ICIC Express Letters
Volume **14**, Number **7**, July **2020**

ICIC International ©2020 ISSN 1881-803X
pp. 721–729

CELLULAR AUTOMATA-BASED PATTERN CLASSIFIER FOR BRAIN-STATE DISCRIMINATION PROBLEM

Rajdeep Chatterjee, Nazma Naskar and Debarshi Kumar Sanyal

School of Computer Engineering
Kalinga Institute of Industrial Technology Deemed to be University
Patia, Bhubaneswar, Odisha 751024, India
{ cse.rajdeep; naskar.preeti; debarshisanyal }@gmail.com

Received September 2019; accepted December 2019

Abstract. *This paper demonstrates the use of classification scheme based on Cellular Automata (CA) on motor-imaginary Electroencephalogram (EEG) signal to identify different brain-states. In Brain-Computer Interface (BCI), imagination or thought of a limb movement is known as motor imagery. It generates brain signals which are similar to the actual limb movements. A popular non-invasive method to record brain activity is electroencephalography. The recorded micro-voltage signals are called electroencephalogram. Over the years, researchers have shown that Multiple Attractor Cellular Automata (MACA) has the inherent ability to act as a k-class natural pattern classifier. In this paper, we introduce an MACA-based classifier to classify motor-imagery of limb movements into left-hand and right-hand movements. The raw EEG signal is first filtered using an elliptic bandpass filter to eliminate different artifacts (noise). Then the following feature extraction techniques are used: Wavelet-based Energy and Entropy (Eng-Ent), Band Power (BP), and Adaptive Auto-Regressive (AAR). Besides the proposed classifier scheme, we also examine Support Vector Machine (SVM) and several ensemble classifier variants as baseline models. Experiments are done using hold-out method as well as 10-fold cross-validation. The empirical results indicate that the MACA-based pattern classifier outperforms the other alternatives in each cases, i.e., hold-out as well as cross-validation. Our proposed MACA-based pattern classifier combined with band power features gives us 91.16% classification accuracy in 10-fold cross-validation. It is, to the best of our knowledge, the highest performance for the BCI Competition II Dataset III (2003) till date.*

Keywords: BCI, Brain-state, Cellular automata, MACA, Classifier, EEG, Motor-imagery

1. **Introduction.** Classification of different brain-states is an essential part of any Brain-Computer Interface (BCI) systems. Based on different brain functionalities, different brain regions are activated and thousands of neurons have been triggered. Due to this neuron activities specific brain waves are generated [1]. One of the popular ways of recording the brain waves is called electroencephalography and the micro-volt electrical signal is known as Electroencephalogram (EEG). Brain-states vary with time, sense and action. Different emotions, cerebral activities or mere imagination cause different brain-states [2, 3]. If a person is thinking or imagining of his/her limb movements while in reality he/she is not performing the movements, in BCI paradigm, it is known as "motor-imagery".

Researchers have been using different feature extraction techniques to generate feature-sets and classifiers to discriminate different brain-states. However, it is found that the best combination of feature extraction and classifier has yet not been achieved. It motivates us to explore new combinations. From our past research, we have concluded that the Adaptive Auto-Regressive (AAR), Band Power (BP), and Wavelet-based Energy and Entropy (Eng-Ent) based features perform consistently well irrespective of different classifiers. In

DOI: 10.24507/icicel.14.07.721

this paper, we have introduced a Cellular Automata (CA)-based classifier to discriminate different brain-states, specifically, motor-imagery EEG signal-based left-hand and right-hand movements [4, 5, 6, 7]. We have also compared the proposed scheme with several baseline classification methods to examine its robustness and consistency [9, 10]. Experimental results show that the proposed scheme achieves state-of-the-art performance.

The paper has been divided into six sections. In Section 2, the necessary theoretical aspects are discussed. Section 3 explains the proposed CA-based classifier. The experimental set-up is provided in Section 4. In Section 5, the results are analyzed and plotted for visualization. Finally, the paper concludes in Section 6.

2. Theoretical Preparation.

2.1. **Feature extraction.** In this step, from a given dataset, we identify the relevant values of necessary information known as “features”. Suitable extraction methods are important to determine in order to obtain the discriminant features. Here, we have used Adaptive Auto-Regressive (**AAR**) parameters, Band Power (**BP**) and Discrete Wavelet Transform (DWT) Energy-Entropy (**Eng-Ent**) features extraction techniques. Filtered raw EEG signals are then transformed into a feature vector (also referred as feature-set). The details of AAR, BP and energy entropy can be found in [11, 12, 13].

The order of autoregressive model is indicated by p. In this paper, the RLS (Recursive Least Square) version of AR models has been used for feature extraction. As the value of p is equal to 6 per electrode, 12 features are extracted using the AAR technique. The total percentage of power presented in a fixed frequency interval is computed in band power [2, 16]. All the brain rhythms have been considered for band power feature-set computation. These are: Delta ($<$ 4 Hz), Theta (4-7 Hz), Alpha (7-13 Hz), Beta (13-25 Hz), and Gamma ($>$ 25 Hz) [1, 17]. As per our previous research, it has been observed that for EEG based motor-imagery classification, the wavelet-based energy-entropy method is best suited for classification problem [13, 18, 19] .

Wavelet is a rapidly decaying wavelike oscillation that has zero-mean. Unlike a sinusoid which extends to infinity, it has a finite duration. Discrete Wavelet Transform (DWT) has been used in this paper to extract energy and entropy features. Here, the input signal has been decomposed three times to its 3rd level. The Daubechies (db) basis function with filter size 4 and third level detail coefficients $D3$ are used to extract features from the input EEG signal [19]. DWT is ideal for downsizing the actual input signal while retaining the properties of the original signal with fewer coefficients. Again, we have performed another step to reduce the dimension of the obtained wavelet energy and entropy [18, 19, 20, 21].

In our study, the size obtained from *AAR*, *BP* and *Eng-Ent* are respectively 12, 10 and 4.

2.2. **Classification.** Different techniques are used for data classification and categorization. A model can be built, only after mapping (training) the obtained class labels in the feature space. Two disjoint sets are formed by splitting the dataset used in classification. The same are referred as training-set and test-set, known as the hold-out technique.

3. Designing of MACA-Based Pattern Classifier.

Let us consider Table 1. There are four point state attractors in MACA of Table 1, further in this paper all CA belong to MACA type, and MACA and CA are used interchangeably. Each attractor, representing each basin, can identify unique memory location. An input pattern traverses through the basin and reaches the attractor state, which identifies class of the pattern. A CA can be used as k-class natural classifier, based on following two facts:

- *Necessary Condition:* Each point state attractor, representing a class, must identify a pattern $\mathcal{P}_i$ from set of patterns to be learnt $\{\mathcal{P}_1, \mathcal{P}_2, \ldots, \mathcal{P}_m\}$;

- *Sufficient Condition:* An attractor is said to represent a *Class XYZ* if it identifies maximum pattern from pattern set XYZ compared to other pattern sets.

A CA can represent maximum k number of classes, if it has k number of attractors. A CA having k number of attractors may represent a less than or equal to k classes.

Table 1. RMTs of the CA $\langle 8, 203, 255, 65 \rangle$ cell rules

Present state:	**111**	**110**	**101**	**100**	**011**	**010**	**001**	**000**	**Rule**
(RMT)	(7)	(6)	(5)	(4)	(3)	(2)	(1)	(0)	
(i) Next State:	d	d	d	d	1	0	0	0	8
(ii) Next State:	1	1	0	0	1	0	1	1	203
(iii) Next State:	1	1	1	1	1	1	1	1	255
(iv) Next State:	d	1	d	0	d	0	d	1	65

In this work, we target the design of a 2-class pattern classifier. As in classification task, two datasets are given to each class. All patterns are converted to binary strings. For n-length converted binary data, we have to generate n-cell CA. Next, attractors are marked according to the maximum number of patterns it identifies from each class. If certain attractor identifies more number of patterns from Class I, then that attractor is said to be represented of Class I, otherwise, Class II. For the identification of a class of patterns, the attractors, representing the classes, need to be stored in memory (see Figure 1). To identify the class of an input pattern p, the CA is loaded with p and updated till it reaches to an attractor. Then, from the attractor and the stored information, one can declare the class of the pattern p. In Figure 1, the class of p is I. However, if there are more than two attractors, then a set of attractors identifies a class.

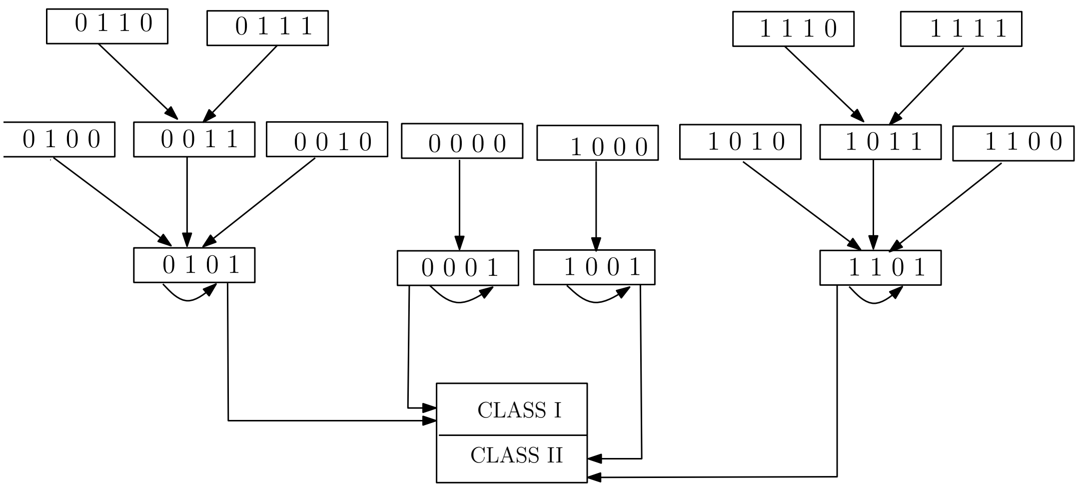

Figure 1. CA based classification strategy

3.1. **Training and testing.** There are two phases in classification training and testing. One set of data is given for training to the classifier. And the other set of data is given for the test. The accuracy measured in test phase is final accuracy of the classifier. The classifier is constructed using the patterns in the training set and next its performance is evaluated with the test set.

3.1.1. *Training.* In training phase of CA based pattern classification, first a CA is identified which gives the highest accuracy. Then, the attractors of the CA are marked according to the number of patterns identified by them. For example, Attractor *Attr* identifies p_1 number of patterns from class I and p_2 number of patterns from class II. If $p_1 > p_2$, then

Attr represents Class I, otherwise Class II. To design 2-class classifier, we select those CAs which have at least two point state attractors where the attractors represent each class. Algorithm 2 and Algorithm 1 are proposed as training algorithms.

$$\textbf{Accuracy} = \frac{\#\text{patterns properly classified}}{\text{Total No. of patterns}} \times 100\% \tag{1}$$

Algorithm 1: Pattern classifier using NULL boundary CA

1: Input: n (Size of CA), dataset1, dataset2.
2: Output: *Accuracy and n-cell CA* $\langle \mathcal{R}_0, \mathcal{R}_1, \dots, \mathcal{R}_{n-1} \rangle$.
3: Generate CA having only point state attractor as described in [4].
4: Read data from data-sets from P_1 and P_2, Set Success $\leftarrow 0$.
5: Identify the attractors for each pattern from P_1 (P_2).
6: Identify and assign each attractor according to the number of patterns identified from the patternsets.
7: Set Success $\leftarrow \sum$ #patternscorrectlyidentified.
8: Set Accuracy $\leftarrow \frac{\text{Success}}{\text{total number of pattern}} \times 100$.
9: Return *Accuracy*, *CA*, *training set of attractor of class 1 (class 2)*.

Algorithm 2: Training phase

1: Input: n (Size of CA), dataset1, dataset2.
2: Output: *Accuracy and n-cell CA* $\langle \mathcal{R}_0, \mathcal{R}_1, \dots, \mathcal{R}_{n-1} \rangle$.
3: *Maxaccuracy* $\leftarrow 0$.
4: **Repeat**
5: CALL ClassifierDesignPointStateAttractor( ) (Algorithm 1).
6: **if** *Maxaccuracy* $<$ *accuracy* **then**
7: Set *Maxaccuracy* $\leftarrow$ *Accuracy*.
8: **end if**
9: Rewrite FinalCA and SetofAttractor.
10: **Until** $i = 0$ to *max_iteration*.

3.1.2. *Testing.* The final accuracy of classifier is obtained in testing phase. The CA obtained from training phase is used to identify the patterns of each dataset. Then, it is checked, whether identified pattern by the attractor and the represented class (as in training phase) of the attractor are same. If so, then it will be accounted as attractor identified correct pattern; otherwise not. Say $A_i \in AtrSet_1$ but identifies as pattern from dataset2 then it is assumed that the pattern is not correctly identified. Algorithm 3 represents the test phase. To Algorithm 3 input the CA size (n) along with $\langle \mathcal{R}_0, \mathcal{R}_1, \dots, \mathcal{R}_{n-1} \rangle$ generated by Algorithm 2, two datasets and two attractor sets. If any pattern is identified by an attractor such that no match exists in both attractor sets, then we use hamming distance. That is, say a pattern $p \in dataset1$ is identified by an attractor Atr. Say $Atr \notin AtrSet_1$ and $Atr \notin AtrSet_2$. So, hamming distance between Atr and all attractors from both the attractor sets is measured, in Algorithm 3 the variable hd stores the value (hamming distance). If it is found that hd is greater than $\lceil \frac{n}{2} \rceil$ with respect to attractor set $AtrSet_1$, then it is counted as attractor of $AtrSet_2$, otherwise $AtrSet_1$. Hence, if hd is less than $\lceil \frac{n}{2} \rceil$ then we say that the pattern is identified correctly. The output of Algorithm 3 is *accuracy*, i.e., percentage of total correctly identified pattern with respect to number of total patterns.

Algorithm 3: Test phase

1: Input: n (Size of CA), dataset1, dataset2, and n-cell CA $\langle \mathcal{R}_0, \mathcal{R}_1, \ldots, \mathcal{R}_{n-1} \rangle$, $AtrSet_1$, $AtrSet_2$.
2: Output: *Accuracy*.

3: Read patterns from dataset1 (resp. dataset2) and store to pattern set P_1 (resp. pattern set P_2).
4: Success $\leftarrow$ 0.
5: **Repeat**
6: Identify the attractor atr_i for p_i.
7: **if** $atr_i \in AtrSet_1$ **then**
8:    set success $\leftarrow$ success + 1.
9: **end if**
10: **if** $((atr_i \notin AtrSet_1)$ & $(atr_i \notin AtrSet_2))$ **then**
11:    Set $hd \leftarrow$ Hamming Distance (atr_i, A_j), where A_j is any attractor within $AtrSet_1$.
12:    **if** $hd < \lceil (\frac{n}{2}) \rceil$ **then**
13:       **Until** pattern $p_i \in P_1$.
14:       Set success $\leftarrow$ success + 1.
15:    **end if**
16: **end if**
17: **Until** pattern $p_i \in P_2$ and attractor set $AtrSet_1$ replaced by $AtrSet_2$
18: Rewrite FinalCA and SetofAttractor.
19: Set accuracy $= \frac{\text{success}}{P_1 + P_2}$.
20: Report Accuracy.

3.1.3. *Complexity analysis of algorithms.* The execution of Step 5 and Step 6 of Algorithm 1 depends on the number of patterns (m), and rest of the steps execute only once. Hence, Algorithm 1 is $O(m)$. Algorithm 2 calls Algorithm 1 for m (any positive integer) times which has time complexity $O(n)$ and remaining statements are $O(1)$. Hence, time complexity of Algorithm 1 is $O(m \times n)$. Algorithm 3 depends on Step 3, which has linear growth; hence it has also time complexity $O(n)$.

4. Experimental Preparation.

4.1. **Dataset.** The EEG dataset used in this paper is Dataset III of BCI Competition II (2003) taken from the Department of Medical Informatics, Institute for Biomedical Engineering, University of Technology, Graz [22]. The recommended sampling frequency is at 128 Hz. We have considered data only from the $C3$ and $C4$ electrodes as they are directly responsible for human left and right hand movements. More relevant information is available in [1, 23].

The whole dataset contains total 280 trials (that is, instances) having equal number trials of left and right hand movements. The first 140 trials are used for training-set and kept the remaining 140 trials for test-set. To eliminate the noise from the raw EEG signal, it is filtered with cut-off frequencies of 0.5 Hz and 50 Hz [13, 18, 23].

4.2. **Other classifiers used.** Cellular automata based pattern classifier has been compared with two variants of SVM classifier and three distinct types of ensemble learning [24]. The details related to the variants of classifiers used and the number of learners used are given in Tables 2, 3 and 4.

Table 2. Variants of classifiers used for hold-out

Classifiers	**Acronyms**	**Kernel Functions**
Support Vector Machine	SVM1	Radial Basis
	SVM2	Polynomial (degree = 3)
Ensemble		**Learner Types**
	ENS1	Subspace
	ENS2	Logitboost
	ENS3	Adaboost

Table 3. Number of learners in ensemble learning used in hold-out

Features	**ENS1**	**ENS2**	**ENS3**
AAR	16	8	64
Eng-Ent	6	18	9
BP	3	6	16

Table 4. Variants of classifiers used for cross-validation and their learner numbers

Classifiers	**Acronyms**	**Kernel Functions**	
Support Vector Machine	SVM1	Radial Basis	
	SVM2	Polynomial (degree = 3)	
Ensemble		**Learner Types**	**Learner Numbers**
	ENS1	Subspace	33
	ENS2	Logitboost	
	ENS3	Adaboost	

4.3. **System configuration.** The implementation of this paper is done using MATLAB 2016a on an Intel(R) Core(TM) i5-6200U CPU 2.40 GHz with 8 GB RAM with 64 bits Windows 10 Professional operating system.

5. **Results and Discussions.** MACA based pattern classifier is implemented once with hold-out set-up and again with 10-fold cross-validation using stratified sampling. In hold-out technique, the first 140 trials (instances) are taken as train-set and the remaining 140 trials as test-set. Three distinct types of feature extraction techniques: AAR, BP and Eng-Ent are used in our study (refer 2.1). *SVM* classifier variants, *SVM1* and *SVM2*, use Radial Basis Function (RBF) and Polynomial (degree = 3) respectively. Similarly, the ensemble learning variants used in our paper are Subspace (*ENS1*), Logit boosting (*ENS2*) and Adaptive boosting (*ENS3*). The used learner number for all the ensemble classifier variants is 33 (it is fixed after a rigorous search and provides consistent results for all the variants).

It is observed that the best three performing classifiers using both hold-out and 10-fold cross-validation are *ENS1*, *ENS3* and our proposed scheme (see Tables 5 and 6). Our MACA-based classifier performs better than every alternative used.

Finally, our proposed CA pattern classifier is compared with few best performing existing classifiers for the BCI Competition II Dataset III (refer [22]). The comparison is presented in a tabular format in Table 7 and shown in Figure 2. We have found that the proposed CA pattern classifier gives accuracy of **91.16%** from band power dataset using 10-fold cross-validation technique. This is, to the best of our knowledge, the highest ever performance obtained from BCI Competition II Dataset III.

Table 5. Accuracy obtained from different datasets using CA based pattern classifier both for hold-out and cross-validation techniques

Technique	**AAR**	**BP**	**Eng-Ent**
Hold-out	84.29	88.49	87.00
10-Fold	87.86	91.16	78.47

Table 6. Accuracies obtained from different classification techniques using AAR, BP and Eng-Ent datasets

Classifier	**Hold-out**			**10-Fold**		
	AAR	**BP**	**Eng-Ent**	**AAR**	**BP**	**Eng-Ent**
SVM1	79.29	50.71	77.86	76.50	49.64	78.42
SVM2	78.57	66.43	77.14	78.29	56.00	71.00
ENS1	82.14	82.14	82.14	77.14	78.29	81.64
ENS2	78.57	80.71	78.57	77.42	76.00	79.29
ENS3	77.14	83.57	79.29	78.77	77.64	80.35
Proposed scheme	84.29	88.49	87.00	87.86	91.16	78.47

Table 7. Comparative analysis of a few best performing techniques on BCI Competition II Dataset III

References	**Methods used**	**Classifiers**	**Accuracy (%)**
Chatterjee et al. [13] (Chatt1)	Wavelet energy-entropy	SVM (kernel: Linear/polynomial)	85.00
Chatterjee and Bandyopadhyay [18] (Chatt2)	Average-power + band-power + Wavelet energy-entropy + RMS + Statistical features (Table V)	MLP	85.71
Lemm et al. [9]	Morlet wavelet	Bayes Quadratic	89.29
Bashar and Bhuiyan [10]	MEMD + STFT	KNN (cosine distance)	90.71
Proposed scheme	10-fold cross-validation + band power	MACA based classifier	**91.16**

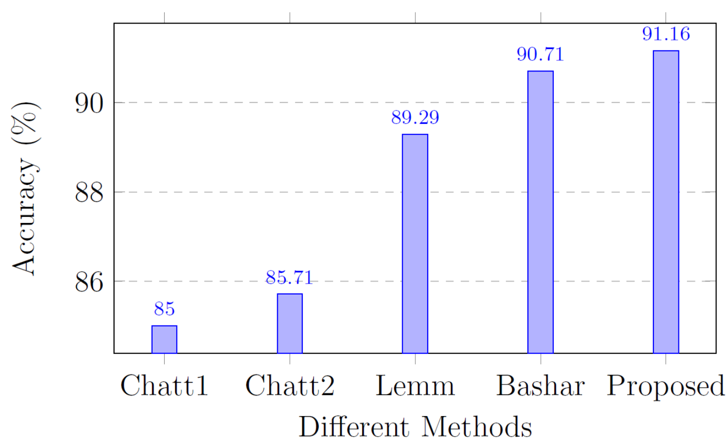

Figure 2. Accuracies of a few best performing techniques on BCI Competition II Dataset III

6. **Conclusion.** EEG based brain-states classification is an important research domain in bio-signal applications. In the past, single and ensemble classifiers have been used to classify motor-imagery signals. In this paper, an MACA-based pattern classifier is used for the first time. It outperforms existing SVM and ensemble-based classifiers in its first adaptation. The results are validated using both hold-out and 10-fold cross-validation techniques. The best performances obtained from hold-out and 10-fold cross-validation techniques are 88.49% and 91.16% respectively with band power dataset. In future, the proposed classifier will be examined with more than one subject in a multi-class configuration.

REFERENCES

[1] G. Pfurtscheller and C. Neuper, Motor imagery activates primary sensorimotor area in humans, *Neuroscience Letters*, vol.239, nos.2-3, pp.65-68, 1997.

[2] N. Brodu, F. Lotte and A. Lécuyer, Comparative study of band-power extraction techniques for motor imagery classification, *2011 IEEE Symposium on Computational Intelligence, Cognitive Algorithms, Mind, and Brain (CCMB)*, pp.1-6, 2011.

[3] R. Chaudhari and H. J. Galiyawala, A review on motor imagery signal classification for BCI, *Signal Processing: An International Journal (SPIJ)*, vol.11, no.2, p.16, 2017.

[4] M. N. Naskar, *Characterization and Synthesis of Cellular Automata with only Point State Attractors*, Ph.D. Thesis, IIEST, Shibpur, 2016.

[5] P. Maji, C. Shaw, N. Ganguly, B. K. Sikdar and P. P. Chaudhuri, Theory and application of cellular automata for pattern classification, *Fundamenta Informaticae*, vol.58, nos.3-4, pp.321-354, 2003.

[6] N. Ganguly, P. Maji, S. Dhar, B. K. Sikdar and P. P. Chaudhuri, Evolving cellular automata as pattern classifier, *International Conference on Cellular Automata*, pp.56-68, 2002.

[7] K. Sutner, Classification of cellular automata, in *Cellular Automata: A Volume in the Encyclopedia of Complexity and Systems Science*, 2nd Edition, Springer, 2018.

[8] A. Datta, R. Chatterjee, D. K. Sanyal and D. Guha, An ensemble classification approach to motor-imagery brain state discrimination problem, *IEEE 2017 International Conference on Infocom Technologies and Unmanned Systems (Trends and Future Directions) (ICTUS)*, pp.322-326, 2017.

[9] S. Lemm, C. Schafer and G. Curio, BCI competition 2003-data set III: Probabilistic modeling of Sensorimotor/spl Mu/rhythms for classification of imaginary hand movements, *IEEE Trans. Biomedical Engineering*, vol.51, no.6, pp.1077-1080, 2004.

[10] S. K. Bashar and M. I. H. Bhuiyan, Classification of motor imagery movements using multivariate empirical mode decomposition and short time Fourier transform based hybrid method, *Engineering Science and Technology, an International Journal*, vol.19, no.3, pp.1457-1464, 2016.

[11] A. Schlögl, D. Flotzinger and G. Pfurtscheller, Adaptive autoregressive modeling used for single-trial eeg classification, *Biomedical Engineering*, vol.42, no.6, pp.162-167, 1997.

[12] A. Schlögl, S. J. Roberts and G. Pfurtscheller, A criterion for adaptive autoregressive models, *Proc. of the 22nd Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, pp.1581-1582, 2000.

[13] R. Chatterjee, T. Bandyopadhyay, D. K. Sanyal and D. Guha, Comparative analysis of feature extraction techniques in motor imagery EEG signal classification, *Proc. of the 1st International Conference on Smart System, Innovations and Computing*, pp.73-83, 2018.

[14] A. Datta and R. Chatterjee, Comparative study of different ensemble compositions in EEG signal classification problem, in *Emerging Technologies in Data Mining and Information Security. Advances in Intelligent Systems and Computing*, A. Abraham, P. Dutta, J. Mandal, A. Bhattacharya and S. Dutta (eds.), Singapore, Springer, 2019.

[15] R. Chatterjee, D. Guha, D. K. Sanyal and S. N. Mohanty, Discernibility matrix based dimensionality reduction for EEG signal, *2016 IEEE Region 10 Conference (TENCON)*, pp.2703-2706, 2016.

[16] W. Klimesch, M. Doppelmayr, H. Russegger, T. Pachinger and J. Schwaiger, Induced alpha band power changes in the human EEG and attention, *Neuroscience Letters*, vol.244, no.2, pp.73-76, 1998.

[17] K. J. Miller, G. Schalk, E. E. Fetz, M. den Nijs, J. G. Ojemann and R. P. N. Rao, Cortical activity during motor execution, motor imagery, and imagery-based online feedback, *Proc. of the National Academy of Sciences*, vol.107, no.9, pp.4430-4435, 2010.

[18] R. Chatterjee and T. Bandyopadhyay, EEG based motor imagery classification using SVM and MLP, *The 2nd International Conference on Computational Intelligence and Networks (CINE)*, pp.84-89, 2016.

[19] R. Chatterjee, T. Bandyopadhyay and D. K. Sanyal, Effects of wavelets on quality of features in motor-imagery EEG signal classification, *International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET)*, pp.1346-1350, 2016.
[20] J. Hu, D. Xiao and Z. Mu, Application of energy entropy in motor imagery EEG classification, *JDCTA*, vol.3, no.2, pp.83-90, 2009.
[21] T. Gandhi, B. K. Panigrahi and S. Anand, A comparative study of wavelet families for EEG signal classification, *Neurocomputing*, vol.74, no.17, pp.3051-3057, 2011.
[22] *Brain Computer Interface Competition II*, Department of Medical Informatics, Institute for Biomedical Engineering, University of Technology Graz, 2004 (Accessed on June 6, 2015).
[23] G. Pfurtscheller and C. Neuper, Motor imagery and direct brain-computer communication, *Proceedings of the IEEE*, vol.89, no.7, pp.1123-1134, 2001.
[24] L. Rokach, Ensemble-based classifiers, *Artificial Intelligence Review*, vol.33, no.1, pp.1-39, 2010.