

An Unsupervised Feature Extraction Method based on CLSTM-AE for Accurate P300 Classification in Brain-Computer Interface Systems

Ramin Afrah (MSc)¹, Zahra Amini (PhD)^{2*}, Rahele Kafieh (PhD)²

¹School of Advanced Technologies in Medicine, Isfahan University of Medical Sciences, Isfahan, Iran

²Medical Image and Signal Processing Research Center, School of Advanced Technologies in Medicine, Isfahan University of Medical Sciences, Isfahan, Iran

ABSTRACT

Background: The P300 signal, an endogenous component of event-related potentials, is extracted from an electroencephalography signal and employed in Brain-computer Interface (BCI) devices.

Objective: The current study aimed to address challenges in extracting useful features from P300 components and detecting P300 through a hybrid unsupervised manner based on Convolutional Neural Network (CNN) and Long Short-term Memory (LSTM).

Material and Methods: In this cross-sectional study, CNN as a useful method for the P300 classification task emphasizes spatial characteristics of data. However, CNN and LSTM networks are combined to modify the classification system by extracting both spatial and temporal features. Then, the CNN-LSTM network was trained in an unsupervised learning method based on an autoencoder to improve Signal-to-noise Ratio (SNR) by extracting main components from latent space. To deal with imbalanced data, an Adaptive Synthetic Sampling Approach (ADASYN) is used and augmented without any duplication.

Results: The trained model, tested on the BCI competition III dataset, including two normal subjects, with an accuracy of 95% and 94% for subjects A and B in P300 detection, respectively.

Conclusion: CNN-LSTM, was embedded into an autoencoder and introduced to simultaneously extract spatial and temporal features and manage the computational complexity of the method. Further, ADASYN as an augmentation method was proposed to deal with the imbalanced nature of data, which not only maintained feature space as before but also preserved anatomical features of P300. High-quality results highlight the suitable efficiency of the proposed method.

Keywords

Brain-Computer Interfaces; P300; Deep Learning; Classification

Introduction

In healthy or disabled individuals, Brain-Computer Interface (BCI) systems can send instructions, operated by brain activity, to external gadgets without the use of a neuro-muscular system [1, 2]. Different types of BCI systems are categorized according to the brain activity

*Corresponding author:
Zahra Amini

Medical Image and Signal Processing Research Center, School of Advanced Technologies in Medicine, Isfahan University of Medical Sciences, Isfahan, Iran
E-mail: zahraamini64@yahoo.com.au

Received: 26 July 2022
Accepted: 23 February 2023

measurement method, such as Near Infrared Spectroscopy (NIRS) [3], Electroencephalography (EEG) [4], and Magnetoencephalography (MEG) [5]. EEG-based BCIs [6] are considered the most popular method due to some advantages, such as ease of use, non-invasiveness, and low cost. Some studies have been conducted on P300-based BCIs, a well-known type of EEG-based BCIs, to modify the performance [7].

Event-related Potentials (ERPs) are a type of EEG signal, generated after specific stimulation, and P300 as one of the most important components of ERP signals, is a positive deflection elicited approximately 300-1000 ms after auditory, somatosensory, or visual stimulation [6, 8, 9]. Anatomically, P300 is more elicited over the midline scalp, and its magnitude increases from the frontal to the parietal lobe [10].

Steps in setting up a P300-based BCI system include user tasks, EEG signal recording, signal pre-processing, feature extraction and translation, signal classification, and feedback to the user interface [11]. The user task is critical for the occurrence of the P300 signal, and a well-designed user task not only provides a strong P300 signal response with a short delay duration but also protects the subject against eye fatigue [12]. After designing a suitable task, EEG signals are recorded, which required preprocessing procedures, such as noise reduction and artifact removal. In the feature extraction stage, the most useful and meaningful features are extracted from the data, and the dimensionality reduction technique is performed if needed. Finally, a proper classifier is used to elicit the P300 wave [13] in the signal classification step based on the obtained features. The created command must then be returned to the user. Typically, criteria, such as accuracy and time of detection, are applied to assess the overall performance of a P300-based BCI system [14].

Farwell and Donchin introduced a well-known P300-based BCI [15], which produces

P300 based on the oddball paradigm. They also generated a speller matrix composed of a 6×6 matrix of symbols. Each time, a row or column is randomly intensified for a short time until each column and row has been intensified once. The user selects the target stimuli by a specific symbol as well as counting the flashing of the row and column that intersect at that desired character [16].

The most important obstacles in P300 detection are considered variability and low SNR [17-19]. Differentiating P300 from other ERP components is a major goal of many studies in this field [20]. Some studies have focused on this field for P300 classification because of the noticeable performance of deep learning methods [21]. Accordingly, deep-learning approaches are firstly surveyed, and conventional methods are also reviewed in the P300 detection studies.

A deep belief network was utilized for P300 classification with an accuracy of 91% on a dataset of 9 subjects [22]. Kshirsagar et al. [23] collected data on 10 subjects with 8 channels of EEG signal to classify P300 in Devanagari script using deep learning methods. They used autoencoder and CNN with an accuracy of 95.82%. Mingfei Liu et al. [24] proposed a deep learning method based on batch normalization for P300 detection on the BCI competition III dataset; a total of 79.09% accuracy for subject B was the best score among acquired results. A deep learning method based on deep belief networks was applied for P300 detection again on the BCI competition III dataset with an accuracy of 86.4% for subject B as the best result was reported [25]. A simple CNN was used, and an accuracy of 86.4% was achieved for the classification of P300 on the BCI competition III dataset [26].

Data were collected from the Fz, Cz, and Pz channels of 15 subjects, and different pipelines were examined for feature extraction plus classification, in which Bayesian linear discriminant analysis with 72.13% accuracy showed the best performance [27]. A method based on

regularized group sparse discriminant analysis in [28] on Pz, Cz, and P8 channels of the BCI competition III dataset was implemented, with an accuracy of 90.00%. A combination of optimization and pattern recognition methods was used for P300 detection, tested on five datasets, with an accuracy of 96.70% as the best performance [29]. The decision tree was used as a classifier on an open dataset from an expert psychologist association, with an accuracy of 99.68% obtained [30].

This study introduces a method to extract spatial and temporal features as two main types of P300 component features by applying the Convolution Long-Short Term Memory AutoEncoder (CLSTM-AE), which is a combination of CNN for spatial and LSTM for temporal feature extraction. Also, an auto-encoder was used as an unsupervised learning approach to reduce the time complexity and information redundancy of the method.

Material and Methods

In this cross-sectional study, a deep learning method was proposed based on both CNN and long short-term memory to extract efficient spatial and temporal features for P300 detection. A new method of augmentation was also used to deal with the imbalanced nature of the P300 dataset based on the Adaptive Synthetic Sampling Approach (ADASYN). The accuracy, sensitivity, recall, and F1-Score were considered to evaluate the performance of the proposed models.

Dataset and pre-processing

In this cross-sectional study, a deep learning method was proposed based on both CNN and long short-term memory to extract efficient spatial and temporal features for P300 detection. A new method of augmentation was also used to deal with the imbalanced nature of the P300 dataset based on the Adaptive Synthetic Sampling Approach (ADASYN). The accuracy, sensitivity, recall, and F1-Score were considered to evaluate the performance of the

proposed models.

Dataset and pre-processing

The brain signals were recorded according to the BCI P300 speller system. The P300 speller is based on the so-called oddball paradigm, showing that uncommon expected inputs cause a positive deflection in the EEG after about 300 ms, called the P300 component. Farwell and Donchin [15] established a P300 speller based on this paradigm by developing a protocol, in which a subject is presented with a 6×6-character matrix, as displayed in Figure 1. For the spelling of a single character, each of the matrix's 12 rows and columns is then intensified in a random sequence (in the sequel, we refer to such a collection of 12 intensifications as a series). The participant was instructed to focus on the character they prefer to spell, in which an EEG P300-evoked potential appears in response to the intensification of a row or column containing the desired character. This series of intensifications is repeated 15 times for each character to cause the spelling method more reliable. The dataset for this competition, which is available on the competition homepage (<https://www.bbc.de/competition/iii/>), was collected from two distinct subjects.

Before digitalization at 240 Hz, signals were bandpass filtered from 0.1 to 60 Hz [15]. The dataset was described in further detail in the



Figure 1: P300 Speller Matrix

BCI competition [30]. The classification of problems addressed is as follows: After stimulation and recording a 64-channel EEG signal for each subject, we wish to predict whether or not such a signal has a P300 component. Therefore, from the raw signal, a period of 1000 ms after stimulation was extracted. Most informative channels for each subject were then selected, including seven channels (Fz, Cz, Pz, C3, C4, PO7, and PO8) for subject A, and eight channels (Cz, C2, C3, T8, FC4, F2, F4, and F8) for subject B. According to the structure of the spelling matrix, only one-sixth of the recorded signals contain P300 (one row and one column from all 12 rows and columns), resulting in severely imbalanced sample sizes of P300 and non-P300 (nP300) data. In this case, the binary classification is prone to biasing towards nP300 samples.

Different types of augmentation methods were evaluated to address severely imbalanced sample sizes of P300 and non-P300 (nP300) data; firstly, random oversampling, as a simple method was implemented, the nature of this procedure is rather random. Although this method preserves the P300 structure and feature space, it merely copies the samples of the preliminary dataset, resulting in overfitting (Figure 2-I). To overcome this problem, other augmentation methods were applied based on synthetic sampling (SMOTE). As the basic one, random SMOTE (Figure 2-II) was applied. However, random SMOTE does not suffer from copying preliminary dataset samples, the output can be completely changed in each run because of its random nature. A version of SMOTE was used based on Support Vector Machine (SVM), whose random part is restricted to support vector samples (Figure 2-III) to control the random characteristics. In this method, augmentation is just applied on support vector samples with a substantial effect on P300 structure and distribution of features in comparison with primitive feature space. To overcome this challenge ADASYN was employed as a method,

focusing on low-density parts of the data to synthesize new samples. However, ADASYN not only does not duplicate the existing dataset samples but also preserves the feature space as before. Further, the outputs of this method do not change in each run. Details of ADASYN are described in depth in [31]. P300 formation and feature space distribution before and after ADASYN augmentation are shown in Figure 2-IV. The proposed ADASYN augmentation method has presented a better result compared to other augmentation methods, depicted in Figure 2, such as SVM SMOTE and random SMOTE.

The signal was sent through a bandpass Chebyshev filter with a cut-off frequency of 0.1-20 Hz to remove undesirable frequencies, and after all baselines were drifted. When the P300 component is extracted from the raw signal, each channel signal is a one-dimensional vector with a length of 240 samples, corresponding to the sampling frequency rate. All channels are cascaded along each other in a larger one-dimension signal to form each segment of data, as indicated in Figure 3a. The channels were transposed and then arranged behind each other as the frames of a video to prepare the data for the CLSTM-AE model (Figure 3b). Before augmentation, 20% of data from each class (750 P300 and 750 nP300) was selected as test data, and the remaining (1700 P300 and 12000 nP300) was augmented as train data in such a way that the two classes were balanced after augmentation. Only 20% of the training data was used for validation, while the remaining 80% was used for training of CLSTM-AE. The amount of data in each train, test, and validation group is mentioned in Table 1.

Method

The historical information is transmitted through a Recurrent Neural Network (RNN) procedure via the chain network structure. The RNN can learn long-term knowledge, but the growth of the vacant distance between the two

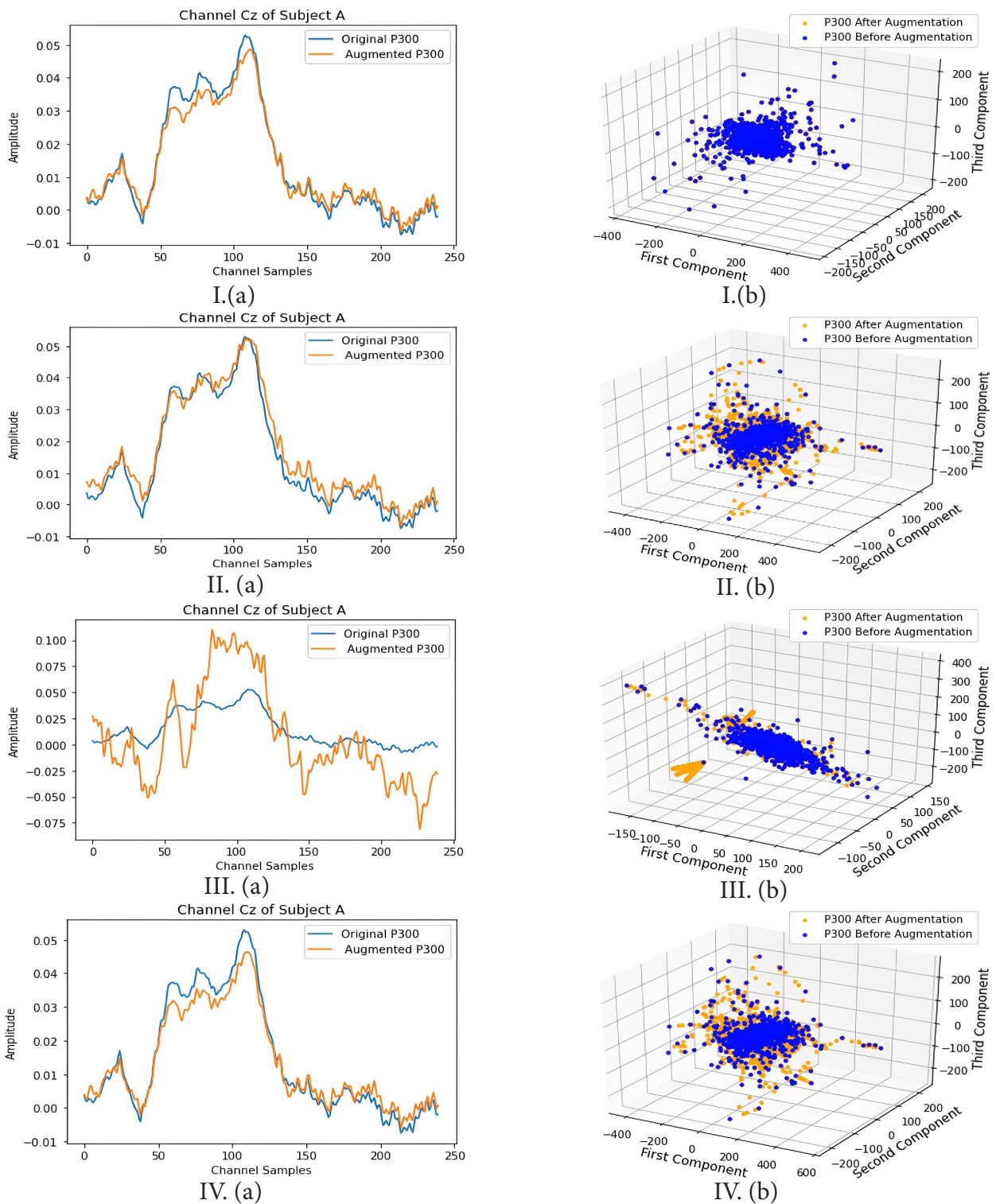


Figure 2: (a) A sample P300 signal structure before and after augmentation, (b) Feature space before and after augmentation in different methods: I. random oversampling, II. SMOTE, III. SMOTE based on SVM, IV. ADASYN. (SMOTE: Synthetic Sampling, SVM: Support Vector Machine, ADASYN: Adaptive Synthetic Sampling Approach)

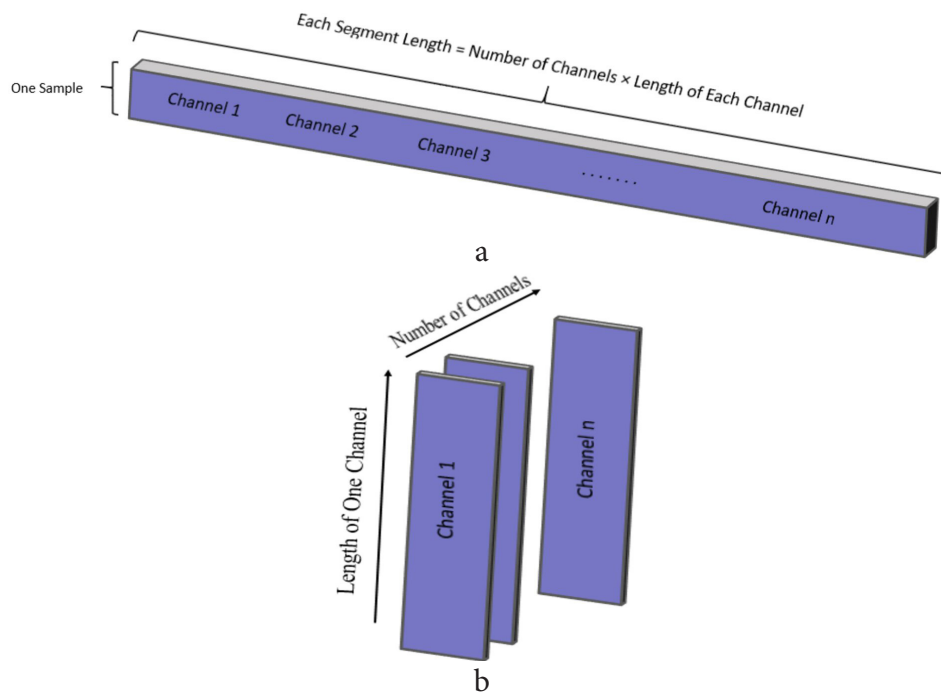


Figure 3: (a) One-dimension presentation of a segment of data. (b) Three-dimension presentation of a segment.

Table 1: Splits of the dataset for train, testing, and validation of the proposed model

Data samples	Number
train	19200
validation	4800
test	1500

memory units may cause memory deterioration and gradient vanishing. Hochreiter and Schmidhuber [32] attempted to solve the problem. The LSTM network was suggested for time series forecasting, considering the gate mechanism and controlling memory information using three separate gates' functions. The internal portion of the LSTM utilizes a nearly complete linkage, causing an issue of duplication of information. The LSTM solely takes into account temporal connections and disregards the spatial correlation within the data. Conversely, CNN is constructed by layering multiple convolutional and pooling

layers, which can efficiently extract spatial information from images without the need for temporal information [33]. However, LSTM uses matrix multiplication on its own, when it's combined with two dimensions CNN, the input is a matrix, not a series. Accordingly, the proposed CLSTM replaces the matrix multiplication with the convolution operation in the LSTM gates to simultaneously extract spatial and temporal features. It captures the benefits of both CNN and LSTM and not only can extract temporal features by LSTM but can also characterize spatial features by CNN. Figure 4 depicts the CLSTM unit in detail.

The enhanced CLSTM unit has three inputs for every gate: the memory information from the former unit, the output from the former unit, and the current time input. The upgraded CLSTM, which is made up of an input gate, a forget gate, an output gate, and a memory unit, can successfully learn time series information.

1) Gate of Forgetting: the gate of forgetting is responsible for selectively discarding

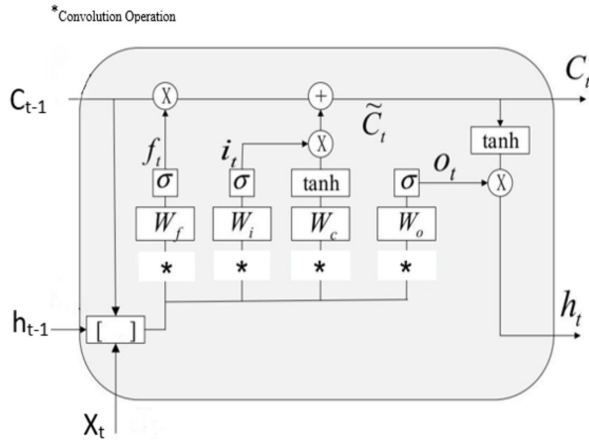


Figure 4: The Convolution Long-Short Term Memory (CLSTM) unit structure

unnecessary information from the memory unit, as follows:

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

where σ is the function of activation, \odot represents the operation of convolution, and x_t represents the input data at a specific point in time. c_{t-1} is the memory unit's prior information, and h_{t-1} is the previous CLSTM unit output. W_f and b_f are the forget gate's weight and bias, respectively [34].

2) The gate of input determines whether or not to incorporate fresh information into the memory unit, consisting of a pair of steps: determining the information, updating via the sigmoid layer, and generating substitute information via the hyperbolic tangent function.

$$i_t = \sigma(W_i \odot [c_{t-1}, h_{t-1}, x_t] + b_i) \quad (2)$$

$$\tilde{C}_t = \tanh(W_c \odot [c_{t-1}, h_{t-1}, x_t] + b_c) \quad (3)$$

Here i_t and \tilde{C}_t refer to the output of the input gate and the memory unit's substitute information, and hyperbolic tangent [34].

3) Memory Cell: It keeps the information up to date.

$$C_t = f_t \cdot C_{t-1} + i_t \cdot \tilde{C}_t \quad (4)$$

Which C_t is the memory unit's stored information [34].

4) Ultimately, based on the prior computa-

tion, the gate of output computes the last output.

$$o_t = \sigma(W_o \odot [c_{t-1}, h_{t-1}, x_t] + b_o) \quad (5)$$

$$h_t = o_t \cdot \tanh(C_t) \quad (6)$$

In this context o_t and h_t denote the current time outputs of the gate of output and CLSTM unit, respectively [34].

Proposed CLSTM-AE Network

Figure 5a exhibits the proposed CLSTM-AE network structure. The encoder section includes CLSTM, BatchNormalization, and pooling layers for encoding input data. De-CLSTM (Deconvolutional LSTM), deconvolutional, BatchNormalization, and upsampling layers are used in the decoder to decode the encoded features. CLSTM-AE includes additional information on past units. This enhances the ability to learn from previous memory. The structure of the network enables efficient feature learning for intricate signal-processing applications.

1) Utilizing Equations, the CLSTM unit utilizes a memory unit and three gates which are employed to extract input data features (1)-(6). CLSTM generates:

$$h_E = F(\sigma(W \odot [c_{t-1}, h_{t-1}, x_t] + b)) \quad (7)$$

Here, F demonstrates the computation of the CLSTM memory cell and three gates. The CLSTM-AE encoder takes into account the CNN network structure, allowing it to compress and extract information from the CLSTM output using convolution and pooling layers. The decoder utilizes a deconvolution layer and an up-sampling layer to improve the quality of data reconstruction [34].

2) Convolution Layer: the result of the convolution layer is:

$$C_i = f(\sum X \odot w_i + b_i) \quad (8)$$

The CLSTM output is represented as X , W_i denotes an activation function, and the rectified linear unit (ReLU), is utilized alongside the i^{th} convolution kernel [34].

3) The pooling layer reduces data dimen-

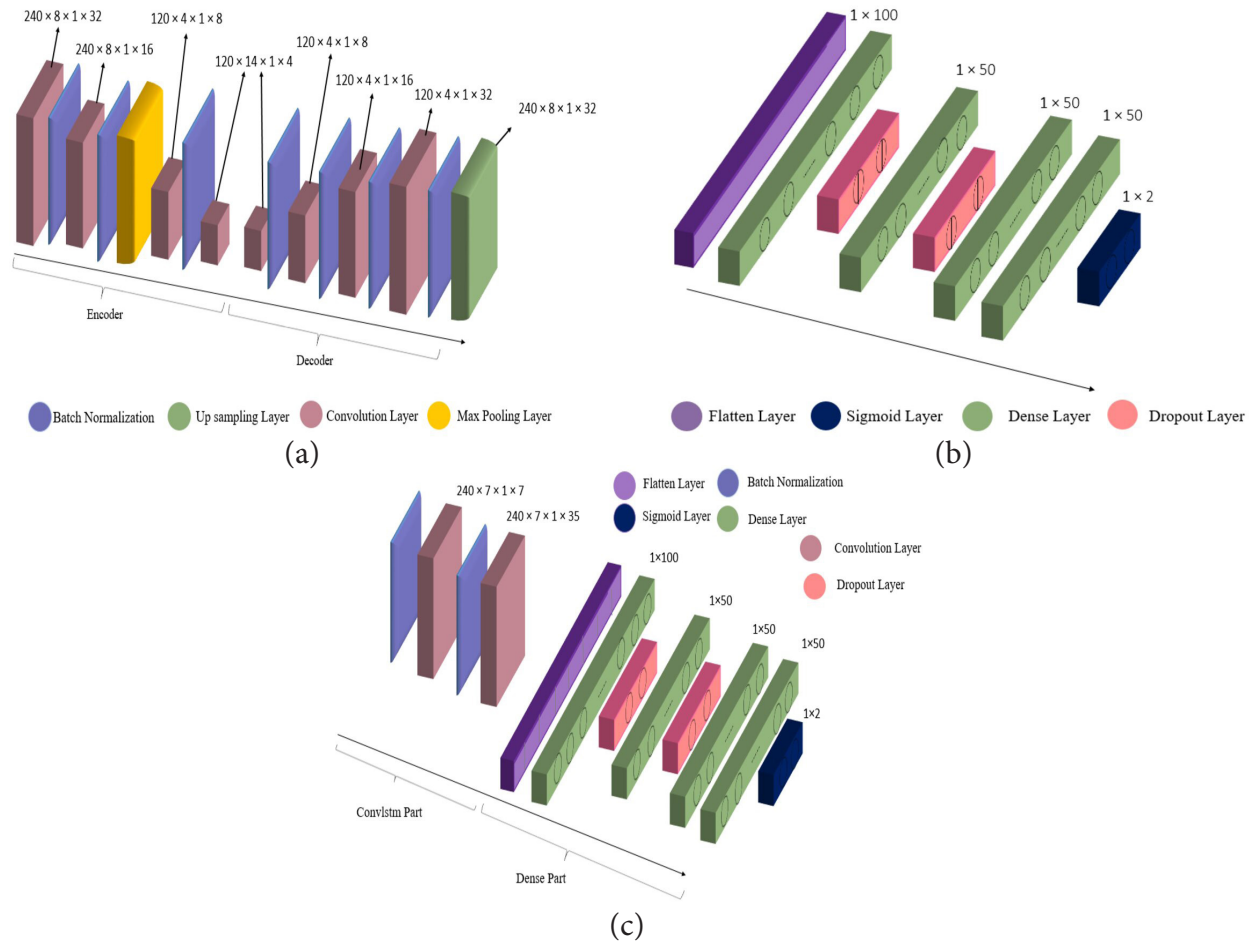


Figure 5: (a) Architecture of proposed CLSTM-AE, (b) Artificial neural network as a classifier, (c) CNN-LSTM classifier (CLSTM-AE: Convolution Long-Short Term Memory AutoEncoder, CNN: Convolutional Neural Network, LSTM Long Short-term Memory)

sionality to improve computational efficiency. The output of the pooling layer is used for the L-length feature in the convolution layers in an i^{th} channel,

$$P_i(m) = m \{C_i(mW, (m+1)W)\} \quad (9)$$

$$0 \leq m \leq \frac{L}{S}$$

where W denotes the window of pooling width, and S shows the size of the stride [34].

4) Layer of up-sampling: This is the inverse of the pooling layer. For the i^{th} feature, the result of up-sampling is:

$$U_k^i = \begin{cases} 0 & k \neq j_k \\ X^i & k = j_k \end{cases} \quad k \in [t \ 2t] \quad t = 12 \dots l \quad (10)$$

In this case, the variable l denotes the length of input features. j_k denotes the location of the maximum value obtained during the max-pooling process, and U_k^i is the k^{th} element of U^i [34].

5) Deconvolution layer: This is the inverse operation of the convolution layer.

$$D_i = \text{ReLu}(\sum X \otimes \overline{w}_i + c_i) \quad (11)$$

Here \otimes refers to the deconvolution computation, and \overline{w}_i which refers to the kernel used for deconvolution [34].

6) DeCLSTM: DeCLSTM has a structure, which is similar to CLSTM, but deconvolution is utilized in place of convolution. The

DeCLSTM output is:

$$h_D = F' \left(\sigma \left(W' \odot [c'_{t-1} \ h'_{t-1} \ x'_t] + b' \right) \right) \quad (12)$$

where F' is the DeCLSTM computation which refers to the components of the memory cell and three gates. h'_{t-1} is the former output of the DeCLSTM unit [34].

CLSTM-AE Training

Table 2 provides an overview of the CLSTM-AE training process, which aims to minimize the loss function by reconstructing data. In this process, the autoencoder utilizes the reconstructed error as its loss function. The Mean Square Error (MSE) (Equation (13)) is employed as the loss function in CLSTM-AE training [34], as follows:

$$E = \frac{1}{m} \sum_{n=1}^m y_n - x_n \quad (13)$$

where m is the number of data points, y_n is

Table 2: The Convolution Long-Short Term Memory AutoEncoder (CLSTM-AE) Training Process [34]

Input: Training data x
1-Establish the hyperparameters
2- Randomly initialize weights and biases.
3- At iteration number N.
4- Enter the training data, X.
5- Minimize the loss function during the training of the CLSTM-AE model.
6- Compute the feature output during the encoding phase, h.
7- Compute the reconstructed output during the decoding phase, y.
8- Compute the reconstructed error, E.
9- Update the decoder's parameters, W_d and B_d .
10-update the encoder's parameters W_d B_d
11-End of For
Output: Feature h and reconstructed data y.

CLSTM-AE: Convolution Long-Short Term Memory AutoEncoder

the predicted data point, and x_n is the original data.

In this research, the CLSTM-AE model is optimized using AdaGrad. Because of its capability of adaptive learning rate. The CLSTM-AE is divided into two parts: encoder and decoder. The data are abstracted into their main components at the end of the encoder section (named bottleneck of autoencoder), using max-pooling and convolution layers. A combination of convolution and LSTM layers is used to extract both the spatial and temporal features. The batch normalization layer is used during training to avoid overfitting. The autoencoder architecture was designed with the fewest layers to reduce the computational burden. The data are then reconstructed as input at the end of the decoder part, which is repeated until the minimum error between the data and its reconstructed version is obtained during 50 epochs. The network has learned to reconstruct the data with minimum error, and the encoder has learned how to extract the main components of data at its final layer. Following training, the main components of data are extracted from the autoencoder's bottleneck. The autoencoder is then saved, and the encoder section is separated from the autoencoder. To classify data into P300 and nP300, test data are fed to the encoder to extract its main components. The extracted features of the final layer of the encoder are flattened and fed into an artificial neural network during 250 epochs for classification, showing dropout to prevent overfitting during the training process and sigmoid as an activation function at its last layer (Figure 5b).

Results

After extracting features of the test data by a trained encoder, they are fed into an artificial neural network in 250 epochs for 15 signal trials of each subject. The accuracy and loss of the model for subject A and subject B are depicted in Figure 6. The model is also evaluated according to other metrics, presented in

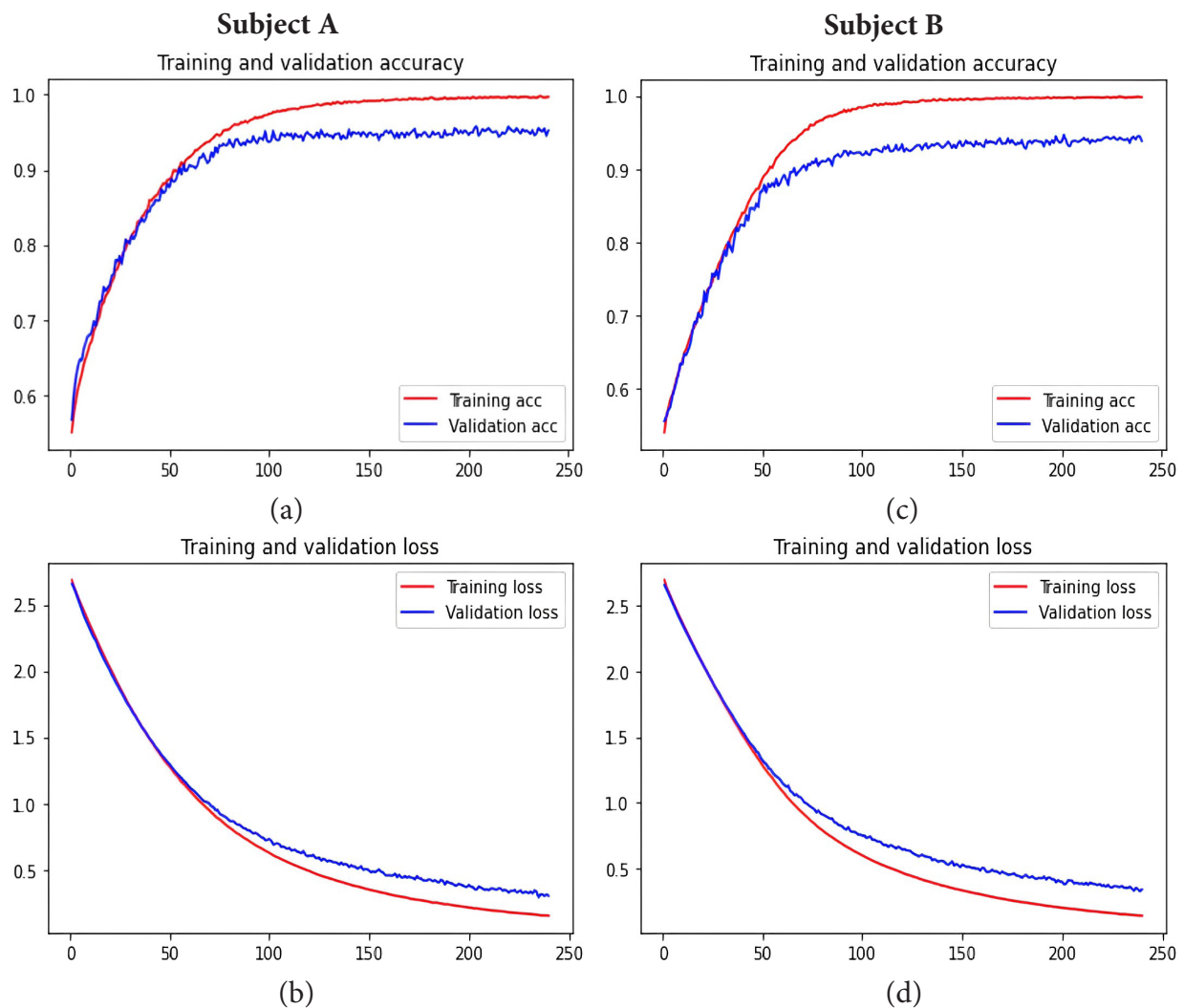


Figure 6: Accuracy (a and c) and loss (b and d) of the proposed method during training and validation for subject A, and subject B, respectively.

Table 3. Furthermore, we also prepared the results of the other state-of-the-art works on the same dataset in Table 4, showing the proposed method has superiority over most of the last outstanding papers.

Further, a CNN-LSTM was implemented by the same dataset to identify the superiority of CLSTM-AE to CNN-LSTM. The architecture of the CNN-LSTM is shown in Figure 5c, and results are presented in Table 3. Regarding accuracy, precision, sensitivity, and F1-Score, CLSTM_AE shows better performance as compared to CNN-LSTM. Additionally, CLSTM-AE extracts an abstraction of the data

reducing the computational burden.

Discussion

The current investigation aimed at employing a 3D input CLSTM_AE model for the P300 detection. The proposed model was implemented based on deep learning principles to redefine the problem of P300 classification from a signal classification into a video classification problem to use both spatial and temporal features of EEG signal in two stages. In the first stage, the autoencoder as an unsupervised feature extractor is used to not only extract but also abstract the principle features

Table 3: Results of CNN-LSTM and CLSTM-AE (the proposed method)

(CNN: Convolutional Neural Network, LSTM Long Short-term Memory, CLSTM-AE: Convolution Long-Short Term Memory AutoEncoder)

	Accuracy (%)		Precision (%)		Sensitivity (%)		F1-Score (%)	
	CNN-LSTM	CLSTM-AE	CNN-LSTM	CLSTM-AE	CNN-LSTM	CLSTM-AE	CNN-LSTM	CLSTM-AE
Subject A	91	95	88	91	98	99	95	95
Subject B	90	94	85	90	98	99	91	94

CNN: Convolutional Neural Network, LSTM Long Short-term Memory, CLSTM-AE: Convolution Long-Short Term Memory AutoEncoder

Table 4: Results of the state-of-the-art works on the same dataset

Reference number	Dataset (Subject)	Accuracy (%)
[24]	A	76.00
	B	80.00
[35]	A	94.20
	B	94.20
[36]	A	81.00
	B	84.00
[37]	A	86.39
	B	86.39
[15]	A	88.40
	B	90.00

in its latent space. In the second stage, it's checked whether a signal contains P300 or not using a neural network as a classifier. One of the main challenges of P300 classification in the P300 speller matrix is the imbalanced nature of data [38]; accordingly, the ADASYN approach was used for augmenting the P300 class. ADASYN has three advantages, including maintaining the primary distribution of features in the feature space after augmentation, maintaining the morphological structure of P300, and the non-randomness characteristic of this method as compared to other synthetic augmentation methods, which causes the stability of augmentation in each run [31]. Accuracy as a metric to survey the number of true positive and true negative classified

samples, precision as a metric to assess the ratio of true positive samples on true and false positive samples, sensitivity to examine the performance of the classifier on true positive samples and false negative samples, and F1-score to evaluate the trade-off between sensitivity and precision are used for performance evaluation. Based on the results in Table 4, the proposed method has superiority over most of the last outstanding papers [15, 24, 36-38].

Conclusion

In this work, a CNN-LSTM model is designed for P300 classification to redefine the signal classification problem to video classification and explore the feasibility of exploiting all the features of the data. Despite the high-quality classification, this method in its original form was computationally intensive. By considering CNN-LSTM's ability to extract both spatial and temporal features in the form of video classification, we sought to preserve this advantage of CNN-LSTM and find a solution to the aforementioned drawbacks. To this end, due to the unsupervised nature of autoencoders, we not only retain the benefits of CNN-LSTM but also abstract the data based on features extracted efficiently from CNN-LSTM, reducing the computational burden and SNR improvement and finally provide more accurate classification results. Also, to solve the problem of unbalanced data, conventional methods repeat the number of segments in classes with a small number of segments. These techniques copy the data that

may cause overfitting which are replaced by ADASYN as a new method for P300 augmentation. Compared to other studies, this work involves changing the view of the problem from signal classification to video classification, using all features of the data in the time domain and extracting spatial and temporal features simultaneously, and using minimal preprocessing algorithms before feature extraction. Furthermore, in the field of ERP, we use a new data augmentation method that preserves the structure and feature space of data as before. Future work could explore more efficient channels, parameter tuning, and other network architectures.

Authors' Contribution

R. Afrah, Z. Amini, and R. Kafieh conceived the idea and analyzed the results. The first draft was written by all authors. The final manuscript was revised by Z. Amini. All the authors read and approved the final version of the manuscript.

Ethical Approval

Approval of all ethical procedures was granted by the Institutional Review Board and Ethics Committees of the National Institute for Medical Research Development under Approval Nos. IR.MUI.RESEARCH.REC.1401.237, Isfahan University of Medical Sciences, Isfahan, Iran.

Funding

This work was supported in part by the Vice Chancellery for Research and Technology of Isfahan University of Medical Sciences under Grants 2401134.

Conflict of Interest

None

References

1. Jamil N, Belkacem AN, Ouhbi S, Lakas A. Non-invasive Electroencephalography Equipment for Assistive, Adaptive, and Rehabilitative Brain-Computer Interfaces: A Systematic Literature Review. *Sensors (Basel)*. 2021;**21**(14):4754. doi: 10.3390/s21144754. PubMed PMID: 34300492. PubMed PMCID: PMC8309653.
2. Amini Z, Abootalebi V, Sadeghi MT. A comparative study of feature extraction methods in P300 detection. 17th Iranian Conference of Biomedical Engineering (ICBME); Isfahan, Iran: IEEE; 2010. p. 1-4.
3. Eastmond C, Subedi A, De S, Intes X. Deep learning in fNIRS: a review. *Neurophotonics*. 2022;**9**(4):041411. doi: 10.1117/1.NPh.9.4.041411. PubMed PMID: 35874933. PubMed PMCID: PMC9301871.
4. Hasbulah MH, Jafar FA, Nordin MH. Fundamental of Electroencephalogram (EEG) Review for Brain-Computer Interface (BCI) System. *Int Res J Eng Technol*. 2019;**6**(5):1017-28.
5. Kobler RJ, Hirata M, Hashimoto H, Dowaki R, Sburlea AI, Müller-Putz GR. Simultaneous decoding of velocity and speed during executed and observed tracking movements: an MEG study. 8th Graz Brain-Computer Interface Conference; Austria: Institute of Neural Engineering Graz BCI; 2019.
6. Abiri R, Borhani S, Sellers EW, Jiang Y, Zhao X. A comprehensive review of EEG-based brain-computer interface paradigms. *J Neural Eng*. 2019;**16**(1):011001. doi: 10.1088/1741-2552/aaf12e. PubMed PMID: 30523919.
7. Kundu S, Ari S. Brain-Computer interface speller system for alternative communication: a review. *IRBM*. 2022;**43**(4):317-24. doi: 10.1016/j.irbm.2021.07.001.
8. Shojaedini SV, Morabbi S, Keyvanpour MR. A New Method to Improve the Performance of Deep Neural Networks in Detecting P300 Signals: Optimizing Curvature of Error Surface Using Genetic Algorithm. *J Biomed Phys Eng*. 2021;**11**(3):357-66. doi: 10.31661/jbpe.v0i0.975. PubMed PMID: 34189124. PubMed PMCID: PMC8236098.
9. Saliassi E, Geerligs L, Lorst MM, Maurits NM. The relationship between P3 amplitude and working memory performance differs in young and older adults. *PLoS One*. 2013;**8**(5):e63701. doi: 10.1371/journal.pone.0063701. PubMed PMID: 23667658. PubMed PMCID: PMC3646823.
10. Li L, Gratton C, Yao D, Knight RT. Role of frontal and parietal cortices in the control of bottom-up and top-down attention in humans. *Brain Res*. 2010;**1344**:173-84. doi: 10.1016/j.brainres.2010.05.016. PubMed PMID: 20470762. PubMed PMCID: PMC2900444.

11. Oralhan Z. A new paradigm for region-based P300 speller in brain computer interface. *IEEE Access*. 2019;**7**:106618-27. doi: 10.1109/ACCESS.2019.2933049.
12. Li Y, Nam CS, Shadden BB, Johnson SL. A P300-based brain-computer interface: Effects of interface type and screen size. *Int J Hum Comput Interact*. 2010;**27**(1):52-68. doi: 10.1080/10447318.2011.535753.
13. Fazel-Rezai R, Ahmad W. P300-based brain-computer interface paradigm design. In: Recent advances in brain-computer interface systems. InTech; 2011.
14. Lu Y, Bi L. EEG signals-based longitudinal control system for a brain-controlled vehicle. *IEEE Trans Neural Syst Rehabil Eng*. 2018;**27**(2):323-32. doi: 10.1109/TNSRE.2018.2889483.
15. Zhang Z, Yu X, Rong X, Iwata M. Spatial-temporal neural network for P300 detection. *IEEE Access*. 2021;**9**:163441-55. doi: 10.1109/ACCESS.2021.3132024.
16. Qu J, Wang F, Xia Z, Yu T, Xiao J, Yu Z, Gu Z, Li Y. A novel three-dimensional P300 speller based on stereo visual stimuli. *IEEE Trans Human-Machine Syst*. 2018;**48**(4):392-9. doi: 10.1109/THMS.2018.2799525.
17. Kim M, Kim J, Heo D, Choi Y, Lee T, Kim SP. Effects of Emotional Stimulations on the Online Operation of a P300-Based Brain-Computer Interface. *Front Hum Neurosci*. 2021;**15**:612777. doi: 10.3389/fnhum.2021.612777. PubMed PMID: 33767615. PubMed PMCID: PMC7987063.
18. Bulat M, Karpman A, Samokhina A, Panov A. Playing a P300-based BCI VR game leads to changes in cognitive functions of healthy adults [Internet]. bioRxiv [Preprint]. 2020 [cited 2020 May 30]. Available from: <https://doi.org/10.1101/2020.05.28.118281>.
19. Rasheed S. A review of the role of machine learning techniques towards brain-computer interface applications. *Mach Learn Knowl Extr*. 2021;**3**(4):835-62. doi: 10.3390/make3040042.
20. Alzahab NA, Apollonio L, Di Iorio A, Alshalak M, Iarlori S, Ferracuti F, et al. Hybrid Deep Learning (hDL)-Based Brain-Computer Interface (BCI) Systems: A Systematic Review. *Brain Sci*. 2021;**11**(1):75. doi: 10.3390/brainsci11010075. PubMed PMID: 33429938. PubMed PMCID: PMC7827826.
21. Zhang X, Yao L, Wang X, Monaghan J, McAlpine D, Zhang Y. A survey on deep learning-based non-invasive brain signals: recent advances and new frontiers. *J Neural Eng*. 2021;**18**(3):031002. doi: 10.1088/1741-2552/abc902. PubMed PMID: 33171452.
22. Cortez SA, Flores C, Andreu-Perez J. Single-trial p300 classification using deep belief networks for a bci system. In: 2020 IEEE XXVII International Conference on Electronics, Electrical Engineering and Computing (INTERCON); Lima, Peru: IEEE: 2020. p. 1-4.
23. Kshirsagar GB, Londhe ND. Improving Performance of Devanagari Script Input-Based P300 Speller Using Deep Learning. *IEEE Trans Biomed Eng*. 2019;**66**(11):2992-3005. doi: 10.1109/TBME.2018.2875024. PubMed PMID: 30307853.
24. Liu M, Wu W, Gu Z, Yu Z, Qi F, Li Y. Deep learning based on batch normalization for P300 signal detection. *Neurocomputing*. 2018;**275**:288-97. doi: 10.1016/j.neucom.2017.08.039.
25. Lu Z, Gao N, Liu Y, Li Q. The detection of p300 potential based on deep belief network. In: 2018 11th international congress on image and signal processing, biomedical engineering and informatics (CISP-BMEI); Beijing, China: IEEE; 2018. p. 1-5.
26. Shan H, Liu Y, Stefanov TP. A Simple Convolutional Neural Network for Accurate P300 Detection and Character Spelling in Brain Computer Interface. Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence (IJCAI-18); IJCAI; 2018. p. 1604-10.
27. Delgado JMC, Achancaray D, Villota ER, Chevalier S. Riemann-Based Algorithms Assessment for Single- and Multiple-Trial P300 Classification in Non-Optimal Environments. *IEEE Trans Neural Syst Rehabil Eng*. 2020;**28**(12):2754-61. doi: 10.1109/TNSRE.2020.3043418. PubMed PMID: 33296306.
28. Wu Q, Zhang Y, Liu J, Sun J, Cichocki A, Gao F. Regularized Group Sparse Discriminant Analysis for P300-Based Brain-Computer Interface. *Int J Neural Syst*. 2019;**29**(6):1950002. doi: 10.1142/S0129065719500023. PubMed PMID: 30880525.
29. Bianchi L, Liti C, Liuzzi G, Piccialli V, Salvatore C. Improving P300 Speller performance by means of optimization and machine learning. *Ann Oper Res*. 2022;**312**:1221-59. doi: 10.1007/s10479-020-03921-0.
30. Srimaharaj W, Chairicharoen R. A novel processing model for P300 brainwaves detection. *J Web Eng*. 2021;**20**(8):2545-70. doi: 10.13052/jwe1540-9589.20815.
31. He H, Bai Y, Garcia EA, Li S. ADASYN: Adap-

- tive synthetic sampling approach for imbalanced learning. In: 2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence); Hong Kong: IEEE; 2008. p. 1322-8.
32. Singari RM, Kankar PK, editors. Music Generation Using LSTM and Its Comparison with Traditional Method. In: Advanced Production and Industrial Engineering. IOS Press; 2022.
33. Alzubaidi L, Zhang J, Humaidi AJ, Al-Dujaili A, Duan Y, Al-Shamma O, et al. Review of deep learning: concepts, CNN architectures, challenges, applications, future directions. *J Big Data*. 2021;**8**(1):53. doi: 10.1186/s40537-021-00444-8. PubMed PMID: 33816053. PubMed PMCID: PMC8010506..
34. Yu J, Liu X, Ye L. Convolutional long short-term memory autoencoder-based feature learning for fault detection in industrial processes. *IEEE Trans Instrum Meas*. 2020;**70**:1-5. doi: 10.1109/TIM.2020.3039614.
35. Sarraf J, Pattnaik PK. A study of classification techniques on P300 speller dataset. *Mater Today Proc*. 2023;**80**(3):2047-50. doi: 10.1016/j.matpr.2021.06.110.
36. Ghazikhani H, Rouhani M. A deep neural network classifier for P300 BCI speller based on Cohen's classtime-frequency distribution. *Turkish J Electr Eng Comput Sci*. 2021;**29**(2):1226-40. doi: 10.3906/elk-2005-201.
37. Dittthapron A, Banluesombatkul N, Ketrat S, Chuangsuwanich E, Wilaiprasitporn T. Universal joint feature extraction for P300 EEG classification using multi-task autoencoder. *IEEE Access*. 2019;**7**:68415-28. doi: 10.1109/ACCESS.2019.2919143.
38. Lee T, Kim M, Kim SP. Improvement of P300-Based Brain-Computer Interfaces for Home Appliances Control by Data Balancing Techniques. *Sensors (Basel)*. 2020;**20**(19):5576. doi: 10.3390/s20195576. PubMed PMID: 33003367. PubMed PMCID: PMC7582676.