*Journal of Autonomous Intelligence* (2019) Volume 2 Issue 4.
doi:10.32629/jai.v2i4.60

## Original Article

# Research on the Key Technologies of Motor Imagery EEG Signal Based on Deep Learning

***Zhuozheng Wang[1*], Zhuo Ma[1], Xiuwen Du[1], Yingjie Dong[1], Wei Liu[1]***

*Beijing University of Technology*

## ABSTRACT

Brain-computer interface (BCI) is an emerging area of research that establishes a connection between the brain and external devices in a completely new way. It provides a new idea about the rehabilitation of brain diseases, human-computer interaction and augmented reality. One of the main problems of implementing BCI is to recognize and classify the motor imagery Electroencephalography(EEG) signals effectively. This paper takes the characteristic data of motor imagery of EEG as the research object to conduct the research of multi-classification method. In this study, we use the Emotiv helmet with 16 biomedical sensors to obtain EEG signal, adopt the fast independent component analysis and the fast Fourier transform to realize signal preprocessing and select the common spatial pattern algorithm to extract the features of the motor imagery EEG signal. In order to improve the accuracy of recognition of EEG signal, a new deep learning network is designed for multi-channel self-acquired EEG data set which is named as min-VGG-LSTMnet in this paper. This network combines Long Short-Term Memory Network with convolutional neural network VGG and achieves the four-classification task of the left-hand, right-hand, left-foot and right-foot lifting movements based on motor imagery. The results show that the accuracy of the proposed classification method is at least 8.18% higher than other mainstream deep-learning methods.

***Keywords:*** *Electroencephalography; Motor Imagery; Convolutional Neural Network; Long Short-term Memory Network*

**ARTICLE INFO**
Received: Oct 29, 2019
Accepted: Feb 23, 2020
Available online: Feb 24, 2020

***CORRESPONDING AUTHOR**
Dr. Zhuozheng Wang, Beijing University of Technology, China;
wangzhuozheng@bjut.edu.cn;

**CITATION**
Zhuozheng Wang, Zhuo Ma, Xiuwen Du, Yingjie Dong, Wei Liu. Research on the Key Technologies of Motor Imagery EEG Signal Based on Deep Learning. Journal of Autonomous Intelligence 2019; 2(4): 1-14. doi: 10.32629/jai.v2i4.60



## 1. Introduction

BCI is an intelligent system that enables users to communicate with external devices such as computers or neural prostheses without the involvement of peripheral nerves and muscles[1]. It has been widely studied recently. The research of BCI has been applied to various aspects of a wide range of fields. In the first place, BCI can assist people better understand, protect and exploit the brain in the field of brain cognition; Additionally, it is able to convert the information sent by the brain into a command of external device to help stroke patients to communicate with the outside world in the field of medical rehabilitation; Further, BCI technology provides the possibility of intelligent weaponry which can remotely control "machine soldiers" in the military field.

A BCI-based system generally records the signals generated by the user's brain and controls a machine by detecting the user's intent through pre-processing, feature extraction, and classification of brain signals[2]. Among the various methods to capture the brain activities, electroencephalography (EEG) is commonly used to collect and feed input signals to BCI systems owing to its non-invasiveness and low cost[3]. EEG is an electrical phenomenon exhibited by the electrophysiological activity of the brain's nerve cells on the surface of the cerebral cortex or scalp. It is most widely and commonly used in motor

imagery-based BCI applications[4].

The classification of motor imagery EEG signal has obtained many great breakthroughs with the effort of researchers at home and abroad. Professor Pfurtscheller[5] designed a BCI system based on Graz I and Graz II theory, which enables motor imagery EEG devices to perform different actions based on classification signals, such as cursor control or character selection operation. Based on the results of Graz's research, Bernhard Obermaier[6] studied 5 different motor imagery tasks of the left hand, right hand, left foot, right foot and brain computing on the lifting movement, and divided the research into different brain task combinations of 2 ,3, 4, and 5 classes. The experimental result shows that the fusion of three classes of different motor imagery tasks obtained the largest information transmission rate. Wan Baikun[7] of Tianjin University adopted two-dimensional time-frequency analysis combined with Fisher analysis to extract features of four different limb parts, and support vector machine was used to recognize. The accuracy rate of that is 85.7%. Xu Xin[8] of Nanjing University of Posts and Telecommunications used common spatial patterns combined with support vector machine to extract and classify the EEG signals in four kinds of motor imagery and the highest accuracy rate is 86.3%.

EEG signal is transient, non-stationary, low signal-to-noise ratio and easy to interfere. Therefore, the difficulty of BCI based on motor imagery is how to extract the most important information and select the optimal model to achieve high-precision recognition and classification. The signal-to-noise of EEG signal is relatively low and susceptible to interference. At present, the noise reduction of EEG signal is mainly achieved by a method of simple filtering, but some important features will lose. Therefore, we need to find a method of data preprocessing that can both reduce noise and retain important information. The existing feature extraction methods have low versatility. For the recognition of two types of tasks, the average accuracy of the algorithm can reach 85%, but the recognition of multi-class tasks is far from satisfactory. Deep learning is a method based on learning the characteristics of sample data. It can be understood as "feature learning" or "representation learning". However, EEG data records the sampling points of the electrode channel over a period of time, which not directly applicable to the traditional deep learning method. In this paper, we design a min-VGG-LSTMnet hybrid deep learning network to solve the problem of low recognition accuracy of multi-class task. It combines Long Short-Term Memory Network with VGGnet, a classic model of Convolutional Neural Network. The min-VGG-LSTMnet achieved high-precision of four-class task based on motor imagery. Compared the performance of the proposed method is with the mainstream deep learning method, the result demonstrates that the accuracy of the proposed network is improved at least 8.18%, and the loss value is reduced by at least 0.0288. The analysis conclusively proves that the proposed min-VGG-LSTMnet is superior to other approaches.

The rest of this paper is arranged as follows: Section 2 introduces related works including signal acquisition, signal preprocessing and feature extraction. Section 3 introduces the design method of deep neural network. Section 4 the experimental results are discussed.

## 2. Materials and Methods

### 2.1 Signal acquisition

According to the endogenous and exogenous stimuli of event-related potential, EEG is divided into two types, spontaneous EEG and induced EEG. The rhythmic changes of EEG signal generally belong to endogenous stimuli, and the frequency is in the range of 0~30Hz, which is divided into five basic bands, $\delta$ band (<4 Hz), θ band (4~8 Hz), $\alpha$ band (8~14 Hz), $\beta$ band (14~30 Hz) and $\gamma$ band (>30 Hz). Especially, $\alpha$ and $\beta$ waves are directly related to the motor imagery BCI study. The increase of the working memory load in the motor imagery is often accompanied with the increase of the power of θ wave, while the $\gamma$ wave is often not considered in the BCI study. Event-Related Synchronization (ERS)[9] refers to the increase of cortical activity in specific frequency bands (such as $\alpha$ and $\beta$ bands), and Event-Related Desynchronization (ERD)[10] refers to the reduction of cortical activity in specific frequency bands (such as $\alpha$ and $\beta$ bands). They mainly reflect the changes of EEG amplitude caused by motor imagery.

The quantization method for different band power is

given by the following formula (1):

$$\frac{ERD}{ERS} = \frac{A-R}{R} \times 100\%, \quad (1)$$

where A is the energy value of the brain band signal, R is the average power of the band, ERD/ERS is the power reduction or increase of the band signal. The negative percentage value represents ERD, and the positive percentage value represents ERS.

In order to analyze the motor imagery EEG signals of different movements and obtain the intrinsic relationship between EEG signal and motor imagery tasks, two different data sets are used in this paper. The first is the official standard competition data set (BCI Competition III data set[11]), which use 118 electrodes at the location of extended international 10-20 system to record EEG signals data of five subjects ("aa", "al", "av", "aw" and "ay") and divided them into training set and test set. The subjects perform one of three imaginary movements, (L)raise left hand, (R)raise right hand, (F) raise right foot. Before t=2s, subjects gaze the computer monitor, keeping arms and feet relaxed and avoiding movement of eyes. When t=2s, the "+" cursor appears on the screen, prompting the subjects to start preparing. After 1s, the "+" cursor on the screen will randomly change to the arrow of left, right and up, and the subjects will perform the same imaginary action of left hand, right hand or right foot. When t=7s, the arrow disappears and the single experiment ends. Each subject contains 140 sample sets per imaginary task and each sample set includes 1520 sample points. The second data set is the EEG data of 20 real students collected in the laboratory. We let 20 subjects carry out the motor imagery tasks, and collect the data of EEG signal. All subjects gave their informed consent for inclusion before they participated in the study. Each subject was divided into four experimental groups as required, including four types of motor imagery tasks, raise the left hand, raise the right hand, raise the left foot, and raise the right foot. First, from the start of the time to 2nd second before, the subject stares at the computer monitor and keeps the arms and feet relaxed, avoiding eyes movement. At 2nd second, a "+" cursor appears in the center of the display screen lasting for about 1 second, the subject is prompted to concentrate on preparation. At 3rd second, the screen randomly displays arrows indicating the left and right directions, and the subject should complete the imaginary motion of left hand, right hand, left foot and right foot according to the direction of the arrow. When the arrow disappears, the subject ends the experiment. The single experiment lasts for about 9 seconds. There will be 128 sample points per second. In order to avoid errors caused by the subjects not entering the test state during the experiment and ensure the accuracy of analysis and processing of EEG signal, we cut off the data for the first 3 seconds and the last 1second, and retain the samples for the middle 5 seconds. The experiment of each subjects lasts 4 minutes and about 3.72 million sample points are obtained.

14-channel EEG collector Emotiv of Emotiv company is used as the signal acquisition instrument in this paper. The most important part of the Emotiv helmet is 16 biomedical sensors. On the one hand, it touches the scalp to sense the nerve signal. On the other hand, it transmits the signal to the computer. The order of the channels of electrode from left to right is AF3, F7, F3, FC5, T7, P7, O1, O2, P8, T8, FC6, F4, F8, AF4, and there are two reference electrodes CMS and DRL. The material of sensor is felt pad soaked in saltwater. Sampling method is sequential. The position of electrode is shown in **Figure 1**:

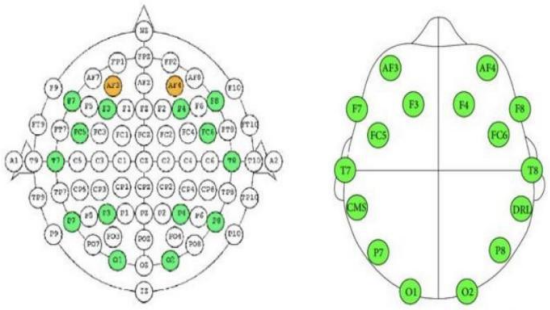


**Figure 1.** Electrode position map.

## 2.2 Motor imagery EEG signal preprocessing

Independent Component Analysis (ICA)[12] as a spatial filtering technique, its coefficients are determined by the statistical correlation of existing data. ICA is a statistical analysis method of blind source separation. "Blind" means unclear source signal or hybrid system. ICA can efficiently extract independent original signals from a mixture of multiple sensors. The FastICA[13] algorithm based on fixed-point iteration is

used to find the non-Gaussian $W^T X$ maximum. FastICA is implemented by the EEGlab, which is an EEG analysis toolbox[14]. It can remove the trials containing artifacts based on statistical features (i.e. variance, kurtosis and maximum). Then the fast Fourier transform method is introduced to make the frequency accurate to 1 Hz and intercepting the time window of the EEG signal from 14 channels.

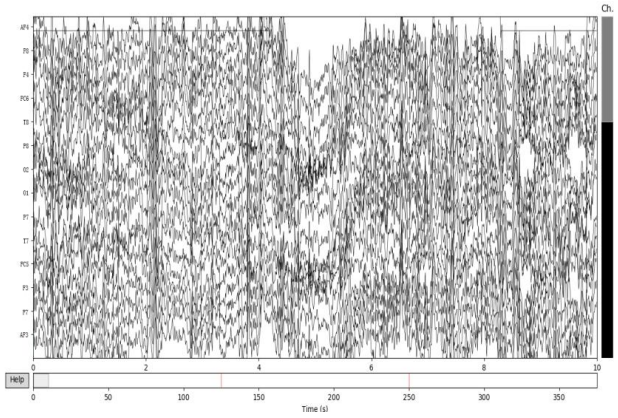

**Figure 2.** Each channel's original data brainwave shape.

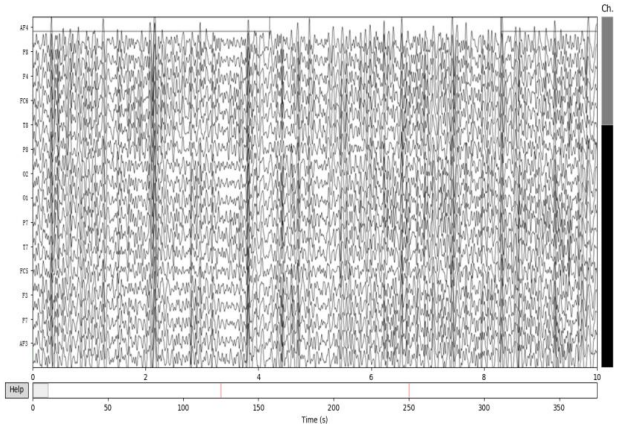

**Figure 3.** Get rid of the fake brainwave shape.

When analyzing EEG signals, the potential activities of electrodes at different spatial positions have a correlation law, which reflects the synchronous and asynchronous electrical activity of cerebral cortex potential. Therefore, the spatial characteristics are very useful to analyze features of EEG signals. EEG are multiple time series of different spatial locations measured on the scalp. The spatial characteristics of EEG signals can be obtained by predicting the mapping position of the electrodes from three-dimensional space to two-dimensional surface. The mapping method uses Equidistant Projection proposed by Azimuthal, namely AEP[15]. The distance from the center of projection to any other point can be preserved by azimuthal equidistant projection.

Assuming A(r, θ, Φ) is a point in the 3-D space, which is projected by the AEP and falls in the point M on the plane of the 2-D image. The equidistant projection model is:

$$\begin{bmatrix} 0 & f & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r \\ \theta \\ \phi \end{bmatrix} = \begin{bmatrix} d \\ \phi \end{bmatrix}, \tag{2}$$

In this paper, the shape of the Emotiv collector can be approximated by a sphere. So, we can use the same method to calculate the projection of the position of electrode on 2-D surface. As shown in **Figure 4**:

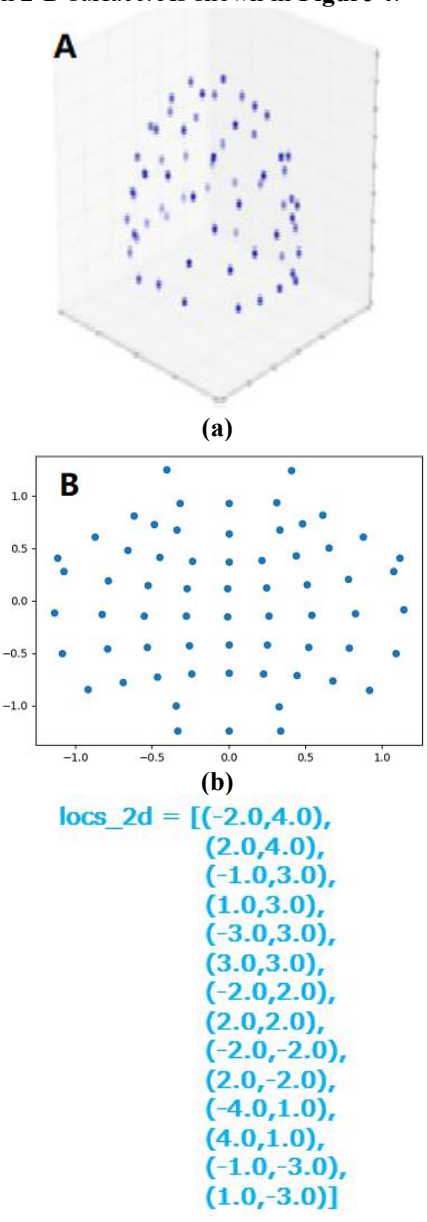


**Figure 4.** AEP 3-D spatial position is converted to 2-D position. **(a)**The position of electrodes in 3-d space. **(b)** AEP orthogonal projection. **(c)** Electrode coordinate.

We use the Clough-Tocher scheme[16] to estimate the value of electrodes. Repeating this process for θ band, α band and β band respectively will produce three corresponding brain topographic maps. And then merging the three brain topographic maps together to form a color image similar to RGB. (three Parameters: height, width, and color depth). Where the color indicates different band, the width and height of the image indicate the spatial distribution of activity on the cerebral cortex corresponding to the electrode of Emotiv, which retains the spatial characteristics of EEG signals. The temporal evolution of brain activity is calculated by image sequences derived from continuous-time windows, which retains the temporal characteristics of EEG signals. The color image is shown in **Figure 5**:

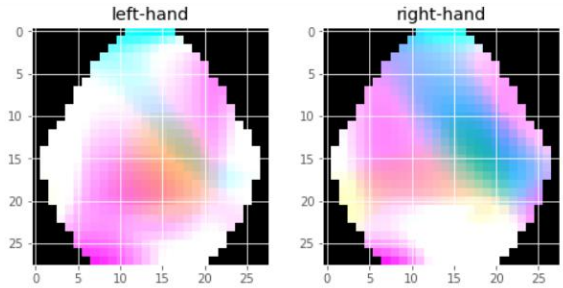


**Figure 5.** Combination of brain activity maps into two-dimensional images of $\theta$, $\alpha$, and $\beta$ ranges.

## 2.2 Common spatial pattern feature extraction method

The most commonly used feature extraction method of detecting ERD/ERS phenomenon is the Common Spatial Pattern (CSP) algorithm, which is a spatial filtering method. The basic principle is to identify features by constructing a spatial filter to make the variance in one type of features is the largest, while the variance in the other type of features is the smallest and get eigenvectors with higher resolution. In the multi-classification problem, the One-Versus-Rest CSP (OVR-CSP)[17] method implements multi-classification feature extraction through the binarization of multi-class problems and count the majority vote of the binary classifier to determine the class label for each test sample.

In data set 1, OVR-CSP divides three types of motor imagery tasks into binary tasks, obtaining three projection matrices and three sets of corresponding spatial features. A given single EEG test set with $N$ channels is represented as a matrix $R$ of size $N \times T$, where $T$ represents the number of samples in each channel in a single experiment, $X_i(i = 1,2,3)$ corresponds to three tasks EEG signals respectively. The normalized covariance matrix $R_i$ of the three types of data is calculated as:

$$R_i = \frac{X_i X_i^T}{trace(X_i X_i^T)}, i = 1,2,3, \tag{3}$$

where $trace$ is the trace operation. The mixed space covariance matrix is as follow:

$$R = \overline{R_1} + \overline{R_2} + \overline{R_3}, \tag{4}$$

where $\overline{R}_i(i = 1,2,3)$ is the average covariance matrix of multiple experiments for three tasks. The eigenvalue decomposition of $R$ is :

$$R = UVU^T, \tag{5}$$

Where $U$ and $V$ represent the eigenvector matrix of $R$ and the eigenvalue diagonal matrix respectively. The eigenvalue diagonal array is arranged in descending order. Whitening matrix $P$ is:

$$P = V^{-\frac{1}{2}}U^T, \tag{6}$$

When using OVR-CSP calculates the projection matrices, one of them is classified into one category, and the other two categories are another category, represented by $R_1^{'}$:

$$R_1^{'} = R_2 + R_3, \tag{7}$$

The two types of signal covariance matrices of the new classification are whitened to:

$$S_1 = P_1 \overline{R_1} {P_1}^T; S_2 = P_1 \overline{R_1^{'}} {P_1}^T, \tag{8}$$

In the above transformation, $S_1$ is a class of covariance, and $S_2$ is classified into another class, which can be written as $S_1 = U_1 V_1 {U_1}^T, S_2 = U_1 V_1^{'} {U_1}^T$, where $U_1$ is a common eigenvector Matrix. $V_1 + V_1^{'} = I$ (unit matrix), when one of the types of eigenvalues is the largest, the other type of eigenvalue is the smallest.

$$F_1 = {D_1}^T P; F_2 = {D_2}^T P, \tag{9}$$

According to the size of the eigenvalues, the first m columns of the eigenvectors are formed into a new matrix $D_1$, and the remaining columns have constituted a matrix $D_2$. $D_1$, $D_2$ and the whitening matrix together form a spatial filter namely F= $[F_1,F_2]$. A new signal $Z_i = F \times X_i$ is obtained by spatial filter projection transformation. We get the eigenvalue of the signal from the following algorithm:

$$f_i = log\left(\frac{\mathrm{var}(Z_i{}^2)}{\sum_1^3 \mathrm{var}(Z_i{}^2)}\right), \quad (10)$$

where $f_i$ is the characteristic coefficient.

The data set is preprocessed to obtain the EEG signals after filtering out the noise. The EVR/ERS feature of the α-band and β-band of EEG signals is extracted by the OVR-CSP method. Combined with multifarious classifiers to complete the three-classification tasks, then we take the category of the maximum probability by voting. Counting the ERD/ERS phenomenon of the left hand, right hand and right foot motor imagery tasks in the data set 1. As shown in **Table 1**. The EEG monitoring process begins by locating the sites for electrode placement based on the international 10–20 system[18]. The motor imagery task is mainly related to the channels C3, Cz and C4, so only consider these three channels. For classification, classical machine learning methods such as support vector machines (SVMs), linear discriminant analysis (LDA), and naive Bayes (NB) algorithms have been commonly used[19]. The traditional classifier is suitable for data sets with small subjects and small data volume, so we choose the data set 1 to test. The average classification performance specific parameters obtained by the experiment with 5-fold cross-validation, the average accuracy of different categories obtained from different classification algorithms and experiments of each subject are shown in **Table 2**.

**Table 1.** Left hand, right hand and right foot motion imaging tasks for ERD/ERS

| **Motion Imaging Tasks** | **C3** | **C4** | **Cz** |
|---|---|---|---|
| Left hand | ERS | ERD | / |
| Right hand | ERD | ERS | / |
| Right foot | ERS | ERS | ERD |

**Table 2.** Classification performance of dataset 1

| **Serial Number** | **SVM** | **Naive Bayes** | **LDA** | **Mean** |
|---|---|---|---|---|
| aa | 0.8253 | 0.7346 | 0.8753 | 0.8117 |
| al | 0.8232 | 0.6875 | 0.8671 | 0.7926 |
| av | 0.8074 | 0.6723 | 0.8426 | 0.7741 |
| aw | 0.7562 | 0.6214 | 0.7923 | 0.7233 |
| ay | 0.7935 | 0.6541 | 0.8537 | 0.7671 |
| Mean | 0.8011 | 0.674 | 0.8462 | Mean |

Deep learning has evolved from machine learning. Compared to feature extraction and classification of traditional machine learning, the biggest improvement of deep learning is the realization of end-to-end data learning, which is suitable for increasing data volume. The application of deep learning for human activity recognition has been effective in extracting discriminative features from raw input sequences acquired from body-worn sensors. Researchers have been adopting deep-learning methods for activity recognition[20]. Deep learning is a learning-based method using a neural network structure with multi hidden layers[21]. The network structure of deep learning designed in this paper are built under the deep learning framework of Keras. There are many deep-learning algorithms. Mainstream deep-learning algorithm includes CNN, RNN, and LSTM.

Convolutional Neural Networks (CNN)[22] is the most popular network algorithm in deep learning. A CNN is a neural network that uses convolution operation instead of traditional matrix multiplication in at least one layer of the network[23]. This paper adopted the method of classifying EEG signal into "video frames" to convert motor imagery EEG signal into image form and applying it to deep learning networks. Firstly, from the network layer structure, CNN is divided into convolutional layer, pooling layer and fully connected layer, which form a complete network structure with feature extraction and classification function through stacking. The two most important features of CNN are local association and parameter sharing. The convolution operation of the convolutional layer can be seen as a mathematical operation of two mutation functions. For one-dimensional convolution, it is often used in signal processing to calculate the delay accumulation of signal. For two-dimensional convolution, the image is represented by the pixels of a two-dimensional matrix. That means given an image $X \in R^{M*N}$, convolution kernel parameters $W \in R^{m*n}$ and $y = WX+b$. Where $X$ is the input of the CNN, $W$ is the weight of the convolution kernel. When the convolution kernel is convolved, the related input region is called the receptive field, and the related output result is called the feature map, the number of feature map is also called depth.

Recurrent Neural Network (RNN)[24] is a neural network with short-term memory ability, which is often used for sequence modeling. In addition to input $X_t$, RNN also has a previous node $S_{t-1}$ of input hidden layer. The output of each layer of RNN is the result of combining the two inputs with matrix $W$ and activation function. The input of RNN is the hidden state of the sequence data x and the last round of the calculated output. Assuming the weight matrix of the hidden layer of the RNN is $W$, and the hidden state is $S$. Expanding the RNN on the time axis, the first layer of the RNN is the input layer. And the input features are passed to the first hidden layer by the neurons of the first layer. The output layer predicts the current time $S_t$ with the weight matrix $V$. The probability value can be predicted by softmax. The parameters $W$ required for each hidden layer calculation are shared parameters.

In principle, RNN can handle such long-term dependency problems. After the network receives the input $X_t$ at time $t$, the value of the hidden layer is $S_t$, the output value is $O_t$, the final result of the network at time ($t$+1) is $O_{t+1}$, which is the result of the current input and all history. This realizes the process of continuous transmission of the RNN of time sequence. However, In an RNN, increasing the data length may induce a gradient error, or a gradient explosive may occur when error parameters are back-propagated. The gradient explosion phenomenon does not meet the training objectives, and a typical RNN does not provide satisfactory results[25]. The long short-term memory network proposed by Hochreiter and Schmidhuber[26] can effectively solve this problem.

Long short-term memory (LSTM) is a variant structure of RNN, which replaces the summation unit of the hidden layer with a recursive substructure memory block, which is better at storing and accessing long dependencies in data. Each memory block contains an input gate, a forget gate and an output gate in the original architecture. In an LSTM unit, the cell state controls the discarding and adding of information through the gate to achieve forgetting and memorizing functions[27]. It can automatically forget or retain the memory of the unit. Using the current input $x^t$ of LSTM and the $h^{t-1}$ passed from the previous state to get four states:

$$\mathrm{z} = tanh\left(\mathrm{w}\frac{x^t}{h^{t-1}}\right), \tag{11}$$

$$\mathrm{z}^i = \sigma\left(\mathrm{w}^i\frac{x^t}{h^{t-1}}\right), \tag{12}$$

$$\mathrm{z}^f = \sigma\left(\mathrm{w}^f\frac{x^t}{h^{t-1}}\right), \tag{13}$$

$$\mathrm{z}^o = \sigma\left(\mathrm{w}^o\frac{x^t}{h^{t-1}}\right), \tag{14}$$

Where $z^i$, $z^f$, $z^o$ are multiplied by the splicing vector and then converted to a value between 0 and 1 by a sigmoid activation function as a gating state. And z is the value converting the result to a value between -1 and 1 through a *tanh* activation function. There are three main phases within LSTM:

(1) The stage of forgetting. The calculated $z^f$ (f represents the forget) is the forget gate to control the previous state $c^{t-1}$, and decide which ones need to stay and forget.

(2) Select the memory phase. Select the memory from input $x^t$. The current input is represented by the previously calculated *z*. The selected gating signal is controlled by $z^i$ (*i* represents information). Adding the results obtained in the above two steps, we can get the next state $c^t$.

(3) Output stage. It is controlled by $z^0$. Similar to the ordinary RNN, the output $y^t$ is often obtained by the change of $h^t$. The state changes are as follows:

$$c^t = z^f \odot c^{t-1} + z^f \odot z, \tag{15}$$

$$h^t = \mathrm{z}^0 \odot \tanh(c^t), \tag{16}$$

$$y^t = \sigma(w^{\cdot} h^t), \tag{17}$$

The updated values are controlled by $z$, $z^f$, $z^0$, $z^i$. In LSTM, there are two iterative values, $c^t$ and $h^t$. $\mathrm{z}^f$ controls the degree of forgetting of $c^t$, while $z^i$ and z control the degree of update of $c^t$, $\mathrm{z}^0$ controls the degree of expression of $c^t$ to $h^t$.

We set some hyperparameters as follows in this paper:

(1) Learning rate. Learning rate is a very important hyper parameter during the network training process, which determines whether the objective function can converge to a local minimum and when it converges to a minimum. If it is set too large, the loss function value *loss* will "explode", and if it is set too small, the training fit takes too long. At present, a gradient descent convergence algorithm is widely used in deep learning. The formula for the gradient descent method is:

$$\omega^* = \omega - \alpha \frac{\partial}{\partial \omega} loss(\omega), \tag{18}$$

The above formula can update the weight ω. Where α is the learning rate, and the initial learning rate α= 0.001 in this experiment.

(2) Optimizer selection. The stochastic gradient descent (SGD) algorithm is as follows:

$$V_{t+1} = \mu V_t + \alpha \nabla \mathrm{L}(\mathrm{W}_t), \tag{19}$$

$$W_{t+1} = W_t - V_{t+1}, \tag{20}$$

where $V_{t+1}$ is the updated value of the network weight in the *t*+1th iteration, and $W_{t+1}$ is the network weight in the *t*+1th iteration.

Adam is an optimizer based exponential decay of the gradient. Calculating the gradient of the *t*-time as follow:

$$g_t = \nabla J(\theta_{t-1}), \tag{21}$$

The exponential moving average of the gradient is calculated as follows:

$$m_t = \beta_1 m_{t-1} + (1-\beta_1) g_t, \tag{22}$$

$$v_t = \beta_2 v_{t-1} + (1-\beta_2) {g_t}^2, \tag{23}$$

where M0 is initialized to 0, $\beta 1$ is the exponential decay rate, usually close to 1. Deviation correction for $m_t$, $v_t$ and the corrected $\widehat{m_t}$, $\widehat{v_t}$ are as follows:

$$\widehat{m_t} = m_t / (1 - {\beta_1}^t), \tag{24}$$

$$\widehat{v_t} = v_t / (1 - {\beta_2}^t), \tag{25}$$

Finally:

$$\theta_t = \theta_{t-1} - \alpha^* \widehat{m_t} (\sqrt{\widehat{v_t}} + \varepsilon), \tag{26}$$

Applying the following operations on Adam we can get the Adabound optimizer:

$$\widehat{\eta_t} = Clip\left(\alpha / \sqrt{v_t}, \eta_l(t), \eta_\mu(t)\right), \tag{27}$$

$$\eta_t = \widehat{\eta_t} / \sqrt{t}, \tag{28}$$

(3) Batch size and epoch. The value of the epoch is determined by fitting the curve. Batch size is often set to 32, 64, 128, etc.

(4) Activation function and weight. The commonly used activation function is RELU. LRELU and ELU are developed on the basis of the RELU function. They are expressed as follows:

$$\mathrm{Re}LU(x) = \begin{cases} x & if\ x > 0 \\ 0 & if\ x \le 0 \end{cases}, \tag{29}$$

$$LReLU(x) = \begin{cases} x_i & if\ x_i > 0 \\ a_i x_i & if\ x_i \le 0 \end{cases}, \tag{30}$$

$$ELU(x)\begin{cases} x & if\ x > 0 \\ a(\exp(x) - 1) & if\ x \le 0, \end{cases} \quad (31)$$

where $x$ is the characteristic of the input and $a$ is the coefficient of the activation.

(5) Batch Normalization(BN). Satisfy the following equation:

$$BN = \frac{X-\bar{\mu}}{\sqrt{\delta+0.01}} \cdot scale + shift, \quad (32)$$

(6) Dropout. In this paper, the dropout values are set to 0.25, 0.5, and 0.75 respectively. The experimental results show that the network training is better at 0.5.

(7) Positive and negative sample ratio. In this paper, the positive and negative samples of the data set are 1:1, and the batch method is used for training.

(8) Gird Search. The number of iterations epoch and batch size can be determined by Gird Search. The final epoch set in this paper is 500 to 1000, and the batch size is 128.

In this paper, the classification accuracy (ACC)[28] combined with the cross-entropy loss function (Loss) are used to evaluate the experimental criteria. ACC is calculated as:

$$ACC = \frac{TP+TN}{TP+FP+FN+TN}, \quad (33)$$

where TP is true positive, FP is false positive, TN is true negative, FN is false negative. The higher the accuracy, the better the classifier. The cross-entropy loss function L is calculated as:

$$L = \sum_{i=1}^{N} y^{(i)} \log\hat{y}^{(i)} + \left(1 - y^{(i)}\right)\log(1 - \hat{y}^{(i)}), \quad (34)$$

where $y^{(i)}$ is the desired output, $\hat{y}^{(i)}$is the actual output. The greater the loss, the larger the gradient.

In order to find the optimal classification method, this paper designs four network structures, namely EEGnet, CNN-2net, min-VGGnet and min-VGG-LSTM -net.

Vernon J. Lawhern[29] of Columbia University of America proposed two EEGnet structures, which directly apply convolutional neural networks to EEG signals analysis. This paper adopted the EEGnet shallow network, and used one-dimensional convolution to classify time-domain EEG signals. We only extract the single-dimensional features of EEG signals. The data size is (*N, T*), where *N* is the number of channels and *T* is the number of samples. The first layer of convolutional layer we use a size with *N*×1 one-dimensional convolution kernel to extract the spatial channel features. Convolution kernels are set 40. The second layer of convolutional layer we use 1×16, the size of each feature map obtained is 1×8. Convolution kernels are set 80. The third layer is a fully connected layer, and the neuron format is set to 256. The fourth layer is the output layer, which contains 4 neurons, which represents the four-class task.

CNN-2net network model framework is shown in **Table 3**:

**Table 3.** CNN-2net network model framework

| **CNN-2net** |
|---|
| 4 weight layers |
| Conv5-32 |
| Maxpool |
| Conv3-64 |
| Maxpool |
| FC-1024 |
| FC-num_classes |
| Softmax |

There are two layers in the structure of convolutional layer, and the size of the convolution kernel can be set casually. The size of the convolution kernel in CNN-2net is 5×5, and the number of convolutions generally increases with the number of network layers. The activation function used in convolutional neural networks is RELU[30]. The adopted optimizer Adam optimizer which most commonly used. The number of neurons and parameters of each layer are adjusted as follows:

The first layer: the input layer inputs a picture corresponding to an array of size 28×28×3.

The second layer: the convolution layer, which uses 32 convolution kernels of size 5 × 5 × 3 to convolve the maps of the input layer, so it contains 32 × 5 × 5 × 3 = 2400 weight parameters. After convolution, the length of

the picture is (28-5+1)/1 = 24, including 32 × 24 × 24 × 3 = 55296 neurons.

The third layer: the pooling layer, samplings each 2×2 area of the previous layer, and selects the maximum value of each area. This layer has no parameters. After sampling, the length and width of each map become half of the original.

The fourth layer: the convolution layer, which uses 32 × 64 convolution kernels of size 5 × 5 × 3 to convolve each map of the previous layer, so it contains 32 × 64 × 5 × 5 × 3 = 153600 weight parameters. After convolution, the length of the picture is (12-5+1)/1 = 8, including 64×8×8×3=12288 neurons.

The fifth layer: the pooling layer, 8 × 8 × 3 map downsampling to 4 × 4 × 3 map. This layer has no parameters.

The sixth layer: the fully connected layer, which connects all the neurons of the output of the pooling layer. This layer has 1024 neurons, and 64×4×4×3×1024=3145728 weight parameter.

The seventh layer: the fully connection layer, which function is similar to the previous fully connection layer. This layer has num_classes neurons, which are related to the task category and 1024×num_classes corresponding parameters.

The eighth layer: the Softmax layer, which is to achieve classification and normalization operations.

The min-VGGnet network is obtained by adjusting the number of the convolution layers and the number of convolution kernels in the VGG network structure. Its structure is shown in **Table 4**:

**Table 4.** Evaluation of the optimal convolution network min-VGGnet

| A | B | C |
|---|---|---|
| 6 weight-layers | 7 weight-layers | 9 weight-layers |
| | | Conv3-32 |
| Conv3-32 | Conv3-32 | Conv3-32 |
| Conv3-32 | Conv3-32 | Conv3-32 |
| | | Conv3-32 |
| | maxpool | |
| Conv3-64 | Conv3-64 | Conv3-64 |
| Conv3-64 | Conv3-64 | Conv3-64 |
| | maxpool | |
| | Conv3-128 | Conv3-128 |
| | maxpool | |
| FC-512 | | |
| FC-num_classes | | |
| Softmax | | |

Three sets of comparative experiments were performed, the difference is the number of layers of the convolution kernel. All convolution kernel sizes are 3×3. Experiment A was a 6-layer structure. The first layer is a stacking of two convolutional layers (Conv3-32), the second layer is further combined with stacking of two convolution layers (Conv3-64), the third layer is maxpool layer. Experiment B is added a convolution layer on the basis of A (Conv3-128). Experiment C differed from B in that the first layer of convolutional layer is stacked with four-layer convolution kernels. The experimental results show that the optimal network structure diagram is the structure of experimental A. The parameters involved in the network structure and training process are described below.

The first layer is the convolution layer, which uses 32 two-layer convolution kernel stack structure with the size of [3 × 3]. The step size padding is set to the mode of "same", and the step size determines the distance moved in the direction of the gradient drop during each iteration. Normally step size is set to 1. Processing input data uses batch standardization, and the training set sample is convoluted. RELU is adopted as the activation function, and the feature map has the same size at the input and output.

The second layer is the pooling layer. This layer uses the maximum pooling function with a size of [2×2] namely maxpool to pool the output of the convolution.

The third layer is the convolution layer. This layer uses 64 two-layer convolution kernel stack structure with a size of [3×3] to further extract deeper features.

The fourth layer is also the pooling layer, that still uses the largest pooling function, namely maxpool

The fifth layer is the fully connected layer, which is a tiled structure. The feature becomes a vector with a size of 1"×" 512 through a fully connected layer. The fully connected layer is a highly purified feature.

The sixth layer is also a fully connected layer, whose purpose is to complete the final classification and determine the final classification according to the number of categories num_classes required by the sample label.

The seventh layer is the Softmax layer, namely the activation function is "Softmax". The final classification is completed.

Long short-term memory network (LSTM) is an improved RNN, which can process long-time sequence information better. Therefore, LSTM is introduced for hybrid network design. The design idea is to use the min-VGGnet structure with better performance in the previous section. In this paper, the total number of parameters in a network with a lower number of neurons in the fully connected layer are retained. Dropout is set to 0.5, we can get the last two fully connected layers. In this hybrid model, the output of per-layer convolutional layer of min-VGGnet performs maximum pooling. Applying one-dimensional convolution and one-dimensional pooling to the output of the convolutional layer, and sending the output after maximum pooling to the LSTM layer. LSTM can capture features of different time patterns. The LSTM takes the input x=($x_1$,⋯,$x_T$) through a form of sequence and calculates the vector sequence of the hidden layer h=($h_1$,⋯,$h_T$) and the output vector y = ($y_1$,⋯,$y_T$). The iteration of t=1 to T is:

$$h_t = H(W_{xh}x_t + W_{hh}h_{t-1} + b_h), \quad (35)$$

$$y_t = W_{hy}h_t + b_y, \quad (36)$$

where $W$, $b$, and $H$ represent the weight matrix, the deviation vector and the hidden layer function respectively. The hidden layer function of LSTM is calculated by the following equations:

$$i_t = \delta(W_{xi}x_t + W_{hi}x_{t-1} + W_{ci}x_{t-1} + b_h), \quad (37)$$

$$f_t = \delta(W_{xf}x_t + W_{hf}x_{t-1} + W_{cf}x_{t-1} + b_f), \quad (38)$$

$$c_t = f_t c_{t-1} + i_t \tanh(W_{xc}x_t + W_{hc}h_{t-1} + b_c), \quad (39)$$

$$o_t = \delta(W_{xo}x_t + W_{ho}x_{t-1} + W_{co}x_{t-1} + b_o), \quad (40)$$

$$h_t = o_t \tanh(c_t), \quad (41)$$

$i_t$, $f_t$, $o_t$, $c_t$ and $h_t$represent input gate, forget gate, output gate and long unit activation vectors, and short unit activation vectors, respectively. The improved feature extraction and classification method of the min-VGGnet and LSTM hybrid structures named min-VGG-LSTMnet in this paper. As shown in **Figure 6**:

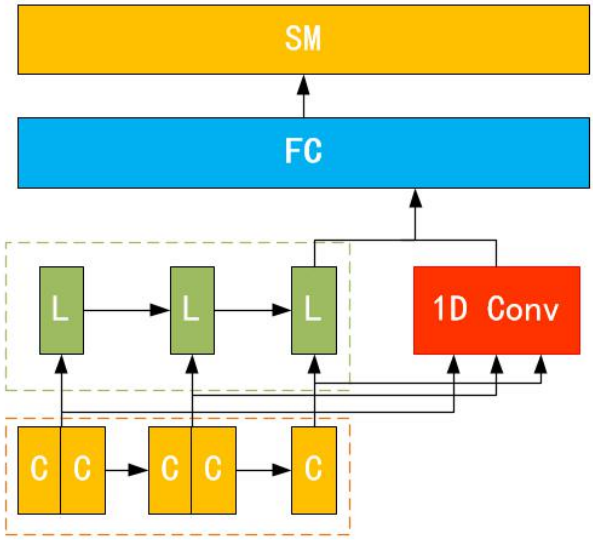


**Figure 6.** CNN combined with LSTM hybrid network min-VGG-LSTMnet

The hybrid structure still uses the structure of min-VGGnet, but the convolutional layer in the min-VGGnet is included in the three-layer stack structure. The first layer is a two-layer stack structure of 32 convolution kernel sizes [3×3], denoted as C1. The second layer is a two-layer stack structure of 64 convolution kernel sizes [3×3], denoted as C2. The third layer is a single-layer structure of 128 convolution kernel sizes [3×3], denoted as C3. LSTM needs to capture different time-mode features across multiple frames, corresponding to three structures of time unit. The inputs at $t$-1, t, and $t$+1 are from the features of shallow, deeper, and deep layers of the min-VGGnet structure, respectively. Each layer in min-VGGnet passes through the largest pooling layer and is transformed into a vector through the reshape layer as the structure of input of the LSTM network, which realizes serial fusion. Finally, LSTM layer and the layer formed by 128 cells get the best results. In the model, a 1-D Conv layer should be added after the output of min-VGGnet. Then

parallelizing the output of the 1-D Conv layer with the output of LSTM. Changing the dimension of the input data through the reshape layer, without the dimension of the sample number (batch size). After fusion, putting them together into the fully connected layer. The fully connection layer still adopts a two-layer stack structure. The first layer is a tile structure, which becomes a 1×512 vector through a fully connected layer. According to the number of num_classes required by the sample label, determining the final number of classifications. Finally entering the Softmax layer and completing the final classification.

## 3. Results

To make the comparison of classification results more intuitive, this paper summarized the values of best cross-entropy loss and classification accuracy of all the above classification models. All experiments are performed on the same computer, which operating system is Win10, 64 bit, CPU is 1.80GHz, RAM is 8GB. The comparison of the results of the network model of the four- classification task of motor imagery is shown in **Table 5:**

**Table 5.** Performance comparison of different classification models under four classification tasks

| Network model | Training Loss | Val Loss | Training ACC | Val ACC | Time(h) |
|---|---|---|---|---|---|
| EEGnet | 0.5979 | 0.5965 | 0.6317 | 0.6776 | 2.32 |
| CNN-2net | 0.4018 | 0.5307 | 0.8291 | 0.7754 | 3.54 |
| min-VGGnet | 0.4177 | 0.5262 | 0.9908 | 0.8312 | 3.91 |
| min-VGG-LSTMnet | 0.3458 | 0.4974 | 1.0 | 0.9130 | 4.16 |

As can be seen from **Table 5**, the min-VGG-LSTMnet has better performance in the four-classification task, and the classification accuracy of the training set and the verification set is higher. The results show that the accuracy of the proposed method is at least 8.18% higher than that of the traditional deep learning method. When the epoch takes 1000 times, the hybrid network model takes more time, but the difference is not big in general.

In order to verify the effectiveness of the proposed method, this paper compared the previous method with the method proposed in this paper. The results are shown in **Table 6**:

**Table 6.** Comparison of different classification models under four classification tasks

| Classification Method | Data Set | Maximum Accuracy | Average Accuracy |
|---|---|---|---|
| References Fisher+SVM | BCI Competition III | 85.7% | 80.95% |
| References CSP+SVM | Self-collected four-class task data set | 86.3% | 80.66% |
| proposed method min-VGG-LSTMnet | collected four-class task data set | 88.56% | 81.52% |

It can be seen from the above table that the proposed method min-VGG-LSTMnet has a classification accuracy of 88.56% on the test set, which is at least 2.26% higher than the method in the literature. Two-dimensional time-frequency analysis combined with Fisher analysis is used to extract the features of the

imaginary movements of the left hand, right hand, foot and tongue in the BCI competition data set, and SVM is used to recognize, which achieves an accuracy rate of 85.7%. A self-collected dataset, which contains the EEG features of the four types of motor imagery tasks of the upper, lower, left and right spheres of the ball. Feature extraction and classification of four types of motor imagery EEG signals use CSP combined with SVM. The highest accuracy is 86.3%. The dataset of this paper is a self-collected data set of four types of motions: left-handed lifting, right-hand lifting, left-foot lifting and right-foot lifting. The min-VGG-LSTMnet network model is used to identify and predict four types of tasks. The highest accuracy is 88.56%.

In summary, the min-VGG-LSTMnet hybrid deep learning network designed in this paper not only realized the four-classification task of motor imagery EEG signal, but also improved the classification accuracy by at least 8.18%, and reduced the loss value by at least 0.0288 compared with the mainstream deep-learning method.

## 4. Discussion

This paper provides an important technical support for the realization of the brain-computer interface through the feature extraction and classification of motor imagery EEG signals, which has great research significance and application value in the fields of medical rehabilitation and new generation human-computer interaction.

The main recommendations and prospects of this paper are as follows:

(1) In the future, the complexity of multi-classification tasks can be further studied in order to obtain more representative motor imagery EEG signal, and the classification algorithm should be enhanced to discriminate between different classification tasks.

(2) The EEG signal is very weak, and it is necessary to continue to explore its intrinsic properties and find a better method of feature extraction. It is not only limited to the feature extraction of motor imagery EEG signal, but also suitable for feature extraction of other types of EEG signal.

(3) At present, the deep learning network is not very good at classifying small data sets. Compared with the simpler model, the advantage of deep learning is that there are enough data to adjust a large number of parameters. However, when the data set is small, there will be overfitting problem. It is hoped that in the future, high-precision classification of small data sets can be and realized.

(4) The classification task could not be performed in real time in this paper. Because the collected data need preprocessing and feature extraction offline. Due to the high complexity of the algorithm, it is impossible to achieve real-time at present, which is one of the important directions for future research. Real-time pre-processing software integration is required to achieve real-time classification.

## References

1. Zied Tayeb; Juri Fedjaev; Nejla Ghaboosi; *et al.* Validating deep neural networks for online decoding of motor imagery movements from EEG signals. Sensors 2019; 19(1): 210.
2. JJ Shih; Krusienski; Dean J; *et al*. Brain-computer interfaces in medicine. Mayo Clinic Proceedings 2012; 87: 268–279.
3. Pfurtscheller G. Functional brain imaging based on ERD/ERS. Vision Research 2001; 41(10-11): 1257-1260.
4. Obermaier B; Neuper C. Information transfer rate in a five-classes brain-computer interface. IEEE Trans Neural Syst Rehabil Eng 2001; 9(3): 283-288.
5. Wan B; Liu Y. Multi-pattern motor imagery recognition based on EEG features. Journal of Tianjin University 2010; 43(10): 895-900.
6. Xu X; Wang N. Feature extraction and classification of EEG signals in four kinds of motion imagination. Journal of Nanjing University of Posts and Telecommunications (Social Science) 2017; 37(06): 18-22.
7. Blankertz B; Müller KR. The BCI competition 2003. IEEE Transactions on Biomedical Engineering 2004; 51(6): 1044-1051.
8. Michel Cotsaftis. The autonomous intelligence challenge. Journal of Autonomous Intelligence 2018; 1(1): 1-1.
9. Manu Mitra. Neural processor in artificial

intelligence advancement. Journal of Autonomous Intelligence 2018; 1(1): 2-14.

10. Delorme A; Makeig S. EEGLAB: an open source toolbox for analysis of single-trial EEG dynamics including independent component analysis. J Neurosci Methods 2004; 134(1): 9-21.
11. Ahmad A; Xavier J. 3D to 2D bijection for spherical objects under equidistant fisheye projection. Computer Vision and Image Understanding 2014; 125: 172-183.
12. Jiaqing Chen; Xiaohui Mu; Yinglei Song; *et al.* Flame recognition in video images with color and dynamic features of flames. Journal of Autonomous Intelligence 2019; 1(1): 11-29.
13. Weide Li, Juan Zhang. An innovated integrated model using singular spectrum analysis and support vector regression optimized by intelligent algorithm for rainfall forecasting. Journal of Autonomous Intelligence 2019; 1(1): 30-45.
14. Krizhevsky A; Sutskever I. Imagenet classification with deep convolutional neural networks. Advances in Neural Information Processing Systems 2012; 1097-1105.
15. Cho K; Van Merriënboer B. Learning phrase representations using RNN encoder-decoder for statistical machine translation. arXiv preprint arXiv 2014; 1406.
16. Graves A; Fernández S. Bidirectional LSTM networks for improved phoneme classification and recognition. International Conference on Artificial Neural Networks 2005; 799-804.
17. Lawhern VJ; Solon AJ. EEGNet: A compact convolutional neural network for EEG-based brain – computer interfaces. Journal of Neural Engineering 2018; 15(5): 056013.
18. Ordóñez F; Roggen D. Deep convolutional and LSTM recurrent neural networks for multimodal wearable activity recognition. Sensors 2016; 16: 115.
19. Yao S; Hu S; Zhao Y; *et al*. Deepsense: A unified deep learning framework for time-series mobile sensing data processing. In Proceedings of the 26th International Conference on World Wide Web, International World Wide Web Conferences Steering Committee 2017; 351 – 360.
20. Okita T; Inoue S. Activity recognition: Translation across sensor modalities using deep learning. In Proceedings of the 2018 ACM International Joint Conference and 2018 International Symposium on Pervasive and Ubiquitous Computing and Wearable Computers 2018; 1462 – 1471.
21. Ha K-W; Jeong J-W. Motor imagery EEG classification using capsule networks. Sensors 2019; 19: 2854.
22. Nicolas-Alonso; Luis F; Gomez-Gil J. Brain computer interfaces, a review. Sensors 2012; 12: 1211 – 1279.
23. Kim D-W; Lee J-C; Park Y-M; *et al*. Auditory brain-computer interfaces (BCIs) and their practical applications. Biomedical Engineering Letters 2012; 2: 13 – 17.
24. Michel Cotsaftis. Autonomous intelligence: An advance level in modern technology. Journal of Autonomous Intelligence 2018; 1(1): 44-44.
25. Yongzhong Lu; Min Zhou; Shiping Chen; *et al.* A perspective of conventional and bioinspired optimization techniques in maximum likelihood parameter estimation. Journal of Autonomous Intelligence 2018; 2(1): 1-12.
26. Lotte F; Bougrain L; Cichocki A; *et al.* A review of classification algorithms for EEG-based brain-com -puter interfaces: A 10-year update. Journal of Neural Engineering 2018; 15: 031005.
27. Park J; Min K; Kim H; *et al*. Road surface classification using a deep ensemble network with sensor feature selection. Sensors 2018; 18: 4342.
28. Zhao R; Yan R; Wang J; *et al*. A hybrid CNN – LSTM algorithm for online defect recognition of CO2 welding. Sensors (Basel) 2017; 17(2).
29. Gao M; Shi G; Li S. Online prediction of ship behavior with automatic identification system sensor data using bidirectional long short-term memory recurrent neural network. Sensors 2018; 18: 4211.
30. Liu C; Wang Y; Kumar K; *et al*. Investigations on speaker adaptation of LSTM RNN models for speech recognition. In Proceedings of the International Conference on Acoustics, Speech and Signal Processing 2016; 5020 – 5024.