Online recognition of emotions using electroencephalography signals

Kevin Asher Altamirano Weiss, Department of Computation Engineering. University of Vale do Itajaí*,* Itajaí, Brazil  
[kev\_asher@edu.univali.br](mailto:kev_asher@edu.univali.br), ramirez@univali.br

Alejandro Rafael García Ramírez  
Department of Computation Engineering. University of Vale do Itajaí*,* Itajaí, Brazil  
[kev\_asher@edu.univali.br](mailto:kev_asher@edu.univali.br), ramirez@univali.br

*Abstract*—Automated brain signal recognition has the potential to bring new experiences, enhancing applications in a diversity of areas. One of the fields of research is the recognition of emotions through electroencephalography (EEG), which shows exclusive advantages compared to other methods. However, research with brain-computer interfaces (BCI) usually presents offline results, leaving a gap in the perspective of a functional system in a production environment. In this work, an online classification system of emotions (positive, neutral, and negative) was developed using open resources. Five machine learning models were trained with the SEED IV dataset, which is labeled with varying emotions. The models were trained and tested using nested cross-validation and grid search to obtain the best hyperparameters. The algorithm implementation in Python was integrated with the OpenBCI software to capture the EEG signals, processing them, and commanding the simulations. The best average accuracy obtained for a single subject was 76.19%, and the average accuracy for all subjects was 57.07%. The average signal processing and prediction time was around 1 millisecond, which demonstrates the potential for applications with real-time restrictions.

Keywords—Emotion recognition, Brain-computer interface, electroencephalography, real time.

# Introduction

The amount of research in applications of brain-computer interfaces (BCI) is constantly growing. Although originally studied within a limited scope in the medical field, such as sensory restoration and motor movement, it has recently expanded to other areas, such as entertainment and education [1]. One of the new applications of BCI systems is the recognition of emotions. The recognition of emotions is part of the branch of affective computing, which aims to process, recognize, and simulate the affective nature of human beings.

There are currently several ways for automated emotion recognition. In [2], facial expressions were recognized to identify seven emotions: joy, sadness, anger, fear, disgust, surprise and neutral. In [3], the recognition of emotions in the valence and arousal axis of the Circumplex model were performed through the classification of patterns found in speech. Both studies present valid techniques for the recognition of emotions but suffer from the limitation of possible simulation or authenticity of the expressed emotion. For example, applications that require a smooth user experience and involve usability tests can easily be compromised if the subjects in trial do not respond with an authentic emotion, which effectively counteracts the experiment [4]. In contrast, the recognition of emotions by means of Electroencephalography (EEG) signals does not suffer from this limitation, since the signals are involuntary and cannot be simulated by the user.

Most research on applications in BCI systems tends to be restricted to manual results of data analysis or processing data in an offline manner, that is, distantly in time from which the data was collected. Although this research strategy is necessary for data exploration and the achievement of discoveries, the number of research papers of practical BCI systems that work online, or in real-time, and that present open source solutions, are scarce.

In this context, research on BCI systems carried out at UNIVALI, and in partner institutions, highlighted the need for an online response. Oldoni et al. in [5] developed a BCI system to measure working memory, aiming to assist professionals in the treatment of patients with aphasia. As the results of the research were analyzed offline, there was a latency for professionals to have access to the data and act from the analysis of the acquired results. Catecati in [6], developed a BCI system to assess user experience when using software. As the results were processed offline, the system took a long time to give feedback to the researchers, hindering the dynamics of the experiments and, consequently, generating a latency that prevented contributing in an agile way in improving the usability of the software.

Given this scenario, this work aims to explore an implementation of a BCI system in an online context, being applied in the recognition of emotions. Three labels were used to classify emotions: positive, neutral, and negative emotions. The implementation described in this work uses open source technologies, to facilitate academic reproduction and reuse.

This paper is structured as follows: section II briefly discusses related works. Section III, presents the dataset used and the details of the system implementation, as well as the pre-processing, feature calculation and model training. Then the development of the integration of the simulation system and analysis of the system execution in real time is described. In section IV the limitations of the system are discussed and in section V the conclusions and perspectives for future works are presented.

# Related works

Several works based on BCI apply to the recognition of emotions. However, as previously mentioned, only a few of them has been implemented online or performed a simulation of the system.

Lan et al. in [7] used four levels of the Circumplex Model to represent the types of emotions classified by the system. The DEAP dataset was used, where fractal dimension was calculated and used as features. An average accuracy of 49.40% was obtained with a threshold of fractal values and max voting. Due to the simplicity of the processing model used, the authors comment on the feasibility of real-time execution.

Hou et al. in [8] classified positive and negative emotions. From the statistical and fractal features, they obtained an accuracy of 91.07% with an SVM classifier. They used a dataset created by themselves. To validate the execution of the system, a simulation interface was developed using a rendered 3D avatar to display the simulated emotions.

Liu et al. in [9] classified positive and negative emotions based on time-frequency features with LDA. An accuracy of 86.63% was obtained with an SVM classifier using a proprietary dataset. A prototype developed for the system simulation allowed the recognition of emotions in real-time. Graphs of valence levels were displayed during the simulation.

# Development

## Analysis and choice of dataset

There were found three datasets for public use and for academic purposes related to the context of recognizing emotions through EEG signals, namely: DEAP. [10], MAHNOB HCI [11] and SEED IV [12]. All datasets feature different sessions and different classes of emotion. In this work, the SEED IV dataset was chosen, as it is more compatible with the objectives of this research.

The SEED IV dataset features 1080 trials with 15 participants and an average duration of 2 minutes each. In each trial, signals from 62 electrodes positioned according to the 10-20 system were recorded, while participants watched videos that induced a set of emotions [12].

## Feature generation

As the designed system needs to work online, the prediction must be carried out as new signals arrive. For a prediction model, this implies the need to define parameters for a sliding window, such as size and overlap. The sliding window runs through the entire dataset, in which the features are calculated, as shown in Fig. 1. The window parameters widely vary in related research and there is no definite consensus on which parameters sizes to use. Thus, preliminary tests were performed with a smaller partition of the dataset, with different window parameters. No significant difference was found between the different parameters. However, the ones that showed the best results were used, such as the window size of 4 seconds and 2 seconds of overlap.

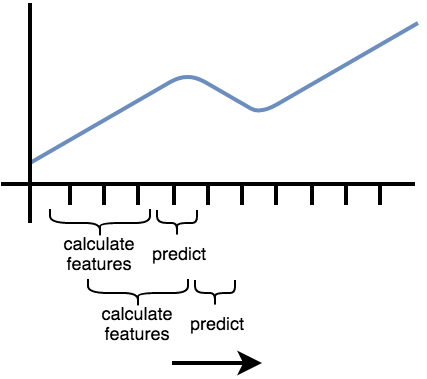


Fig. 1. Sliding window to calculate the features for prediction.

Source: Adapted from [13].

For each frequency band of EEG signals (gamma, beta, alpha, theta, and delta), the spectral density was calculated using the FFT, which was used as an input features to train the machine learning models. According to the experiments carried out by the authors of the SEED IV dataset, Zheng et al. in [12], the configuration with only 6 electrodes was used. This configuration presented results equivalent to the configuration of 62 electrodes [12]. Thus, a total of 30 features are generated per window instance, originating from the signals of 6 electrodes and features of 5 frequency bands.

Before starting the process of feature generation, a 1-50Hz bandpass filter was applied to remove artifacts of unwanted frequencies, as well as the use of a Hann window to minimize spectral leakage effects.

## Models and Hyperparameters

Multiple machine learning models have been tested to predict emotions. Table I specifies the models and hyperparameters used to implement the system in this work.

Table I

Models and Hyperparameters

| Model | Hyperparameters |
| --- | --- |
| Logistic Regression | C |
| Neural Network (MLP) | Nº of layers, neurons and alpha |
| SVM | Kernel, C and Gamma |
| LDA | N/A |
| Random Forest | Nº of estimators |

Since the calculated characteristics have an associated time dependency, it is not possible to shuffle all the data in the dataset, as is usually done before training machine learning models. This restriction exists because the trained model must be tested in trials never seen before, and the shuffling of the dataset will mix samples from different trials causing a data leak between the training and test datasets. Therefore, shuffling the dataset needs to be performed at the trial level and not at the sample level.

To alleviate over-fitting problems when training the models, the 5-fold cross-validation was used. To determine the best models, grid search was used to determine the best hyperparameter combinations. During the grid search execution, a second 5-fold cross-validation layer is performed to avoid over-fitting issues when finding the best hyperparameters.

This training strategy is called nested cross validation and was applied to all trained models. Table II, shows the average accuracy of the models used for the different sessions for each subject.

Analyzing the results of Table II it is possible to verify that there is no significant difference between the different models tested. For example, accuracy for subject 3 shows similar results through the different models. In contrast, the accuracy among participants varies widely. For example, the models have an accuracy of around 40% for subject 1, and around 75% for subject 15. Therefore, the models show a correlation that is highly subject-dependent.

Table II

Average accuracies of the models for each subject.

| Subject | LDA | Random Forest | Logistic Regression | Neural Network (MLP) | SVM |
| --- | --- | --- | --- | --- | --- |
| 1 | 40,03 | 44,41 | 40,49 | 39,19 | 41,31 |
| 2 | 64,36 | 65,92 | 62,83 | 60,62 | 63,78 |
| 3 | 46,22 | 44,6 | 43,47 | 42,24 | 44,9 |
| 4 | 59,56 | 67,28 | 61,42 | 61,31 | 64,49 |
| 5 | 52,86 | 56,09 | 50,41 | 50,02 | 50,85 |
| 6 | 64,58 | 68,92 | 67,97 | 66,18 | 67,56 |
| 7 | 58,53 | 61,71 | 59,93 | 54,74 | 60,84 |
| 8 | 60,31 | 62,07 | 62,39 | 60,01 | 61,66 |
| 9 | 55,17 | 52,82 | 45,17 | 53,66 | 52,56 |
| 10 | 52,22 | 56,93 | 54,39 | 52,97 | 52,88 |
| 11 | 49,71 | 45,09 | 48,2 | 43,4 | 49,8 |
| 12 | 42,11 | 46,5 | 41,32 | 42,33 | 44,29 |
| 13 | 54,97 | 52,16 | 52,03 | 49,17 | 54,03 |
| 14 | 52,68 | 55,69 | 54,55 | 52,44 | 53,95 |
| 15 | 73,67 | 75,93 | 76,19 | 71,23 | 76,18 |

## Software Development

The OpenBCI platform [14] simulated the acquisition of EEG signals. It provides several tools integrated within the acquisition boards developed by the OpenBCI. Since the OpenBCI was developed on the Processing platform, it uses Java as a programming language to create custom graphical interfaces, including custom widgets. In this work, a custom widget was developed to display the current emotion being predicted by the system.

The prototyping and development environment of the prediction system was carried out in Python. Therefore, it was necessary to develop a communication layer between the two execution contexts. In this work, a TCP connection between the OpenBCI widget and Python was developed. It follows the client-server model. The OpenBCI acts as a client, which forwards the EEG data to a Python server, which is constantly waiting for new data. The Python server responds with the classification results. Fig. 2 illustrated the exchange of messages between the two contexts.

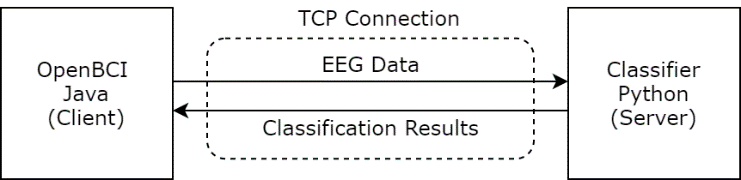


Fig. 2. Communication protocol between the two execution contexts.

## Simulation

Before running the simulation, the EEG data from a trial are transformed into a format accepted by OpenBCI. Then, the data file is selected and imported into the OpenBCI to start the simulation, as illustrated in Fig. 3.

Fig. 4 displays the simulation of an ongoing trial. The emotions widget is displayed in the lower right quadrant, which shows a negative emotion, in this case.

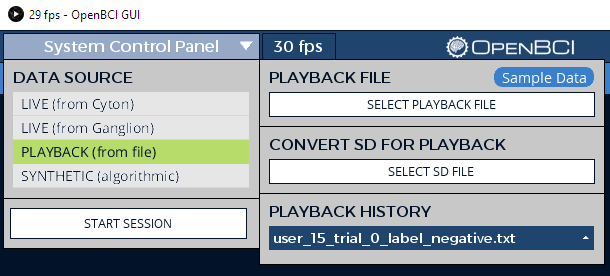


Fig. 3. Simulation setup.



Fig. 4. Simulation in progress.

## Execution time

The implementation of the classifier system was performed in the Python language, using the Scikit-Learn library. The pre-processing of the six data channels was carried out with the libraries of Scipy and Numpy. The implementation takes advantage of the vectorized operations available in the libraries, to efficiently process multiple data channels. The pre-processing execution time, added to the calculation of the characteristics and the prediction (using an SVM model), was, on average, close to 1 millisecond. It shows the capacity for developing BCI applications in real-time. The results were obtained with an Intel Core i5-7400 @ 3.00Ghz processor.

# Limitations

One of the challenges of processing EEG signals is the precise removal of artifacts. EEG signals are subject to several and different types of artifacts that arise from internal physiological occurrences and external sources. Despite the existence of several algorithms detecting and removing artifacts, most of them are not suitable for an online application, since they have unsupervised nature or high processing complexity [15].

In this work, the implementation of an artifact removal algorithm was not performed due to time restrictions. A bandpass filter was applied to remove unnecessary frequencies, but there are still unwanted artifacts in the frequencies used. The artifacts and their removal will be addressed in future works.

To illustrate this issue, the spectrograms of two subjects with different performance results from the trained models were compared. Fig. 5 shows the spectrogram of a trial from subject 1, with visible artifacts delimited by rectangles in red. The artifacts have the characteristic shape of the frequency domain transformation of the Dirac Delta function, which represents an impulse in the time domain. The artifact was probably due to a sudden reaction from the subject. The effect of this type of artifact produces outliers in the dataset, which can negatively affect the performance of trained models. As shown in Table II, subject 1 exhibits low performance in all models.

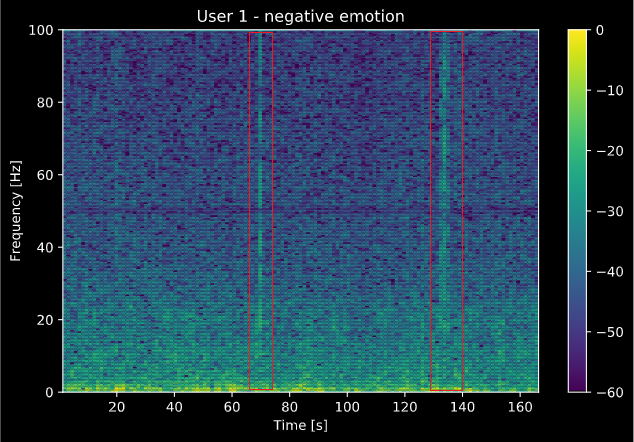


Fig. 5. Normalized spectrogram of the FT7 electrode with spectral density in dB of subject 1. In this trial, a negative emotion is stimulated.

In contrast, the trial with subject 15 (who presented the best performance among all subjects, according to Table II), shown in Fig. 6, does not show the discussed artifact illustrated in Fig. 5. However, it contains non-easily perceptible artifacts that cannot be detected from the spectrogram.

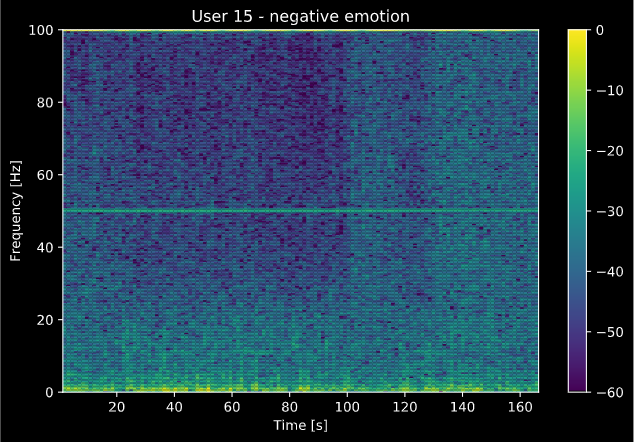


Fig. 6. Normalized spectrogram of the FT7 electrode with spectral density in dB of subject 15.

# Conclusions

In this work, open source technologies were explored to develop an online BCI system to recognize emotions. The OpenBCI platform was used to develop the application via simulation, and five machine learning models were used searching for the best classification results. The SEED-IV dataset was used to train the models, using spectral density. The best average accuracy obtained for a subject was 76.19%, and the average accuracy among all subjects was 57.07%. The variation observed among the classification results indicates that they are highly subject-dependent. The implemented system is modular, allowing its use in other BCI projects, and not only for the recognition of emotions. In addition, preliminary analysis of the system's processing time was performed, demonstrating the potential use in applications with real-time processing restrictions. Future work may address the recognition of emotions or other applications of BCI using the algorithms provided by this project, which will allow a faster iteration on the development cycle of new studies.

##### acknowledgemens

This study was financed by FAPESC, grant numbers 2015TR300 and 2020TR729, and by CNPq, grant number 315338/2018-0. This study was also financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Finance Code 001.

##### References

[1] R. P. N. Rao, Brain-computer interfacing: An introduction. 2011.

[2] E. Lozano-Monasor, M. T. López, A. Fernández-Caballero, and F. Vigo-Bustos, “Facial Expression Recognition from Webcam Based on Active Shape Models and Support Vector Machines,” in IWAAL 2014, Nov. 2014, pp. 147–154, doi: 10.1007/978-3-319-13105-4\_23.

[3] G. Trigeorgis et al., “Adieu features? End-to-end speech emotion recognition using a deep convolutional recurrent network,” in ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings, 2016, doi: 10.1109/ICASSP.2016.7472669.

[4] T. S. Rached and A. Perkusich, “Emotion Recognition Based on Brain-Computer Interface Systems,” Sch. Enviromental Sci., 2012.

[5] M. L. Oldoni, E. J. Legal, A. Rafael, and G. Ramirez, “Working Memory Load Measurement on Aphasia patients through a Brain Computer Interface : Theoretical Considerations and Preliminary Results,” pp. 1–11, 2019.

[6] T. Catecati, “AVALIAÇÃO DA SATISFAÇÃO DO USUÁRIO EM TEMPO REAL POR MEIO DO USO DE ELETROENCEFALOGRAFIA,” UDESC, 2018.

[7] Z. Lan, Y. Liu, O. Sourina, and L. Wang, “Real-time EEG-based User ’ s Valence Monitoring,” 2015.

[8] X. Hou, Y. Liu, O. Sourina, and W. Mueller-Wittig, “CogniMeter: EEG-based Emotion, Mental Workload and Stress Visual Monitoring,” Proc. - 2015 Int. Conf. Cyberworlds, CW 2015, pp. 153–160, 2016, doi: 10.1109/CW.2015.58.

[9] Y. J. Liu, M. Yu, G. Zhao, J. Song, Y. Ge, and Y. Shi, “Real-time movie-induced discrete emotion recognition from EEG signals,” IEEE Trans. Affect. Comput., vol. 9, no. 4, pp. 550–562, 2018, doi: 10.1109/TAFFC.2017.2660485.

[10] S. Koelstra et al., “DEAP: A database for emotion analysis; Using physiological signals,” IEEE Trans. Affect. Comput., vol. 3, no. 1, pp. 18–31, 2012, doi: 10.1109/T-AFFC.2011.15.

[11] R. L. Wilson, “Ethical Issues of Brain Computer Interfaces (BCI),” in IACAP 2013, 2013, pp. 1–18, [Online]. Available: http://www.iacap.org/proceedings\_IACAP13/paper\_41.pdf.

[12] W. L. Zheng, W. Liu, Y. Lu, B. L. Lu, and A. Cichocki, “EmotionMeter: A Multimodal Framework for Recognizing Human Emotions,” IEEE Trans. Cybern., vol. 49, no. 3, pp. 1110–1122, 2019, doi: 10.1109/TCYB.2018.2797176.

[13] M. Christ, N. Braun, J. Neuffer, and A. W. Kempa-Liehr, “Time Series FeatuRe Extraction on basis of Scalable Hypothesis tests (tsfresh – A Python package),” Neurocomputing, 2018, doi: 10.1016/j.neucom.2018.03.067.

[14] OpenBCI, “OpenBCI,” OpenBCI, 2013. .

[15] M. K. Islam, A. Rastegarnia, and Z. Yang, “Methods for artifact detection and removal from scalp EEG: A review,” Neurophysiol. Clin. Neurophysiol., 2016, doi: 10.1016/j.neucli.2016.07.002.