Applied Soft Computing 122 (2022) 108811

Contents lists available at ScienceDirect

Applied Soft Computing

journal homepage: www.elsevier.com/locate/asoc

Data augmentation for Convolutional LSTM based brain computer interface system

Kahoko Takahashi [a], Zhe Sun [b,*], Jordi Solé-Casals [c,d,e,*], Andrzej Cichocki [f], Anh Huy Phan [f], Qibin Zhao [g], Hui-Hai Zhao [h], Shangkun Deng [i], Ruggero Micheletto [a,*]

[a] Cognitive Information Science Lab., Yokohama City University, 22-2 Seto, Kanazawa Ward, Yokohama, Japan
[b] Computational Engineering Applications Unit, Head Office for Information Systems and Cybersecurity, RIKEN, 2-1 Hirosawa, Wako-shi, Saitama, Japan
[c] Data and Signal Processing Research Group, Department of Engineering, University of Vic–Central University of Catalonia, Vic 08500, Catalonia, Spain
[d] Department of Psychiatry, University of Cambridge, Cambridge CB2 3EB, United Kingdom
[e] College of Artificial Intelligence, Nankai University, Tianjin 300071, China
[f] Laboratory Tensor networks and deep learning for applications in data mining, Skolkovo Institute of Science and Technology, Moscow, Russia
[g] Tensor Learning Team, RIKEN Center for Advanced Intelligence Project, Tokyo, Japan
[h] Alibaba Quantum Laboratory, Alibaba Group, Beijing 100102, China
[i] College of Economics and Management, China Three Gorges University, Yichang, China

ARTICLE INFO

Article history:
Received 20 July 2020
Received in revised form 15 March 2022
Accepted 27 March 2022
Available online 8 April 2022

Keywords:
Convolutional LSTM neural network
Data augmentation
Brain computer interface
Empirical Mode Decomposition

ABSTRACT

Electroencephalogram (EEG) is a *noninvasive* method to detect spatio-temporal electric signals in human brain, actively used in the recent development of Brain Computer Interfaces (BCI). EEG's patterns are affected by the task, but also other variable factors influence the subject focus on the task and result in noisy EEG signals difficult to decipher. To surpass these limitations methods based on artificial neural networks (ANNs) are used, they are inherently robust to noise and do not require models. However, they learn from examples and require lots of training data-sets. This will increase costs, need research time and subjects effort. To reduce the number of experiments necessary for network training, we devised a methodology to provide artificial data from a limited number of training data-sets. This was done by applying Empirical Mode Decomposition (EMD) on the EEG frames and intermixing their Intrinsic Mode Function (IMFs). We experimented on motor imagery (MI) tests where participants were asked to imagine movement of the left (or right) arm while under EEG recording. The EEG data were firstly transformed using the Morlet wavelet and then fed to an originally designed Convolutional Neural Network (CNN) with long short term memory blocks (LSTM-RNN). The introduction of artificial frames improved performances when compared with standard algorithms. The artificial frames become advantageous even when the number of available real frames was only of 7 or 8. In a test with two subjects (200 recordings for each subject), we reached an accuracy better than 88% for both subjects. Improvements due to the artificial data were especially noticeable for the under-performing subject, whose EEG had lower accuracy. Imagination recognition accuracy was about 89% with 360 training frames, in which 300 were artificially created starting from 60 real ones. We believe this methodology of synthesizing artificial data may contribute to the development of novel and more efficient ways to train neural networks for brain computer interfaces.

1. Introduction

Brain computer interface (BCI) devices provide means to send commands of various complexity to computers without using any verbal or tactile actions. These systems, like many others (e.g., fNIRS, ECoG, etc.) are able to read brain signals and recognize specific commands. In simple words, a BCI device connects between the user and the actual machinery with using brain signals. A BCI apparatus helps people with handicaps, it may also contribute to the realization of devices with applications in a multitude of fields [1–3].

However, brain signals are generated by the electric activity of billions of neurons firing in a random, uncorrelated manner. The signals are tiny in intensity (few microvolts) and of low frequency, thus they are difficult to analyze and classify. Moreover,

* Corresponding authors.
E-mail addresses: zhe.sun.vk@riken.jp (Z. Sun), jordi.sole@uvic.cat (J. Solé-Casals), ruggero@yokohama-cu.ac.jp (R. Micheletto).

https://doi.org/10.1016/j.asoc.2022.108811
1568-4946/

to assign meaning to specific EEG events, we have to understand subjects differences and variability within same subjects [4].

In this sense, it is very important to realize a reliable and robust algorithm able to recognize and classify properly EEG patterns. Most used in the past are linear classifiers, like linear discriminant analysis (LDA), regularized LDAs and support vector machines (SVMs) that generally outperforms other classifiers [5–7]. Recently, artificial neural networks (ANNs) emerged as useful classifier, because they are characterized by robustness to noise. When properly trained, they can overcome subject variability and can adapt to extremely non-linear behavior [8]. However, to be used for brain–computer interfaces, ANN should be trained labeling each EEG pattern to the correct mental state in order to obtain the optimal ANN response [9,10]. The necessity of a great amount of calibration data makes the training of ANNs inconvenient, especially when dealing with subjects of age or with disabilities. Generating artificial data by randomly sampling the available dataset is a possible strategy [11].

In this study we propose a new method that uses a Convolutional LSTM network to analyze EEG patterns with a reduced number of training sets. The subjects are asked to imagine movement of the left or right hand (motor imagery tests) while EEG are being recorded for two seconds. Each of these training set is labeled accordingly and used for the network learning phase. We developed an original method to synthesize artificial training data starting from a limited number of real tests on the subjects and then using empirical mode decomposition [12,13] (EMD), mixing in an opportune way the data intrinsic mode functions (IMF) as described in details in Section 2.3.

The imagination of a motor action can induce recognizable brain signals. If we analyze the spectral content, we notice the emergence of recognizable frequencies in correspondence to particular mental states [14]. For example, θ rhythm (4–7 Hz) is found during sleep and low brain activities and β rhythm (13–35 Hz) in alertness and motor activity. Brain waves show not only rhythmical patterns, but have also more structured behaviors. For instance, in preparation of a motor action, frequencies slowly reduce in amplitude. These frequencies show dynamical properties. The recall of cognitive event, emotional events or motor actions can accelerate/decelerate or suppress/enhance α and β rhythms [15] in a complex way, the phenomenon is called event related de-synchronization. Experiments confirm that even the simple perception of a motor action performed by a third person, can result as suppression of β waves [16] in the EEG signal. Electroencephalogram events are mapped to specific areas of the brain. For example, different body movements result to signals in localized in regions of various shape and extensions on the skull. The area related to hand movements is especially large and easy to map with EEG. For this reason, many imagery EEG experiments are done asking subjects to imagine motor actions involving their hands or arms. One of the main difficulties in detecting correctly EEG signals is noise filtering. An approach is to firstly filter each EEG channel with a band-pass, then a spatial filter is applied, the cluster obtained are projected on positions on the skull to be identified. This method, or those derived from it, are called common spatial pattern (CSP) technique [9]. They are widely used to reduce noise, but noticeably the band-pass filtering used are obtained by trial and error in a case-by-case fashion.

2. Methods

As mentioned above, in this study we want to improve previous methodologies using highly adaptable, robust to noise and trainable convolutional neural networks (CNN) [17], structured with convolution layers, pooling layers and rectification layers. The input of the net at a certain instant is the EEG voltage,

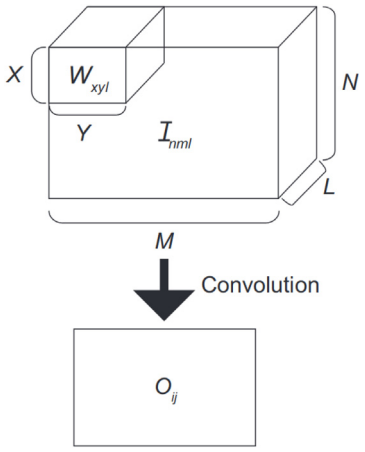

Fig. 1. The sketch of the structure of the CNN used. I_{nml} represents the *image* (or *frame*) extracted from the EEG potential data, n and m are coordinates on the scalp and l is the frame number. The array W is the convolution filter, it is smaller and have three components x, y and l. The first two are spatial components running over the wider range n and m. The latter l represents the time (frame). This operation extracts the *features* associated to the filter W_{xyl}.

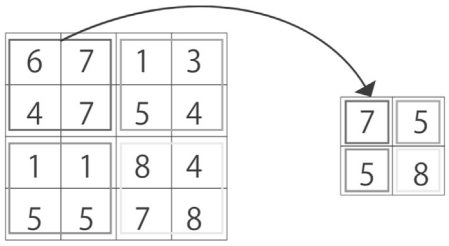

Fig. 2. A batch of time dependent EEG data are filtered by a convolution layer (see Fig. 1), the resulting frame is reduced in size by pooling the maximum value for each four adjacent data as in this simple sketch.

structured in an $N \times M$ grid where N and M represent, loosely, the coordinates along the scalp. In this way, we can interpret the EEG data at a certain instant of time as an *image*, or *frame*, of size $N \times M$. Since our data are time series coming from multiple EEG channels, we have a multidimensional input that is of size $N \times M \times L$, where L is the frame number (trials), in a certain EEG data batch.

The convolution process filters each of these L images with a filter W that runs over the N and M coordinates. The size of the filter is X by Y by the frame number L. The operation results in a new array O_{ij} given by the expression:

$$O_{ij} = \Sigma_x \Sigma_y \Sigma_l W_{xyl} I_{(i+x-1)(j+y-1)l} + b \tag{1}$$

The array O_{ij} represents the EEG signal map, convoluted with the filter W and integrated along all the frames available in the particular data batch. The data size and CNN structure are introduced in Table 1. This can be called the *feature map* due to the filter W. See Fig. 1 for a simple schematic representation of the convolution process (the filter W is optimized in order to obtain the best classification performances, in other words the filter is actually a variable).

The O_{ij} frame is as large as the original data, a *pooling* layer in the network is introduced to reduce the size of it. We used the max pooling procedure, each four adjacent data in the frame is substituted by one single data corresponding to the maximum of the four (see Fig. 2).

For better optimization with only positive numbers, a rectified linear unit (ReLU) layer is added. This operator simply sets any

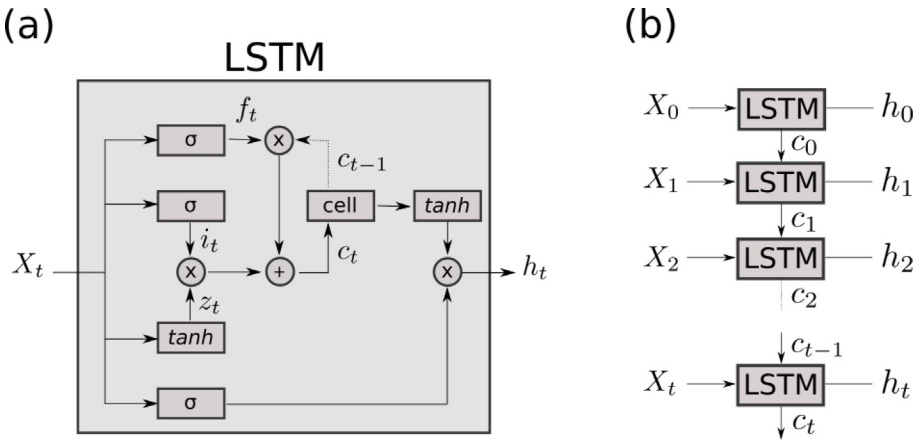

Fig. 3. In panel (a) the scheme of the LSTM block. A memory cell passes its value C_t to the next time step depending on calculations involving the values of the *gates* f, i and z. In (b) the time explicit structure of the network.

negative values to zero and leave untouched all the other data. With this configuration the network can learn to classify a special set of training data. These are data that are EEG recordings from subjects, with the correct left or right imagery label manually inserted by a human.

2.1. The long short-term memory cell

The so called Long short-term memory (LSTM) [18] is a particular recurrent neural network (RNN) that is used to analyze time series data. An RNN can memorize previous data in the time-series and use them for the optimization process. Compared to a simple RNN, a LSTM network has longer memorization time and this results in successful applications to natural language processing and speech recognition. It is widely used for time series and signals processing. In this study we use a LSTM network with the forget gate introduced by Gers [19].

Calling the input signal at time t as X_t, the corresponding LSTM block output is h_t. The latter depends not only on X_t but also on the previous output h_{t-1}. This is valid for all point in time X_t, X_{t-1}, X_{t-2} etc. The LSTM block involves memory *gates* that have a function to selectively pass information to neurons. During the learning phase, the gates actually extend or shorten the memory length to maximize the fit with the desired output.

The scheme of the LSTM block we used is as in Fig. 3(a).

The first output gate acts as a standard RNN network, so its output is:

$$o_t = \sigma(W_o x_t + R_o h_{t-1} + b_o) \tag{2}$$

where σ is a sigmoid function, W_o and R_o are weight arrays and b_o is a bias constant. The *forget* gate f_t and the input gates i_t and z_t have similar structure:

$$f_t = \sigma(W_f x_t + R_f h_{t-1} + b_f) \tag{3}$$

$$i_t = \sigma(W_i x_t + R_i h_{t-1} + b_i) \tag{4}$$

and

$$z_t = tanh(W_i x_t + R_i h_{t-1} + b_i) \tag{5}$$

These three elements are connected to the *memory* cell c_t through element-wise addition or multiplication modules as in Fig. 3. That is:

$$c_t = i_t \times z_t + f_t \times c_{t-1} \tag{6}$$

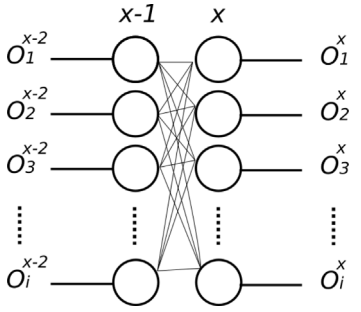

Fig. 4. The core structure of an artificial neural network. Layer $(x-1)$ and layer x are *fully connected*. The output of one layer becomes the input of the next. The output $O^{(x-2)}$ of layer $(x-2)$ is the input of neurons in layer $(x-1)$ and so on. Eq. (8) transfers values from one layer to the next through the weights W_{ij}.

The function of the memory cell is to keep the c_t value or not. Depending on this decision c_{t-1} calculation is executed and finally the output is:

$$z_t = o_t \times tanh(c_t) \tag{7}$$

As in any RNN the LSTM block acts on each X_t, as schematically shown in Fig. 3(b).

2.2. Optimization

We use here a fully connected artificial neural network with features extraction filters and memory blocks. The specific type of memory we use is called long short-term memory (LSTM). Hereafter we describe briefly some details about the structure of our network.

The fully connected network is structured in layers. Within each layer, neurons are not connected to each other. However each neuron of the layer is feeding and receiving information to all the neurons of the adjacent layers (Fig. 4). The output of each layer is transferred to the next layer through the weights Wij. In this way the output for the neuron i of layer x is:

$$O_i^x = \mathscr{L}(I_i^x) \tag{8}$$

where I_i^x is the neuron activation $\Sigma_i^N W_{ij}^{x-1} O_i^{x-1} - \theta_i^{x-1}$. Here θ is a threshold and N the number of neurons in the layer. The logistic function $\mathscr{L}$ is $\frac{1}{1+e^{-x}}$.

The optimization of the weights W_{ij} is obtained by the back-propagation method [20]. The *cost function* that estimates the error between a target Y_i and the actual output O_i is the *cross*

entropy derived from the logistic regression theory [21]. For each layer x we define the cost as:

$$E = -\Sigma_i[Y_i^x \ln O_i^x + (1 - Y_i^x)\ln(1 - O_i^x)] \tag{9}$$

The new weights W_{ij} are optimized by adding a small term proportional to the gradient:

$$\frac{dE}{dW_{ij}^{x-1}} = \frac{dE}{dO_i^x}\frac{dO_i^x}{dI_i^x}\frac{dI_i^x}{dW_{ij}^{x-1}} \tag{10}$$

From Eq. (9) we can calculate these partial derivatives :

$$\begin{aligned}\frac{dE}{dO_i^x} &= -\frac{Y_i^x}{O_i^x} + \frac{1 - Y_i^x}{1 - O_i^x} \\ \frac{dO_i^x}{I_i^x} &= O_i^x(1 - O_i^x) \\ \frac{dI_i^x}{W_{ij}^x} &= O_i^{x-1}\end{aligned} \tag{11}$$

multiplying these together few terms go away and finally we have:

$$\frac{dE}{dW_{ij}^{x-1}} = O_i^{x-1}(O_i^x - Y_i^x) \tag{12}$$

We multiply this term by the logistic derivative (second in Eq. (11)), so ΔW_{ij}^{x-1} will be bigger around the center and minimal on the extremes of it:

$$\Delta W_{ij}^{x-1} = -\varepsilon O_i^{x-1}(O_i^x - Y_i^x)O_i^x(1 - O_i^x) \tag{13}$$

where ε is a variable *learning rate* given by:

$$\varepsilon = \varepsilon_0 \frac{1}{1 + \beta N_e} \tag{14}$$

Here ε_0 is the initial learning rate and β the *learning attenuation*. N_e are the *epochs*, the number of times the learning algorithm have to pass through the entire training dataset to update its weights (an epoch can be divided in batches of data extracted at random from the set).

To speed up the convergence we introduce a momentum [22, 23] term α. Overall, considering that

$$\Delta W_{ij}^{x-1} = W_{ij}^{x-1}(t + 1) - W_{ij}^{x-1}(t)$$

at time step t the new $W_{ij}^x(t + 1)$ will be:

$$W_{ij}^{x-1}(t+1) = W_{ij}^{x-1}(t) - \varepsilon O_i^{x-1}(O_i^x - Y_i^x)O_i^x(1-O_i^x) + \alpha W_{ij}^{x-1}(t) \tag{15}$$

2.3. Empirical mode decomposition

To improve network training efficiency and reduce the number of tedious and time-consuming tests with humans, we created artificial EEG frames from a number of real motor imaginary experiments. Empirical mode decomposition has been used for EEG signal processing [13]. We decomposed the EEG frames using Empirical Mode Decomposition and created new ones intermixing their intrinsic mode functions (IMF) [24]. A number of labeled homogeneous EEG data are decomposed and randomly mixed, creating new artificial data, each different from one another, but still compatible to the same label because obtained from different sets of the same motor-imagery contents.

The time series $EEG(t)$ is decomposed in the sum of n intrinsic mode functions and a trend component $r_n(t)$ as

$$EEG(t) = \Sigma_{k=1}^{n} IMF_k(t) + r_n(t) \tag{16}$$

The main characteristics of the $IMF_k(t)$ function are that (1) the difference between the number of peaks and the number of zero-crossing points should be one or zero, (2) the envelope of the peaks and minimum points on average should be zero.

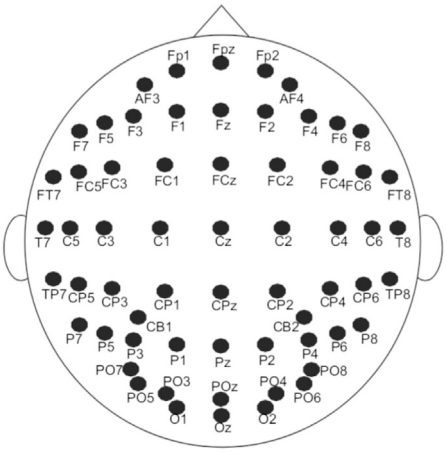

Fig. 5. The positions and channel naming of the 62 electrodes used in the experiment.

2.4. Synthesis of artificial frames

We use the data obtained by Phan [25] on two subjects (S1 and S2) using an EEG of 62 electrodes placed like in Fig. 5. The measure was made for two seconds, 200 times per subject. The EEG sampling rate was set to 500 Hz, so each dataset was an array of size 1000 × 62 (500 Hz for 2 s, 62 channels).

Of the 200 total datasets, we used 60 of them as training, and the remaining 140 for validation. Since we intend to generate artificial training data with our original method, we call these 60 datasets the *real* training set and the other that we will create *artificial* training sets.

The artificial frames are constructed using an iterative procedure. As introduced above, firstly we decompose the original signal $EEG(t)$ in its *empirical modes* using the intrinsic mode function $IMF(t)$ and the *trend component* $r_n(t)$ [12]. E is an operator that does the spline interpolation of all the maxima and all the minima of the signal and the apex T represents the transpose of a vector, the decomposition is done accordingly to the following steps:

1. A function $s(t) = r_{i-1}(t)$ is defined, i begins with $i = 1$ and $r_0(t) = EEG(t)$.
2. Calculate the upper and lower envelopes $(E^u[s(t)], E^l[s(t)])$
3. We define the average of the upper and lower envelope as $m(t) = (E^u[s(t)] - E^l[s(t)])/2$.
4. The intrinsic mode function (IMF) candidate $h(t)$ is defined as the difference between the average $m(t)$ and the signal: $h(t) = s(t) - m(t)$.
5. $s(t)$ is reassigned as $s(t) = h(t)$ and the procedure returns to point (2) until the ratio $\frac{s(t)^T s(t)}{m(t)^T m(t)}$ reaches 40 dB.
6. Then the first intrinsic mode function $IMF_i(t) = h(t)$ is found.
7. The residual is defined as the difference between the original signal and the current IMF $r_i(t) = r_{i-1}(t) - IMF_i(t)$
8. The procedure is repeated from (2) until the ratio $\frac{r_{i-1}(t)^T r_{i-1}(t)}{r_i(t)^T r_i(t)}$ reaches 50 dB.

In this way the EEG(t) pattern is decomposed in n intrinsic mode functions plus a residual, as in Eq. (16). To create new artificial EEG frames, we choose random IMFs and use Eq. (16) to build new artificial frames. We have, of course, to mix frames belonging to the same group of imagery data (left hand or right one). In Fig. 6 we show a simplified scheme of the procedure.

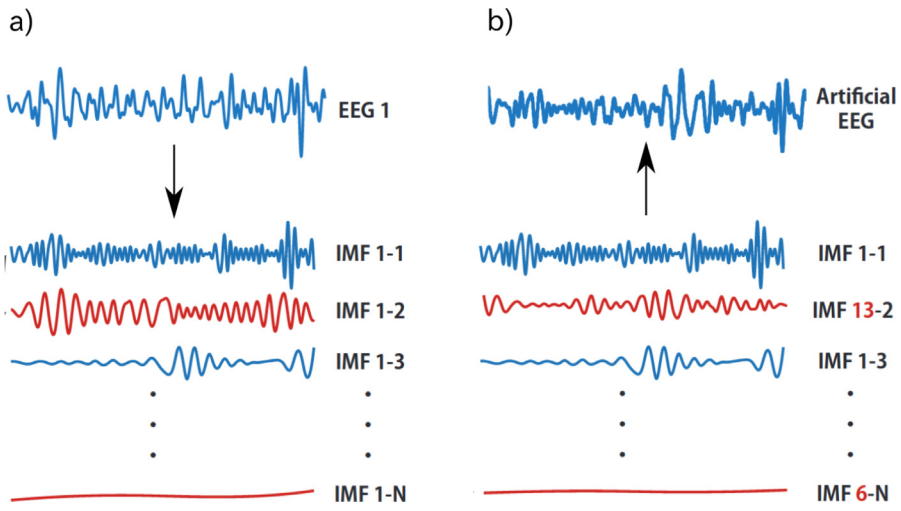

Fig. 6. A sketch on how the IMF decomposition works. In (a) is shown the EEG time series decomposed to N intrinsic mode functions. In (b) an artificial frame is created composing various intrinsic mode functions chosen from random EEGs of the same imagery dataset (in the figure IMF $i-j$ means the jth intrinsic mode function from the ith EEG).

2.5. Time resolved frequency analysis

EEG signals can be analyzed by their frequency and their position in space. Since a simple Fourier analysis would critically depend on the time-window length, we used a Morlet based wavelet analysis. The so-called *mother wavelet* is a function that is variable in shape, enlarging at low frequencies and getting sharper at higher. In this way the analysis is balanced and more suitable for wide frequency range signals as our imagery EEG time series. The general wavelet form we used is the following:

$$W_\psi(a,b) = \frac{1}{\sqrt{a}} \int_{-\infty}^{\infty} s(\tau)\psi^*\left(\frac{\tau - b}{a}\right)d\tau \tag{17}$$

where b is the time reference and a is the scaling parameter. As the *mother* wavelet function we use the Morlet function below:

$$\psi_{Morlet}(t) = \frac{1}{\sqrt{\pi\beta}} \cdot e^{i2\pi f_c t} \cdot e^{-\frac{t^2}{\beta}} \tag{18}$$

here f_c is the center frequency and β is the band width parameter. The smooth shape of this function is typical of biological and physiological reactions, often used in brain research (see a plot in Fig. 7).

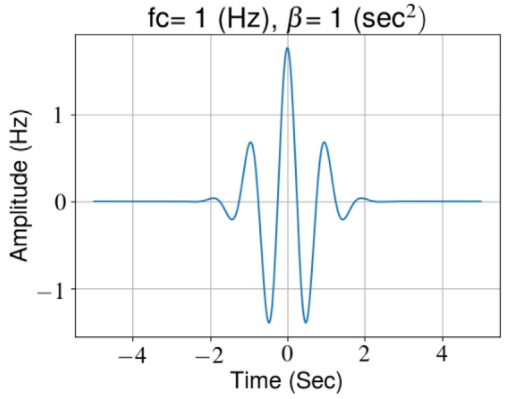

Fig. 7. The shape of the mother wavelet *Morlet*, Eq. (18), with $f_c = 1$ Hz and $\beta = 1\ s^2$.

2.6. The convolutional neural network

We used two different network configurations in this study [17,26]. The first is a CNN network that categorizes the result of EEG wavelet analysis, the second it is a Convolutional LSTM network classifying the spectrum of the same EEG analysis [27] (see Fig. 8).

The CNN used is of the same structure as the one used by Kwon for the recognition and classification of emotions by electroencephalographic analysis [28]. Before the signal is fed to the network it is converted to its time–frequency distribution (TFD) described above. The conversion is done through Fourier or wavelet transformation where an 8–30 Hz spectrum is shown for an interval of 2 s. Given our setup constraint of 500 Hz sampling rate (corresponding to 2 ms integration time), we have chosen 20 samples to calculate each spectrum. Thus, each wavelet time-window has a duration of 40 ms and there are a total of 50 of them (Fig. 9). Each time series is a 1000 points vector (2 s of EEG recording at 500 Hz sampling rate). There are 62 EEG channels so the network input is a bi-dimensional array of size 62×1000. After the wavelet/Fourier transformation it becomes a $23 \times 50 \times 62$ three dimensional array that is fed to the CNN network. The structure of the CNN network is given in Table 1.

In our network the kernel filter (that is the filter operating on the input bidimensional array of data) is of size 3×5 with 1×1 stride size (the strides are the number of pixels by which the kernel filter moves over the data, 1×1 means that the filter moves one step vertically and one horizontally) and has 100 filters. The pooling kernel is instead 3×3 with 2×2 stride size. The network used here has fewer layers than that of Kwon [28], and a reduced number of layers improves computational efficiency and this has no detrimental effects on our study since we have a lower number of training data. After the second pooling, the output is transformed to a one-dimensional vector of length 3600. This is fed to a fully connected network with two output. The two labels values are the vector [1, 0] or [0, 1], corresponding to the two motor imagery thought of moving the right or left arm respectively. In the network training phase,

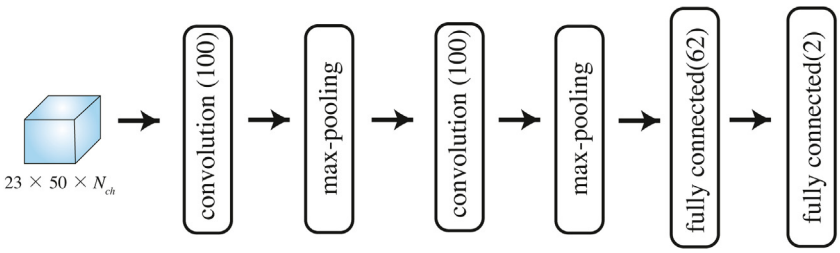

Fig. 8. A simplified sketch of the CNN used in this study. The input is a three dimensional array in which the first dimension is the wavelet transformation output is a vector of 23 values (8–30 Hz interval divided in 23 bins), the second is the number of time-windows (of 40 ms duration for a total of 2000 ms) and the third the EEG channels on the scalp ($N_{ch} = 62$ channels).

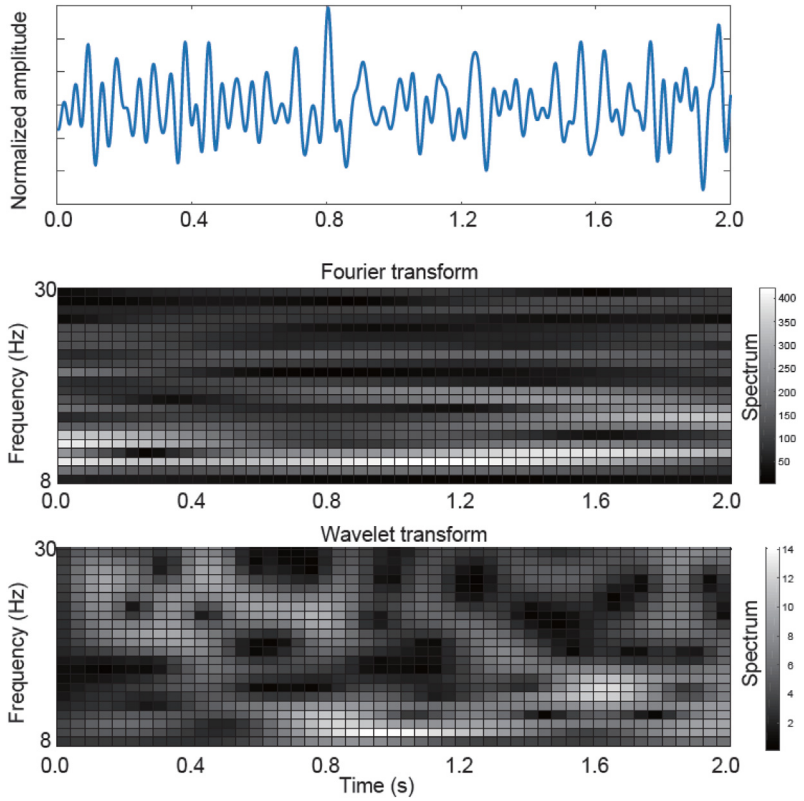

Fig. 9. The wavelet (top panel Fourier) transformation results in images of size 50 × 23. The normalized time-series is time-windowed in chunks of 40 ms duration, with 20 samples in each of them. The integration time is 2 ms (500 Hz sampling rate) and since the total recording time is of 2 s, there are a total of 50 time-windows, each corresponds to pixels on the horizontal axis. The transformation spectrum results in 23 points ranging from 8 to 30 Hz on the vertical axis.

Table 1
The CNN structure and data sizes as implemented with the KERAS library [29]. The bi-dimensional convolutional network output is filtered by a rectified linear unit (ReLU) and fed to a pooling layer. The procedure is repeated and the result is transformed in a flat vector with the Flatten operator. This vector finally is fed to a regular ANN network composed by two densely connected layers rectified by a ReLU unit. Weights are optimized using the Adam optimizer. The layer type is a mnemonic and in parenthesis is the actual library function used.

Layer (type)	Output shape
convolution_1 (Conv2D)	(23, 50, 100)
activation_1 (ReLU)	(23, 50, 100)
pooling_1 (MaxPooling2D)	(11, 24, 100)
convolution_2 (Conv2D)	(9, 20, 100)
activation_2 (ReLU)	(9, 20, 100)
pooling_2 (MaxPooling2D)	(4, 9, 100)
flatten_1 (Flatten)	(3600)
layer_1 (Dense)	(62)
activation_3 (ReLU)	(62)
layer_2 (Dense)	(2)

a function proportional to the output distance from these labels will be used to optimize all the CNN network variable parameters.

To train the CNN network we used 60 of the 200 EEG recording to create artificial training data. The remaining 140 measurements were used for the network testing and validation. In a first test, using 30 EEG measurements, we created two sets of 20 and 30 artificial training data for a total of 50 (30+20) or 60 (30+30) training sets. In another test, the set of 60 EEG recording was used to create sets of 120, 300 and 600 artificial data. The CNN in this case was trained with a total of 180, 360 and 660 sets. Once the training was complete, the CNN performance was tested and validated for subject 1 and 2 using the 140 untrained recording.

2.7. CNN and LSTM network

With the single CNN structure it is not possible to enhance the time dependence of the EEG signal, thus here we constructed an enhanced CNN-LSTM hybrid network as shown in Fig. 10. For the experiment we used the 45 EEG channels shown in Fig. 11. Like in

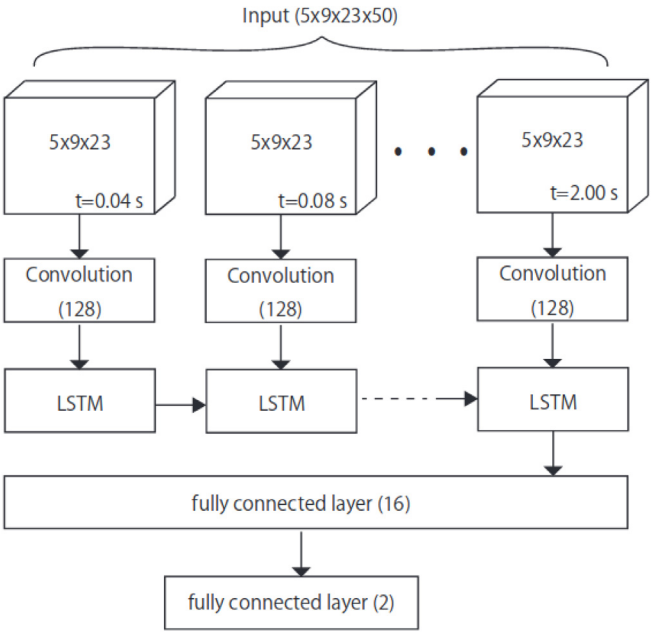

Fig. 10. A sketch of the Convolutional LSTM used in this study.

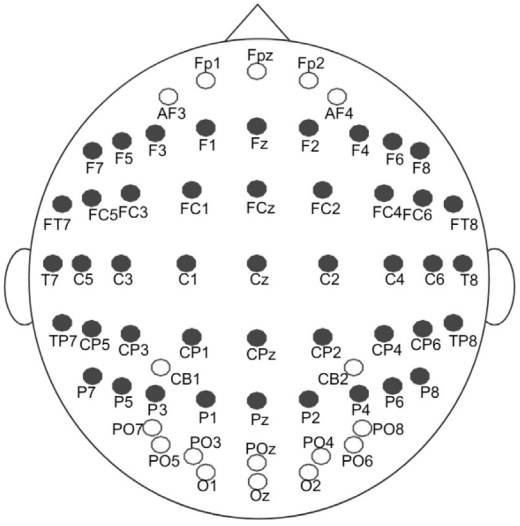

Fig. 11. The 45 channels used in the CNN-LSTM experiments are shown with the filled dots.

the previous CNN analysis, after the wavelet transformation, we have a tensor of dimension (23 × 50 × 45). The 45 channels are spatially organized in a (5 × 9) array (see Fig. 11). We have one of this array for each of the 23 resulting frequencies and each of the 50 time-windows. The data tensor is of size (9 × 5 × 23 × 50). In Fig. 12 is shown an example of one of the 5 × 9 EEG frames. In the CNN+LSTM configuration, the first 40 ms chunk of input data is given to the CNN and subsequently fed to the LSTM network. The output of it is given to a standard fully connected ANN. Like in the previous case, the network final layer is composed by two neurons, one corresponds to the "left arm" movement label, the other for "right arm" one. The optimization process proceeds like in any CNN, updating the network weights accordingly to an optimization algorithm. Once the network is optimized, we evaluate its performances with untrained input data and compare the network output with the subject actual imagery label (see Table 2).

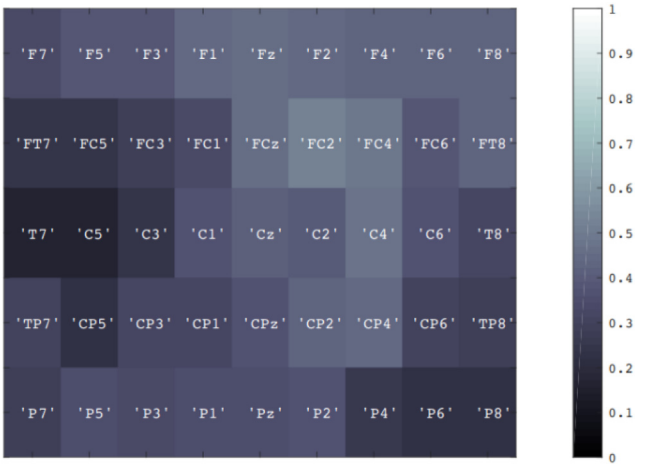

Fig. 12. An example of the normalized EEG input, organized in a (5 × 9) array. Channel labels correspond to Fig. 11.

Table 2
Parameters we are using in our CNN+LSTM neural network.

Layer (type)	Output shape
time_distributed_1 (Time Distributed)	(50, 7, 3, 128)
time_distributed_2 (Time Distributed)	(50, 2688)
lstm_1 (LSTM)	(256)
dense_1 (Dense)	(16)
activation_1 (Activation)	(16)
dense_2 (Dense)	(2)
activation_2 (Activation)	(2)

Table 3
The accuracy for the CNN with a short time-window Fourier transformation (STFT) and wavelet transformations with different f_b. In gray the best results, the three trials are performed with 60 training sets and 140 evaluation sets (subject 1).

	Accuracy for each trial			Average
	#1	#2	#3	
fb = 1	87.9%	88.6%	88.6%	88.4%
fb = 2	88.6%	89.3%	88.6%	88.8%
fb = 3	88.6%	86.4%	88.6%	87.9%
fb = 4	90.0%	88.6%	85.0%	87.9%
fb = 5	90.0%	89.3%	87.1%	88.8%
fb = 6	88.6%	87.1%	88.6%	88.1%
STFT	81.4%	83.6%	86.4%	83.8%

3. Results

Firstly we compare the network accuracy for the CNN with the time windowed Fourier transformation (short time Fourier transformation, STFT) against the one with the wavelet with $f_b = 1$ up to $f_b = 6$. Training is done with 200 sets of imagery data, 140 are used for training and 60 are reserved for accuracy evaluation. Results are shown in Table 3 for subject 1 and Table 4 for subject 2.

Noticeably, using wavelet transformation gives better accuracy up to 5% for subject 1 and 1% for subject 2 (grayed in the tables). We did not notice strong dependence on the parameter f_b, so hereafter, we use a network with wavelet spectral pre-processing with $f_b = 6$.

To test our CNN+LSTM network we firstly train the system with 60 real EEG labeled data. Then we create 120, 300 and 600 artificial data with the methodology explained above and use them again as training sets. In total, the network is trained with 180, 360 and 660 sets.

Then, we use from 6, 12 and 30 real EEG data to create artificial frames in order to obtain a total of 60 training sets. We train the

Table 4
The accuracy for the CNN with a short time-window Fourier transformation for subject 2.

	Accuracy for each trial			Average
	#1	#2	#3	
fb = 1	85.0%	88.6%	85.7%	86.4%
fb = 2	85.7%	87.1%	85.0%	85.9%
fb = 3	87.1%	85.0%	87.9%	86.7%
fb = 4	87.9%	88.6%	85.7%	87.4%
fb = 5	86.4%	88.6%	85.7%	86.9%
fb = 6	85.7%	87.1%	87.1%	86.7%
STFT	86.4%	85.7%	87.1%	86.4%

CNN+LSTM network again with those. This procedure is done for subject 1 and subject 2.

In the next four tables we compare the accuracy of a CNN network with a CNN with a long short-term memory cell (LSTM). In Tables 5 and 6 are shown the results for the bare CNN network for subject one and two respectively. Then the same results are shown for the CNN+LSTM network again for the two subjects, Tables 7 and 8.

4. Discussion

We compared our results with the work of Cichocki and Phan (2010) where similar motor-imagery brain computer interface datasets were classified using the CSP method, an advanced tensor decomposition approach [25]. We notice that the CNN+LSTM network shows an increase of 5.18% for subject one and a decrease of 3.57% for the second subject. However, as it is evident from the results in Tables 5 and 6, when we add 600 artificial training data with our method the accuracy raises for both subjects. In particular, the accuracy of 88.21% for subject 2, and 89.46% obtained with subject1, are excellent values similar to previous motor-imagery results [25]. Compared to the CNN network performance, our proposed CNN+LSTM system increases accuracy of 3.21% for subject one and 0.36% for subject two (Table 9). Considering both our tests with CNN and CNN+LSTM, the additional EMD artificial data in the network training produce better or, depending on subjects, slightly better accuracy compared to past literature accuracy on EEG motor-imagery tests.

4.1. CNN and CNN+LSTM compared

In Fig. 13 we show the effect of our artificial frames on the training process, for both networks CNN and CNN+LSTM compared. For subject one, the first 30 and 60 real EEG frames (zero artificial training sets) result in an increase for both CNN and CNN+LSTM models. However, with 30 frames CNN ha slightly less accuracy than CNN+LSTM and the opposite happens with 60 training frames. A similar behavior is observed for subject two: for 30 training sets CNN+LSTM has better performances, for 60 the CNN network improves and the two models have almost the same accuracy. We can say that for a low number of training data, the CNN+LSTM performs better than the CNN, whereas when the data are abundant the two networks appear to be equivalent. The LSTM memory appears to be critical when the training data are scarce, whereas this become unimportant when the training is plenty.

4.2. The dependence on training sets number

In Fig. 14 we show the CNN+LSTM accuracy response when trained with an increasing number of real EEG frames.

Clearly the performance gets better with training, however the improvement is reduced for higher number of training frames.

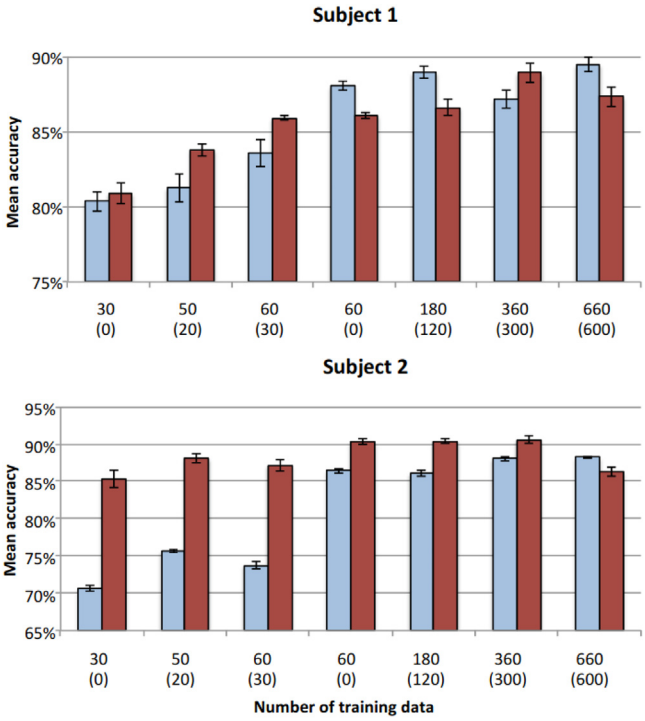

Fig. 13. The effect of EMD frames on the CNN (light color) and CNN+LSTM (dark color) networks. Accuracy compared for both subjects (see Tables 5–8). The introduction of the LSTM produces improved precision in almost all cases. On the vertical axis the accuracy is shown in percentage, averaged over four experiments with identical conditions and different random seeds. On the horizontal axis the numbers of total training frames are shown, below in parentheses the numbers of artificial frames are included in the count.

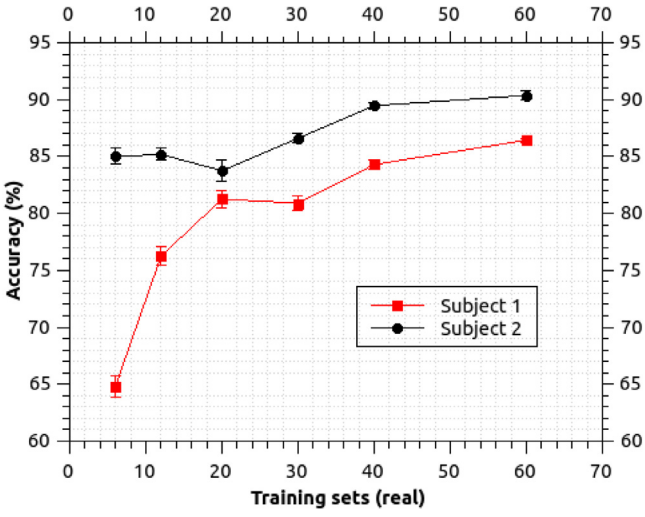

Fig. 14. The accuracy of the CNN+LSTM network response in function of real training data. The accuracy improves with the number of training sets, for subject 1 this effect is especially dominant. Data are extracted from Tables 7 and 8 (no artificial data).

For both subjects we observe a lower accuracy when the network training is scarce. Increasing the number of training set, as expected, improves the overall accuracy, nevertheless it remains noticeably lower for subject 1. Since for equal number of training frames, the network does not perform equally with the two subjects, we can speculate that subject 1 was distracted or less focused on the task. In other words, he/her had other thoughts

Table 5
Classification accuracy with a CNN network (subject 1). The simulation is repeated 4 times using different random seeds. The number of real and artificial data are changed as in the table. The validation tests are done with 140 EEG recording. The grayed cell emphasizes the best accuracy.

Number of training data			Accuracy of each validation				Average	SE	SD
Real data	Artificial data	Total	#1	#2	#3	#4			
30	0	30	80.00%	80.71%	82.14%	78.57%	80.36%	1.29%	0.64%
30	20	50	83.57%	78.57%	82.14%	80.71%	81.25%	1.85%	0.92%
30	30	60	84.29%	85.71%	80.71%	83.57%	83.57%	1.82%	0.91%
60	0	60	88.57%	87.14%	88.57%	87.86%	88.04%	0.59%	0.30%
60	120	180	87.86%	89.29%	90.00%	88.57%	88.93%	0.80%	0.40%
60	300	360	87.86%	87.86%	87.86%	85.00%	87.14%	1.24%	0.62%
60	600	660	90.00%	90.00%	90.00%	87.86%	89.46%	0.93%	0.46%

Table 6
Classification accuracy with a CNN network (subject 2). Data are shown in the same fashion as Table 5.

Number of training data			Accuracy of each validation				Average	SE	SD
Real data	Artificial data	Total	#1	#2	#3	#4			
30	0	30	72.14%	70.71%	70.00%	70.00%	70.71%	0.87%	0.44%
30	20	50	76.43%	75.51%	75.71%	75.00%	75.66%	0.51%	0.26%
30	30	60	75.00%	72.86%	74.29%	72.86%	73.75%	0.93%	0.46%
60	0	60	85.71%	87.14%	87.14%	85.71%	86.43%	0.71%	0.36%
60	120	180	85.00%	87.14%	86.43%	85.71%	86.07%	0.80%	0.40%
60	300	360	87.14%	88.57%	87.86%	88.57%	88.04%	0.59%	0.30%
60	600	660	87.86%	88.57%	87.86%	88.57%	88.21% (a)	0.36%	0.18%

Table 7
Classification accuracy with Convolutional LSTM neural network (subject 1). The number of validation sets is 140. The grayed cell emphasizes the best parameters. The data in bold are those used in the plot of Fig. 13.

Number of training data			Accuracy of each validation				Average	SE	SD
Real data	Artificial data	Total	#1	#2	#3	#4			
6	0	6	65.71%	67.14%	62.14%	64.29%	64.82%	1.85%	0.92%
12	0	12	75.00%	74.29%	78.57%	77.14%	76.25%	1.70%	0.85%
20	0	20	81.43%	82.14%	82.85%	78.57%	81.25%	1.63%	0.81%
30	**0**	**30**	81.43%	78.57%	81.43%	82.14%	**80.89%**	1.37%	0.69%
40	0	40	83.57%	83.57%	85.00%	85.00%	84.29%	0.71%	0.36%
6	44	50	65.71%	68.57%	65.71%	56.43%	64.11%	4.58%	2.29%
12	38	50	75.71%	78.57%	76.43%	77.14%	76.96%	1.06%	0.53%
30	**20**	**50**	85.00%	83.57%	83.57%	82.86%	**83.75%**	0.78%	0.39%
6	56	60	67.14%	63.57%	57.14%	59.29%	61.79%	3.86%	1.93%
12	48	60	78.57%	76.43%	78.57%	80.71%	78.57%	1.52%	0.76%
20	40	60	84.29%	82.86%	84.29%	80.71%	83.04%	1.46%	0.73%
30	**30**	**60**	85.71%	85.71%	86.43%	85.71%	**85.89%**	0.31%	0.15%
40	20	60	86.43%	86.43%	85.71%	86.43%	86.25%	0.31%	0.15%
60	**0**	**60**	86.43%	86.43%	85.71%	85.71%	**86.07%**	0.36%	0.18%
60	**120**	**180**	85.00%	87.86%	87.14%	86.43%	**86.61%**	1.06%	0.53%
60	**300**	**360**	90.71%	89.29%	88.57%	87.14%	**88.93%**	1.29%	0.64%
60	**600**	**660**	86.43%	85.71%	88.57%	88.57%	**87.32%**	1.28%	0.64%

that were perturbing the motor-imagery EEG patterns, this was reflected on the network ability to recognize them (Fig. 14).

In Fig. 15 we plot the reciprocal of the training sets number against the accuracy. For subject 1, there is a clear linear correlation, with regression coefficient of $R^2 = 0.97$. In this reciprocal space, the negative inclination and position of the intercept suggests the theoretical accuracy limit for an infinite number of training sets, for subject 1 this is about 87%. In the case of the second subject, we observe the same negative trend, however linearity is less prominent, especially for the last two points of the curve. Those correspond to tests with less training data where the network seems to produce much better accuracy than with subject 1. However, considering only the first four points (more than 20 training sets) the linear trend is clear, it shows a fit of $R^2 = 0.98$ and an intercept of about 93%. This is suggesting a better accuracy for this subject in the theoretical case of a great number of training sets.

Using the same reciprocal plot, we can visualize the effect of artificial frames on the network accuracy (see Fig. 16). Here we use an increasing number of real EEG frames used for training the network and add a variable number of artificially generated frames in order to reach a total of 60 training sets. The plot shows the accuracy using only real frames with filled dots, and the accuracy with the real and artificial ones with asterisks. Clearly, the artificial frames enhance the overall accuracy of the network, the only exception is when the number of real frames is very small. The inversion point is indicated by the arrows (0.13–0.14 corresponds to about 7–8 frames). This inversion is expected since the artificial frames are created from real EEG data. When the number of those is limited, the new frames are synthesized by a small number of intrinsic functions and the random mixing these in the empirical mode decomposition presumably ends up in reproducing a bad copy of the original EEG frames, actually degrading the network ability to classify properly the EEG patterns. On the other hand, when the number of available real EEG frames surpasses the threshold approximately indicated by the arrows, the effect of the intrinsic mode functions intermixing begins to introduce positive effects on the CNN's classification ability.

Table 8
Classification accuracy with Convolutional LSTM neural network (subject 2). The grayed cell emphasizes the best parameters.

Number of training data			Accuracy of each validation				Average	SE	SD
Real data	Artificial data	Total	#1	#2	#3	#4			
6	0	6	83.57%	83.57%	86.43%	86.43%	85.00%	1.43%	0.71%
12	0	12	85.71%	85.00%	86.43%	83.57%	85.18%	1.06%	0.53%
20	0	20	84.29%	82.14%	86.43%	82.14%	83.75%	1.78%	0.89%
30	**0**	**30**	87.86%	86.43%	85.71%	86.43%	**86.61%**	0.78%	0.39%
36	0	36	90.71%	88.57%	87.86%	88.57%	88.93%	1.07%	0.54%
40	0	40	89.29%	88.57%	90.00%	90.00%	89.46%	0.59%	0.30%
30	**20**	**50**	87.14%	87.86%	87.14%	90.00%	**88.04%**	1.17%	0.59%
6	54	60	83.57%	78.57%	85.71%	85.00%	83.21%	2.79%	1.39%
12	48	60	90.00%	86.43%	87.14%	87.14%	87.68%	1.37%	0.69%
20	40	60	85.71%	87.14%	85.00%	83.57%	85.36%	1.29%	0.64%
30	**30**	**60**	88.57%	85.00%	86.43%	88.57%	**87.14%**	1.52%	0.76%
40	20	60	89.29%	90.00%	90.00%	89.29%	89.64%	0.36%	0.18%
60	**0**	**60**	89.29%	90.71%	91.43%	90.00%	**90.36%**	0.80%	0.40%
60	**120**	**180**	90.71%	89.29%	90.71%	90.71%	**90.36%**	0.62%	0.31%
60	**300**	**360**	90.00%	89.29%	90.71%	92.14%	**90.54%**	1.06%	0.53%
60	**600**	**660**	87.86%	84.29%	86.43%	86.43%	**86.25%**	1.28%	0.64%

Table 9
The comparison of our results with CSP method. The CNN+LSTM approach improved the accuracy for both subjects.

Method	Accuracy	
	Subject 1	Subject 2
CSP	82.86%	90.00%
CNN	88.04%	86.43%
CNN + LSTM	86.07%	90.36%

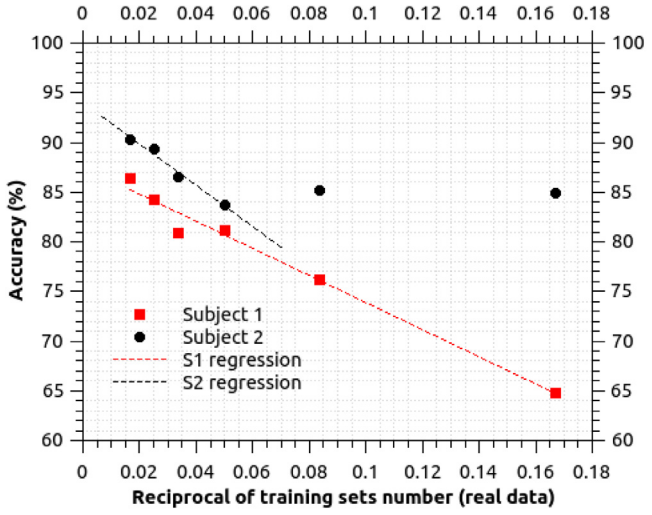

Fig. 15. The network classification accuracy against the reciprocal of the number of training sets. The intercept with the vertical axis indicates the theoretical accuracy limit for an ideal infinite number of training sets. Subject1 seems to converge to a maximum accuracy of about 87%–88%. However, the other subject data show linearity only when the network is well trained. In fact the last two points corresponding to 12 and 6 training sets only seem to depart from linearity. The regression line excluding these two suggests a theoretical limit of about 93%.

5. Conclusions

In this study we compared a convolutional neural network (CNN) and one with a long short-term memory block (CNN+LSTM) in the task of recognizing brain wave spatial–temporal patterns in motor-imagery experiments. Because of the scarce availability of real EEG recordings on human subjects, we proposed a method to synthesize artificial EEG frames from subject's real recording, and evaluated the effect of them on the network performances. The main findings of this study are the following:

- The CNN and CNN+LSTM networks we proposed performed similarly or better than previous method used in literature for the same motor-imagery tasks. In particular, our results compare well with CSP, having, depending on subjects, similar or better results (Table 9).
- The convolutional network with long short-term memory blocks (CNN+LSTM) improves results of a similar network without the LSTM block. This is apparent especially when the number of available training data is less (Fig. 13).
- The addition of artificial frames tends to improve the accuracy of a CNN+LSTM network, and this improvement is prone to reach a plateau for greater number of artificial frames added (Fig. 13).
- The (accuracy) vs (reciprocal of training sets) space shows a quasi linear behavior that can be used to infer the network accuracy limits (Figs. 15 and 16).

Our study demonstrates for the first time that with a CNN+LSTM network, it is possible to improve the EEG pattern recognition by introducing artificial frames realized with intermixing intrinsic mode functions in an empirical mode decomposition process. This helps in better training neural networks for classification of motor-imagery tasks, with less training sets and better accuracy. Our work may pave the way for the realization of next generation brain computer interfaces or other future devices that rely upon brain-wave analysis.

CRediT authorship contribution statement

Kahoko Takahashi: Conceptualization, Methodology, Investigation, Software, Writing. **Zhe Sun:** Methodology, Investigation, Supervision, Software, Reviewing and editing. **Jordi Solé-Casals:** Methodology, Investigation, Supervision, Software, Reviewing and editing. **Andrzej Cichocki:** Methodology, Supervision. **Anh Huy Phan:** Methodology, Software. **Qibin Zhao:** Methodology, Data collection. **Hui-Hai Zhao:** Methodology, Investigation. **Shangkun Deng:** Methodology. **Ruggero Micheletto:** Methodology, Investigation, Supervision, Software, Reviewing and editing.

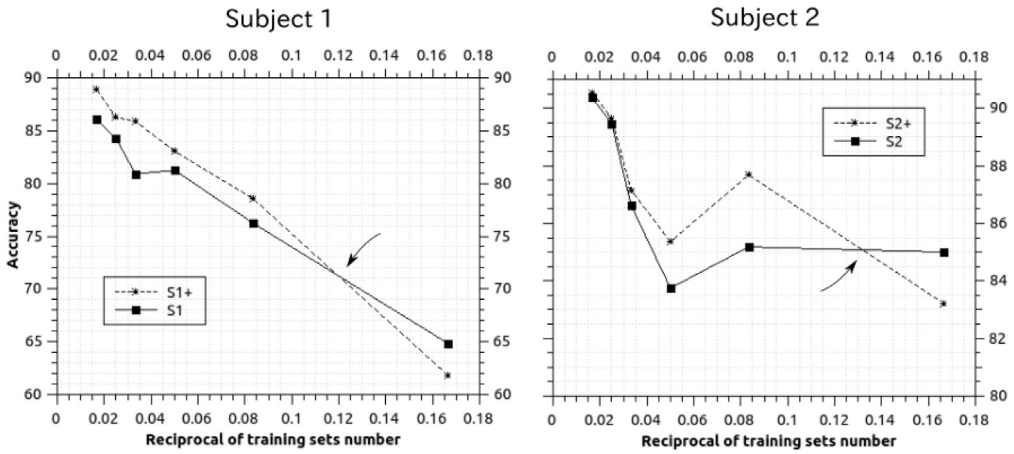

Fig. 16. A comparison of the CNN+LSTM network accuracy with and without artificial frames. The dotted line and "*" symbol display the accuracy when artificial frames are used (S1+ and S2+). These are advantageous for both subjects, however improvement is especially noticeable for the subject whose EEG shows lower accuracy (subject 1). In that case the artificial frames bring the network performance to the same level of subject 2 (>85% for 60 or more frames). The arrows indicate the point where the use of artificial frames becomes advantageous, about 0.13–0.14, corresponding to more than 7–8 frames. On the horizontal axis we have the reciprocal of the training set number from real data(1/6, 1/12, 1/20, 1/30, 1/40 and 1/60). The number of artificial frames added to those is enough to reach 60 frames in total with the exception of 60 real frames where we added 300 artificial ones (6+54, 12+48, 20+40, 30+30, 40+20 and 60+300). Data are extracted from Tables 7 and 8.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

J.S-C. contribution was partially supported by the University of Vic - Central University of Catalonia, Spain (No. R0947) and is also based upon COST Action CA18106, supported by COST (European Cooperation in Science and Technology). The work of A. C and A.H P. was partly supported by Ministry of Science and Higher Education grant No. 075-10-2021-068, and the joint projects: Artificial Intelligence for Life between SKOLTECH and the University of Sharjah.

References

[1] J.R. Wolpaw, N. Birbaumer, D.J. McFarland, G. Pfurtscheller, T.M. Vaughan, Brain–computer interfaces for communication and control, Clin. Neurophysiol. 113 (6) (2002) 767–791.
[2] B.J. Lance, S.E. Kerick, A.J. Ries, K.S. Oie, K. McDowell, Brain–computer interface technologies in the coming decades, Proc. IEEE 100 (Special Centennial Issue) (2012) 1585–1599.
[3] J. V., F. Lotte, M. Tangermann, Brain-computer interfaces: Beyond medical applications, Computer 45 (4) (2012) 26–34.
[4] G.J. N. Luis Fernando, Brain computer interfaces, a review, Sensors 12 (2) (2012) 1211–1279.
[5] F. Lotte, L. Bougrain, A. Cichocki, M. Clerc, M. Congedo, A. Rakotomamonjy, F. Yger, A review of classification algorithms for EEG-based brain–computer interfaces: a 10 year update, J. Neural Eng. 15 (3) (2018) 031005.
[6] K. Muller, C.W. Anderson, G.E. Birch, Linear and nonlinear methods for brain-computer interfaces, IEEE Trans. Neural Syst. Rehabil. Eng. 11 (2) (2003) 165–169.
[7] S. Soman, et al., High performance EEG signal classification using classifiability and the twin SVM, Appl. Soft Comput. 30 (2015) 305–318.
[8] Z. Dalin, Y. Lina, Z. Xiang, W. Chen, S. Wang, B. Robert, B. Boualem, Cascade and parallel convolutional recurrent neural networks on EEG-based intention recognition for brain computer interface, in: Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence (AAAI18, Humans and AI), February 2-7, 2018, New Orleans, la, USA., AAAI, 2018, pp. 1703–1710.
[9] H. Ramoser, J. Muller-Gerking, G. Pfurtscheller, Optimal spatial filtering of single trial EEG during imagined hand movement, IEEE Trans. Rehabil. Eng. 8 (4) (2000) 441–446.
[10] K. Shiu, S. Alok, M. Kabir, T. Tatsuhiko, A deep learning approach for motor imagery EEG signal classification, in: 2016 3rd Asia-Pacific World Congress on Computer Science and Engineering, APWC on CSE, IEEE, 2016, pp. 34–39.
[11] S. Soman, S. Saxena, et al., Eigensample: A non-iterative technique for adding samples to small datasets, Appl. Soft Comput. 70 (2018) 1064–1077.
[12] C. Josep, O. Rupert, G. Christoph, S. Jordi, A new method to generate artificial frames using the empirical mode decomposition for an EEG-based motor imagery BCI, Front. Neurosci. 12 (2018) 308.
[13] S.J.D.J. Z. Zhiwen, C. Andrzej, Y. Zhenglu, S. Zhe, A novel deep learning approach with data augmentation to classify motor imagery signals, IEEE Access 7 (2019) 15945–15954.
[14] P. Nunez, E. Nunez, R. Srinivasan, A. Srinivasan, O.U. Press, Electric Fields of the Brain: The Neurophysics of EEG, Oxford University Press, 2006.
[15] P.J. A., The functional significance of mu rhythms: translating "seeing" and "hearing" into "doing", Brain Res. Rev. 50 (1) (2005) 57–68.
[16] G. Pfurtscheller, F.L. da Silva, Event-related EEG/MEG synchronization and desynchronization: basic principles, Clin. Neurophysiol. 110 (11) (1999) 1842–1857.
[17] S.R. Tibor, S.J. Tobias, F.L.D. Josef, G. Martin, E. Katharina, T. Michael, H. Frank, B. Wolfram, B. Tonio, Deep learning with convolutional neural networks for EEG decoding and visualization, Hum. Brain Map. 38 (11) (2017) 5391–5420.
[18] S. Hochreiter, J. Schmidhuber, Long short-term memory, Neural Comput. 9 (8) (1997) 1735–1780.
[19] F.A. Gers, J. Schmidhuber, Recurrent nets that time and count, in: Proceedings of the IEEE-INNS-ENNS International Joint Conference on Neural Networks. IJCNN 2000. Neural Computing: New Challenges and Perspectives for the New Millennium, Vol. 3, IEEE, 2000, pp. 189–194.
[20] D.E. Rumelhart, G.E. Hinton, R.J. Williams, Learning representations by back-propagating errors, Nature 323 (1986) 533–536.
[21] Y. LeCun, Y. Bengio, G. Hinton, Deep learning, NATURE 521 (7553) (2015) 436–444.
[22] X. Yu, N.K. Loh, W.C. Miller, New acceleration technique for the back-propagation algorithm, in: IEEE International Conference on Neural Networks, San Francisco, California, 28 March-1 April 1993, IEEE, 1993, pp. 1157–1161.
[23] C.C. Yu, B.D. Liu, A backpropagation algorithm with adaptive learning rate and momentum coefficient, in: 2002 International Joint Conference on Neural Networks, IJCNN '02, Honolulu, HI; United States; 12 May 2002, IEEE, 2002, pp. 1218–1223.
[24] N. Huang, Z. Shen, S. Long, M. Wu, H. Shih, Q. Zheng, N. Yen, C. Tung, H. Liu, The empirical mode decomposition and the Hilbert spectrum for nonlinear and non-stationary time series analysis, Proc. Royal Soc. A-Math. Phys. Eng. Sci. 454 (1971) (1998) 903–995.
[25] A.H. Phan, A. Cichocki, Tensor decompositions for feature extraction and classification of high dimensional datasets, Nonlinear Theory Appl., IEICE 1 (1) (2010) 37–68.
[26] V.J. Lawhern, A.J. Solon, N.R. Waytowich, S.M. Gordon, C.P. Hung, B.J. Lance, EEGNet: a compact convolutional neural network for EEG-based brain-computer interfaces, J. Neural Eng. 15 (5) (2018).

[27] P. Wang, A. Jiang, X. Liu, J. Shang, L. Zhang, LSTM-based EEG classification in motor imagery tasks, IEEE Trans. Neural Syst. Rehabil. Eng. 26 (11) (2018) 2086–2095.

[28] Y.-H. Kwon, S.-B. Shin, S.-D. Kim, Electroencephalography based fusion two-dimensional (2D)-convolution neural networks (CNN) model for emotion recognition system, Sensor 18 (5) (2018) 1383.

[29] F. Chollet, et al., Keras, 2015, https://keras.io.