Performance Analysis of Ensemble Methods for Multi-class Classification of Motor Imagery EEG Signal

Saugat Bhattacharyya*, Amit Konar*, D.N. Tibarewala†, Anwesha Khasnobish† and R. Janarthanan‡

*Dept. of Electronics & Telecommunication Engineering
Jadavpur University, Kolkata, India-700032
Email: saugatbhattacharyya@live.com, konaramit@yahoo.co.in

†School of Bioscience & Engineering
Jadavpur University, Kolkata, India-700032
Email: biomed.ju@gmail.com, anweshakhasno@gmail.com

‡Computer Science Department
TJS Engineering College, Chennai, India
Email: srmjana_73@yahoo.com

***Abstract*—Recent advances in the field of Brain-computer Interfacing (BCI) has opened wide potentials in neuro-rehabilitative applications. Electeroencephalography (EEG) is the most frequently used brain measure in BCI research. Mental states are distinguished from classifiers which uses features extracted from the raw EEG as inputs. Ensemble classifiers combine a number of classifiers or learners to improve the classification results. It is more suited for multi-class classification of time-varying EEG signal. In this paper, we have used AdaBoost, LPBoost, RUSBoost, Bagging and Random Subspaces for classification of 3-class motor imagery EEG data. For this purpose, we have employed adaptive autoregressive coefficients as features and feed forward neural network (FFNN) as the base learner of the ensemble methods. The results show that the classification accuracies of the ensemble classifiers except RUSBoost performs better than a single FFNN classifier.**

***Keywords*—Motor imagery, Electroencephalography, Multi-class classification, Ensemble methods, Adaptive Autoregressive Parameter, Feed Forward Neural Network.**

I. Introduction

Recent advances in brain-computer interfacing (BCI) have successfully decoded brain signals of a person to their corresponding mental states. Movement related brain signals, also known as *motor imagery* signals, are one of the commonly researched brain states in BCI [1] which aims to provide rehabilitation to persons with physical disabilities, like amyotropic lateral sclerosis, cervical spinal injury, paralysis and amputee [2]. Researchers have successfully decoded left-hand, right-hand, foot and tongue imagery [3] to drive mobile [4] and humanoid [5] robots, and wheelchairs [6], [7] for such patients. Electroencephalography (EEG) is the most commonly used brain measure by researchers in BCI because it is non-invasive, easy-to-use, easy availability, portability and good temporal resolution [1], [8].

BCI technology is primarily composed of three components: i) pre-processing, ii) features extraction, and, iii) classification [9]. The pre-requisite of any BCI-driven device is the good recognition rate of the classifiers for the features extracted. EEG being a non-stationary, non-gaussian complex signal [8], requires the use of various time- , frequency- , time-frequency and non-linear signal processing algorithms, like wavelet transforms [10], band power estimates [11], adaptive autoregressive parameters [12], hjorth parameters [13] and approximate entropy [14], to extract relevant information from the raw data. These information, known as *features* are fed as inputs to the classifiers, which produces the corresponding brain states as the output. Classifiers like support vector machine (SVM), linear discriminant analysis (LDA), nave Bayesian (NB), neural networks (NN) and k-nearest neighbor (kNN) [15], are known to yield good recognition accuracy for two class classification, but their performance are not at par for multi-class classification. To improve the results on multi-class classification, techniques like one-against-one (OAO), one-against-all (OAA) and error correction code (ECC) [1] were implemented with the classifiers.

As the number of classes in a multi-class problem rises, the number of training sets (of high dimensionality) becomes comparatively smaller. It is known that classifiers trained on small training set becomes biased and has large variance due to the insufficient estimation of related parameters and thus, such classifiers are termed as '*weak*'. Ensemble classifier builds many such *weak* classifiers, known as *base learners* and combines the results of these classifiers to yield an outcome [16]. Some commonly used ensemble methods are *Bagging*[17], *Boosting*[18] and *Random Subspaces*[19].

Ensemble methods are suited for EEG classification for the following two reasons. First, the dimensionality of the EEG is often high and one of the pre-requisites of BCI is to train the classifier as fast as possible, thus, the training set also must be small. Second, EEG is a time-varying signal, and thus, it becomes hazardous to employ a single trained classifier to recognize the classes of the unknown (incoming) features [16]. In spite of these advantages, ensemble studies has yet to gain a foothold in BCI research and very few studies exists on this matter. In this paper, we compare the performance of some standard ensemble methods: *AdaBoost*, *LPBoost*, *RUSBoost*, *Bagging* and *Random Subspaces*, to decode three motor imagery tasks: elbow-, shoulder- and finger-movement, from their respective EEG signals. This comparison further facilitates the selection of appropriate ensemble method for future use in multi-class EEG classification. Here, we have employed *Adaptive Autoregressive Parameter* for features extraction and *Feed Forward Neural Network* as the base learner.

The rest of the paper is divided into the following sections: Section II gives a brief description of the ensemble methods employed in this study. Section III describes the experiments undertaken to acquire the EEG signal and its resultant features. Section IV gives a comparison on the performance of the ensemble methods employed in this study. Concluding remarks are mentioned in Section V.

II. Ensemble Classifiers

To measure and compare the performance of the ensemble classifiers, we have employed feed forward neural network as the common base learners. In this section, we review the ensemble algorithms and the base learner employed in this study.

A. AdaBoost

The AdaBoost (adaptive boosting) family of algorithm, developed by Freund and Schapire [20], is the most influential boosting algorithm. Here, the performance of the *weak (base) learners* are enhanced effectively by calling the learner algorithm repeatedly of different distribution of the training data, specifically the weights of each training data. Initially the weights of each training data are uniform. After each iterations, the easily classified patterns are assigned lower weights and the difficult patterns are assigned higher weights, thus, increasing the focus of the learners towards the difficult ones. After every iterations the base learners prepares a new prediction rule and after N iterations, N prediction rules are prepared to construct the final distance discriminant, by which the unknown patterns can be recognized. The final prediction rule is equal to the weighted majority vote of all predictors and the final accuracy of the classifier is effectively boosted. In this study, we have employed the AdaBoost.M2 extension which employs the one-against-one strategy for classification and it minimizes the pseudo-loss of the whole process.

B. LPBoost

LPBoost (linear programming boosting) introduced by Demiriz *et al.* [21], is a variant of AdaBoost Algorithm. It performs multiclass classification by attempting to maximize the minimal margin of the training set so that a low generalization error is obtained. LPBoost maximizes the minimal margin iteratively through a sequence of linear programming problems. LPBoost typically creates ensembles with many learners having weights that are of smaller order of magnitude from other learners. At each iteration, the optimization problem becomes increasingly constrained and thus, slow to solve.

C. RUSBoost

RUSBoost (random under sampling boosting) [22] is a hybrid data sampling/boosting algorithm designed to improve the performance of learners trained on skewed (unbalanced) data. It employs random undersampling technique which randomly removes data from the majority class. Initially, the weights of each training data are selected to be $1/m$, where m is the number of training instances. Then, each learner is iteratively trained as follows. First, random undersampling is applied to remove $T\%$ of the majority class, until it becomes a minority in the new temporary training set and a new weight distribution is prepared. Both the new training set and weight distribution is passed to the weak learner and the pseudo-loss is calculated. Next, the weight distribution is updated by the pseudo-loss and normalized and the updated weights are used for the next iteration. After N iterations, a weighted majority vote of the learners is the final result.

D. Bagging

The Bagging predictor (bootstrap aggregrating) proposed by Breiman [17] integrates the bootstrap sampling technique to manipulate training data. Bootstrap sampling [23] is employed to obtain the training subsets for training the base learner. At each iteration, Ntrain samples are selected randomly with replacement from the original training set of Ntrain samples to learn an individual classifier. Uniform majority voting of classifiers are aggregated to predict the test sample of an ensemble.

E. Random Subspaces

Random Subspaces, introduced by Ho [19], constructs individual classifiers from randomly selected feature subspaces. This method solves the problem of curse of dimensionality and thus, is applicable for high dimensional dataset. This method compensates for the possible deficiency of accuracies and thus merits to a high ensemble diversity [19]. In this method, feature subspaces are selected at random from the original feature space, and individual classifiers are created based on those attributes of the feature subspaces. The outputs from each individual classifiers are combined by uniform majority voting to yield the final prediction.

F. Base Learner: Feed Forward Neural Network

Neural Networks [24] mimics the working of the biological neuron for automated pattern recognition applications. Feed Forward Neural Networks (FFNN) consists of three types of layers: input, hidden and output layers. The artificial neurons in each layer are connected to the neurons of the next layer only in the forward direction, i.e., signals from the i^{th} layer can only propagate to layers greater than i. The neurons in the hidden and output layers receive weighted output from neurons of the previous layers. The weights are adjusted after each learning iterations so that the error between observed and desired output is minimized. For the purpose of this study, the number of hidden layers selected for all ensembles is 10.

III. Experiments and Methods

The experiments designed for this study required the subject to imagine moving their index finger, elbow and shoulder, when instructed by a visual cue. Seven right-handed subjects (four female and three male), in the age group of 255 years, performed the experiment in a single session of 90 trials (30

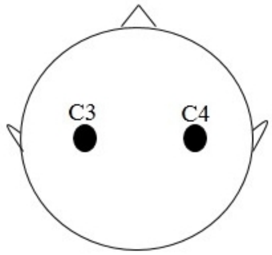

Fig. 1. Electrode locations of C3 and C4 electrode, based on the International 10-20 Electrode System.

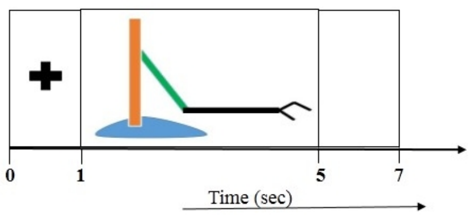

Fig. 2. Timing scheme for a trial in the visual stimuli. In this example, the subject is instructed to imagine moving his shoulder.

trials for each movement). The motor imagery signals from the subjects were recorded using a 19 channel EEG amplifier (NeuroWin, Make-NASAN). Based on the nature of the experiment, we have selected the C3 and C4 electrodes (Fig. 1) for our study because these electrode locations coincides with the movement activation areas (primary motor cortex, supplementary motor area and pre-motor area) of the brain [25]. Further in this section, we discuss about the visual stimuli designed and the features employed in this study.

A. Design of the visual stimuli

The generic structure of the visual stimuli are as follows: In the first 30 seconds of a session, the subject is asked to relax during which the baseline EEG of the subject is recorded, which is followed by 90 trials of 7 seconds each. Each trial begins with a fixation '+' for 1 second, as an instruction to the subject to get ready and focus on the screen. Then, the subject is instructed to imagine moving their index finger, elbow or shoulder of their right hand for 4 seconds based on the visual cue display on the screen. Each trial ends with a blank screen for 2 seconds during which the subject is asked to relax. An example of the timing scheme of a trial is shown in Fig. 2.

B. Filtering the EEG signal

It is known from standard literature [1] that motor imagery signals are dominant in the alpha (8-12 Hz) and central beta (16-24 Hz) band. Thus, for this study, we have designed an IIR elliptical filter of bandwidth 8-24 Hz to filter the EEG signals acquired from the amplifier. An elliptical filter is selected because it has good frequency domain characteristics of sharp roll off and good attenuation of the pass- and stop-band ripples.

C. Feature Extraction: Adaptive Autoregressive Parameter

An autoregressive model (AR) is suitable for stationary signals and thus, it is not a suitable for EEG feature extraction

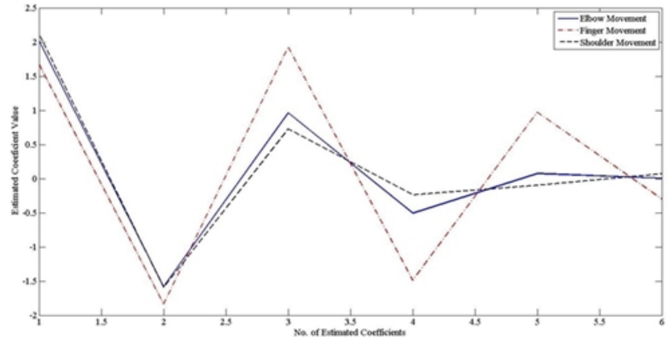

Fig. 3. Average AAR estimates of the EEG data obtained from channel C3 for finger(-.), elbow(-) and shoulder (–) movement over 10 instances.

as it is a non-stationary signal. For extraction of EEG features, an AR model is allowed to vary with time which is known as the Adaptive Autoregressive model (AAR) [26]. An AAR model is defined as follows,

$$y_k = a_{1,k}y_{k-1} + a_{2,k}y_{k-2} + \ldots + a_{p,k}y_{k-p} + X_t \quad (1)$$

where, X_t is a pure random noise process with zero mean and variance σ_x^2, p is the model order and $a_{i,k}$ are the time-varying AR parameters, or adaptive autoregressive parameters.

An AR model assumes the EEG to be the filtered white noise X_t. X_t is the new input to the model, and the past p samples is used to calculate the rest of the equation. Thus X_t is called the innovation process and is orthogonal to all past values. In practice, the AAR parameter a_k are only estimated values $\hat{a}_k$. If the estimates are near true value, the prediction error will be close to the innovation process, i.e.

$$e_k = y_k - \hat{a}_{k-1}^T y_{k-1} \quad (2)$$

Hence, the prediction error is independent of all previous samples y_{k-i}, $i > 0$. Based on these assumptions, AAR estimation algorithms are available. For this study the Least Mean Square approach [27] was implemented for estimation and the estimated AAR coefficients $\hat{a}_k$ are used as the features. After extensive experimentation, the order p of AAR model is selected as 6. Thus, the feature vector is arranged in the following fashion: 90 trials $\times$ 2 electrodes $\times$ 6 coefficients. Fig. 3 and 4 indicates the average of the AAR estimates over 10 instances of the three movements: finger, elbow and shoulder for the two electrodes C3 and C4, respectively.

IV. Results and Discussions

The features prepared in the previous section are fed as inputs to the ensemble classifiers. The aim of the classifiers is to discriminate the incoming features into one of the three motor imagery classes: finger movement, elbow movement and shoulder movement. The performance of the classifiers are measured over a test sample for each dataset using two metrics: classification accuracy and computational time taken (C.T.). For this computation of classification accuracy, we have used *k*-fold cross-validation technique [15], where the total dataset is divided into *k* different partitions. For *k* iterations, the *k*-th

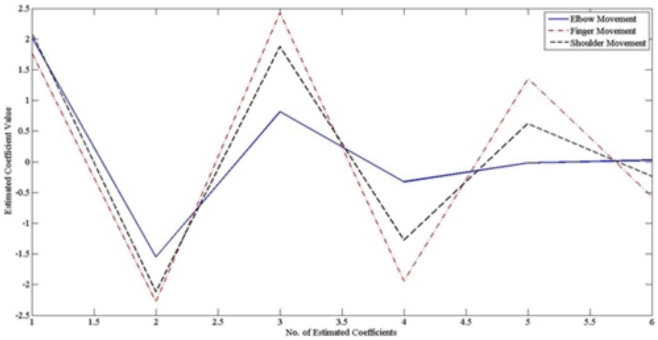

Fig. 4. Average AAR estimates of the EEG data obtained from channel C4 for finger(-.), elbow(-) and shoulder (–) movement over 10 instances.

TABLE I. COMPARISON OF THE CLASSIFICATION ACCURACIES AND COMPUTATIONAL TIME TAKEN

Subject ID	ADA	LP	RUS	Bag	RS	FFNN
1	**97.78**	94.44	64.44	87.78	87.78	66.67
2	**90.00**	94.44	56.67	82.22	82.22	61.11
3	**90.00**	83.33	65.56	71.11	71.11	70.00
4	**96.67**	65.56	65.56	88.89	88.89	65.67
5	**87.78**	81.11	61.11	72.22	72.22	60.00
6	**78.89**	72.22	57.78	58.89	58.89	57.78
7	78.89	**82.22**	48.89	68.89	68.89	55.00
Mean	**88.57**	81.90	60.00	75.71	75.71	62.32
C.T. (sec)	9.95	10.00	9.97	9.79	9.97	8.45

partition of the dataset is used as test dataset and the rest of the *(k-1)*-th partition is used as training dataset. The average of the *k* accuracies of the classifier is used for comparison among other classifiers in this study.

Table I shows the comparison of the average classification accuracies of the five ensemble classifiers: AdaBoost (ADA), LPBoost (LP), RUSBoost (RUS), Bagging (Bag), Random Subspaces (RS) and a single FFNN classifier, over k=10. As noted from Table I, all the ensemble classifiers except RUSBoost yields better result than a single FFNN classifier. Also, AdaBoost yields the best results for all the datasets and shows a rise of 26.25% from a single FFNN classifier.

The computational speed for this study has been measured in MATLAB version 7.9 environment. The specification of the system where the computations of the experiment took place are as follows: Processor- Intel Core i7, 3.40 GHz and 4 GB RAM. The average computational time taken (C.T.) by the each classifiers, measured for a single test sample is given in Table I. It is noted from Table I that AdaBoost takes the minimum amount of time to perform the computations among the other ensembles.

We have statistically validated our result by employing Friedman Test [27]. Here, this test compares the relative performance of the different ensemble classifier employed. The null hypothesis here, states that all the algorithms are equivalent, so their ranks r_j should be equal. The Friedman statistic, is distributed accordingly to χ_F^2 with $k-1$ degrees of freedom.

$$\chi_F^2 = \frac{12N}{k(k+1)}[\sum_j r_j^2 - \frac{k(k+1)^2}{4}] \tag{3}$$

where, k is number of classifers for comparison and N is the number of datasets. It is noted from Table I that here, k=6 and N=7 and we consider the classification accuracies to be the basis for ranking, which is shown in Table II.

TABLE II. COMPARISON OF THE CLASSIFICATION ACCURACIES AND COMPUTATIONAL TIME TAKEN

Subject ID	ADA	LP	RUS	Bag	RS	FFNN
1	1	2	6	3.5	3.5	5
2	1	2	6	3.5	3.5	5
3	1	2	6	3.5	3.5	5
4	1	5.5	5.5	2.5	2.5	4
5	1	2	5	3.5	3.5	6
6	1	2	5.5	3.5	3.5	5.5
7	2	1	6	3.5	3.5	5
r_j	1.14	2.36	5.71	3.35	3.35	5.07

Now, from Table II, we obtain $\chi_F^2 = 28.246$ ¿ $\chi^2_{6,0.05} = 12.592$. So, the null hypothesis, claiming that all the algorithms are equivalent, is wrong to a level of 5% confidence interval and, therefore, the performances of the algorithms are determined by their ranks only. It is clear from Table II, the rank of AdaBoost is 1, claiming AdaBoost outperforms all the other classifier algorithms.

V. CONCLUSION

This paper presents a performance analysis on the following ensemble techniques: AdaBoost, LPBoost, RUSBoost, Bagging and Random Subspaces, with Feed Forward Neural Network as the base classifier and Adaptive Autoregressive Parameters as the features. The results, thus obtained, shows a significant improvement in the accuracy when compared to a single FFNN classifier. AdaBoost yields the best result in terms of classification accuracy (88.57%) and computational time (9.95 sec). The results suggest that ensemble methods significantly improves the accuracies during multi-class classification of non-stationary signals like EEG. Further study in this direction will aim to optimize the feature selection, extraction and classification techniques to be implemented in real time application of Brain-Computer Interfacing.

ACKNOWLEDGMENT

The author would like to thank University Grants Commission, India, University of Potential Excellence Programme (Phase II) in Cognitive Science, Jadavpur University and Council of Scientific and Industrial Research, India.

REFERENCES

[1] G. Townsend, B. Graimann, and G. Pfurtscheller, "Continuous EEG classification during motor-imagery simulation of an asynchromous BCI," *IEEE Trans. Neural Sys. Rehab Engg.*, vol. 12, no. 2, pp. 258-265, 2004.

[2] G. Dornhege, Towards Brain-Computer Interfacing, MIT Press, 2007.

[3] C. Brunner, R. Scherer, B. Graimann, G. Supp, and G. Pfurtscheller, "Online control of a brain-computer interface using phase synchronization," *IEEE Trans. Biomed. Engg.*, vol. 53, no. 12, pp. 2501-06, 2006.

[4] S. Bhattacharyya, A. Sengupta, T. Chakraborti, D. Banerjee, A. Khasnobish, A. Konar, D.N. Tibarewala, R. Janarthanan, "EEG controlled remote robotic system from motor imagery classification," In: *3rd Int. conf on Com., Comm. & Networking Tech. (ICCCNT) 2012*, pp. 1-8, 26-28 July 2012.

[5] Y. Chae, J. Jeong, and S. Jo, "Toward brain-actuated humanoid robots: Asynchronous direct control using an EEG-based BCI," *IEEE Trans. Robotics*, vol. 28, no. 5, pp. 1131-1144, 2012.

[6] J. Long, Y. Li, H. Wang, T. Yu, J. Pan, and F. Li, “A hybrid brain computer interface to control the direction and speed of a simulated or real wheelchair,” *IEEE Trans. Neural Sys. Rehab. Eng.*, vol. 20, no. 5, pp. 720-729, Sept. 2012.

[7] F. Galan, M. Nuttin, E. Lew, P.W. Ferrez, G. Vanacker, J. Philips, and J. del R. Millan, “A brain-actuated wheelchair: asynchronous and non-invasive brain-computer interfaces for continuous control of robots,” *Clin. Neurophy.*, vol. 119, pp. 2159-2169, 2008.

[8] S. Sanei, and J.A. Chambers, EEG Signal Processing, John Wiley & Sons, 2008.

[9] S.G. Mason, and G.E. Birch, “A General Framework for Brain-Computer Interface Design”, *IEEE Trans. Neural Sys. & Rehab. Eng.*, vol. 11, no. 1, pp. 70-85, 2003.

[10] T. Yamaguchi, M. Fujio, K. Inoue, and G. Pfurtscheller, “Wavelet Analysis of EEG signals during Motor Imagery,” In: *Int Conf. Wavelet Analysis & Pattern recog. 2008*, pp. 454-59, Aug 2008.

[11] P. Herman, G. Prasad, T.M. McGinnity, and D. Coyle, “Comparative Analysis of Spectral Approaches to Feature Extraction for EEG based Motor Imagery Classification,” *IEEE Trans. Neural Sys. Rehab. Engg.*, vol. 16, no. 4, pp. 317-326, 2008.

[12] A. Schlogl, “Dynamic Spectral Analysis based on Auto-regressive model with time-varying coefficients,” *IEEE-EMBC and CMBEC*, 1995.

[13] C. Vidaurre, N. Kramer, B. Blankertz, and G. Pfurtscheller, “Time Domain Parameters as a feature for EEG-based brain-computer interfaces,” *Neural Networks*, vol. 22, pp. 1313-1319, 2009.

[14] Y. Fusheng, H. Bo, and T. Qingyu, Approximate entropy and its application to bio-signal analysis, In: Nonlinear Biomedical Signal Processing, Vol 2: Dynamic Analysis and Modeling, M.Akay, Ed., IEEE Press Series on Biomedical Engineering, NY (USA), pp. 72-92.

[15] E. Alpaydin, Introduction to machine learning, MIT Press, 2004.

[16] S. Sun, C. Zhiang, D. Zhang, “An experimental evaluation of ensemble methods for EEG signal classification,” *Pattern Recognition Letters*, vol. 28, pp. 2157-2163, 2007.

[17] L. Breiman, “Bagging Predictors,” *Machine Learning*, vol. 24, no. 2, pp. 123-140, 1996.

[18] P. Xiaomei, Z. Chongxun, X. Jin, and B. Guangyu, “Adaboost for improving classification of left and right hand motor imagery tasks,” In: *1st Int. Conf. on Neural Interface and Control*, Wuhan, China, pp. 10-13, May 2005.

[19] T.K. Ho, “The random subspace method for constructing decision forests,” *IEEE Trans Pattern Analysis Mach. Int.*, vol. 20, no. 8, pp. 832-844, 1998.

[20] Y. Freund, and R.E. Shapire, “A decision-theoretic generalization of online learning and an application to boosting,” *J. Comput. Syst. Sci.*, vol. 55, no. 1, pp. 119-139, 1997.

[21] A. Demiriz, K.P. Bennett, and J. Shawe-Taylor, “Linear programming boosting via column generation,” *J. Machine Learning*, vol. 46, no. 1-3, pp. 225-254, 2002.

[22] C. Seiffert, T.M. Khoshgoftaar, J.V. Hulse, and A. Napolitano, “RUSBoost: Improving classification performance when training data is skewed,” In: *19th Int. Conf. on Pattern Recognition (ICPR 2008)*, Tampa, FL, pp. 1-4, Dec. 2008.

[23] B. Efron, and R. Tibshirani, An introduction to the bootstrap. Boca Raton, FL: Chapman & Hall/CRC, 1993.

[24] S. Haykin, Neural Networks: A comprehensive foundation, 2nd ed., Prentice Hall, 1998.

[25] J.C. Tamraz, and Y.G. Comair, Atlas of regional anatomy of brain using MRI, Springer Berlin-Heidelberg, pp. 117-138, 2006.

[26] A. Schlogl, The Electroencephalogram and the Adaptive Autoregressive model: Theory and Applications, Doctoral thesis work submitted to vorgelegt an der Technischen Universitat Graz, 2000.

[27] J. Demšar, “Statistical comparisons of classifiers over multiple data sets,” *J. Mach. Learn. Res.*, vol. 7, pp. 1-30, 2006.