

# Deep Learning Methods for EEG Signals Classification of Motor Imagery in BCI

Muhammad Fawaz Saputra<sup>1</sup>, Noor Akhmad Setiawan<sup>2</sup>, Igi Ardiyanto<sup>3</sup>

**Abstract**—EEG signals are obtained from an EEG device after recording the user's brain signals. EEG signals can be generated by the user after performing motor movements or imagery tasks. Motor Imagery (MI) is the task of imagining motor movements that resemble the original motor movements. Brain Computer Interface (BCI) bridges interactions between users and applications in performing tasks. Brain Computer Interface (BCI) Competition IV 2a was used in this study. A fully automated correction method of EOG artifacts in EEG recordings was applied in order to remove artifacts and Common Spatial Pattern (CSP) to get features that can distinguish motor imagery tasks. In this study, a comparative studies between two deep learning methods was explored, namely Deep Belief Network (DBN) and Long Short Term Memory (LSTM). Usability of both deep learning methods was evaluated using the BCI Competition IV-2a dataset. The experimental results of these two deep learning methods show average accuracy of 50.35% for DBN and 49.65% for LSTM.

**Keyword**—Electroencephalograph, Motor Imagery, Mu, Beta, Brain Computer Interface, Deep Learning, Deep Belief Networks, Long Short Term Memory.

## I. INTRODUCTION

Electroencephalograph (EEG) is a device, instrument, or machine used to record, collect and measure brain signals originating from the electrical activity of nerves in different parts of the brain simultaneously [1], [2]. EEG signals are signals obtained from an EEG device after recording the user's brain signals. EEG signals can be generated by the user after performing motorized or motor imagery tasks. The task of motor movements was carried out by the user to produce brain signals such as hand movements, leg movements, and blinking of the eyes. Motor Imagery (MI) is the task of imagining motoric movements resembling the original motor movements.

Brain Computer Interface (BCI) is a technology that allows users to interact with the external environment through direct connections between the brain and output devices using brain signals [3]. BCI bridges interactions between users and applications in performing tasks. In this study, BCI applications translate motor imagery tasks performed by users into commands that can be understood by machines. The BCI

application allows users to control one or more external devices by imagining the task to be performed. EEG signals that can make users control external devices are generally divided into several frequency ranges, namely delta, theta, alpha, beta, and gamma. Delta has a frequency range of less than 4 Hz, theta has a frequency range of 4-8 Hz, alpha has a frequency range of 8-13 Hz, beta has a frequency range of 13-30 Hz, gamma has a range of more than 30 Hz [4]–[6]. There is one more type of signal, that is mu signal, mu have a frequency range that is almost the same as alpha, which is 8-13 Hz, the difference is alpha activity appears in the posterior and occipital parts, whereas mu appear from the sensorimotor region [7]. This frequency determines which area was related to the motor imagery. Motor imagery are in mu frequency range and beta [8]. Therefore, in this study bandpass filtering was applied in the 8-30 Hz frequency range. In other studies, mu are in the frequency range 8-12 Hz and beta is in the range of 18-26 Hz [9]–[11]. Differences in this case are natural, then can be used as learning in the future to determine the frequency range. The value of this frequency range varies when brain activity was associated with any motor task even though the movement was not needed, it will affect that value. When this unnecessary motion affects the value of the EEG signal, this was called noise on the EEG signal. Motor movements are different from motor imagery, motor movements such as moving the left hand, right hand, or leg are converted into commands to control external devices, while the motor imagery in the hand by imagining an imaginary hand is in front of our eyes and then moved like the original hand. This motor imagery recording is then translated by the BCI application in order to do the task. In a BCI system, a good classifier is important to get high accuracy.

Deep learning methods are still rarely explored in EEG-based BCI systems, because they are quite difficult to apply to the development of a perfect EEG classification framework due to various impact factors, such as noise, correlation between channels, and high-dimensional EEG data [12]. Deep Belief Network (DBN) is one part of Deep learning. Some works have introduced DBN into the EEG signal classification [13]–[15]. The application of DBN for EEG signal classification from motor imagery is still rare. Meanwhile, other deep learning classifiers, Long Short Term Memory (LSTM), are still relatively small for EEG signals classification. Several studies use LSTM for EEG signals classification [16], [17]. The application of LSTM to the classification of EEG signals from motor imagery is also still rare. Based on this, the aim is to explore the DBN and LSTM methods for EEG signal classification from motor imagery. These two methods were compared to get the best method among these two deep learning methods.

<sup>1</sup> Student, Department of Electrical and Information Engineering, Faculty of Engineering, Universitas Gadjah Mada, Grafika Street 2 Yogyakarta 55281 Indonesia (e-mail: muhammad.fawaz.s@mail.ugm.ac.id)

<sup>2,3</sup> Lecturer, Department of Electrical and Information Engineering, Faculty of Engineering, Universitas Gadjah Mada, Grafika Street 2 Yogyakarta 55281 Indonesia (e-mail: noorwewe@ugm.ac.id; igi@ugm.ac.id)

## II. PRE-PROCESSING

### A. Data Description and Analysis

Brain Computer Interface (BCI) Competition IV 2a was used in this study. This data includes EEG signals from nine subjects, namely A01 - A09, and each subject provides two recording sessions. The data consists of 25 channels, which include 22 EEG channels and three monopolar EOG channels. EOG channels are provided for further applications of artifact processing methods. These signals include four different MI tasks, namely the left hand (class 1), right hand (class 2), both feet (class 3) and tongue (class 4). Two sessions on different days were recorded for each subject. Each session was comprised of six runs separated by short breaks. One run consists of 48 trials (twelve for each of the four possible classes), yielding a total of 288 trials per session.

In this study, two classes were used, namely the right hand motor imagery, and the left hand motor imagery. Thus, 24 trials in two classes, and the total trials for six runs are 144 trials. It means 72 trials within rejected trial in each class. The EEG dataset consists of two types of training data and test data, which the EEG training dataset was used to produce a classifier model, and then an EEG test dataset was used to get the classification accuracy. Classification accuracy was achieved by matching it with the label which is the output of the classifier. After finishing analyzing the data, the pre-processing stage was applied and feature extraction stage before classification. Fig. 1 shows the stages in flow diagram of method used in this study.

### B. A Fully Automated Correction Method of EOG Artifacts in EEG Recordings

A fully automated correction method of EOG artifacts in EEG recordings was developed by [18]. In this study, a fully automated correction method of EOG artifacts in EEG recordings was applied [19].

The following linear models was assumed with three spatial (horizontal, vertical, and radial) EOG components, including 23 channel numbers (left EOG), 24 channel numbers (middle EOG), 25 channel numbers (right EOG). See the following models, presented as (1).

$$a(t) = j(t) + \{o_1(t), o_2(t), o_3(t)\} b \quad (1)$$

where  $a(t)$  is the recorded value of channel  $ch$  at time  $t$ ,  $j$  is the source signal without artifact contamination,  $o_1$ ,  $o_2$ , dan  $o_3$  indicate the noise source  $O$  of the three spatial EOG components. And  $b$  indicate the weight of EOG artifacts at the EEG channel. Extending this to more EEG channels and using matrix notation, can be written in (2).

$$A_{T \times M} = J_{T \times M} + O_{T \times N} B_{N \times M}. \quad (2)$$

The indices indicate the size of each matrix. Signal  $A$  and  $J$  have  $T$  time points dan  $M$  channels. The noise  $O$  have  $n$  components.  $B$  indicate the weights from each EOG component to each EEG channel. To obtain the corrected signal, the following formulas are used.

$$J = A - O B. \quad (3)$$

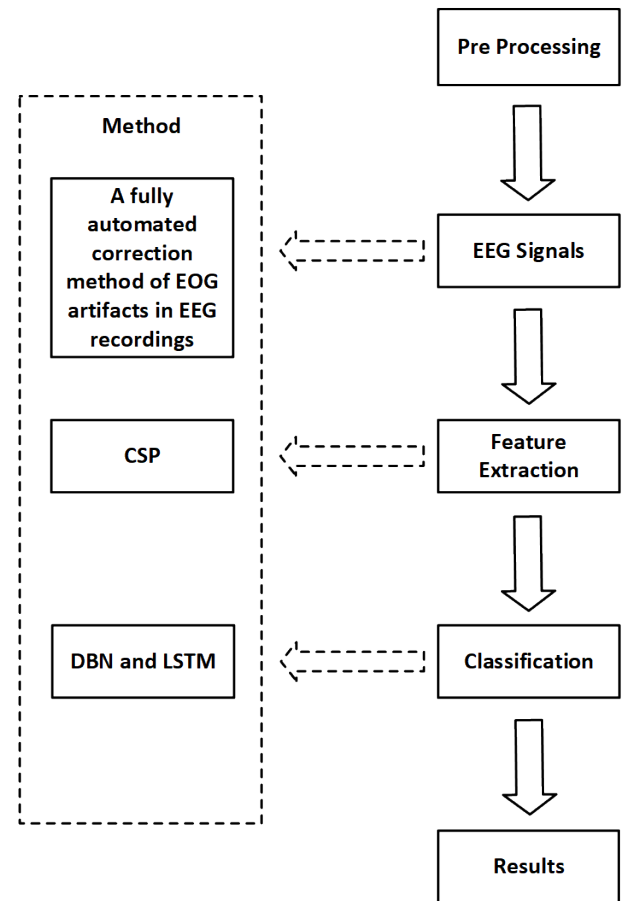


Fig. 1 Flow diagram of method used in this study.

Since the signal  $J$  and the noise  $O$  are independent, then:

$$O^T J = O^T A - O^T O B. \quad (4)$$

With  $O^T J = 0$ , then (5) is obtained.

$$B = O^T O^{-1} O^T A = D_{NN}^{-1} D_{NY} \quad (5)$$

where  $D_{NN} = O^T O$  is the auto-covariance from EOG channel and  $D_{NY} = O^T A$  is the cross-covariance between EEG and EOG channel.

### C. Spectral Perturbation Analysis and Band Pass Filter

Spectral perturbation analysis was used to know which range that motor imagery was produced. Band pass filter is useful to eliminate noise from signals that pass a certain range, so that the signal does not interfere with the signal processing. In this study butterworth band pass filters was applied in frequency range of mu 8-13 Hz and beta 13-30 Hz. This frequency range was applied because both motor imagery tasks are in mu and beta. Fig. 2 and Fig. 3 show the dominant range on channel C3 and C4 at the duration of both motor imagery tasks.

## III. FEATURE EXTRACTION

In this study, the Common Spatial Pattern (CSP) was used [20]. Features extracted from two original EEG sample classes using CSP. CSP serves to determine spatial filters that maximize the variance of signals in one class and

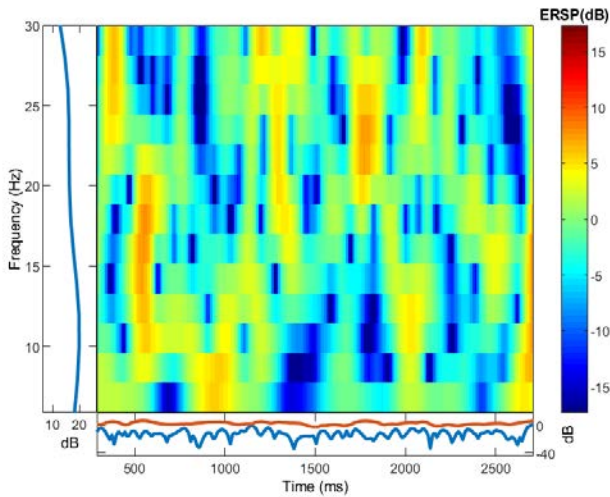


Fig. 2 ERSP of left region (C3).

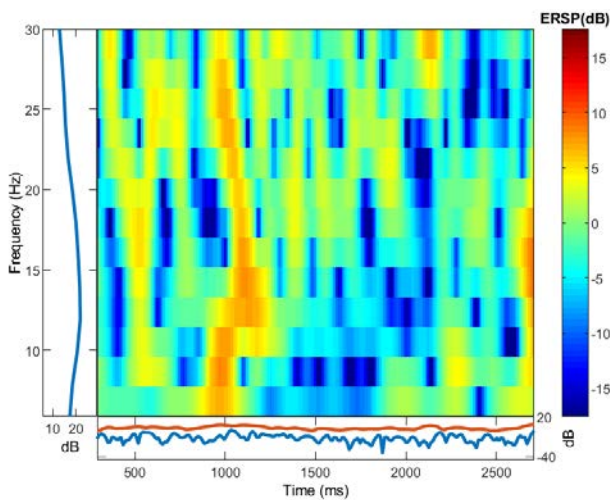


Fig. 3 ERSP of right region (C4).

simultaneously minimize the variance of signals in other classes. Assume the original EEG signal was represented as a matrix  $J \in R^{M \times T}$ , where  $M$  is the number of channels,  $T$  is the sample point of each channel. The CSP operation process is shown as follows.

Spatial covariance from EEG signal data is calculated using (6).

$$C = \frac{JJ^T}{\text{trace}(JJ^T)}. \quad (6)$$

To separate the two different classes of motor imagery, spatial covariances was made from the two data classes respectively. After getting the average covariance  $\bar{C}_1$  dan  $\bar{C}_2$ , a composite spatial covariance was obtained.

$$\bar{C} = \bar{C}_1 + \bar{C}_2. \quad (7)$$

$\bar{C}$  decomposed into (8).

$$\bar{C} = V_c \Sigma V_c^T \quad (8)$$

In this case,  $V_c$  is a feature vector that was decomposed.  $\Sigma$  is a diagonal matrix consisting of eigenvalues. The eigenvalues

are arranged in descending order, then transform them with whitening.

$$Q = \sqrt{\Sigma^{-1}} V_c^T. \quad (9)$$

Covariance matrices can then be changed to the following equation.

$$\left. \begin{aligned} Q \bar{C}_i Q^T &= S_i = V \Sigma_i V^T \\ \Sigma_1 + \Sigma_2 &= I \end{aligned} \right\} \quad (10)$$

To get the projection matrix the equation is done:

$$W = (V^T Q)^T. \quad (11)$$

After that, single trial EEG data can be changed to the following equation.

$$Z = W J \quad (12)$$

EEG signals can be projected on the first  $n$  and the last  $n$  from  $W$  after whitening process. This means, the first  $n$  column calculated as the first half  $M$  and the rest of column as the next half  $M$ , where  $M$  is the number of channels dan  $n$  is half of total channels. The first  $n$  column and the last  $n$  column from  $W$  can be calculated in row vectors  $Z_p = (p = 1, 2, \dots, n, M - n + 1, \dots, M)$ , from  $Z$  which maximizes the difference in variance between the groups associated with the largest eigen values in  $\Sigma_1$  dan  $\Sigma_2$ . The signals are contained in the first and rows  $Z$  in equation (12), due to  $W$ . The features are obtained by the following equation.

$$f_p = \frac{\log f_{var}(Z_p)}{\sum_{i=1}^{2n} \log f_{var}(Z_i)} \quad (13)$$

The  $f_{var}(Z_p)$  symbol shows the variance of total data and  $f_{var}(Z_i)$  show  $i$  variance of data.

#### IV. CLASSIFICATION

##### A. Deep Learning

Deep learning is part of machine learning, because the initial purpose of machine learning, making computers, and artificial intelligence is to make a computer system that works like a human brain. Humans are competing to realize this. Deep learning is a neural network that has at least two hidden layers [21]. From the development of a neural network that has many hidden layers, the Deep Belief Network (DBN) and Long Short Term Memory (LSTM) are developed.

##### B. Deep Belief Network (DBN)

Deep Belief Network (DBN) developed by Geoffrey E. Hinton, Simon Osindero, dan Yee-Whye Teh in 2006 [22]. In this study, DBN was used [23]. For algorithms, DeeBN toolbox was used [24]. In DBN,  $v$  representing feature vector that only contain one channel feature. The RBM defines the joint distribution there, assuming as a unit seen in DBN and  $h$ , hidden units can be calculated in (14).

$$p(v) = \frac{\sum_u e^{-E(v, h)}}{\sum_u \sum_g e^{-E(u, h)}}. \quad (14)$$

$E$  is an energy function defined as (15).

$$E(v, h) = \sum_{j \in \text{visible}} a_j v_j - \sum_{j \in \text{visible}} b_j h_j - \sum_{i, j} v_j h_j \omega_{ij} \quad (15)$$

where  $v_j h_j$  is binary status if visible unit  $i$  and hidden unit  $j$ ,  $a_i, b_j$  is their bias and  $\omega_{ij}$  is the weight between them. Network provides a probability for each possible pair of vectors that are visible and hidden through this energy function.

$$p(v, h) = \frac{1}{Z} e^{-E(v, h)} \quad (16)$$

The probability given by the network to training data can be optimized by adjusting the weights and biases to reduce that energy. The derivative of the log vector training probability with respect to weight was calculated as follows.

$$\begin{aligned} \frac{\partial \log p(v)}{\partial \omega_{ij}} &= \frac{\sum_{v \in D} \partial \log p(v)}{\partial \omega_{ij}} \\ &= E_{data} \left[ \frac{\partial E(v, h)}{\partial \omega_{ij}} \right] \\ &\quad - E_{model} \left[ \frac{\partial E(u, g)}{\partial \omega_{ij}} \right] \end{aligned} \quad (17)$$

where the first item is expectation of  $\partial E(v, h) / \partial \omega_{ij}$  responds to training set  $D$  and hidden variables sampled according to the conditional distribution of the dataset on  $p(h | v)$ , given training samples randomly selected,  $v$ , the binary state,  $h_j$ , from each hidden unit,  $j$ , set to 1 with probability as in (18).

$$p(h_j = 1 | v) = \sigma(b_j + \sum v_j \omega_{ij}) \quad (18)$$

where  $\sigma(x)$  is the sigmoid logistic function  $1/(1 + \exp(-x))$ .  $v_i h_j$  then an unbiased sample.

The way to get an unbiased sample from a unit that looks similar to a hidden unit is that with no direct connection between units seen in RBM, hidden vectors are defined as (19).

$$p(v_i = 1 | h) = \sigma(\alpha_i + \sum_j h_j \omega_{ij}) \quad (19)$$

The vectors are combined with binary vectors from the class label for training the RBM classifier, shared data distribution and class labels. The energy function becomes (20).

$$E(v, l, h) = - \sum_i \alpha_i v_i - \sum_j b_j h_j - \sum_{i, j} \omega_{ij} \sum_y c_y l_y \sum_{y, j} \omega_{yj} h_j l_y. \quad (20)$$

In this equation, 1 is the binary class label and  $\omega_{ij}$  is the weights between hidden units and label units.

### C. Long Short Term Memory (LSTM)

Long Short Term Memory (LSTM) is part of Recurrent Neural Network (RNN). LSTM was developed by Holreiter and Schmidhuber in 1997 [25]. In this study, LSTM was used

TABLE I  
EXPERIMENTAL RESULTS

| Subject        | Classification Accuracy (%) |       |
|----------------|-----------------------------|-------|
|                | DBN                         | LSTM  |
| A01            | 48.15                       | 48.15 |
| A02            | 51.85                       | 51.85 |
| A03            | 48.15                       | 51.85 |
| A04            | 50                          | 50    |
| A05            | 50                          | 50    |
| A06            | 52.38                       | 47.62 |
| A07            | 50                          | 50    |
| A08            | 50                          | 50    |
| A09            | 52.63                       | 47.37 |
| <b>Average</b> | 50.35                       | 49.65 |

[26]. LSTM has three gates that have the ability to delete or add information to a state. The first gate is the forget gate to decide what information to discard from the cell state, this decision is made by the sigmoid layer.

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \quad (21)$$

The second gate is an input gate consisting of the sigmoid layer to decide which value to update, and the layer that creates a new vector of updated values as described in (22) and (23).

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \quad (22)$$

$$\tilde{C}_t = \tanh(W_c \cdot [h_{t-1}, x_t] + b_c) \quad (23)$$

Then, the cell state was updated from equations (21), (22), and (23) by (24).

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t \quad (24)$$

Finally, the output of the current state will be calculated based on the state of the updated cell and the sigmoid layer which determines which part of the cell state will be the final result as described in equations (25) and (26).

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \quad (25)$$

$$h_t = o_t * \tanh(C_t) \quad (26)$$

In these equations,  $\sigma$  is the sigmoid activation function that presses the number into the range (0,1),  $\tanh$  is the activation function of hyperbolic tangent which suppresses the number into the range (-1,1),  $W_f, W_i, W_c, W_o$  is the weight matrix,  $x_t$  is the input vector,  $h_{t-1}$  shows the past hidden state and  $b_f, b_i, b_c, b_o$  are bias vectors.

### D. Experimental Results

In this experiment, a laptop with the Windows 10 operating system with an Intel Core i7-4510U (2.6 GHz) CPU with 4 GB of RAM was used, the experiment was conducted in Matlab 2019a. The experimental result are shown in Table I.

Comparison between DBN and LSTM in the same dataset and modelset. Data was randomized using 5-Cross Validation for nine subjects to obtain optimal accuracy results. According to [27], the DBN algorithm is set with the following ranges of

parameters: hidden nodes = 5-100, learning rate = 0.1-0.9, weight decay = 0.1-0.9, momentum = 0.001-0.01. For DBN, number of epochs = 10 and mini batch size = 20 were set. The LSTM algorithm was applied with the number of epochs = 10 according to [28], while the number of inputs and other parameters adjusted to data, with the following parameters: input size = 22, number of hidden units = 100, mini batch size = 27.

## V. CONCLUSIONS

This paper has evaluated two different deep learning methods as classifiers using dataset of motor imagery from BCI Competition IVa. Deep belief Network (DBN) and Long Short Term Memory (LSTM) are tested to find the most efficient classifier in terms of accuracy results. DBN has shown itself to be the most efficient classifier with the slight average accuracy difference 50.35% compared to LSTM which achieves 49.65% accuracy.

LSTM get the lower accuracy value than DBN in average. Accuracy values that have been presented are intended for comparison of classification methods. However, the level of accuracy is not high enough. In future work, to overcome this problem, different data filtering methods and feature extraction methods must be considered to obtain higher accuracy. Larger subject populations should be considered as a consideration for the method performance hypothesis better than.

From the results of this experiment, DBN gets a higher accuracy value than LSTM, the results of this hypothesis can be used in future work as a choice of methods for EEG signal classification using the deep learning method.

## REFERENCES

- [1] R. Wang, D. Li, J. Wang, L. Cai, and L. Shi, "Synchrony Analysis Using Different Cross-Entropy Measures of the Electroencephalograph Activity in Alzheimer's Disease," *Proc. - 2016 9th Int. Congr. Image Signal Process. Biomed. Eng. Informatics, CISP-BMEI 2016*, 2016, pp. 1541–1545.
- [2] K. Umezawa, T. Saito, T. Ishida, M. Nakazawa, and S. Hirasawa, "An Electroencephalograph-Based Method for Judging the Difficulty of a Task Given to a Learner," *Proc. - IEEE 17th Int. Conf. Adv. Learn. Technol. ICAALT 2017*, 2017, pp. 384–386.
- [3] Y.J. Kim, N.S. Kwak, and S.W. Lee, "Classification of Motor Imagery for Ear-EEG Based Brain-Computer Interface," *2018 6th Int. Conf. Brain-Computer Interface, BCI 2018*, 2018, pp. 1–2.
- [4] R.N. Roy, S. Charbonnier, and S. Bonnet, "Detection of Mental Fatigue Using an Active BCI Inspired Signal Processing Chain," *IFAC Proc. Volumes*, Vol. 19, No. 3, pp. 2963-2968, 2014.
- [5] C. Rahmad, R. Ariyanto, and D. Rizky, "Brain Signal Classification using Genetic Algorithm for Right-Left Motion Pattern," *Int. J. Adv. Comput. Sci. Appl.*, Vol. 9, No. 11, pp. 247–251, 2018.
- [6] S. Sanei and J.A. Chambers, *EEG Signal Processing*, Hoboken, USA: John Wiley & Sons Ltd., 2007.
- [7] H.M. Hobson and D.V.M. Bishop, "Mu Suppression – A Good Measure of the Human Mirror Neuron System?," *Cortex*, Vol. 82, pp. 290–310, 2016.
- [8] S. Shahid, R.K. Sinha, and G. Prasad, "Mu and Beta Rhythm Modulations in Motor Imagery Related Post-stroke EEG: A Study Under BCI Framework for Post-stroke Rehabilitation," *BMC Neurosci.*, Vol. 11, No. S1, pp. 1–2, 2010.
- [9] F. Pichiorri, F. De Vico Fallani, F. Cincotti, F. Babiloni, M. Molinari, S.C. Kleih, C. Neuper, A. Kübler, and D. Mattia, "Sensorimotor Rhythm-based Brain-Computer Interface Training: The Impact on Motor Cortical Responsiveness," *J. Neural Eng.*, Vol. 8, No. 2, pp. 1-9, 2011.
- [10] X. Yong and C. Menon, "EEG Classification of Different Imaginary Movements within the Same Limb," *PLoS One*, Vol. 10, No. 4, pp. 1-24, 2015.
- [11] J.R. Wolpaw and D.J. Mcfarland, "Control of a Two-dimensional Movement Signal by a Noninvasive Brain-Computer Interface in Humans," *Proc. Natl. Acad. Sci. USA*, Vol. 101, No. 51, pp. 17849–17854, 2004.
- [12] M. Dai, D. Zheng, R. Na, S. Wang, and S. Zhang, "EEG Classification of Motor Imagery Using a Novel Deep Learning Framework," *Sensors (Switzerland)*, Vol. 19, No. 3, pp. 1–16, 2019.
- [13] S. Amari, "A Multichannel Deep Belief Network for the Classification of EEG Data," *J. Soc. Mech. Eng.*, Vol. 90, No. 823, pp. 758–759, 2017.
- [14] L.T. Xuyen, L.T. Thanh, D.V. Viet, T.Q. Long, N.L. Trung, and N.D. Thuan, "Deep Learning for Epileptic Spike Detection," *VNU J. Sci. Comput. Sci. Commun. Eng.*, Vol. 33, No. 2, pp. 1–13, 2018.
- [15] W.L. Zheng, J.Y. Zhu, Y. Peng, and B.L. Lu, "EEG-based Emotion Classification Using Deep Belief Networks," *Proc. - IEEE Int. Conf. Multimed. Expo*, 2014, pp. 1-6.
- [16] N. Michielli, U.R. Acharya, and F. Molinari, "Cascaded LSTM Recurrent Neural Network for Automated Sleep Stage Classification Using Single-channel EEG Signals," *Comput. Biol. Med.*, Vol. 106, pp. 71–81, 2019.
- [17] S. Jawed, H.U. Amin, A.S. Malik, and I. Faye, "EEG Visual and Non-visual Learner Classification Using LSTM Recurrent Neural Networks," *Proc. 2018 IEEE EMBS Conf. Biomed. Eng. Sci. IECBES 2018*, 2018, pp. 467–471.
- [18] A. Schlögl, C. Keinrath, D. Zimmermann, R. Scherer, R. Leeb, and G. Pfurtscheller, "A Fully Automated Correction Method of EOG Artifacts in EEG Recordings," *Clin. Neurophysiol.*, Vol. 118, No. 1, pp. 98–104, 2007.
- [19] H. Mo and Y. Zhao, "Motor Imagery Electroencephalograph Classification Based on Optimized Support Vector Machine by Magnetic Bacteria Optimization Algorithm," *Neural Processing Letters*, Vol. 44, No. 1, pp. 185-197, Aug. 2016.
- [20] L. Duan, Z. Hongxin, M.S. Khan, and M. Fang, "Recognition of Motor Imagery Tasks for BCI Using CSP and Chaotic PSO Twin SVM," *J. China Univ. Posts Telecommun.*, Vol. 24, No. 3, pp. 83–90, 2017.
- [21] A. Craik, Y. He, and J.L. Contreras-Vidal, "Deep Learning for Electroencephalogram (EEG) Classification Tasks: A Review," *J. Neural Eng.*, Vol. 16, No. 3, pp. 1-28, 2019.
- [22] G.E. Hinton, S. Osindero, and Y.-W. Teh, "A Fast Learning Algorithm for Deep Belief Nets," *Neural Comput.*, Vol. 18, pp. 1527–1554, 2006.
- [23] X. An, D. Kuang, X. Guo, Y. Zhao, and L. He, "A Deep Learning Method for Classification of EEG Data Based on Motor Imagery," *Proc. 10th Int. Conf. Intelligent Computing in Bioinformatics (ICIC 2014)*, 2014, pp. 203-210.
- [24] M.A. Keyvanrad and M.M. Homayounpour, "A Brief Survey on Deep Belief Networks and Introducing a New Object Oriented Toolbox (DeeBNet)," Lab. for Intelligent Multimedia Processing (LIMP), Amirkabir University of Technology, Tehran, Iran, Tech. Report, 2014, pp. 1–27.
- [25] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," *Neural Computation*, Vol. 9, No. 8, pp. 1735–1780, 1997.
- [26] S. Alhagry, A.A. Fahmy, and R.A. El-Khoribi, "Emotion Recognition based on EEG using LSTM Recurrent Neural Network," *Int. J. Adv. Comput. Sci. Appl.*, Vol. 8, No. 10, pp. 8–11, 2017.
- [27] M.-H. Horng, "Fine-Tuning Parameters of Deep Belief Networks Using Artificial Bee Colony Algorithm," *DEStech Trans. Comput. Sci. Eng.*, pp. 69–72, 2018.
- [28] B. Nakisa, M.N. Rastgoo, A. Rakotonirainy, F. Maire, and V. Chandran, "Long Short Term Memory Hyperparameter Optimization for a Neural Network Based Emotion Recognition Framework," *IEEE Access*, Vol. 6, pp. 49325–49338, 2018.