International Journal of Innovative Computing, Information and Control
Volume **9**, Number **3**, March **2013**
ICIC International ©2013 ISSN 1349-4198
pp. 1145–1157

COMPARING COMMON MACHINE LEARNING CLASSIFIERS IN LOW-DIMENSIONAL FEATURE VECTORS FOR BRAIN COMPUTER INTERFACE APPLICATIONS

Onder Aydemir and Temel Kayikcioglu

Department of Electrical and Electronics Engineering
Karadeniz Technical University
61080 Trabzon, Turkey
{ onderaydemir; tkayikci }@ktu.edu.tr

Received October 2011; revised February 2012

Abstract. *There are lots of classification and feature extraction algorithms in the field of brain computer interface. It is significant to use optimal classification algorithm and fewer features to implement a fast and accurate brain computer interface system. In this paper, we evaluate the performances of five classical classifiers in different aspects including classification accuracy, sensitivity, specificity, Kappa and computational time in low-dimensional feature vectors extracted from EEG signals. For our experiments we used BCI Competition 2003 Data Set III and Data Set Ia. Classifiers were compared on 61 different datasets which were created with a combination of extracted features. When classifiers were ranked based on the average values of performance metrics, we conclude that the NB and the SVM classifiers are shown to be good candidates for pattern classification for low dimensional feature vectors. On the other hand, it can obviously mention that decision tree classifier provides the worst performance. We believe that this paper has a significant contribution in the field of classifier for brain computer interface applications.*

Keywords: Brain computer interface, Classification accuracy, Classification performance, Computational time, Kappa, Low-dimensional feature vector, Sensitivity, Specificity

1. Introduction.

The brain computer interface (BCI) provides a new communication channel for subjects to interact with the external world without using their muscles. Electroencephalogram (EEG) based BCI systems analyze electrical brain activity recorded from electrodes placed on the subject's scalp and extract features to determine the intents of the user. Features are then translated into control signals that are used to control external devices (e.g., an electromechanical arm, a wheelchair) [1].

Input signals of an EEG based BCI system are naturally non-stationary, have poor signal to noise ratio, dependent on physical or mental tasks, and contaminated with various artifacts, such as electromyogram (EMG) and electrooculogram (EOG). All these disadvantages motivate the researchers substantially to improve the speed and accuracy of all components of the communication system between the brain and a BCI output device. Hence, it is significant to use optimal classification algorithm and low dimensional feature set to implement a fast and accurate brain computer interface system [2-4].

Selection of the most appropriate classifier is a critical problem in brain computer interface applications. In literature, several classification algorithms have been tested for specific features such as linear discriminate analysis (LDA) [5,6], k-nearest neighbor (k-NN) [7], support vector machines (SVM) [7], and neural networks [8]. Most of them have evaluated the performance of classifier just in terms of classification accuracy (CA). On the other hand, feature vector dimension influences classifier's performance [9]. So, in order to propose the most appropriate classifier, it is essential to predict properties of feature set, such as whether it is low or high dimension. There are also some studies in literature which compare the performance of different classifiers. Tran et al. reported the empirical comparison of the reduced multivariate model and its extensions using hyperbolic, ramp and step basis functions, SVM, k-NN and multilayer perceptron network (MLP) [10]. They considered classification accuracy, computational time and storage requirement metrics to evaluate performance of classifiers. Their results showed that SVM is the best classifier with an average accuracy rate of 83.69%. In another approach, Dixon and Brereton used six synthetic data sets to compare five different classifiers (Euclidean distance to centroids, LDA, Quadratic Discriminant Analysis, radial basis function kernel support vector machine (RBF-SVM) and Learning Vector Quantization) performance [11]. According to their results, they recommended to look at the data structure prior to model building to determine the suitable classifier. However, they did not test their proposed methods with a real data set. Lorena et al. investigated the use of nine supervised machine learning algorithms (repeated incremental pruning to produce error reduction, genetic algorithm for rule set production, decision trees, random trees, k-NN, naïve Bayes (NB), Logistic Regression, SVM and MLP) to model the potential distribution of 35 plant species from Latin America [12]. They calculated only AUC (Area under the ROC Curve) metric to evaluate the classifier's effectiveness. As a consequence of their results, random trees showed the best performance, while k-NN showed the worst. In another classifier based study, Furdea et al. compared four classification algorithms, including stepwise linear discriminant analysis, shrinkage linear discriminant analysis, linear support vector machine and RBF-SVM, with the aim to find a suitable classifier to distinguish 'yes-' or 'no-thinking' [13]. They achieved the highest classification accuracy of 68.8% with RBF-SVM. Among those studies none of them have investigated classifiers in terms of low-dimensionality and also they have generally used only CA metric to compare performances of classifiers. This paper, based on the theory of optimal classifier and fewer features, contributes to the literature a statistical approach for comparative performance assessment of five classifiers which are commonly used in EEG-based brain computer interface applications. We evaluate the performances of classifiers in different aspects including CA, sensitivity (SE), specificity (SP), Kappa (κ) and computational time (CT).

For our study, we used BCI Competition 2003 Data Set III and Data Set Ia which are described in the following section. For Data Set III we extracted six features by calculating alpha frequency (8-13 Hz), beta frequency (13-20 Hz) and total frequency band powers of the signals and for Data Set Ia three features were extracted by calculating gamma frequency band power of the signals. Then, we classified the signals with five classifiers including k-NN, SVM, LDA, NB and decision tree (DT).

The paper is organized as follows. Section 2 describes the materials and methods. The results are provided in Section 3. The conclusions and discussions are given in Section 4.

2. Materials and Methods.

2.1. Data set description.
Our algorithm is performed on the BCI Competition 2003 Data Set III, which was taken from a single healthy female subject at the University of Technology Graz, and Data Set Ia, which was taken from a single healthy subject at the University of Tuebingen. Following subsections described the data set in detail.

2.1.1. *Description of BCI Competition 2003 Data Set III.*
Brain activity was recorded with three bipolar EEG channels (C3, Cz, C4) with sampling frequency of 128 Hz and it was filtered between 0.5 and 30 Hz (Figure 1(a)). The recording length of a trial was

set to 9 seconds. The first 2 seconds were quiet. At $t = 2$ seconds, an acoustic stimulus indicated the beginning of the trial, and a cross ("+") was displayed for 1 seconds. Then, at $t = 3$ seconds, an arrow (left/right) was displayed as a cue. The subject was asked to use imagination of left or right hand movements to move the feedback bar into the direction of the cue. The order of left and right cues was random. During the experiment the subject sat in a relaxing chair with armrests. We decided to use the last 6 seconds, while the first 3 seconds is the preparation period in which no event happened.

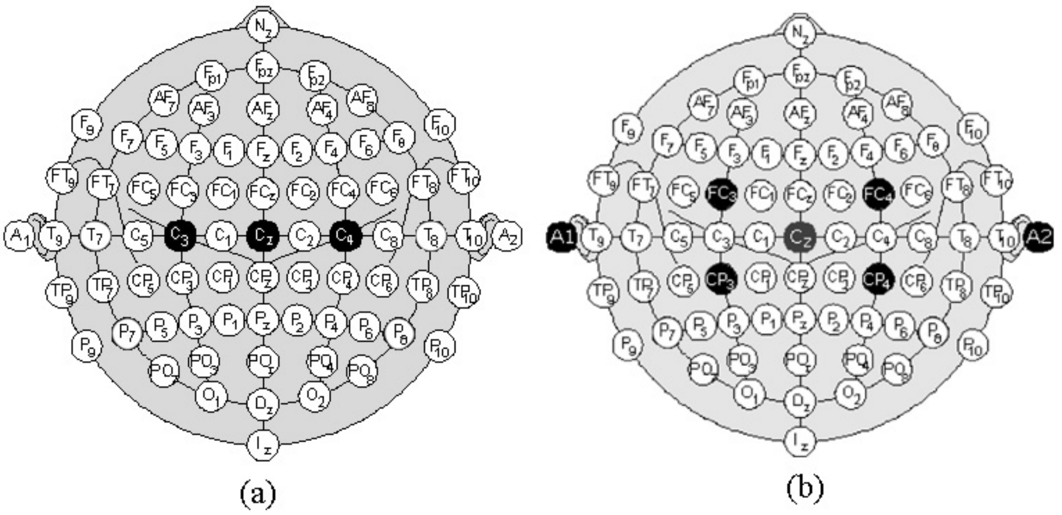

FIGURE 1. Montage of EEG electrodes as international 10-20 system. (a) Montage for Data Set III signals and (b) montage for Data Set Ia signals.

The experimental data set consists of 140 trials for training set (70 trials for right hand movement, RHM and 70 trials for left hand movement, LHM) and 140 trials for test set (70 trials for RHM and 70 trials for LHM). For further information about the data set, please refer to [14,15].

2.1.2. *Description of BCI Competition 2003 Data Set Ia.* The subject was asked to move a cursor up and down on a computer screen, while his/her slow cortical potentials (SCPs) were recorded. The subject received visual feedback of his/her SCPs, which were corrected for vertical eye movements. It is observed that cortical positivity (negativity) led to a downward (upward) movement of the cursor on the screen. Brain activity was recorded from six different channels with sampling frequency of 256 Hz. Six EEG electrodes are (A1, A2, FC3, CP3, FC4 and CP4) located according to the International 10-20 system as shown in Figure 1(b) and are referenced to the vertex electrode Cz. Each trial had duration of 6 seconds. During every trial, the task was visually presented by a highlighted goal at either the top or bottom of the screen to indicate up or down from 0.5 seconds onward until the end of the trial. The visual feedback was presented from 2nd second to 5.5 seconds. Only this 3.5-second interval is provided for training and testing in each trial.

The experimental data set consists of 268 trials for training set (135 trials for cursor up movement, CUM and 133 trials for cursor down movement, CDM) and 293 trials for test set (147 trials for CUM and 146 trials for CDM). For further information about the data set, please refer to [16].

2.2. **Feature extraction.** Feature extraction is a crucial step in BCI system, because its capability directly influences the performance of the classifier. However, it requires a lot of research to extract useful features using the existing feature extraction methods or from a newly developed method.

In this study power spectral density (PSD) technique is used as feature extraction method. This technique has always been a popular method for classifying EEG signals [17-19]. The first step in EEG classification is to determine if the signals have distinguishable features in their power spectrum. With a close examination, we observed that the alpha, the beta and the total band powers of the Data Set III signals, recorded from C3 and C4, show difference between left and right hand movement imaginations. On the other hand Mensh et al. showed that gamma band power (24-37 Hz) of the Data Set Ia signals can be used as features to classify CUM and CDM [20]. We used this band power in our paper for feature extraction.

In order to obtain band power (BP) of the signals, first we calculated the fast Fourier transform (FFT) coefficients as follows:

$$X(k) = \frac{1}{N}\sum_{n=0}^{N-1} x(n)e^{-j2\pi(k-1)(n-1)/N}, \quad k = 0, 1, \ldots, N \tag{1}$$

where N is number of EEG samples taken for analysis, $x(n)$ is the EEG signal, $X(k)$ is the kth FFT coefficient. Then, BP is computed by:

$$BP\Big|_{f_{LOW}}^{f_{UPPER}} = \sum \left\| X(k)\Big|_{f_{LOW}}^{f_{UPPER}} \right\|^2 \tag{2}$$

where $X(k)|_{f_{LOW}}^{f_{UPPER}}$ denotes FFT coefficients between low cutoff frequency (f_{LOW}) and upper cutoff frequency (f_{UPPER}). For the alpha band $f_{LOW} = 8$ Hz and $f_{UPPER} = 13$ Hz, for the beta band $f_{LOW} = 13$ Hz and $f_{UPPER} = 20$ Hz and for the total band $f_{LOW} = 0.5$ Hz and $f_{UPPER} = 30$ Hz (because EEG data set was filtered between 0.5 and 30 Hz when it was recorded). For the gamma band $f_{LOW} = 24$ Hz and $f_{UPPER} = 37$ Hz.

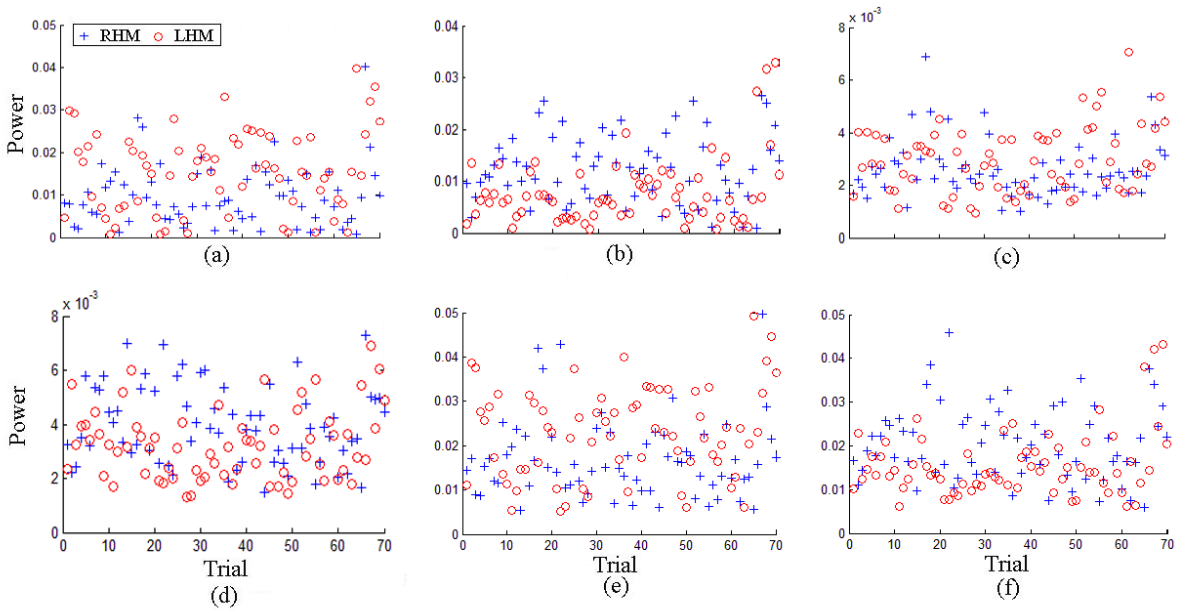

FIGURE 2. Band powers. (a) Alpha BP of channel C3, (b) alpha BP of channel C4, (c) beta BP of channel C3, (d) beta BP of channel C4, (e) total BP of channel C3 and (f) total BP of channel C4.

The calculated band powers of the training set of Data Set III are illustrated in Figure 2. In this figure , Figure 2(a) shows the alpha BP of channel C3, which is defined as feature 2 (f1), Figure 2(b) shows the alpha BP of channel C4, which is defined as feature 2 (f2), Figure 2(c) shows the beta BP of channel C3, which is defined as feature 3 (f3), Figure 2(d) shows the beta BP of channel C4, which is defined as feature 4 (f4), Figure 2(e) shows the total BP of channel C3, which is defined as feature 5 (f5), Figure 2(f) shows the total BP of channel C4, which is defined as feature 6 (f6). The horizontal axis is the index of trial numbers, and the vertical axis shows the values of power. Plus indicates the trials of RHM, and circle denotes the trials of LHM. As seen from the figure, the band powers for the RHM and the LHM trials have clustered in the opposite directions. Based on this strong clue, we considered these values can be selected as features to classify the two tasks.

The calculated gamma band powers of the training set of Data Set Ia are illustrated in Figure 3. In this figure , Figure 3(a) shows the gamma BP of channel A2, which is defined as feature 7 (f7), Figure 3(b) shows the gamma BP of channel CP3, which is defined as feature 8 (f8) and Figure 3(c) shows the gamma BP of channel CP4, which is defined as feature 9 (f9). The horizontal axis is the index of trial numbers, and the vertical axis is the values of power. Plus points show the trials of CUM, and circles denote the trials of CDM.

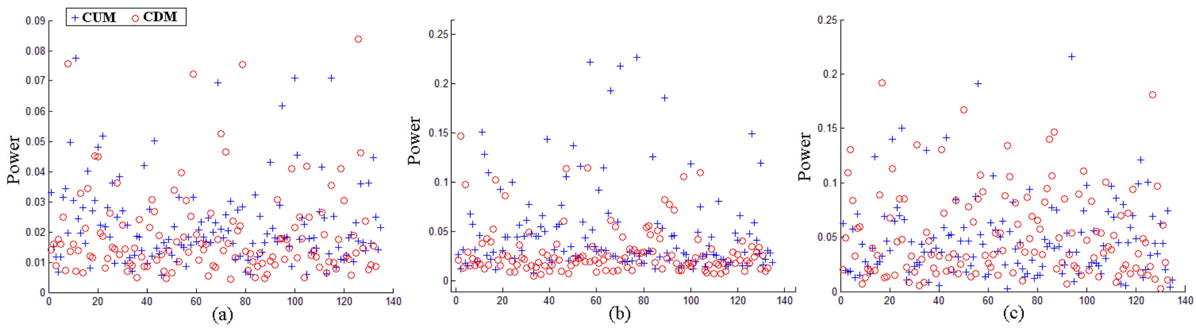

FIGURE 3. Band powers. (a) Gamma BP of channel A2, (b) gamma BP of channel CP3 and (c) beta BP of channel CP4.

2.3. **Classification algorithms.** A classifier is an algorithm which has to be trained with labeled training examples to be able to distinguish new unlabeled examples between a fixed set of classes. The general framework of training and testing process of the classification procedure is illustrated in Figure 4. From each trial, features are extracted to form a feature vector which is used as the representation of corresponding trial. Feature vector set is obtained by extracting features from training trials, and then used to train classifier. In the testing phase, trained classifier decides the class label according to extracted feature vector from corresponding test trial.

In our study we trained k-NN, SVM, LDA, NB and DT classification algorithms by using all training data set. In the following subsections, we briefly review aspects of the five classifiers.

2.3.1. *k-Nearest Neighbor.* The k-NN classifier is a common classification algorithm, which determines a testing sample's class by the majority class of the k closest training samples [21,22]. This is illustrated with a simple example in Figure 5, which shows data records, each with two attributes that are representations of two classes of data (blue and red). In this case $k = 5$. The unlabeled test trial would be labeled by the category of the class red, because four out of its five closest samples (neighbors) are red. It is worth the mention that the performance of a k-NN algorithm depends on the distance metric

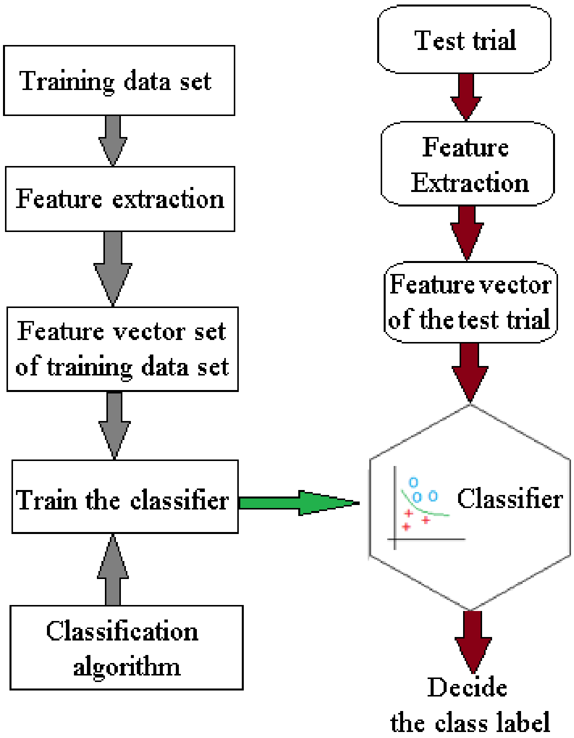

Figure 4. The general framework of training and testing process

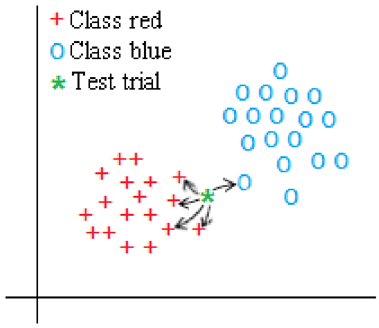

Figure 5. A simple example of the k-NN algorithm

and the value of k. In our study, we used Euclidean distance metric and leave-one-out cross-validation (LOOCV) technique to determine the best value of k to maximize the classification performance. The k value was searched in interval between 1 and 15, with step size of 1. Appendix describes in detail the LOOCV technique.

2.3.2. *Support vector machine.* An SVM performs classification tasks by constructing the best hyper plane in a multidimensional space by finding the maximum possible margin [23,24]. In this paper, the SVM classification framework is implemented by the following expression:

$$f(x) = sign\left(\sum_{i=1}^{T} \alpha_i y_i K(x, x_i) + b\right), \tag{3}$$

where $f(x)$ is the decision function, T is the number of trials, $\alpha_i \in R$ are the Lagrangian multipliers obtained by solving a quadratic optimization problem, $y_i \in \{1, -1\}$ are the class labels, b is the bias and $K(x, x_i)$ is the Kernel function. Although there are many alternatives for the Kernel function existing, we utilized most commonly used radial basis function kernel as:

$$K(x, x_i) = \exp\left(-\frac{\|x - x_i\|^2}{2\sigma^2}\right). \tag{4}$$

We have chosen this kernel due to the fact that the number of hyper parameters of this kernel is smaller than those of other kernels. This kernel function is specified by the scaling factor σ. To determine optimum value for the scaling factor, the same validation procedure is used in the k-NN classification algorithm. The most appropriate σ value was searched in interval between 0.1 and 1.5 (step size 0.1) with the same validation procedure used in the k-NN classification algorithm.

2.3.3. *Linear discriminant analysis.* LDA classifies two classes based on the assumption that both classes are under normal distribution with equal covariance matrices. The separating hyper plane is obtained by finding the projection of the labeled training data that maximizes the distance between the two classes' means and minimizes the interclass variance. The main aim is to solve the problem

$$y = w^T x + w_0, \tag{5}$$

where x is the feature vector. The vectors w and w_0 are determined by maximization of the interclass means and minimization of interclass variance [25].

2.3.4. *Naïve Bayes.* Naïve Bayes classifier is a simple probabilistic algorithm based on applying Bayes' theorem [26,27] with naïve independence assumptions. Consider a set of training trials where each trial is made up from m discrete-valued features and a class from a finite set C. The naïve Bayes classifier can probabilistically predict the class of an unknown trial using the available training trial set to calculate the most probable output. The most probable class C_{NB} of an unknown trial with the conjunction $A = a_1, a_2, \ldots, a_m$ is calculated by

$$C_{NB} = \arg\max_{c \in C} p(c \backslash A). \tag{6}$$

2.3.5. *Decision tree.* This algorithm constructs a decision tree with branch(es) and node(s) based on feature vector set. The decision tree begins with a root node r derived from whichever variable in the feature space minimizes a measure of the impurity of the two sibling nodes. The measure of the impurity at node r, denoted by $im(r)$, is as shown in the following equation:

$$im(r) = -\sum_{i=1}^{m} p(w_i \backslash r) \log p(w_i \backslash r). \tag{7}$$

where $p(w_i \backslash r)$ is the proportion of patterns x_i allocated to class w_i at node r. Each none-terminal node is then divided into two further nodes, r_1 and r_2, such that p_1, p_2 are the proportions of entities passed to new nodes r_1, r_2 respectively. The most appropriate division is that which maximizes the difference given in Equation (8).

$$\Delta im(d, r) = im(r) - p_1 im(r_1) - p_2 im(r_2) \tag{8}$$

The decision tree grows until a phase is reached in which there is no significant decrease in the measure of impurity when a further additional division d is implemented. When this phase is reached, the node r is not sub-divided further, and automatically becomes a terminal node. The class w_i, associated with the terminal node r is that which maximizes the conditional probability $p(w_i \backslash r)$. Eventually, in testing phase, test samples are classified using the calculated optimal decision tree model. For a detailed description of the method, please see [28]. In this paper, to construct the decision tree default configuration of the classregtree function in Statistics Toolbox, Matlab R2010b was used.

2.4. **Performance metrics for classifiers.**

2.4.1. *Classification accuracy.* If we define class labels of the binary (two-class) prediction problem as positive and negative, classifier has the following four possible outcomes:

- True positive (TP): The number of positive samples correctly predicted.
- True negative (TN): The number of negative samples correctly predicted.
- False positive (FP): The number of positive samples incorrectly predicted.
- False negative (FN): The number of negative samples incorrectly predicted.

Classification accuracy is defined as the percentage of the number of trials classified correctly in the test set over the total trials. It is calculated by:

$$CA = \frac{TP + TN}{TP + TN + FP + FN} \times 100 \tag{9}$$

In our study, we defined the RHM and the CUM imageries as the positive samples and the LHM and the CDM imageries as the negative samples.

2.4.2. *Sensitivity and specificity.* Sensitivity and specificity are calculated by the following formulae, respectively:

$$SE = \frac{TP}{TP + FN} \times 100 \tag{10}$$

$$SP = \frac{TN}{TN + FP} \times 100 \tag{11}$$

For the Data Set III the sensitivity refers to the ratio of correctly classified RHMs to the total population of RHM cases, whereas specificity is the ratio of correctly classified LHMs to the total population of LHM cases. On the other hand, for the Data Set Ia the sensitivity refers to the ratio of correctly classified CUMs to the total population of CUM cases, whereas specificity is the ratio of correctly classified CDMs to the total population of CDM cases.

2.4.3. *Kappa.* Kappa statistics is defined as the proportion of correctly classified samples after accounting for the probability of chance agreement. It is calculated by:

$$Kappa = \frac{P(D) - P(E)}{1 - P(E)} \tag{12}$$

where $P(D)$ denotes the proportion of overall agreement and $P(E)$ is the probability of expected agreement by chance. The Kappa coefficient value is ranged between 1 and -1, which corresponds to a perfect and a completely wrong classification, respectively. A Kappa coefficient with value 0 means that the performance is equal to random guess.

2.4.4. *Computational time.* We computed the training and testing times of the classifiers. All the runtime experiments were conducted on a PC with Intel ® Core ™ i7 CPU 1.73 GHz, 4 GB RAM.

3. **Results.** The BCI Competition 2003 Data Set III and Data Set Ia were tested with five classifiers using the different combinations of the features (57 combinations for BCI Competition Data Set III and 4 combinations for BCI Competition Data Set Ia). The results of the classifiers in terms of four metrics including CA, SE, SP and κ for Data Set III are given in Table 1. In the table, the best results of the metrics are written in boldface and averages of the four metrics are given in the last line. In case of using f1, f3, f4 and f6 features together, NB classifier provided the best CA, SP and κ performance which are 82.9%, 88.6% and 0.66, respectively. The best SE was obtained as 88.6% when f2 and f3 feature pair classified by using the k-NN algorithm. The worst case was obtained when f1 and f3 feature pair classified by using DT algorithm. In this case CA, SE, SP and κ were calculated as 52.9%, 61.4%, 44.3% and 0.06, respectively. According to the averages of four metrics NB classifier achieves the highest average values of CA, SP and κ which are 76.4%, 77.8% and 0.53, respectively. The highest average value of SE is obtained as 75.7% by k-NN algorithm.

The results of the classifiers in terms of the four metrics for Data Set Ia are given in Table 2. The best results of the metrics are written in boldface and averages of the four metrics are given in the last line. In case of using f7 and f8 features together, SVM classifier provided the best CA and κ performance which are 76.1% and 0.52, respectively. The best SE was obtained as 85.0% when f8 and f9 feature pair were classified using the k-NN algorithm. On the other hand, the best SP was obtained as 88.4% by LDA and NB classifiers. However, the worst case was obtained when f7 and f9 feature pair were classified using DT algorithm. In this case CA, SE, SP and κ were calculated as 59.0%, 56.5%, 61.6% and 0.18, respectively. According to the averages of the four metrics, SVM classifier achieves the highest average values of CA and κ which are 72.1% and 0.44, respectively. The highest average value of SE is obtained as 74.8% by k-NN algorithm. Additionally, the highest average value of SP is obtained as 86.3% by NB algorithm.

We also computed the computational times of the classifiers for training and testing stages. Table 3 presents the average computational times of the both stages (the values are given in seconds). It can be seen in the table that the fastest training and testing times were obtained as 0.005 CPU seconds with LDA and NB classifiers. Conversely, the slowest time for the training and testing stages was obtained by SVM.

Table 4 presents the most suitable classifiers for each metric according to the averages of the metrics on the data sets. To determine the winner of the classifier we counted the number of times each classifier was repeated throughout the each row. For the BCI Competition Data Set III, the NB was selected the most suitable classifier for four metrics (CA, SP, κ and CT). For the BCI Competition Data Set Ia, the NB and the SVM were selected the most suitable classifiers for two metrics (for the SVM; CA and κ, for the NB; SP and CT). Consequently, the overall winners of the classifiers are determined as the NB and the SVM.

4. **Conclusion and Discussion.** This paper evaluated the performances of five classifiers (k-NN, SVM, LDA, NB and DT) in different aspects including CA, SE, SP, κ and CT in low-dimensional feature vectors extracted from the BCI Competition 2003 Data Set III and Data Set Ia. The classifiers were compared on a total of 61 different datasets which were created with a combination of extracted features.

The results showed that NB algorithm achieved much better performance for Data Set III, however, SVM and NB achieved much better performance for Data Set Ia. The experiments proved that it is difficult to propose a firm classification algorithm. Based on the results from Tables 1 and 2, it seems selection of the most appropriate classifier highly depends on structure of the data set. Furthermore, when classifiers were ranked

Table 1. Performances of the classifiers for Data Set III

Features	**k-NN**				**SVM**				**LDA**				**NB**				**DT**			
	CA	**SE**	**SP**	**κ**	**CA**	**SE**	**SP**	**κ**	**CA**	**SE**	**SP**	**κ**	**CA**	**SE**	**SP**	**κ**	**CA**	**SE**	**SP**	**κ**
f1-f2	78.6	75.7	81.4	0.57	78.6	77.1	80.0	0.57	77.9	77.1	78.6	0.56	78.6	78.6	78.6	0.57	78.6	78.6	78.6	0.57
f1-f3	65.0	70.0	60.0	0.30	65.0	67.1	62.9	0.30	65.7	61.4	70.0	0.31	65.7	65.7	65.7	0.31	52.9	61.4	44.3	0.06
f1-f4	69.3	70.0	68.6	0.39	75.7	72.9	78.6	0.51	80.0	78.6	81.4	0.60	76.4	78.6	74.3	0.53	62.9	67.1	58.6	0.26
f1-f5	63.6	67.1	60.0	0.27	65.0	68.6	61.4	0.30	65.0	62.9	67.1	0.30	65.7	62.9	68.6	0.31	65.0	62.9	67.1	0.30
f1-f6	78.6	74.3	82.9	0.57	79.3	75.7	82.9	0.59	78.6	74.3	82.9	0.57	79.3	80.0	78.6	0.59	75.0	78.6	71.4	0.50
f2-f3	73.6	**88.6**	58.6	0.47	72.1	80.0	64.3	0.44	79.3	81.4	77.1	0.59	75.7	74.3	77.1	0.51	65.0	82.9	47.1	0.30
f2-f4	63.6	77.1	50.0	0.27	67.9	72.9	62.9	0.36	71.4	70.0	72.9	0.43	69.3	68.6	70.0	0.39	60.0	72.9	47.1	0.20
f2-f5	79.3	75.7	82.9	0.59	77.1	77.1	77.1	0.54	76.4	74.3	78.6	0.53	77.1	75.7	78.6	0.54	71.4	61.4	81.4	0.43
f2-f6	65.0	78.6	51.4	0.30	70.7	74.3	67.1	0.41	71.4	72.9	70.0	0.43	72.1	70.0	74.3	0.44	54.3	62.9	45.7	0.09
f3-f4	67.9	77.1	58.6	0.36	66.4	74.3	58.6	0.33	70.0	70.0	70.0	0.40	70.0	65.7	74.3	0.40	65.7	77.1	54.3	0.31
f3-f5	63.6	64.3	62.9	0.27	63.6	60.0	67.1	0.27	63.6	57.1	70.0	0.27	65.7	67.1	64.3	0.31	56.4	55.7	57.1	0.13
f3-f6	70.7	82.9	58.6	0.41	70.7	74.3	67.1	0.41	75.0	74.3	75.7	0.50	76.4	75.7	77.1	0.53	65.7	75.7	55.7	0.31
f4-f5	69.3	67.1	71.4	0.39	75.7	72.9	78.6	0.51	76.4	75.7	77.1	0.53	74.3	75.7	72.9	0.49	67.1	67.1	67.1	0.34
f4-f6	69.3	77.1	61.4	0.39	66.4	68.6	64.3	0.33	69.3	70.0	68.6	0.39	70.0	70.0	70.0	0.40	53.6	58.6	48.6	0.07
f5-f6	77.9	74.3	81.4	0.56	76.4	74.3	78.6	0.53	78.6	77.1	80.0	0.57	77.9	77.1	78.6	0.56	77.1	82.9	71.4	0.54
f1-f2-f3	79.3	78.6	80.0	0.59	79.3	80.0	78.6	0.59	77.9	77.1	78.6	0.56	81.4	82.9	80.0	0.63	75.7	80.0	71.4	0.51
f1-f2-f4	78.6	77.1	80.0	0.57	79.3	78.6	80.0	0.59	77.9	77.1	78.6	0.56	80.7	75.7	85.7	0.61	78.6	80.0	77.1	0.57
f1-f2-f5	77.9	74.3	81.4	0.56	79.3	78.6	80.0	0.59	77.9	77.1	78.6	0.56	77.1	78.6	75.7	0.54	77.1	74.3	80.0	0.54
f1-f2-f6	78.6	72.9	84.3	0.57	80.0	78.6	81.4	0.60	80.7	80.0	81.4	0.61	80.7	78.6	82.9	0.61	80.0	78.6	81.4	0.60
f1-f3-f4	70.0	70.0	70.0	0.40	72.9	74.3	71.4	0.46	77.9	78.6	77.1	0.56	75.7	78.6	72.9	0.51	65.7	70.0	61.4	0.31
f1-f3-f5	65.0	67.1	62.9	0.30	66.4	65.7	67.1	0.33	65.0	61.4	68.6	0.30	65.7	64.3	67.1	0.31	55.7	58.6	52.9	0.11
f1-f3-f6	78.6	75.7	81.4	0.57	78.6	80.0	77.1	0.57	79.3	74.3	84.3	0.59	80.0	80.0	80.0	0.60	77.1	77.1	77.1	0.54
f1-f4-f5	69.3	72.9	65.7	0.39	75.0	72.9	77.1	0.50	76.4	75.7	77.1	0.53	74.3	72.9	75.7	0.49	70.0	80.0	60.0	0.40
f1-f4-f6	79.3	75.7	82.9	0.59	79.3	78.6	80.0	0.59	78.6	74.3	82.9	0.57	80.7	75.7	85.7	0.61	75.0	78.6	71.4	0.50
f1-f5-f6	77.9	72.9	82.9	0.56	77.1	74.3	80.0	0.54	78.6	77.1	80.0	0.57	75.7	77.1	74.3	0.51	78.6	82.9	74.3	0.57
f2-f3-f4	72.1	85.7	58.6	0.44	72.1	78.6	65.7	0.44	75.0	75.7	74.3	0.50	71.4	68.6	74.3	0.43	62.9	77.1	48.6	0.26
f2-f3-f5	79.3	75.7	82.9	0.59	75.0	74.3	75.7	0.50	75.7	72.9	78.6	0.51	77.1	78.6	75.7	0.54	72.9	67.1	78.6	0.46
f2-f3-f6	71.4	85.7	57.1	0.43	75.0	80.0	70.0	0.50	78.6	78.6	78.6	0.57	73.6	72.9	74.3	0.47	65.0	77.1	52.9	0.30
f2-f4-f5	78.6	75.7	81.4	0.57	80.0	80.0	80.0	0.60	77.9	74.3	81.4	0.56	78.6	72.9	84.3	0.57	71.4	62.9	80.0	0.43
f2-f4-f6	69.3	84.3	54.3	0.39	67.9	75.7	60.0	0.36	67.9	70.0	65.7	0.36	70.0	68.6	71.4	0.40	60.0	71.4	48.6	0.20
f2-f5-f6	80.7	77.1	84.3	0.61	80.0	78.6	81.4	0.60	77.9	75.7	80.0	0.56	79.3	77.1	81.4	0.59	72.1	65.7	78.6	0.44
f3-f4-f5	69.3	70.0	68.6	0.39	72.1	72.9	71.4	0.44	75.7	75.7	75.7	0.51	73.6	74.3	72.9	0.47	65.7	71.4	60.0	0.31
f3-f4-f6	70.0	82.9	57.1	0.40	70.7	77.1	64.3	0.41	73.6	74.3	72.9	0.47	73.6	70.0	77.1	0.47	65.7	75.7	55.7	0.31
f3-f5-f6	78.6	75.7	81.4	0.57	77.9	77.1	78.6	0.56	78.6	77.1	80.0	0.57	80.0	82.9	77.1	0.60	74.3	72.9	75.7	0.49
f4-f5-f6	78.6	74.3	82.9	0.57	79.3	78.6	80.0	0.59	77.9	77.1	78.6	0.56	78.6	72.9	84.3	0.57	77.9	81.4	74.3	0.56
f1-f2-f3-f4	80.0	78.6	81.4	0.60	79.3	78.6	80.0	0.59	77.9	77.1	78.6	0.56	81.4	77.1	85.7	0.63	76.4	81.4	71.4	0.53
f1-f2-f3-f5	77.9	74.3	81.4	0.50	78.6	77.1	80.0	0.57	78.6	78.6	78.6	0.57	75.6	77.1	74.3	0.51	72.9	74.3	71.4	0.46
f1-f2-f3-f6	80.0	77.1	82.9	0.60	78.6	80.0	77.1	0.57	78.6	77.1	80.0	0.57	81.4	77.1	85.7	0.63	75.7	78.6	72.9	0.51
f1-f2-f4-f5	80.0	74.3	85.7	0.60	79.3	78.6	80.0	0.59	77.1	75.7	78.6	0.54	79.3	80.0	78.6	0.59	76.4	75.7	77.1	0.53
f1-f2-f4-f6	77.9	75.7	80.0	0.56	77.9	77.1	78.6	0.56	77.1	75.7	78.6	0.54	77.9	74.3	81.4	0.56	79.3	80.0	78.6	0.59
f1-f2-f5-f6	77.9	74.3	81.4	0.56	79.3	78.6	80.0	0.59	80.7	80.0	81.4	0.61	77.9	77.1	78.6	0.56	78.6	74.3	82.9	0.57
f1-f3-f4-f5	67.9	71.4	64.3	0.36	71.4	74.3	68.6	0.43	77.1	77.1	77.1	0.54	71.4	72.9	70.0	0.43	67.1	65.7	68.6	0.34
f1-f3-f4-f6	79.3	77.1	81.4	0.59	77.1	77.1	77.1	0.54	79.3	74.3	84.3	0.59	**82.9**	77.1	**88.6**	**0.66**	77.1	77.1	77.1	0.54
f1-f3-f5-f6	78.6	74.3	82.9	0.57	78.6	80.0	77.1	0.57	78.6	77.1	80.0	0.57	73.6	74.3	72.9	0.47	76.4	77.1	75.7	0.53
f1-f4-f5-f6	77.1	72.9	81.4	0.54	77.1	74.3	80.0	0.54	77.9	77.1	78.6	0.56	77.9	78.6	77.1	0.56	78.6	82.9	74.3	0.57
f2-f3-f4-f5	80.0	77.1	82.9	0.60	77.9	78.6	77.1	0.56	75.7	72.9	78.6	0.51	80.0	75.7	84.3	0.60	72.1	62.9	81.4	0.44
f2-f3-f4-f6	70.7	81.4	60.0	0.41	72.9	78.6	67.1	0.46	72.1	75.7	68.6	0.44	74.3	71.4	77.1	0.49	63.6	77.1	50.0	0.27
f2-f3-f5-f6	80.7	77.1	84.3	0.61	77.1	77.1	77.1	0.54	77.1	74.3	80.0	0.54	82.1	78.6	85.7	0.64	70.7	61.4	80.0	0.41
f2-f4-f5-f6	79.3	75.7	82.9	0.59	77.9	77.1	78.6	0.56	78.6	75.7	81.4	0.57	75.7	72.9	78.6	0.51	72.1	65.7	78.6	0.44
f3-f4-f5-f6	78.6	75.7	81.4	0.57	78.6	78.6	78.6	0.57	77.1	77.1	77.1	0.54	82.1	77.1	87.1	0.64	72.9	72.9	72.9	0.46
f1-f2-f3-f4-f5	78.6	74.3	82.9	0.57	78.6	80.0	77.1	0.57	78.6	78.6	78.6	0.57	79.3	81.4	77.1	0.59	73.6	75.7	71.4	0.47
f1-f3-f4-f5-f6	77.9	72.9	82.9	0.56	78.6	77.1	80.0	0.57	77.9	77.1	78.6	0.56	78.6	80.0	77.1	0.57	77.1	77.1	77.1	0.54
f1-f2-f4-f5-f6	78.6	75.7	81.4	0.57	78.6	77.1	80.0	0.57	80.7	82.9	78.6	0.61	81.4	77.1	85.7	0.63	77.9	75.7	80.0	0.56
f1-f2-f3-f5-f6	77.1	75.7	78.6	0.54	77.9	77.1	78.6	0.56	81.4	80.0	82.9	0.63	80.7	81.4	80.0	0.61	72.9	71.4	74.2	0.46
f1-f2-f3-f4-f6	78.6	75.7	81.4	0.57	78.6	80.0	77.1	0.57	77.1	75.7	78.6	0.54	80.0	75.7	84.3	0.60	76.4	80.0	72.9	0.53
f2-f3-f4-f5-f6	77.9	75.7	80.0	0.56	79.3	78.6	80.0	0.59	77.9	74.3	81.4	0.56	79.3	75.7	82.9	0.59	70.7	61.4	80.0	0.41
f1-f2-f3-f4-f5-f6	76.4	75.7	77.1	0.53	77.9	78.6	77.1	0.56	81.4	82.9	80.0	0.63	79.3	75.7	82.9	0.59	73.6	72.9	74.3	0.47
Averages	74.9	75.6	74.2	0.50	75.4	74.7	74.7	0.51	76.2	75.1	77.4	0.53	76.4	75.0	77.8	0.53	70.6	72.8	68.4	0.41

Table 2. Performances of the classifiers for Data Set Ia

Features	***k*-NN**				**SVM**				**LDA**				**NB**				**DT**			
	CA	**SE**	**SP**	κ	**CA**	**SE**	**SP**	κ	**CA**	**SE**	**SP**	κ	**CA**	**SE**	**SP**	κ	**CA**	**SE**	**SP**	κ
f7-f8	74.1	75.5	72.6	0.48	**76.1**	67.4	84.9	**0.52**	69.6	51.0	**88.4**	0.39	69.6	51.0	**88.4**	0.39	62.5	63.3	61.6	0.25
f7-f9	64.9	55.1	74.7	0.30	71.3	68.0	74.7	0.43	66.9	49.7	84.3	0.34	65.9	46.9	84.9	0.32	59.0	56.5	61.6	0.18
f8-f9	71.0	**85.0**	56.9	0.42	66.2	70.1	62.3	0.32	69.6	66.0	73.3	0.39	70.3	57.1	83.6	0.41	65.5	70.8	60.3	0.31
f7-f8-f9	71.8	83.7	59.6	0.43	74.7	80.3	69.2	0.50	72.4	60.5	84.3	0.45	72.0	55.8	**88.4**	0.44	62.1	72.8	51.4	0.24
Averages	70.5	74.8	66.0	0.41	72.1	71.5	72.8	0.44	69.6	56.8	82.6	0.39	69.5	52.7	86.3	0.39	62.3	65.9	58.7	0.25

Table 3. Average computational times of the classifiers

Stage	***k*-NN**	**SVM**	**LDA**	**NB**	**DT**
Training	5	27	0.005	0.005	0.04
Testing	0.02	0.04	0.005	0.005	0.02

Table 4. Winner of the classifiers

Data set	**CA**	**SE**	**SP**	κ	**CT**	**Winner(s)**
BCI Competition Data Set III	NB	*k*-NN	NB	LDA&NB	LDA&NB	**NB**
BCI Competition Data Set Ia	SVM	*k*-NN	NB	SVM	LDA&NB	**SVM&NB**

based on the average values of performance metrics as given in Table 4, we conclude that the NB and the SVM classifiers are shown to be good candidates for pattern classification for low dimensional feature vectors. However, it can obviously mention that DT provides the worst performance.

The experiments also showed that if a classifier has any tune parameter, it becomes time consuming especially in training phase. Compared with the other classifiers, based on the results from Table 3, although low-dimensional feature vectors are used, SVM takes much more time to be trained. The testing times of the k-NN and the SVM classification algorithms are 4 and 8 times longer than those of the LDA and the NB algorithms, respectively.

In our approach, we have selected the classifiers which are often used in BCI applications. We believe that this is a significant contribution in the field of classifier for BCI.

Acknowledgements. The authors would like to thank all the researchers who are associated with the data acquisition and processing of the data set. This work was supported by the Alcatel-Lucent Turkey and the TUBITAK (The Scientific and Technological Research Council of Turkey), Project No: EEEAG-110E035.

We thanks, Assistant Professor Muhammad Muddassir Malik from the Department of Computing School of Electrical Engineering and Computer Science, at the National University of Sciences and Technology (NUST), Pakistan for extensively proofreading this manuscript.

REFERENCES

[1] J. R. Wolpaw, N. Birbaumer, D. J. McFarland, G. Pfurtscheller and T. M. Vaughan, Brain-computer interfaces for communication and control, *Clin. Neurophysiol.*, vol.113, pp.767-791, 2002.

[2] M. Cheng, X. Gao, S. Gao and D. Xu, Design and implementation of a brain-computer interface with high transfer rates, *IEEE Trans. Biomed. Eng.*, vol.49, no.10, pp.1181-1186, 2002.

[3] G. Pfurtscheller et al., Self-paced operation of an SSVEP-based orthosis with and without an imagery-based "brain switch": A feasibility study towards a hybrid BCI, *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol.18, no.4, pp.409-414, 2010.

[4] I. Guler and E. D. Ubeyli, Multiclass support vector machines for EEG-signals classification, *IEEE Transactions on Information Technology in Biomedicine*, vol.11, no.2, pp.117-126, 2007.

[5] A. Schlogl, F. Lee, H. Bischof and G. Pfurtscheller, Characterization of four-class motor imagery EEG data for the BCI-competition, *J. Neural Eng.*, vol.2, no.14-22, 2005.

[6] D. Coyle, G. Prasa and T. M. McGinnity, A time-series prediction approach for feature extraction in a brain-computer interface, *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol.13, no.4, pp.461-467, 2005.

[7] M. Besserve, J. Martinerie and L. Garnero, Improving quantification of functional networks with EEG inverse problem: Evidence from a decoding point of view, *Neuroimage*, vol.55, no.4, pp.1536-1547, 2011.

[8] Hema C. R., Paulraj M. P., Nagarajan R. Sazali Yaacob, A. H. Adom, Brain machine interface: A comparison between fuzzy and neural classifiers, *International Journal of Innovative Computing, Information and Control*, vol.5, no.7, pp.1819-1827, 2009.

[9] S. Wang et al., A feature selection method based on improved fisher's discriminant ratio for text sentiment classification, *Expert Systems with Applications*, vol.38, no.7, pp.8696-8702, 2011.

[10] Q. L. Tran et al., An empirical comparison of nine pattern classifiers, *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol.35, no.5, pp.1079-1091, 2005.

[11] S. J. Dixon and R. G. Brereton, Comparison of performance of five common classifiers represented as boundary methods: Euclidean distance to centroids, linear discriminant analysis, quadratic discriminant analysis, learning vector quantization and support vector machines, as dependent on data structure, *Chemometr. Intell. Lab.*, vol.95, no.1, pp.1-17, 2009.

[12] A. C. Lorena et al., Comparing machine learning classifiers in potential distribution modeling, *Expert Systems with Applications*, vol.38, no.5, pp.5268-5275, 2011.

[13] A. Furdea et al., A new (semantic) reflexive brain-computer interface: In search for a suitable classifier, *Journal of Neuroscience Methods*, vol.203, no.1, pp.233-240, 2012.

[14] S. C. Lin, Y. C. I. Chang and W. N. Yang, Meta-learning for imbalanced data and classification ensemble in binary classification, *Neurocomputing*, vol.73, no.1-3, pp.484-494, 2009.

[15] C. Neuper, A. Schlögl and G. Pfurtscheller, Enhancement of left-right sensorimotor EEG differences during feedback-regulated motor imagery, *J. Clin. Neurophysiol.*, vol.16, no.4, pp.373-82, 1999.

[16] B. Blankertz et al., The BCI competition 2003, *IEEE Trans. Biomed. Eng.*, vol.51, pp.1044-1051, 2004.

[17] L. Svoboda, A. Stancak and P. Sovka, Detection of cortical oscillations induced by SCS using power spectral density, *Radioengineering*, vol.16, no.4, pp.38-45, 2007.

[18] S. Shahid and G. Prasad, Bispectrum-based feature extraction technique for devising a practical brain-computer interface, *J. Neural Eng.*, vol.8, no.2, 2011.

[19] S. Aydin, Determination of autoregressive model orders for seizure detection, *Turk. J. Electr. Eng. Comput. Sci.*, vol.18, no.1, pp.23-30, 2010.

[20] B. D. Mensh, J. Werfel and H. S. Seung, BCI competition 2003 – Data set Ia: Combining gamma-band power with slow cortical potentials to improve single-trial classification of electroencephalographic signals, *IEEE Trans. Biomed. Eng.*, vol.51, pp.1052-1056, 2004.

[21] R. O. Duda, P. E. Hart and D. G. Stork, *Pattern Classification*, Wiley, New York, 2001.

[22] J.-Y. Jiang and S.-J. Lee, FMLKNN: A fuzzy membership function based k-nearest neighbor approach for multi-label classification, *ICIC Express Letters*, vol.5, no.4(A), pp.1069-1075, 2011.

[23] V. N. Vapnik, *Statistical Learning Theory*, Wiley, New York, 1998.

[24] C. Shi, Y. Wang and H. Zhang, Faults diagnosis based on support vector machines with parameters optimization, *ICIC Express Letters, Part B: Applications*, vol.2, no.6, pp.1245-1252, 2011.

[25] K. R. Muller, C. W. Anderson and G. E. Birch, Linear and nonlinear methods for brain-computer interfaces, *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol.11, no.2, pp.165-169, 2003.

[26] T. M. Mitchell, *Machine Learning*, McGraw-Hill, Singapore, 1997.

[27] A. Adam, M. I. Shapiai, Z. Ibrahim, M. Khalid and L. W. Jau, Development of a hybrid artificial neural network – Naïve bayes classifier for binary classification problem of imbalanced datasets, *ICIC Express Letters*, vol.5, no.9(A), pp.3171-3175, 2011.

[28] L. Breiman et al., *Classification and Regression Trees*, Chapman & Hall, Boca Raton, 1993.

Appendix. Leave-One-Out Cross-Validation. In LOOCV technique, the training phase is performed using T-1 trials, where T is the total number of trials, and the validation is carried out using the excluded trial. If this trial is misclassified an error is counted. This is repeated T times, each time excluding a different trial. On the other hand, because we had 140 trials for Data Set III, for 140 times, 1 of the trial-data was chosen as the validation trial and the classifier was trained by the remaining 139 trials, and then the trained classifier was applied on the 1 validation data. The same procedure was also applied for Data Set Ia.

We used LOOCV technique to estimate the most appropriate classifier parameter, which provides the highest average value of the four metrics (CA, SE, SP and κ) result on validation set. Average of the four metrics was computed as follows:

$$Performance_{Avg} = \frac{CA + SE + SP + \text{Kappa} \times 100}{4}. \tag{13}$$

We utilized LOOCV, since it makes the best use of the available data and avoid the problems of random selections.