- [3] K. Sayed, M. Kamel, M. Alhaddad, H. M. Malibary, Y. M. Kadah, "Characterization of phase space trajectories for Brain-Computer Interface", *Biomedical Signal Processing and Control*, Vol. 38, 2017, pp. 55-66.
- [4] Y. Guo, Y. Zhang, Z. Chen, Y. Liu, W. Chen, "EEG classification by filter band component regularized common spatial pattern for motor imagery", *Biomedical Signal Processing and Control*, Vol. 59, 2020, p. 101917.
- [5] B. O. Olcay and B. Karaçalı, "Evaluation of synchronization measures for capturing the lagged synchronization between EEG channels: A cognitive task recognition approach", *Computers in Biology and Medicine*, Vol. 114, 2019, p. 103441.
- [6] J. Jin, Y. Miao, I. Daly, C. Zuo, D. Hu, A. Cichocki, "Correlation-based channel selection and regularized feature optimization for MI-based BCI", *Neural Networks*, Vol. 118, 2019, pp. 262-270.
- [7] P. Gaur, R. B. Pachori, H. Wang, G. Prasad, "A multi-class EEG-based BCI classification using multivariate empirical mode decomposition based filtering and Riemannian geometry", *Expert Systems with Applications*, Vol. 95, 2018, pp. 201-211.
- [8] O. Landau, A. Cohen, S. Gordon, N. Nissim, "Mind your privacy: Privacy leakage through BCI applications using machine learning methods", *Knowledge-Based Systems*, Vol. 198, 2020, p. 105932.
- [9] J. Feng et al. "Towards correlation-based time window selection method for motor imagery BCIs", *Neural Networks*, Vol. 102, 2018, pp. 87-95.
- [10] A. Barachant, S. Bonnet, M. Congedo, C. Jutten, "Classification of covariance matrices using a Riemannian-based kernel for BCI applications", *Neurocomputing*, Vol. 112, 2013, pp. 172-178.
- [11] Y. Li, P. P. Wen, "Clustering technique-based least square support vector machine for EEG signal classification", *Computer methods and Programs in Biomedicine*, Vol. 104, No. 3, 2011, pp. 358-372.
- [12] N. Lu, T. Yin, "Motor imagery classification via combinatory decomposition of ERP and ERSP using sparse nonnegative matrix factorization", *Journal of Neuroscience Methods*, Vol. 249, 2015, pp. 41-49.
- [13] J. Lee, J.-Y. Jeong, C.-H. Jun, "Markov blanket-based universal feature selection for classification and regression of mixed-type data", *Expert Systems with Applications*, Vol. 158, 2020, p. 113398.
- [14] F. Haider, S. Pollak, P. Albert, S. Luz, "Emotion recognition in low-resource settings: An evaluation of automatic feature selection methods", *Computer Speech & Language*, Vol. 65, 2021, p. 101119.
- [15] G. H. de Rosa, J. P. Papa, X.-S. Yang, "A nature-inspired feature selection approach based on hypercomplex information", *Applied Soft Computing*, Vol. 94, 2020, p. 106453.
- [16] C. Li, X. Luo, Y. Qi, Z. Gao, X. Lin, "A new feature selection algorithm based on relevance, redundancy and complementarity", *Computers in Biology and Medicine*, Vol. 119, 2020, p. 103667.
- [17] C.-F. Tsai, Y.-T. Sung, "Ensemble feature selection in high dimension, low sample size datasets: Parallel and serial combination approaches", *Knowledge-Based Systems*, Vol. 203, 2020, p. 106097.
- [18] L. Breiman, "Random forests", *Machine Learning*, Vol. 45, No. 1, pp. 5-32, 2001.
- [19] A. Zelaia, I. Alegria, O. Arregi, B. Sierra, "A multiclass/multilabel document categorization system: Combining multiple classifiers in a reduced dimension", *Applied Soft Computing*, Vol. 11, No. 8, 2011, pp. 4981-4990.
- [20] P. Davis, F. Aziz, M. T. Newaz, W. Sher, L. Simon, "The classification of construction waste material using a deep convolutional neural network", *Automation in Construction*, Vol. 122, 2021, p. 103481.
- [21] S. C. Lemon, J. Roy, M. A. Clark, P. D. Friedmann, W. Rakowski, "Classification and regression tree analysis in public health: methodological review and comparison with logistic regression", *Annals of Behavioral Medicine*, Vol. 26, No. 3, 2003, pp. 172-181.
- [22] L. Breiman, "Bagging predictors", *Machine Learning*, Vol. 24, No. 2, 1996, pp. 123-140.

- [23] R. E. Banfield, L. O. Hall, K. W. Bowyer, W. P. Kegelmeyer, "A comparison of decision tree ensemble creation techniques", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 29, No. 1, 2006, pp. 173-180.
- [24] G. Martinez-Munoz, D. Hernández-Lobato, A. Suárez, "An analysis of ensemble pruning techniques based on ordered aggregation", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 31, No. 2, 2008, pp. 245-259.
- [25] D. Zhang, X. Zhou, S. C. H. Leung, J. Zheng, "Vertical bagging decision trees model for credit scoring", *Expert Systems with Applications*, Vol. 37, No. 12, 2010, pp. 7838-7843.
- [26] M. A. M. Joadder, S. Siuly, E. Kabir, H. Wang, Y. Zhang, "A new design of mental state classification for subject independent BCI systems", *IRBM*, Vol. 40, No. 5, 2019, pp. 297-305.
- [27] Y. Park, W. Chung, "Frequency-optimized local region common spatial pattern approach for motor imagery classification", *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, Vol. 27, No. 7, 2019, pp. 1378-1388.
- [28] A. Hekmatmanesh, H. Wu, F. Jamaloo, M. Li, H. Handroos, "A combination of CSP-based method with soft margin SVM classifier and generalized RBF kernel for imagery-based brain computer interface applications", *Multimedia Tools and Applications*, Vol. 79, No. 25, 2020, pp. 17521-17549.
- [29] S. Selim, M. Tantawi, H. Shedeed, A. Badr, "A Comparative Analysis of Different Feature Extraction Techniques for Motor Imagery Based BCI System", *Proceedings of the Joint European-US Workshop on Applications of Invariance in Computer Vision*, 2020, pp. 740-749.
- [30] W. Hang, W. Feng, S. Liang, Q. Wang, X. Liu, K.-S. Choi, "Deep stacked support matrix machine based representation learning for motor imagery EEG classification", *Computer Methods and Programs in Biomedicine*, Vol. 193, 2020, p. 105466.
- [31] L. He, Y. Hu, Y. Li, D. Li, "Channel selection by Rayleigh coefficient maximization based genetic algorithm for classifying single-trial motor imagery EEG", *Neurocomputing*, Vol. 121, 2013, pp. 423-433.
- [32] D. Fattahi, B. Nasihatkon, R. Boostani, "A general framework to estimate spatial and spatio-spectral filters for EEG signal classification", *Neurocomputing*, Vol. 119, 2013, pp. 165-174.
- [33] N. Lu, T. Li, J. Pan, X. Ren, Z. Feng, H. Miao, "Structure constrained semi-nonnegative matrix factorization for EEG-based motor imagery classification", *Computers in Biology and Medicine*, Vol. 60, 2015, pp. 32-39.
- [34] Z. Qiu, J. Jin, H.-K. Lam, Y. Zhang, X. Wang, A. Cichocki, "Improved SFFS method for channel selection in motor imagery based BCI", *Neurocomputing*, Vol. 207, 2016, pp. 519-527.
- [36] K. K. Kandaswamy, G. Pugalenth, K.-U. Kalies, E. Hartmann, T. Martinetz, "EcmPred: Prediction of extracellular matrix proteins based on random forest with maximum relevance minimum redundancy feature selection", *Journal of Theoretical Biology*, Vol. 317, 2013, pp. 377-383.
- [35] Y. Saeys, T. Abeel, Y. Van de Peer, "Robust feature selection using ensemble feature selection techniques", *Proceedings of the Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, 2008, pp. 313-325.
- [37] X. Li, W. Chen, Q. Zhang, L. Wu, "Building auto-encoder intrusion detection system based on random forest feature selection", *Computers & Security*, Vol. 95, 2020, p. 101851.
- [38] E. Izquierdo-Verdiguier, R. Zurita-Milla, "An evaluation of Guided Regularized Random Forest for classification and regression tasks in remote sensing", *International Journal of Applied Earth Observation and Geoinformation*, Vol. 88, 2020, p. 102051.
- [39] X. Ji, B. Yang, Q. Tang, "Seabed sediment classification using multibeam backscatter data based on the selecting optimal random forest model", *Applied Acoustics*, Vol. 167, 2020, p. 107387.
- [40] X. Ma, X. Sun, "Sequence-based predictor of ATP-binding residues using random forest and mRMR-IFS feature selection", *Journal of Theoretical Biology*, Vol. 360, 2014, pp. 59-66.
- [41] A. Srivastava, S. Ghosh, N. Anantharaman, V. K. Jayaraman, "Hybrid biogeography based simultaneous feature selection and MHC class I peptide binding prediction using support vector machines and random forests", *Journal of Immunological Methods*, Vol. 387, No. 1-2, 2013, pp. 284-292.

- [42] L. Breiman, J. H. Friedman, R. A. Olshen, C. J. Stone, "Classification and regression trees, wadsworth statistics", Probability Series, Belmont, California: Wadsworth, 1984.
- [43] C. Wang, A. Wang, J. Xu, Q. Wang, F. Zhou, "Outsourced privacy-preserving decision tree classification service over encrypted data", *Journal of Information Security and Applications*, Vol. 53, 2020, p. 102517.
- [44] D. Niu, K. Wang, L. Sun, J. Wu, X. Xu, "Short-term photovoltaic power generation forecasting based on random forest feature selection and CEEMD: A case study", *Applied Soft Computing*, Vol. 93, 2020, p. 106389.
- [45] M. Seera, C. P. Lim, "Online motor fault detection and diagnosis using a hybrid FMM-CART model", *IEEE Transactions on Neural Networks and Learning Systems*, Vol. 25, No. 4, 2013, pp. 806-812.
- [46] J. Cho, X. Li, Z. Gu, P. U. Kurup, "Recognition of explosive precursors using nanowire sensor array and decision tree learning", *IEEE Sensors Journal*, Vol. 12, No. 7, 2011, pp. 2384-2391.
- [47] A. Suárez, J. F. Lutsko, "Globally optimal fuzzy decision trees for classification and regression", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 21, No. 12, 1999, pp. 1297-1311.
- [48] Y. Wu, Y. Ke, Z. Chen, S. Liang, H. Zhao, H. Hong, "Application of alternating decision tree with AdaBoost and bagging ensembles for landslide susceptibility mapping", *Catena*, Vol. 187, 2020, p. 104396.
- [49] W. Chen, H. Hong, S. Li, H. Shahabi, Y. Wang, X. Wang, B. Bin Ahmad, "Flood susceptibility modeling using novel hybrid approach of reduced-error pruning trees with bagging and random subspace ensembles", *Journal of Hydrology*, Vol. 575, 2019, pp. 864-873.
- [50] F. Li, C.-H. Chen, G. Xu, L. P. Khoo, Y. Liu, "Proactive mental fatigue detection of traffic control operators using bagged trees and gaze-bin analysis", *Advanced Engineering Informatics*, Vol. 42, 2019, p. 100987.
- [51] H. Hong et al. "Landslide susceptibility mapping using J48 Decision Tree with AdaBoost, Bagging and Rotation Forest ensembles in the Guangchang area (China)", *Catena*, Vol. 163, 2018, pp. 399-413.
- [52] Y. Himeur, A. Alsalemi, F. Bensaali, A. Amira, "Robust event-based non-intrusive appliance recognition using multi-scale wavelet packet tree and ensemble bagging tree", *Applied Energy*, Vol. 267, 2020, p. 114877.
- [53] Z. Xiong, Y. Cui, Z. Liu, Y. Zhao, M. Hu, J. Hu, "Evaluating explorative prediction power of machine learning algorithms for materials discovery using k-fold forward cross-validation", *Computational Materials Science*, Vol. 171, 2020, p. 109203.
- [54] A. Luque, A. Carrasco, A. Martín, A. de las Heras, "The impact of class imbalance in classification performance metrics based on the binary confusion matrix", *Pattern Recognition*, Vol. 91, 2019, pp. 216-231.
- [55] D. Simon, D. L. Simon, "Analytic confusion matrix bounds for fault detection and isolation using a sum-of-squared-residuals approach", *IEEE Transactions on Reliability*, Vol. 59, No. 2, 2010, pp. 287-296.
- [56] W. Castro, J. Oblitas, M. De-La-Torre, C. Cotrina, K. Bazán, H. Avila-George, "Classification of cape gooseberry fruit according to its level of ripeness using machine learning techniques and different color spaces", *IEEE Access*, Vol. 7, 2019, pp. 27389-27400.
- [57] Y. Zhou et al. "Predictors of response to repeated ketamine infusions in depression with suicidal ideation: an ROC curve analysis", *Journal of Affective Disorders*, Vol. 264, 2020, pp. 263-271.
- [58] A. Ben-David, "About the relationship between ROC curves and Cohen's kappa", *Engineering Applications of Artificial Intelligence*, Vol. 21, No. 6, 2008, pp. 874-882.
- [59] G. Tripepi, K. J. Jager, F. W. Dekker, C. Zoccali, "Diagnostic methods 2: receiver operating characteristic (ROC) curves", *Kidney International*, Vol. 76, No. 3, 2009, pp. 252-256.
- [60] F. Lei, P. Dupuis, O. Durrieu, G. Zissis, P. Maussion, "Acoustic resonance detection using statistical methods of voltage envelope characterization in metal halide lamps", *IEEE Transactions on Industry Applications*, Vol. 53, No. 6, 2017, pp. 5988-5996.
- [61] Z. Li, N. Wang, Y. Li, X. Sun, M. Huo, H. Zhang, "Collective efficacy of support vector regression with smoothness priority in marine sensor data prediction", *IEEE Access*, Vol. 7, 2019, pp. 10308-10317.
- [62] A. J. Fowle, C. D. Binnie, "Uses and abuses of the EEG in epilepsy", *Epilepsia*, Vol. 41, 2000, pp. S10-S18.