Postprint of: Kurowski A., Mrozik K., Kostek B., Czyżewski A., Method for Clustering of Brain Activity Data Derived from EEG Signals, FUNDAMENTA INFORMATICAE, Vol. 168, iss. 2-4 (2019), pp. 249-268, DOI: 10.3233/FI-2019-1831

Fundamenta Informaticae 168 (2019) 1–20
DOI 10.3233/FI-2019-1800
IOS Press

Method for Clustering of Brain Activity Data Derived from EEG Signals

Adam Kurowski
Multimedia Systems Department
Faculty of Electronics
Telecommunications and Informatics
Gdansk University of Technology, Gdansk, Poland

Katarzyna Mrozik
Multimedia Systems Department
Faculty of Electronics
Telecommunications and Informatics
Gdansk University of Technology, Gdansk, Poland

Bozena Kostek*
Audio Acoustics Laboratory
Faculty of Electronics
Telecommunications and Informatics
Gdansk University of Technology, Gdansk, Poland
bozena.kostek@pg.edu.pl

Andrzej Czyzewski
Multimedia Systems Department
Faculty of Electronics
Telecommunications and Informatics
Gdansk University of Technology, Gdansk, Poland

Abstract. A method for assessing separability of EEG signals associated with three classes of brain activity is proposed. The EEG signals are acquired from 23 subjects, gathered from a headset consisting of 14 electrodes. Data are processed by applying Discrete Wavelet Transform (DWT) for the signal analysis and an autoencoder neural network for the brain activity separation. Processing involves 74 wavelets from 3 DWT families: Coiflets, Daubechies and Symlets. Euclidean distance between clusters normalized with respect to the standard deviation of the whole set of data are used to separate each task performed by participants. The results of this stage allow for an assessment of separability between subsets of data associated with each activity performed by experiment participants. The speed of convergence of the training process employing deep learning-based clustering is also measured.

Keywords: brain activity, brain-computer interfaces, EEG, data clustering, machine learning, deep learning

*Address for correspondence: Audio Acoustics Laboratory, Faculty of Electronics, Telecommunications and Informatics, Gdansk University of Technology, Gdansk, Poland

1. Introduction

Recent years brought several research studies associated with classification of EEG signals related to brain activity or to emotional state recognition [1, 2, 3, 4, 5]. However, a computational model that is able to assign acquired signals to activity or emotion is not yet available, even though some researchers obtained encouraging classification effectiveness (approx. 70% and higher) [5, 6]. The main difficulty in comparing these approaches lies in the problem of utilizing different ways of acquiring signals, different processing methods, classifiers, etc. Moreover, both the number of subjects, as well as time related to performing various tasks, differ between studies. A variety of problems associated with the signal quality, the size of training set or the choice of a classification algorithm employed for classification of EEG signals was presented in the review paper of Lotte et al. [6]. A very thorough review of the state-of-the-art of BCI (Brain-Computer Interfaces) systems was also presented a few years ago by Nicolas-Alonso and Gomez-Gil, showing that a plethora of approaches is utilized along with a wide range of applications already being available in this domain [7]. Still, the most challenging issues are the pre-processing and feature extraction, as they need extracting meaningful information from a large volume of poor-quality data contaminated by artifacts and noise [6, 8]. Several research papers is concentrated on the problem of finding new ways of feature extraction and proposing solutions tailored to various tasks. Examples of such studies may be a classification of affective states employing features related to the energy of EEG signals and to connectivity parameters [8] or classification of spontaneous motor-related tasks with the use of subset of best-performing features pre-selected from a broader set of parameters [9]. Also, there is a need for personalized HCI (Human Computer Interaction) or BCI models allowing for the users interaction adaptation according to the feedback from the user [10].

Therefore, the objective of this work was to assess whether is it possible to create a feature extraction algorithm which would be useful for data clustering and for unsupervised classification of EEG signals. An expected result of this approach is defining an effective feature vector for an automatic EEG signals classification, derived from the brain activity analysis. Generation of such a vector should not be performed by a human, but be performed by the machine learning algorithm. Therefore, in this work we would like to present a way of machine learning-based extraction of features from EEG signals, which is based on a deep learning algorithm. Such feature vectors (vectors of parameters) may be used as a parametric representation of EEG signal for algorithms performing analysis and processing of such signals. Moreover, due to the nature of the autoencoder this transformation to the domain of parameters is reversible within the margin defined by the error obtained during the training of the autoencoder neural network. This idea is commonly used in the field of face recognition for preparation of feature vectors utilized by the machine learning classifier. Such a set of features is then processed by another classifier in order to i.e. associate images of faces with a specific person [11, 12, 13]. In this field, extraction of features is typically performed by a convolutional neural network (CNN). Parts of the network responsible for feature extraction are trained on large datasets, and then they are usually re-used in many tasks related to the face recognition problem. Such an idea is called transfer learning. The feature extraction part of the network is trained only once. Only the classifier has to be trained on the task-specific data. The similar, associated with EEG signals solution can potentially be utilized to improve the performance of brain-computer interfaces, which are applicable to controlling a variety of electronic devices used for many potential fields of applications. Particularly, it may create new

possibilities for people, who suffer from injuries, which would, in turn, help them to communicate in other ways that those employing standard modalities, like speech or gestures. It may also provide a way of controlling electronic devices in environments in which communication is difficult. Feature extraction algorithms presented in this paper for the first stage of processing utilize a typical approach, i.e. Discrete Wavelet Transform (DWT) and in addition they are based on the concept of an autoencoder employing a neural network, which is capable of performing the dimensionality reduction of data [14], which was originally developed for the purpose of data compression [16]. In the study carried out, the compression dimensionality reduction properties of the autoencoder algorithm principle is applied to generate a small set of features associated with frames of the EEG signal gathered from a headset. The generated stream of feature values passes through the clustering algorithm, which may then be used for the purpose of EEG signals classification or for finding out whether the mental state of the examined subject changed during the study. Electroencephalographic (EEG) signals were acquired from 23 (19 males, 4 females) participants aged between 23 and 44 years. Participants did not have knowledge of tasks to be done in the course of experiments and therefore, they also did not practice those tasks before the data acquisition session. Also, all subjects were healthy, and they did not reported any neurological disorders which could have an influence on outcomes obtained in the study. Results were analyzed with the use of the wavelet-based feature extraction and of an autoencoder neural network as a parameterization tool. Tests were performed for 74 mother wavelets, 15 of them were further chosen for some more detailed analyses. Finally, in order to evaluate separability between task-related classes, a comparison of distances normalized with respect to the standard deviation of the whole dataset was made for the sets of feature vectors associated to each task performed by the experiment participants.

2. Signal acquisition method

The input data were gathered for the signal processing algorithms during sessions lasting for approximately 20 minutes. Each session consisted of three tasks performed by the subjects:

- relaxation with closed eyes - before this task execution the subjects were asked to relax in order to induce mental states associated with such an activity,
- watching a music video - music genre chosen for this task was folk metal due to the engaging and lively character of this music style which creates an opportunity to induce mental states that were being of an opposite type to those ones in the preceding task,
- playing a logic game, the game used at this stage of the experiment was Netwalk logic computer game, in which a player had to rotate elements of the board in such a manner, that a connection was being established between a single central element called server and multiple peripheral elements called computers.

Each task lasted for approximately 5 minutes. Data were collected with the use of scripts written in Python programming language extended by libraries provided by the headset manufacturer in order to capture the raw EEG signals from the headset. The hardware used for the data acquisition was

Emotiv EPOC headset, which acquires signals from 14 electrodes assigned according to the 10-20 placement standard: AF3, F7, F3, FC5, T7, P7, O1, O2, P8, T8, FC6, F4, F8 and AF4. The electrodes used in this type of the headset were saline-based ones. The sampling rate was equal to 128 S/s. Data acquired from the headset were processed with the use of a separate set of scripts, also written in the Python programming language. The transmission of signals acquired by the device was wireless.

3. Processing algorithms and analyses

Processing of data gathered from the EEG acquisition headset was also implemented in Python programming language. Libraries used for the numerical calculations performed on signals gathered from measurement were as follows: NumPy, SciPy and PyWalets. Scikit-learn and TensorFlow, employed as machine learning and deep learning libraries [15, 16, 17]. In the experiment, 74 mother wavelets were evaluated with respect of their ability to provide spontaneous discrimination between the three of the tasks assigned to participants. Research was conducted for three chosen families of wavelets: Coiflets (1-17), Daubechies (1-38) and Symlets (2-20). The numbers in brackets refer to all available orders of wavelets from those types which are contained in PyWavelets library. The families selected for the study were examples of groups commonly used in EEG signal processing [18, 19].

First, the input data had to be split into groups of recorded excerpts associated with each task performed by participants. The length of raw data gathered in the process of the experimental sessions was 6 hours and 3 minutes. For each of 23 files containing data gathered from 23 participants, the first and the last 10 seconds were removed to eliminate artifacts occurring due to possible movements shortly before and shortly after the recording started. Above mentioned artifacts were associated with putting the headset on or with removing it or correcting its position on the participants head. Then, each recording was edited in such a way that it was split into 3 even parts, with a 5-second margin separating each part. The scheme of signal editing is depicted in Figure 1.

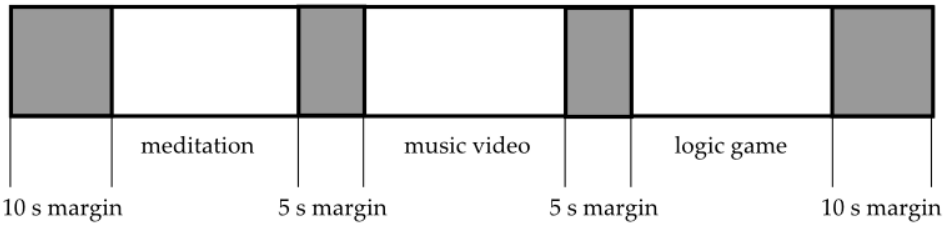


Figure 1. Scheme of splitting the data into task-related fragments

After splitting recordings into three class-associated fragments, each of those fragments was further divided into sets of 5-second long frames. Therefore, taking into account the sampling rate of 128 S/s, the frame contained 640 samples-long fragments of signals obtained from 14 EEG channels. Frames were taken with the overlap ratio of 0.5. Finally, each of 3 classes of activity consisted of 2778 examples which gives the total of 8334 examples for the training of the autoencoder neural network.

Frames were transformed with the use of DWT (Discrete Wavelet Transform). Each channel was decomposed with employing each of 74 analyzed mother wavelets. The result was a set of coefficients organized in the form of 2D matrix, as it is depicted in Figure 2.

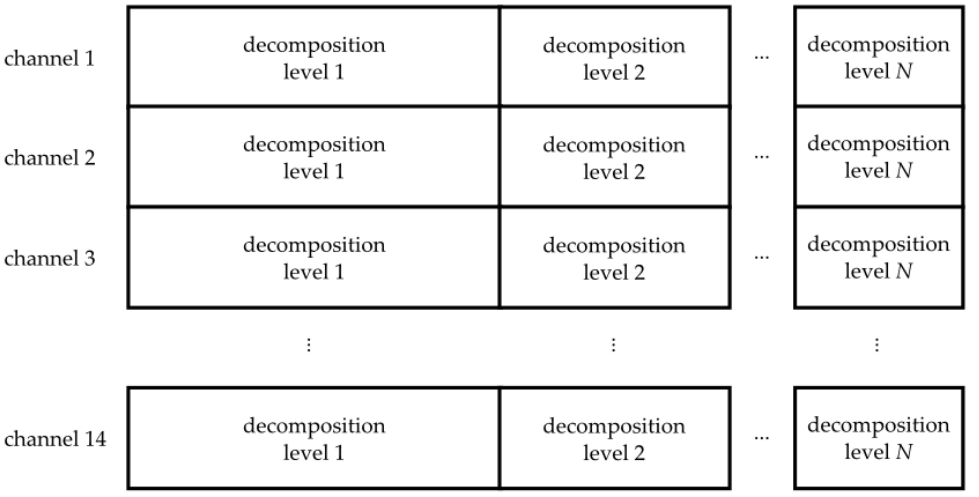


Figure 2. Structure of frame after calculation of the DWT coefficients for each of 14 channels, and for each of N levels of decomposition. The number of levels of the decomposition changes with a type of the mother wavelet used

The number of levels of decomposition is denoted as N, therefore the length of the channel depicted in the Figure 2. depened on the choice of the mother wavelet. In case of our research the number of levels of decompositions varied from 3 to 10. The width of the frame varies from 642 to 869 samples, the height was always equal to 14, the number representing EEG channel signals. The last step was normalization of the data. Based on the literature review [20] we decided to use z-score of input data due to its simplicity. z-score calculation is performed with respect to the following formula:

$$z = \frac{x - x'}{\delta_x}, \tag{1}$$

where z is the normalized feature vector, x is an unnormalized feature vector, x' is the mean value of the vector x , and δ is the standard deviation of this vector. Next, a minimal value of z is subtracted from z.

In the next step, a set of training examples was shuffled and the training of the convolutional autoencoder started. The autoencoder is a special machine learning-based method which can be applied to the feature extraction process. Autoencoders, in addition to general tasks related to dimensionality reduction and feature extraction, can perform tasks such as sound synthesis or pre-training of deep learning models [21, 22]. The network in the autoencoder consists of two parts: encoder function $h = f(x)$ and decoder function $r = g(h)$, where: x denotes an input, h is called the hidden layer. The

aim of the decoder function is to produce the reconstruction of the original function $r = g(f(x)) = x$ with the least possible amount of deformation. The structure of the corresponding type of a neural network is presented in Figure 3.

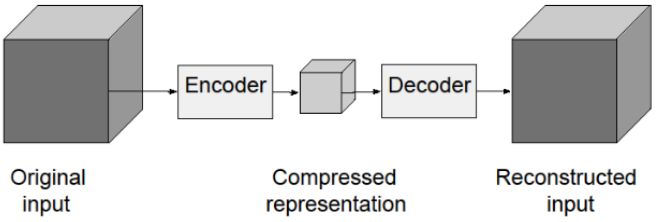


Figure 3. Structure of autoencoder neural network

This kind of the network may reduce dimensionality in the following way: both the input and the reconstructed output have the same dimensions as the vector which should be compressed, i.e. the dimension should be equal to N. The encoder part has n outputs which are elements of the vector h, which is then used for the reconstruction of the original vector by the decoder network. The network may also be utilized for the extraction of n features associated with vector of N samples. Autoencoder neural networks can also be used for processing and dimensionality reduction of EEG signals and feature sets associated with such signals [23, 24]. In the case of our study, the autoencoder contains convolutional layers in both the encoder and decoder in order to enhance its ability to calculate features from 2D results of the DWT-based data preprocessing. The structure of the neural network was prepared by testing the speed of training, layers were added as long as their addition increased efficiency of training in terms of the acquired loss function value decrease per 20 first training epochs. The training epoch is understood as a full iteration completed over all examples from the training set. Loss function is a cost metric used in the process of learning of the neural network. Its choice depends on the type of a problem solved. In example, for tasks of signal reconstruction, a common choice is

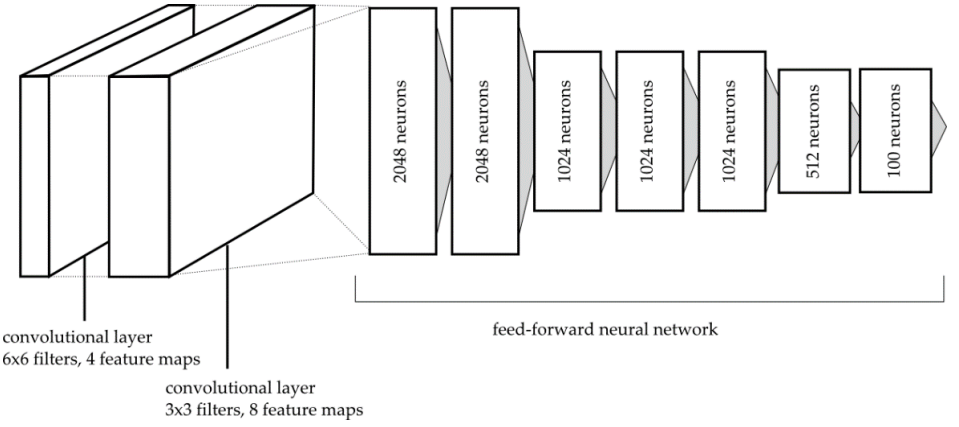


Figure 4. Structure of the encoder network used in the study

a mean square error of output acquired from an autoencoder. The structure of the encoder and the decoder parts of the autoencoder network employed is depicted in Figures 4 and 5.

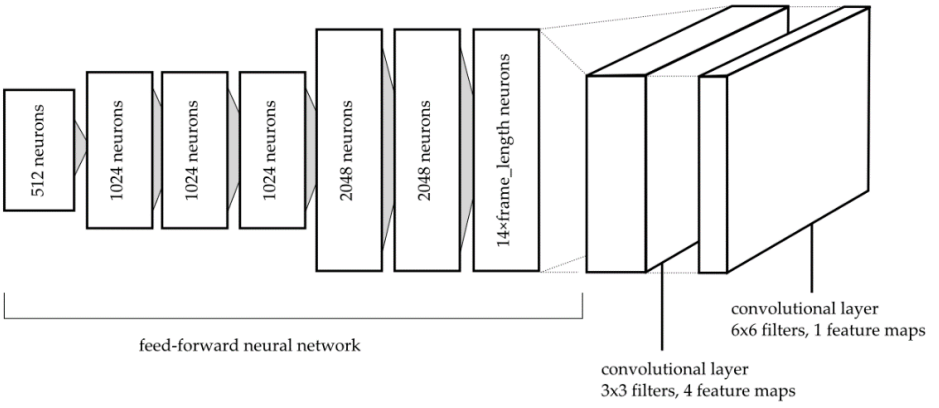


Figure 5. Structure of the decoder network used for the study. Length of frame varies between 642 and 869 samples, therefore also the exact number of neurons in the last layer of the dense neural network depends on the type of the chosen wavelet and on the length of frame resulting from this choice

As can be seen in Figures 4 and 5, the autoencoder consists of two convolutional layers and seven dense layers. An activation function in each case is chosen to be ReLU. In decoder, a so called transposed convolutional layers were used in order to provide the decoder ability of performing the inverse transformation as the encoding part of the neural network. The AdaM optimizer was chosen as a learning optimization algorithm. The autoencoder settings were as follows: the learning rate hyperparameter (α) was set to 10^{-5}, β_1 was equal to 0.999, β_2 to 0.99 and ϵ to 10^{-8}. The mean square error calculated for a pair of 2D input and output data was chosen as an optimized loss function. The length of the representation vector passed from the encoder to the decoder is set as equal to 100. In turn, the batch size used for the training of the autoencoder was set equal to 32. The training was stopped in on one of 3 following terms:

- if the target value of loss was achieved,
- if the maximum number of epochs was achieved,
- if for 10 consecutive samples an absolute change of loss was smaller than 10^{-7}.

The experiment was conducted in two following phases: the broad search for best performing wavelets from a set of 74 potential candidate mother wavelets and the precise stage which was performed for preselected mother wavelets obtained from the broad search stage. For the broad search stage, the maximum number of epochs was set to 75 and target error was set to 10^{-3} and the precise learning stage for which maximal number of training epochs was set to 1000 and target error was lowered to 10^{-5}. After the first stage of the experiment is completed, an evaluation of performance of clustering is conducted. A measure r_v for each task is calculated according to the following formula:

$$r_v\{c\} = var(x_{x \in c})/var(x), \tag{2}$$

where c denotes one of the tasks performed by participants, $var(x_{x \in c})$ is the variance of points calculated from the input data associated with a selected task, which stands for $x_{x \in c}$, $var(x)$ represents variance of all points obtained from the encoder part of the autoencoder network. Also an overall measure, taking into account all classes, is introduced. It is calculated with respect to the following formula:

$$r_g = \max_c(r_v\{c\}) \tag{3}$$

As a result of the execution of the first stage of the experiment four groups of wavelets were chosen for the second stage of the experiment:

- group of three wavelets enabling to obtain three lowest values of r_v for a meditation scenario,
- group of three wavelets enabling to obtain three lowest values of r_v for music video scenario,
- group of three wavelets enabling to obtain three lowest values of r_v for logic game scenario,
- group of six wavelets which provide the lowest maximum value of r_g.

In the second part of the experiment, only wavelets chosen with respect to the criteria above were used. The maximum number of iterations was increased to 1000 and the target error was lowered to 10^{-5}. For the outcome of this stage of the experiment some additional measures were calculated. The first one refers to cumulated confidence interval borders for the variance estimate of each class-related dataset and the sum of these datasets:

$$s^2_{cl,c} = \sum_k \frac{ns^2_{k,c}}{\chi^2_{1-\alpha/2,n-1}}, s^2_{cr,c} = \sum_k \frac{ns^2_{k,c}}{\chi^2_{\alpha/2,n-1}}, \tag{4}$$

where c denotes the name of the data associated with the currently processed class. There are four classes: meditation, logic game, music video, and a general one, which refer to the data samples from the merged first three classes. The standard deviation estimation denoted as s, s^2 represents the unbiased variance estimator, n is the number of points in the set associated with the chosen class, k is an index of a point in the selected set, $\chi^2_{1-\alpha/2,n-1}$ denotes a critical value of χ^2 probability distribution for the level of significance equal to α with $n-1$ degrees of freedom. The parameter with subscript cl denotes the lower boundary of the confidence interval, cr stands for the higher one.

Another measure calculated for the final stage of processing was the cumulated uncertainty of the mean value used for the calculation of the centroid of each class-related group of points denoted as $\Delta d(c)$:

$$\Delta d(c) = \sum_k t_{1-\alpha/2,n-1} \frac{s_{k,l}}{\sqrt{n-1}}, \tag{5}$$

where c one of three analysed classes, $t_{1-\alpha/2,n-1}$ is the critical value of the t-student distribution acquired for the significance level of α and $n-1$ degrees of freedom. Variance and centroid coordinates estimates may be used to define a probable range of values for the distance between class-related

groups of points. To do that, first, an estimate of the range for the distance between centroids should be calculated. The lower estimate of the distance $d_l(c_i, c_j)$ is given by:

$$d_l(c_i, c_j) = \sqrt{\sum_k \min_{p_1,p_2} \{ \big(x'\{k, c_i\} - x'\{k, c_j\} + p_1 \Delta d(c_i) + p_2 \Delta d(c_j)\big)^2 \}}, \tag{6}$$

where x'_c is an estimate of k-th element in a vector representing the vector of points belonging to a class c, and p_1, p_2 are values optimized in the process of finding the minimum value of the optimized expression. The upper estimate of the distance $d_r(c_i, c_j)$ is given by:

$$d_r(c_i, c_j) = \sqrt{\sum_k \max_{p_1,p_2} \{ (x'_{k,c_i} - x'_{k,c_j} + p_1 \Delta d(c_i) + p_2 \Delta d(c_j))^2 \}}, \tag{7}$$

where s_1, s_2 are used for optimizing the expressions (5) and (6) in order to get the minimal or maximal possible value of distance if the confidence interval, while the significance level equal to α is considered. Finally, the estimated range containing the distance between clusters normalized with respect to the estimated standard deviation of the general set associated with EEG data can be calculated:

$$d_\delta \in \langle \frac{d_l(c_i, c_j)}{s_{cr,c}}; \frac{d_r(c_i, c_j)}{s_{cl,c}} \rangle, \tag{8}$$

where the normalized distance between the centroid of data clusters is denoted as d_δ.

4. Results

The experiment was divided into two parts. The first one involved a short training of the autoencoder employing each of 74 wavelets from three families: Coiflets, Daubechies and Symlets. A short training had maximum limit of epochs set to 75 and the target loss value set to 0.001. An evaluation with the use of $r_v(class)$ ratio was then performed for each variant of the autoencoder based upon a particular wavelet. Calculations were performed with the use of the Nvidia DGX station and the Nvidia Geforce GTX 1080 graphics card. Results of this stage of study are shown in Table 1. Wavelet names are denoted in the form of abbreviation, i.e.: coif stands for Coiflets, db for Daubechies, and sym for Symlets. Also, a number denoting the order of the wavelet is provided.

If the subset of vectors associated with a particular class has a lower variance than the set containing all available vectors, there is a chance that it happens due to the presence of clusters. Initially, it was intended to choose 3 wavelets having lowest values of r_v from each task-related group for further processing and six wavelets associated with 6 smallest maximal values chosen among 3 tasks calculated as: max$\{r_v(logicgame)$,$r_v(meditation)$,$r_v(musicvideo)$ $\}$. However, it can be seen in Table 1 that the coif12 wavelet in case of meditation scenario was associated with the value of r_v 73 times smaller than the second-best candidate wavelet - db18, what means that probably the r_v value obtained for coif12 is an outlier observation. For other scenarios the ratio calculated by dividing r_v associated with second best and the best wavelet does not exceed 2.

Table 1. Values of r_v obtained for the best 10 wavelets chosen for each group are shown in first three columns (logic game, meditation and music video), the max. val. column contains results chosen with respect to the maximum value criterion (a smaller value is better). Bold values denote cases chosen for further, more precise training.

ID	logic game		meditation		music video		max. val.	
	name	r_v	name	r_v	name	r_v	name	r_v
1	**sym7**	**0.233**	**coif12**	**0.005**	**coif12**	**0.584**	**coif1**	**1.132**
2	**sym19**	**0.313**	**db18**	**0.365**	**sym13**	**0.869**	**sym3**	**1.145**
3	**sym8**	**0.336**	**db17**	**0.460**	**db18**	**0.872**	**db3**	**1.154**
4	sym16	0.484	**sym20**	**0.477**	**sym11**	**0.944**	**sym4**	**1.160**
5	sym18	0.486	db20	0.492	sym9	0.989	**sym2**	**1.168**
6	sym12	0.546	db28	0.548	sym5	0.998	**coif3**	**1.170**
7	db2	0.557	db10	0.556	sym4	1.004	coif8	1.171
8	sym10	0.580	db8	0.617	sym2	1.023	db15	1.176
9	sym15	0.593	db11	0.642	sym17	1.026	coif7	1.176
10	db14	0.606	db30	0.642	db3	1.030	db19	1.178

Therefore, in every group containing coif3 wavelets, the fourth candidate wavelet was chosen for further tests in the case if coif12 turned out to be an outlier and value of r_v associated with meditation scenario and coif12 wavelet was a result of lack of convergence of a learning algorithm. For the second stage of the experiment, 15 wavelets were chosen. Details concerning the learning process are shown in Tables 2 and 3.

Table 2. Detailed outcomes of learning process for each wavelet chosen for the second stage of the experiment (denotations as before).

ID	logic game			meditation		
	name	no. epochs	loss	name	no. epochs	loss
1	**sym7**	**56**	**0.00092**	**coif12**	**75**	$149 \cdot 10^3$
2	**sym19**	**45**	**0.00089**	**db18**	**68**	**0.00081**
3	**sym8**	**52**	**0.00089**	**db17**	**46**	**0.00096**
4	sym16	43	0.00090	**sym20**	**75**	**0.00175**
5	sym18	75	0.00485	db20	70	0.00091
6	sym12	50	0.00097	db28	75	0.00189
7	db2	16	0.00099	db10	75	0.00106
8	sym10	44	0.00094	db8	27	0.00087
9	sym15	47	0.00090	db11	57	0.00084
10	db14	57	0.00097	db30	75	0.00151

Table 3. Detailed outcomes of learning process for each wavelet chosen for the second stage of the experiment (denotations as before).

ID	music video			min-max criterion		
	name	no. epochs	loss	name	no. epochs	loss
1	**coif12**	**75**	$149 \cdot 10^3$	**coif1**	**12**	**0.00095**
2	**sym13**	**47**	**0.00087**	**sym3**	**15**	**0.00098**
3	**db18**	**68**	**0.00081**	**db3**	**25**	**0.00086**
4	**sym11**	**66**	**0.00099**	**sym4**	**23**	**0.00086**
5	sym9	75	0.01762	**sym2**	**31**	**0.00099**
6	sym5	31	0.00096	**coif3**	**61**	**0.00098**
7	sym4	23	0.00086	coif8	75	0.00187
8	sym2	31	0.00099	db15	56	0.00096
9	sym17	38	0.00079	coif7	34	0.00093
10	db3	25	0.00086	db19	56	0.00086

Tables 2 and 3 present the outcomes of the learning process, i.e. the number of iterations of the learning algorithm (epochs), and the final loss value. The course of the learning process differed with respect to the wavelet used for the preparation of input parameters. The number of epochs required to

Table 4. The final obtained value of loss function, number of iterations of training algorithm and r_v values associated with each class after the second stage of training process.

ID	wavelet name	loss	no. epochs	task r_v			max val. r_v
				logic game	meditation	music video	
1	**coif1**	0.0000099	129	0.777	1.136	**1.017**	**1.136**
2	**coif3**	0.0000189	502	1.100	**0.629**	1.188	1.188
3	**coif12**	0.0002299	1000	1.107	0.685	1.157	**1.157**
4	**db3**	0.0000098	253	**0.712**	1.177	**1.061**	1.177
5	**db17**	0.0000297	598	1.097	**0.630**	1.211	1.211
6	db18	0.0000344	666	1.105	0.653	1.194	1.194
7	**sym2**	0.0000099	127	**0.630**	1.196	1.132	1.196
8	**sym3**	0.0000099	90	**0.670**	1.199	**1.086**	1.199
9	sym4	0.0000099	316	1.032	0.735	1.162	1.162
10	sym7	0.0000230	373	1.052	0.670	1.208	1.208
11	sym8	0.0000180	796	1.106	0.653	1.170	1.170
12	**sym11**	0.0000308	382	1.117	**0.626**	1.204	1.204
13	**sym13**	0.0000325	271	1.097	0.713	1.155	**1.155**
14	sym19	0.0000315	408	1.068	0.631	1.229	1.229
15	sym20	0.0000342	167	1.110	0.639	1.204	1.204

meet one of the stop conditions was in range from 12 up to the maximal allowed value in the first stage of the experiment, i.e. 75. Also in this case the coif12 wavelet may also be identified as the one associated with the significantly larger training error than other wavelets, what may identify a problem with convergence. The detailed number of epochs needed for reaching one of the possible stop conditions, together with the final obtained value of loss function and r_v associated with parameters obtained from the trained encoder are presented in Table 4.

The number of epochs differed from relatively small number of 90 epochs for the sym3 wavelet up to maximum allowed number of 1000 epochs observed in the case of the coif12 wavelet what further confirms the hypothesis of problems with convergence being the cause of a small value of r_v observed in the outcomes of the first stage of the experiment. Wavelets marked in bold font were found to be one of the three wavelets providing the lowest value of r_v associated with each class.

Table 5. Mean distance between vectors associated with clusters related to the tasks performed by subjects, lowest estimated value of distance is denoted as distance l, the highest estimated value as distance r.

scenario	logic game - music video		meditation - music video		meditation - logic game	
	distance l	distance r	distance l	distance r	distance l	distance r
coif12	0.0099	**0.0914**	0.2051	0.3552	0.1952	0.3433
coif1	0.0346	0.1172	0.2620	0.4171	0.2267	0.3763
coif3	0.0113	0.0928	**0.2812**	**0.4319**	**0.2733**	**0.4205**
db17	0.0119	0.0952	0.2328	0.3830	0.2213	0.3670
db18	0.0342	0.1182	0.2081	0.3576	0.1740	0.3190
db3	0.0314	0.1257	0.2138	0.3626	0.1734	0.3154
sym11	0.0271	0.1130	0.2191	0.3675	0.1937	0.3369
sym13	0.0265	**0.1618**	**0.1614**	**0.3016**	**0.1378**	**0.2674**
sym19	0.0107	0.0945	0.2574	0.4091	0.2476	0.3933
sym20	0.0281	0.1196	0.2024	0.3508	0.1739	0.3172
sym2	**0.0367**	0.1206	0.1906	0.3390	0.1540	0.2989
sym3	0.0318	0.1283	0.1997	0.3498	0.1561	0.3005
sym4	0.0272	0.1094	0.2617	0.4159	0.2346	0.3833
sym7	0.0218	0.1127	0.2563	0.4071	0.2361	0.3794
sym8	0.0228	0.1054	0.2549	0.4037	0.2412	0.3856
best 3, meditation	0.0113	0.0928	0.2812	0.4319	0.2733	0.4205
best 3, logic game	0.0357	0.1221	0.1932	0.3419	0.1553	0.2999
best 3, music video	0.0318	0.1274	0.2051	0.3549	0.1626	0.3065
best 3, min-max	0.0258	0.1586	0.1643	0.3049	0.1418	0.2724
all	0.0118	0.0942	0.2794	0.4300	0.2714	0.4184
all, PCA-reduced	**0.0096**	0.0944	0.2795	0.4299	0.2712	0.4184
min	0.0096	0.0914	0.1614	0.3016	0.1378	0.2674
max	0.0367	0.1618	0.2812	0.4319	0.2733	0.4205

The next step of analysis was calculation of the distance between task-related clusters normalized with respect to the estimated variance of data sets. Calculation was performed according to Equation 7. Minimum and maximum values of the estimated normalized distance are shown in Table 5. Values from Table 5 are depicted in graphical form in Figure 6.

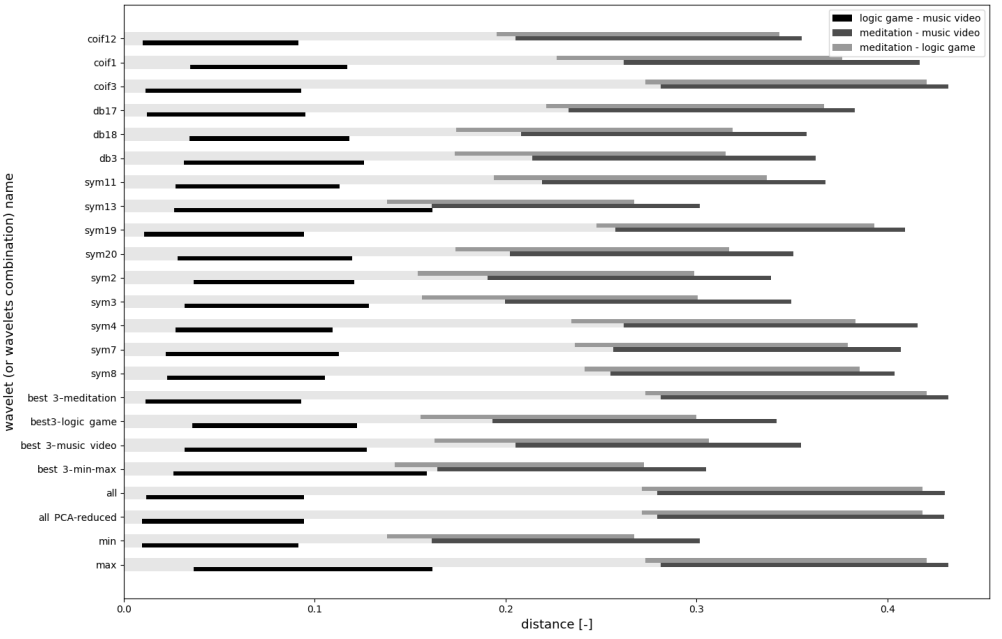
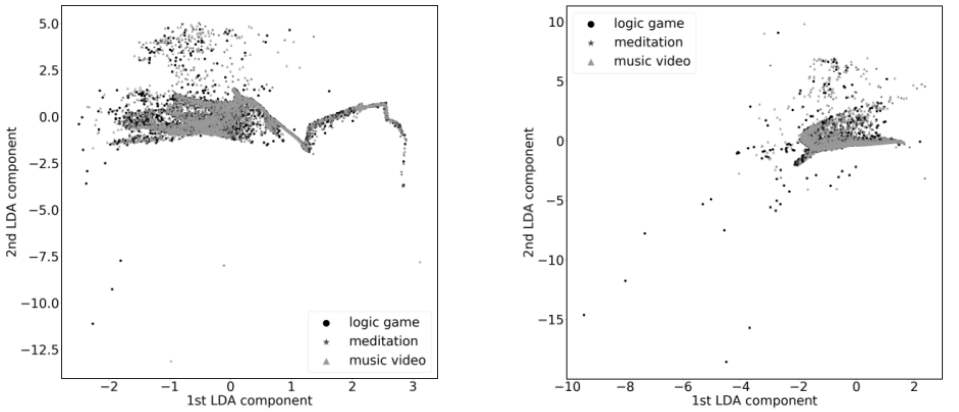


Figure 6. Graphical visualization of the data from Table 5

The estimate of the lowest value of the normalized distance between groups of vectors associated with each task varies from 0.0096 to 0.2674. The range for the upper border of the estimate interval is from 0.0367 to 0.4205. Calculation of the confidence interval borders associated with absolute distances and estimates of variances was performed with assumed level of confidence equal to 0.05. In the course of the experiment, also distances in spaces created by concatenation of multiple vectors were investigated Scenarios denoted as best 3 are acquired by merging three 100-parameter long vectors obtained from the three wavelets associated with the lowest values of r_v. Scenario denoted as all was performed for vectors obtained from merging vectors originating from all 15 wavelets each resulting vector was 1500-parameters long. Also, influence of dimensionality reduction performed with PCA on those 1500-parameters long vectors was investigated. In the case of PCA reduced vectors 100 most significant components were chosen to form reduced parameter vectors.

An analysis with the use of multiple wavelets did not improve performance in terms of the obtained normalized distance between clusters. However, if visualization of clusters is performed the placement of points representing vectors of parameters changes significantly. A linear discriminant analysis (LDA) was used for the purpose of data visualization and to make such a comparison possible. LDA is an algorithm which allows for obtaining a projection of set existing in a high-dimensional space onto a

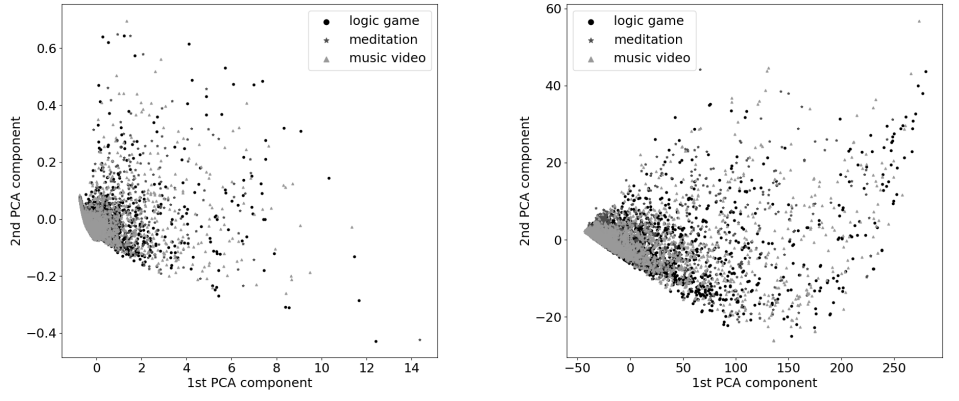
low-dimensional space which would maximize the separation between classes in the low-dimensional space. A low-dimensional projection may be then used for visualization of high-dimensional data with particular emphasis on separability of data classes. In case of visualization, the target space is two-dimensional. Visualizations prepared for two wavelets providing highest values for each of performed tasks are shown in Figures 7 and 8 with the use of LDA and PCA projection of the decision space onto a 2D plane respectively.



(a) sym2, a wavelet providing highest distance for music video task

(b) coif3, a wavelet providing highest distance for meditation and logic game tasks

Figure 7. Examples of visualizations of the single wavelet related data sets obtained with the use of LDA



(a) sym2, a wavelet providing highest distance for music video task

(b) coif3, a wavelet providing highest distance for meditation and logic game tasks

Figure 8. Examples of visualizations of the single wavelet related data sets obtained with the use of PCA

Projections of vectors on a 2D plane, calculated in the course of the experiment are characterized by irregular shape. In both cases, an area containing a dense cloud of points related to all three groups can be identified. Also, both of them contain sparse areas of outliers. Most of those outliers are associated with one group, in the case of both wavelets outliers tend to be associated with music video class. The PCA-based visualization shows a more compact structure of the cluster and distinction between classes is less visible. Therefore, this is a premise that there are meaningful differences between classes (as they are visible on the LDA-based visualization), but there are no obvious separate

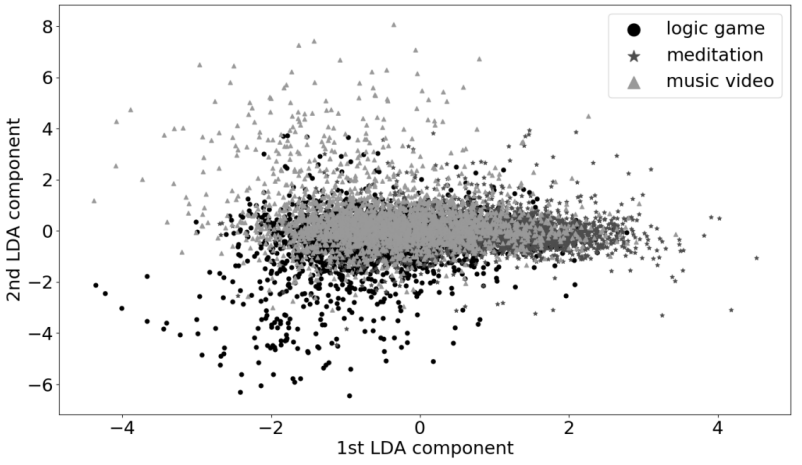


Figure 9. An example of the data sets obtained with the use of the LDA for the set associated with merged parameters obtained from all 15 wavelets from the 2nd stage of the study

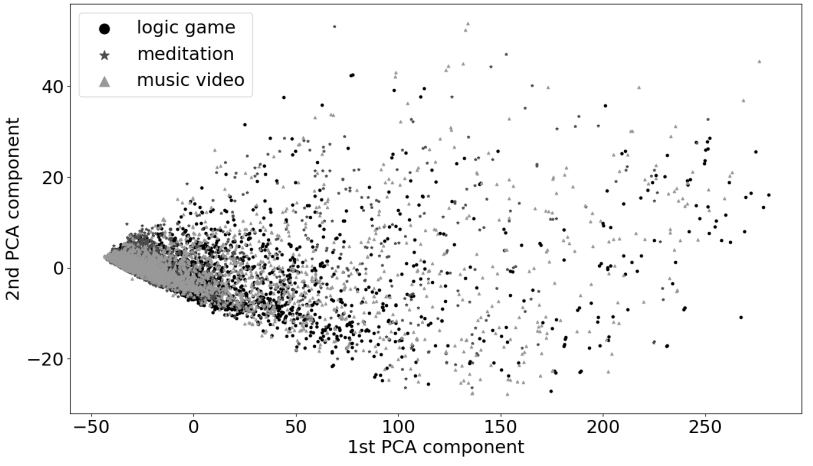


Figure 10. An example of the data sets obtained with the use of the PCA for the set associated with merged parameters obtained from all 15 wavelets from the 2nd stage of the study

clusters found for each classes, which holds true for both LDA and PCA based visualizations. A similar observation can be made if visualizations calculated from 1500 parameters-long vectors obtained by joining vectors originating from all 15 wavelets are considered. They are shown in Figure 9 and 10.

The pattern of outliers in the case of visualization from Figure 9 is more evident than in the case of visualizations shown in Figure 7. In the case of visualization depicted in Figure 9, outliers are placed above the main cluster position near the zero value of 2nd LDA component axis. This fact may be a consequence of the way the LDA algorithm works, as it finds the way of projecting the decision space onto a visualization plane, which allows for the best possible separation on the target plane. In Figure 10, a visualization is based on two most significant PCA components, which preserve most the of variance present in the high-dimensional source dataset. As it preserves Euclidean distances between training instances, this visualization allows depicting correctly geometric relations between points, whereas LDA accents differences between entities positioned in the decision space. In the LDA-based depiction of training examples, the part of plane with negative values of the second LDA component, mostly logic game related outliers can be found. Points associated with meditation tend to be placed on the right part of the center cluster. This observation leads to the conclusion that there are task-specific vectors occurring only during the execution of this task, however, they do not occur continuously, and they are rather rare when compared to more frequently observed vectors placed near to the center of the visualization plane. This relation is especially visible in the LDA projection of data, positioning of training examples on PCA-based visualization also allows to see groups of points belonging majorly to one class, but they are less prominent if compared to the LDA-based depiction.

5. Discussion

We found in our research that time frames extracted from the recorded EEG signals associated with performing 3 selected tasks may be clustered by the unsupervised machine learning algorithm, yielding interesting results. The distance differs for each of the investigated pairs of tasks. For the logic game and for the music video the distance range calculated by obtaining a confidence interval is smaller than for the two other pairs of activities. This can possibly make it harder to discern those two activities with the use of the machine learning-based classifiers. The distance calculated from the data also depends on the type of the wavelet type used, which may be a guideline for the choice of a wavelet suitable for distinguishing between two activities. We also found that an outlier pattern may be used for the differentiation between tasks, especially if multiple wavelets are used.

If compared to other research studies in the EEG-based analysis field, a work of Faust et al. from 2015 should be mentioned in the context of the choice of a suitable mother wavelet for various tasks [25]. In their review paper, a group of wavelets suitable for detection of epilepsy was presented. The db4 wavelet was found to be the most effective in the investigated research, however other wavelet types return satisfactory results. For the tasks investigated in our paper, the Daubechies wavelet of the similar order - db3 wavelet was found to provide a moderate separation while compared to the best performing wavelet coif3. Our paper gives also some insight into the process of training of a machine learning algorithm for the three selected wavelet types: Daubechies, Coiflets and Symlets

of various orders. Based on the already mentioned work by Faust et al., wavelets of the Symlet and Coiflet families occurred in two works only while compared with the Daubechies type (26 times).

There are also other examples of works in the literature associated with identification of emotional states (Aguinaga et al., 2015, [26]), classification of epileptic seizures (Ocak, 2008, [27]) or feature extraction from the EEG signals obtained from the participants performing various tasks (Amin et al., 2015, [28]). Amins et al. study is the most similar one to ours in terms of assumptions and conditions of the EEG signals acquisition. However, all mentioned works concentrate on assessing the performance of a DWT-based system employing one or more classifiers. In the case of works of Ocak [27] and Amin et al. [28] no direct analysis of data structure and separability of data frames or mental states observed in the course of experiments was performed. A visual assessment of separability was provided for the emotion-related mental state classification in the study of Aguinaga et al. [26]. The work of Ocak [27] provides an interesting way of optimization of a feature vector by selecting the best parameters with the use of the genetic algorithm. In our approach we tried to acquire the best performing set of parameters by selecting an appropriate mother wavelet which allows for the most efficient training of an autoencoder neural network. This is an important feature in the context of applicability of the approach proposed for the unsupervised classification of a given task or a mental state.

Autoencoder neural networks are investigated in the context of the analysis of the EEG signals, but also as classification or data preprocessing methods. Examples of such studies are recent works of Li et al. [29], Jirayucharoensak et al. [30] and Tabar and Halici [31]. These works concentrate on testing the performance of the autoencoder networks compared to other machine learning-based classification solutions, but they contain, however, no separability analyses. Meanwhile, our work provides insight into properties of the wavelets belonging to 3 selected families which influence the speed of training measured in epochs needed for acquiring the target value. The target value may be defined in terms of loss, distance between groups of vectors associated with tasks performed by participants or placement of outlier vectors in the feature space. The results obtained led to conclusions associated with the fact that outlier vectors associated with different tasks performed by participants tend to be placed in different areas of the decision space as it can be observed in Figure 9. Therefore, we can suppose that it is possible to define mental states associated with the tasks performed by participants in a similar manner to the mental states defined in the work by Aguinaga et al. [26].

6. Conclusions and future work

Results presented in our work allow for drawing multiple conclusions. In the first stage of the experiment wavelets from the Symlet family were found most prominently in the group of ten best wavelets defined with regards of obtaining a low value of r_v for the logic game task. Wavelets from Daubechies family were more likely to be found in the group associated with a lower value of r_v, if meditation task is considered. This means that the type of the wavelet chosen for the feature extraction should depend on the task classified. Low values of r_v may be a consequence of a problematic convergence of the training algorithm which could be observed in the case of coif12 wavelet behavior during the first stage of the experiment. The low variance is probably a consequence of the presence of outliers which

occurred due to the lack of convergence of the training algorithm, what resulted in a high value of loss shown in Table 3. The speed of convergence of the training process measured in terms of epochs needed to reach a target loss value is also dependent on the choice of the mother wavelet.

A single wavelet may lead to detection of noticeable differences in distances between sets associated with task-related clusters of points in the encoding space. However, if Figures 7 and 9 are considered, it can be observed that the analysis employing all preselected wavelets allows for obtaining an interesting pattern of points. This pattern also includes distances between centroids of the sets which are lying within the confidence interval with the lower boundary greater than 0.4, however knowing the distribution of the outlier points seems also to be interesting. Outliers of the logic game group are in the topmost part of Figure 9, whereas in the lower part outliers associated with music video are positioned. On the right side of this figure a dominance with meditation class members can be seen. This observation may enable a hypothesis forming that participants during the time they performed assigned tasks were involved in one basic mental state and occasionally during performing such a task they switched to the outlier state which is specific to the performed task, only. Therefore, it can be identified as an outlier in a specific part of the LDA-based projection of decision space in a plane depicted in Figure 9.

More research is needed to confirm above hypotheses, however, it may serve as a guideline for the design of a system for the classification of mental state of test participants. Such a system should take into account a situation where data frames associated with a given mental state are scarce in the input signal and they do not occur constantly, but in certain time intervals. Future research efforts should investigate possibilities of constructing such a system and assessing its performance.

Acknowledgments:

The project was funded by the National Science Centre on the basis of the decision number DEC-2014/15/B/ST7/04724. It was also partially supported by Statutory Funds of Electronics, Telecommunications and Informatics Faculty, Gdansk University of Technology.

References

[1] Bashivan P, Rish I, Yeasin M, Codella N. Learning Representations from EEG with Deep Recurrent-Convolutional Neural Networks. 2015. pp. 1–15. doi:10.1080/03610928808829796. 1511.06448, URL http://arxiv.org/abs/1511.06448.

[2] Kumar N, Kumar J. Measurement of Cognitive Load in HCI Systems Using EEG Power Spectrum: An Experimental Study. *Procedia Computer Science*, 2016. **84**:70–78. doi:10.1016/j.procs.2016.04.068. URL http://dx.doi.org/10.1016/j.procs.2016.04.068.

[3] Schirrmeister RT, Springenberg JT, Fiederer LDJ, Glasstetter M, Eggensperger K, Tangermann M, Hutter F, Burgard W, Ball T. Deep learning with convolutional neural networks for EEG decoding and visualization. *Human Brain Mapping*, 2017. **38**(11):5391–5420. doi:10.1002/hbm.23730. 1703.05051.

[4] Yildirm N, Varol A. A research on estimation of emotion using EEG signals and brain computer interfaces. *2nd International Conference on Computer Science and Engineering, UBMK 2017*, 2017. pp. 1132–1136. doi:10.1109/UBMK.2017.8093523.

[5] Zhang X, Yao L, Zhang D, Wang X, Sheng QZ, Gu T. Multi-Person Brain Activity Recognition via Comprehensive EEG Signal Analysis. 2017. doi:10.475/123. 1709.09077, URL http://arxiv.org/abs/1709.09077.

[6] Lotte F, Congedo M, Lécuyer A, Lamarche F, Arnaldi B. A review of classification algorithms for EEG-based brain-computer interfaces. *Journal of Neural Engineering*, 2007. **4**(2). doi:10.1088/1741-2560/4/2/R01.

[7] Nicolas-Alonso LF, Gomez-Gil J. Brain computer interfaces, a review. *Sensors*, 2012. **12**(2):1211–1279. doi:10.3390/s120201211. s120201211.

[8] Arnau-González P, Arevalillo-Herráez M, Ramzan N. Fusing highly dimensional energy and connectivity features to identify affective states from EEG signals. *Neurocomputing*, 2017. **244**:81–89. doi:10.1016/j.neucom.2017.03.027.

[9] Millán JDR, Franzé M, Mouriño J, Cincotti F, Babiloni F. Relevant EEG features for the classification of spontaneous motor-related tasks. *Biological Cybernetics*, 2002. **86**(2):89–95. doi:10.1007/s004220100282.

[10] Rosales R, Castañón-Puga M, Lara-Rosano F, Evans R, Osuna-Millan N, Flores-Ortiz M. Modelling the interruption on HCI using BDI agents with the fuzzy perceptions approach: An interactive museum case study in Mexico. *Applied Sciences (Switzerland)*, 2017. **7**(8):1–18. doi:10.3390/app7080832.

[11] Cao X, Wipf D, Wen F, Duan G, Sun J. A practical transfer learning algorithm for face verification. *Proceedings of the IEEE International Conference on Computer Vision*, 2013. pp. 3208–3215. doi:10.1109/ICCV.2013.398.

[12] Ng Hw, Nguyen VD, Vonikakis V, Winkler S. Deep Learning for Emotion Recognition on Small Datasets Using Transfer Learning. *ICMI '15 Proceedings of the 2015 ACM on International Conference on Multimodal Interaction*, 2015. pp. 443–449. doi:10.1145/2818346.2830593. URL http://dl.acm.org/citation.cfm?id=2830593.

[13] Noroozi M, Favaro P. Unsupervised learning of visual representations by solving jigsaw puzzles. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2016. **9910 LNCS**:69–84. doi:10.1007/978-3-319-46466-4_5. 1603.09246.

[14] Wang Y, Yao H, Zhao S. Auto-encoder based dimensionality reduction. *Neurocomputing*, 2016. **184**:232 – 242. doi:https://doi.org/10.1016/j.neucom.2015.08.104. RoLoD: Robust Local Descriptors for Computer Vision 2014, URL http://www.sciencedirect.com/science/article/pii/S0925231215017671.

[15] Oliphant TE. Python for scientific computing. *Computing in Science and Engineering*, 2007. **9**(3):10–20. doi:10.1109/MCSE.2007.58. 1507.04592.

[16] PyWavelets. Internet website of the Python library. http://pywavelets.readthedocs.io. Accessed: 2017-12-19.

[17] Abadi M, Agarwal A, Barham P, Brvedo E, Chen Z, Citro C, Corrado G, Davis A, Dean J. TensorFlow: Large-scale machine learning on heterogeneous systems. 2015. doi:10.1038/nn.3331. arXiv:1603.04467v2, URL http://download.tensorflow.org/paper/whitepaper2015.pdf.

[18] Al-Qazzaz N, Hamid Bin Mohd Ali S, Ahmad S, Islam M, Escudero J. Selection of Mother Wavelet Functions for Multi-Channel EEG Signal Analysis during a Working Memory Task. *Sensors*, 2015. **15**(11):29015–29035. doi:10.3390/s151129015. URL http://www.mdpi.com/1424-8220/15/11/29015/htm.

[19] Sanei S, Chambers JA. EEG Signal Processing, volume 1. 2007. ISBN 9780470025819. doi:10.1002/9780470511923.

[20] Jayalakshmi T, Santhakumaran A. Statistical Normalization and Backpropagation for Classification. *International Journal of Computer Theory and Engineering*, 2011. **3**(1):89–93. doi:10.7763/IJCTE.2011.V3.288. URL http://www.ijcte.org/papers/288-L052.pdf.

[21] Gluege S, Boeck R, Wendemuth A. Auto-Encoder Pre-Training of Segmented-Memory Recurrent Neural Networks. *European Symposium on Artificial Neural Networks*, 2013. (April):29–34.

[22] Sarroff AM, Casey M. Musical Audio Synthesis Using Autoencoding Neural Nets. *Proceedings of the International Computer Music Conference*, 2014. **1**(September):14–20.

[23] Wang W, Huang Y, Wang Y, Wang L. Generalized autoencoder: A neural network framework for dimensionality reduction. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, 2014. pp. 496–503. doi:10.1109/CVPRW.2014.79.

[24] Helal MA, Eldawlatly S, Taher M. Using Autoencoders for Feature Enhancement in Motor Imagery Brain-Computer Interfaces. *2017 13Th Iasted International Conference on Biomedical Engineering (Biomed)*, 2017. pp. 89–93. doi:10.2316/P.2017.852-052.

[25] Faust O, Acharya UR, Adeli H, Adeli A. Wavelet-based EEG processing for computer-aided seizure detection and epilepsy diagnosis. *Seizure*, 2015. **26**:56–64. doi:10.1016/j.seizure.2015.01.012. URL http://dx.doi.org/10.1016/j.seizure.2015.01.012.

[26] Aguinaga AR, Lopez Ramirez MA, Baltazar Flores MdR. Classification model of arousal and valence mental states by EEG signals analysis and Brodmann correlations. 2015. **6**(6):230–238.

[27] Ocak H. Optimal classification of epileptic seizures in EEG using wavelet analysis and genetic algorithm. *Signal Processing*, 2008. **88**(7):1858–1867. doi:10.1016/j.sigpro.2008.01.026.

[28] Amin HU, Malik AS, Ahmad RF, Badruddin N, Kamel N, Hussain M, Chooi WT. Feature extraction and classification for EEG signals using wavelet transform and machine learning techniques. *Australasian Physical & Engineering Sciences in Medicine*, 2015. **38**(1):139–149. doi:10.1007/s13246-015-0333-x. URL http://link.springer.com/10.1007/s13246-015-0333-x.

[29] Li J, Struzik Z, Zhang L, Cichocki A. Feature learning from incomplete EEG with denoising autoencoder. *Neurocomputing*, 2015. **165**:23–31. doi:10.1016/j.neucom.2014.08.092. 1410.0818, URL http://dx.doi.org/10.1016/j.neucom.2014.08.092.

[30] Jirayucharoensak S, Pan-Ngum S, Israsena P, Jirayucharoensak S, Pan-Ngum S, Israsena P. EEG-Based Emotion Recognition Using Deep Learning Network with Principal Component Based Covariate Shift Adaptation. 2014. doi:10.1155/2014/627892,10.1155/2014/627892.

[31] Tabar YR, Halici U. A novel deep learning approach for classification of EEG motor imagery signals. *Journal of Neural Engineering*, 2017. **14**(1). doi:10.1088/1741-2560/14/1/016003.