doi:10.20944/preprints202106.0016.v1

Article

Advanced LabVIEW Applications aimed for the Acquisition, Processing and Classification of the Electroencephalographic Signals Used in a Brain-Computer Interface System

Oana Andreea RUSANU [1]

* Correspondence: oana.rusanu@unitbv.ro; Tel.: +40 785 149 820

Abstract: This paper proposes several LabVIEW applications to accomplish the data acquisition, processing, features extraction and real-time classification of the electroencephalographic (EEG) signal detected by the embedded sensor of the NeuroSky Mindwave Mobile headset. The LabVIEW applications are aimed at the implementation of a Brain-Computer Interface system, which is necessary to people with neuromotor disabilities. It is analyzed a novel approach regarding the preparation and automatic generation of the EEG dataset by identifying the most relevant multiple mixtures between selected EEG rhythms (both time and frequency domains of raw signal, delta, theta, alpha, beta, gamma) and extracted statistical features (mean, median, standard deviation, route mean square, Kurtosis coefficient and others). The acquired raw EEG signal is processed and segmented into temporal sequences corresponding to the detection of the multiple voluntary eye-blinks EEG patterns. The main LabVIEW application accomplished the optimal real-time artificial neural networks techniques for the classification of the EEG temporal sequences corresponding to the four states: 0 - No Eye-Blink Detected; 1 - One Eye-Blink Detected; 2 – Two Eye-Blinks Detected and 3 – Three Eye-Blinks Detected. Nevertheless, the application can be used to classify other EEG patterns corresponding to different cognitive tasks, since the whole functionality and working principle could estimate the labels associated with various classes.

Keywords: brain-computer interface, EEG signal, artificial neural networks, LabVIEW application, features extraction, eye-blinks detection, EEG headset

1. Introduction

Brain-Computer Interface is a multidisciplinary research field, which comprises achievements in related scientific and technical areas: artificial intelligence, computer science, mechatronics [1-3], signal processing, neuroscience and psychology [4]. The aim of a brain-computer interface system is related to helping people with neuromotor disabilities, who cannot communicate with the outside environment by using natural paths, such as muscles and peripheral nerves. These patients have suffered a cerebral vascular accident or severe injuries to the spinal cord so that they have lost the ability to move their upper and/or lower limbs. Other reasons which provoked their impairments are related to awful diagnosis: amyotrophic lateral sclerosis or locked-in syndrome. An innovator solution able to provide an alternative way of regaining their independence and confidence is the Brain-Computer Interface. This is a thought-provoking field with a rapid evolution because of its applications based on brain-controlled mechatronics devices (wheelchairs [5-8], robot arm [9-10], robot hand [11], mobile robots [12], household items [13] and smart home [14]) or mind-controlled virtual keyboards [15] or 3D simulations. The working principle underlying a Brain-Computer Interface is consisting of the following

main phases: acquisition, processing and classification of signals related to brain patterns triggered by the execution of certain cognitive tasks, followed by the translation of the detected biopotentials and transmission of commands to the controlled applications.

Electroencephalography (EEG) is the most convenient method for the acquisition of signals across the scalp, by using dry or wet electrodes which allow the detection of the neuronal bio-potentials. It is non-invasive, portable, inexpensive and advantageous thanks to its high temporal resolution. There were also produced mobile EEG headsets aimed for research [16] or entertainment. Further, the EEG acquired signal is undergone some processing methods for noise reduction and filtering of the most significant ranges of frequencies. Then, the EEG bio-potentials should be classified by applying artificial intelligence techniques so that it will result different categories of signal patterns according to the cognitive task executed by the user. Thus, the user should accomplish some tasks requiring a mental effort: imagine the movement of the right or left limb [17-18], mentally execute some arithmetic operations, focus his/her own attention [19] on a single idea, relax or meditate, mentally counting from 3 to 3 and visualize the resizing of 3D figures. Each cognitive task is characterized by a specific EEG signals pattern, which is assigned the corresponding class. The classification process is facilitated by machine learning techniques, such as: neural networks, supported vector machines, logistic regression and others. Every class is associated to a particular command that is transmitted to the target application, for example: a brain-controlled motorized wheelchair, a neuroprosthesis, a medical system aimed for rehabilitation [20] or the mind-controlled movement of the cursor on the computer screen.

A recently published paper [21] presents the classification of the EEG signal acquired from the NeuroSky Mindwave Mobile headset. An Android mobile app was developed so that it can facilitate the Bluetooth connection between smartphone and NeuroSky. Alpha (8 – 13 Hz) and Beta (13 – 30 Hz) EEG rhythms have been chosen. The features that provided an increased accuracy are the following: power spectral density, spectral centroids, standard deviation and energy entropy. These signals and features are associated with the execution of some mental exercises: quick solving math and relaxing (doing nothing). The classifiers involved in this experiment are the following: Linear Discriminant Analysis, Support Vector Machines and K-Nearest Neighbor. To this research, 8 subjects have been invited to collaborate and a measurement protocol has been defined.

Another recently published paper [22] presents a brain-computer interface application aimed for people with movement disabilities. It has been used a portable headband called Muse, released in 2016. The phases underlying the implementation of the application are the following: data acquisition, signal processing, features extraction, an algorithm for the translation and transferring of commands to the controlled device. The relevant mental tasks are consisting of: short and long eye-blinks. The eye-blink is an artefact that is easy to detect across the raw EEG signal. The research was focused on the development of a software application which enabled the data acquisition from the Muse headset, signals processing and their translation into commands. There were conducted experiments with 21 people with age between 20 – 60 years. The time duration of the training was measured and compared between the participants to the research study.

Another significant recent research [23] focused on the classification of two tasks of imagination that generated specific brainwaves patterns. It was used a single-electrode headset, called NeuroSky Mindset, to detect event related potentials (ERPs) corresponding to the cognitive process, for example: reflection to a specific image or character. During the tasks of imagination, the brain has undergone a workload that can be measured and classified into two classes so that it will be associated with the answers true/false given to specific questions. An initial phase was related to the EEG signals processing, whose purpose was related to interference removal and noise reduction. Then, it was executed the process of essential features extraction, performed in OpenVIBE software. There were used the following techniques: Discrete Wavelet Transform (DWT), signal average and simple DSP (Digital Signal Processing). After that, there were trained some classification models, which involved: support vector machines (SVM) and artificial neural networks with multilayer perceptron (MLP). Weka software was used to evaluate the SVM and MLP models and compare them for selection of the highest classification accuracy. Data were collected from 8 healthy students (average age: 19.6 years), who were shown different experimental paradigms composed of: symbolic and text cues. By focusing his/her attention on a particular cue, the user could answer true/false to a question. The maximum accuracy achieved with the proposed method was 91%.

This paper proposes a novel LabVIEW application aimed for the acquisition, processing, and classification of the electroencephalographic (EEG) signals necessary to the implementation of a Brain-Computer Interface system (BCI). Although the EEG signals classification process is mainly focused on the detection of multiple voluntary eye-blinks, the implemented algorithm provides flexibility and robustness so that it can be successfully applied to the recognition of different signal patterns corresponding to other mental tasks related to the Brain-Computer Interface research filed.

The development of the LabVIEW application is based on the State Machine design pattern, so that it enables the running of the following sequences:

- Manual and automatic mode for the EEG signals acquisition;
- The EEG signal processing and the preparation of temporal sequences for EEG dataset;
- The generation of training and testing EEG dataset based on multiple mixtures between the selected EEG signals and the extracted features;
- The training of a model based on the Artificial Neural Networks (ANN) through the EEG signal classification process by setting certain values for hyperparameters or searching the optimized ones;
- The testing phase of the trained model based on Artificial Neural Networks (ANN) so that it will be possible to measure the accuracy and precision of the classification process;
- The deployment phase by applying the Neural Network model on the EEG signals acquired in real-time ◎ an independent novel LabVIEW application was developed.

Regarding the main contributions brought by the main application presented in this paper, it was proposed a novel approach by leveraging the LabVIEW graphical programming environment to develop several original virtual instruments aimed to acquire, process and classify the EEG signal acquired from the embedded sensor of the portable EEG NeuroSky headset. It was also proposed a unique and robust experimental paradigm involving various stages accomplished by the software system based on LabVIEW.

Another achievement of the research conducted in this paper is related to the development of an independent LabVIEW application aimed for the classification of EEG signals in real-time. Previous related papers are mainly focused on the following stages: data acquisition, processing [24], feature extraction [25-26], training of classification models applied on the acquired data and testing the obtained models on different datasets. The final phase corresponding to applying the trained and tested classification model on data acquired in real-time is rarely presented in the previously published articles [27-28]. Moreover, the current paper describes in detail the preparation of training or testing dataset, which is an important phase in the machine learning based application.

The purpose of the current research is to provide an efficient solution that integrates the following sections: data acquisition, processing and real-time classification of the EEG signals. The working principle underlying the LabVIEW application could also enable the detection of some brain patterns, for example, P300 and SSVEP EEG potentials [29-30] related to various cognitive tasks. Currently, the developed system was tested on the recognition of the EEG patterns related to the classification of multiple voluntary eye-blinks, which are detected as artefacts across the raw EEG signal. According to scientific literature [31], the eye-blinks were considered precise control signals in a brain-computer interface application.

Regarding the structure of the current paper, the following sections are related to providing some detailed insights about the materials and methods necessary to develop the LabVIEW application, showing and analyzing the results and finally, there are stated some conclusions about the overall project work and future research directions are highlighted.

2. Materials and Methods

2.1. LabVIEW graphical programming environment

This paper proposes a novel software application developed in the LabVIEW graphical programming environment [32], which provides efficient solutions for helping researchers, professors, students and engineers in their projects and related activities. LabVIEW proved successful results regarding the development of virtual instruments for the automation of industrial processes, data acquisition, signal analysis, image processing, interactive simulations, command and control systems, testing and measurement systems and education. Moreover, LabVIEW enables the possibility to create versatile applications based on artificial intelligence or machine learning techniques, by taking advantage of various toolkits containing useful functions. Therefore, LabVIEW offers the benefit of allowing rapid and simple communication with several acquisition devices, hardware systems aimed for data processing and actuators. The programming language underlying LabVIEW is called 'G' and it was introduced in 1986 by Jeff Kodosky and James Truchard. The software applications developed by using LabVIEW are called 'virtual instruments' (vi). The graphical user interface designed in LabVIEW is known as 'Front Panel'. It can include various appealing elements, such as: controls (input variables) of different types (numerical: knobs, dials, meters, gauges, pointer slides; Boolean: push buttons, slide switch, toggle switch, rocker button; string: text boxes) and indicators (output variables) of different types (numerical: waveform graphs, charts, XY graphs; Boolean: LEDs). Like in procedural programming languages, LabVIEW offers the advantage to use data structures, for example, arrays and clusters. The source code implemented in LabVIEW is represented by the 'Block Diagram', which provides the possibility of using both various structures (while loop, for loop, switch case) and functions ('Bundle by Name', 'Build Array', 'Search and replace string') in order to design compact, maintainable, scalable and robust programming paradigms, such as: State Machine, Producer/Consumer, Event Handler or Master/Slave.3.

2.2. NeuroSky Mindwave Mobile – Second Edition (released in 2018)

The EEG signal is detected from the single embedded sensor of NeuroSky Mindwave Mobile (second edition, released in 2018) headset [33], which is based on a ThinkGear chipset that enables advanced functionalities. These features are available by accessing the 'NeuroSky' LabVIEW toolkit, which includes various functions allowing: the acquisition of raw EEG signal, the extraction of the EEG rhythms (delta: 0 – 3 Hz, theta: 4 – 7 Hz, alpha: 8 -12 Hz, beta: 13 – 30 Hz and gamma: > 30 Hz), the measurement of attention and meditation level and the calculation of eye-blinking strength. NeuroSky headset has only an embedded sensor, which should be placed on the forehead (FP1 location according to the 10-20 System) or the frontal cerebral lobe of the user. Likewise, the headset contains a clip that represents the reference or the ground necessary to close the electrical circuit and it should be attached to the earlobe. Moreover, the second version of NeuroSky Mindwave Mobile headset distinguishes by the other releases through its design, providing a flexible arm to get a comfortable placement on the user's forehead and achieve samples in higher precision of the EEG bio-potentials. Likewise, NeuroSky chipset increases the accuracy of the EEG signal due to its embedded advanced filters aimed for noise reduction. Further benefits are given by the technical features: sampling rate is equal to 512 Hz, the range of bandwidth is from 0.5 to 100 Hz, it supports SPI and I2C interfaces and its resolution is equal to 16 bits. According to scientific literature [34-35], NeuroSky Mindwave headset was frequently used in the research due to its inexpensive cost and good results regarding the high accuracy of the acquired EEG signal.

2.3. An overview of the proposed LabVIEW application based on a State Machine paradigm involving the acquisition, processing and classification of the EEG signal detected from the embedded sensor of Neurosky

The main original contribution of this paper is revealed by the implementation of the proposed LabVIEW application comprising of several novel virtual instruments aimed for the integration of the three stages underlying the development of a portable brain-computer interface system: the acquisition, the processing and the classification of the EEG signal detected from the embedded sensor of the NeuroSky Mindwave Mobile headset. The proposed LabVIEW application is consisting of a State Machine paradigm accomplishing the following functionalities (Figure 1):

Manual Mode of data acquisition for real-time monitoring and displaying the EEG signal (raw, delta, theta, alpha, beta and gamma) both in time and frequency domain;

Automatic Mode of data acquisition for recording the EEG temporal sequences associated with certain cognitive tasks necessary for the preparation of the EEG datasets;

Processing the obtained EEG temporal sequences by the extraction of statistical features and the assignment of proper labels corresponding to each of the four classes: 0 – No Eye-Blink; 1 – One Eye-Blink; 2 – Two Eye-Blinks and 3 – Three Eye-Blinks;

The automatic generation of a series of EEG datasets based on the proposed mixtures between the EEG signals (raw, delta, theta, alpha, beta and gamma) in time and frequency domains and the extracted statistical features (arithmetic mean, median, mode, skewness and others);

The training of a neural networks model either by setting specific hyperparameter or by searching the optimized hyperparameters applied on each EEG dataset delivered from the previous stage;

The evaluation of each trained neural networks model by running it for the classification of another EEG dataset that can be delivered by using a similar procedure as it was previously described regarding the proposed mixtures between EEG signals and statistical features.

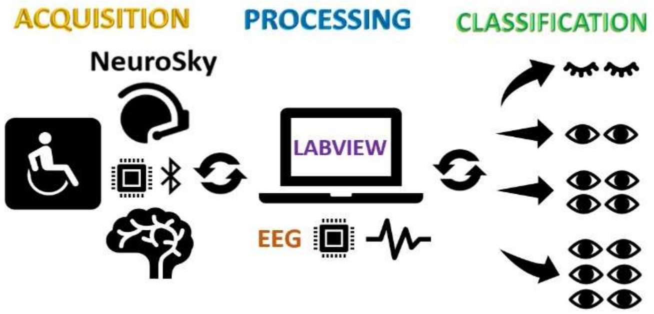
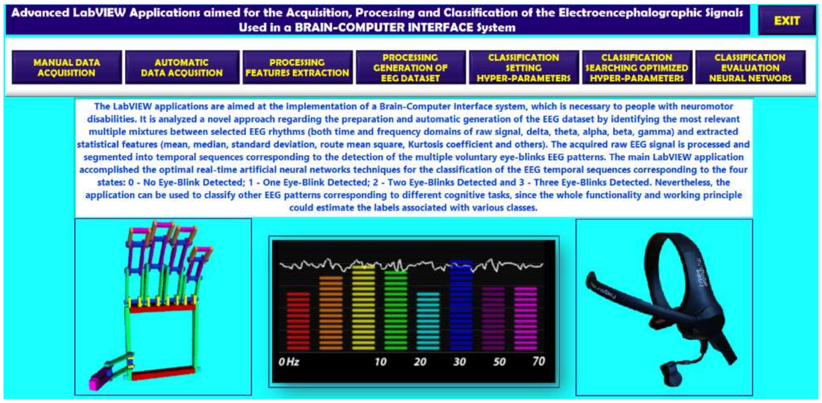


Figure 1. An overview on the proposed LabVIEW application aimed for the acquisition, processing and classification of the electroencephalographic signal used in a brain-computer interface

Each of the above functionalities can be accomplished by the selection of the corresponding virtual button to open the tab or graphical windows consisting of customized settings and options. Moreover, each of these tabs is comprising of a button for returning to the main Configuration graphical window shown in the Figure 2.



Advanced LabVIEW Applications aimed for the Acquisition, Processing and Classification of the Electroencephalographic Signals Used in a BRAIN-COMPUTER INTERFACE System

The LabVIEW applications are aimed at the implementation of a Brain-Computer Interface system, which is necessary to people with neuromotor disabilities. It is analyzed a novel approach regarding the preparation and automatic generation of the EEG dataset by identifying the most relevant multiple mixtures between selected EEG rhythms (both time and frequency domains of raw signal, delta, theta, alpha, beta, gamma) and extracted statistical features (mean, median, standard deviation, route mean square, Kurtosis coefficient and others). The acquired raw EEG signal is processed and segmented into temporal sequences corresponding to the detection of the multiple voluntary eye-blinks EEG patterns. The main LabVIEW application accomplished the optimal real-time artificial neural networks techniques for the classification of the EEG temporal sequences corresponding to the four states: 0 - No Eye-Blink Detected; 1 - One Eye-Blink Detected; 2 - Two Eye-Blinks Detected and 3 - Three Eye-Blinks Detected. Nevertheless, the application can be used to classify other EEG patterns corresponding to different cognitive tasks, since the whole functionality and working principle could estimate the labels associated with various classes.

EXIT
MANUAL DATA ACQUISITION
AUTOMATIC DATA ACQUSITION
PROCESSING FEATURES EXTRACTION
PROCESSING GENERATION OF EEG DATASET
CLASSIFICATION SETTING HYPER-PARAMETERS
CLASSIFICATION SEARCHING OPTIMIZED HYPER-PARAMETERS
CLASSIFICATION EVALUATION NEURAL NETWORS
0 Hz
10
20
30
50
70

Figure 2. A sequence of the Front Panel showing the Configuration graphical window displaying the virtual buttons corresponding to all the functionalities based on a State Machine paradigm

2.4. The manual mode of data acquisition and the EEG signal processing

A significant function, included by the LabVIEW 'NeuroSky' toolkit and used in the development of the application presented in this paper, is called: 'ThinkGear Read – MultiSample Raw (EEG)'. This function enables the raw EEG signal acquisition and returns an array containing a specific number of numerical values. This number is given by the input parameter called 'Samples to Read' and it can be assigned to a numerical value, for example: 512, 256, 128, 64 or other. It should be taken into consideration that 'Samples to Read' parameter does not have the same meaning with the 'Sampling Frequency'. According to technical specifications, in case of NeuroSky chipset, the sampling frequency is a fixed value, established to 512 Hz, so that it can be acquired a number of 512 samples in one second. Therefore, by setting 'Samples to read = 512', it results a single buffer or 1D array containing 512 numerical values. Otherwise, by setting 'Samples to read = 256', there are returned two buffers or 2 x 1D arrays, each of them containing 256 numerical values. In LabVIEW, 1D array is a matrix with one dimension, meaning either one row and many columns or one column and many rows. Other functions allowing the communication between LabVIEW and NeuroSky headset were linked as it follows (Figure 3): 'Clear Connections' (used to reset all previous connections), 'Create Task' (used for the initial settings of serial data transfer: port name, baud rate, data format), 'Start Task' (used to start the connection or open the communication port), 'Signal Quality' (used to display the value characterizing the percentage of signal quality), 'Read MultiSample Raw' (used to acquire an array of samples of EEG raw signal) and 'Clear Task' (used to close the connection or the communication port).

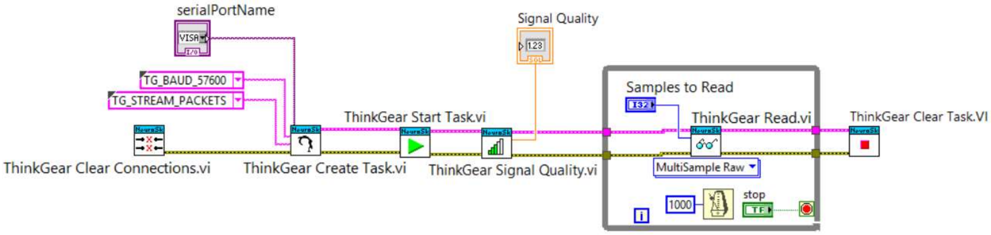


Figure 3. A sequence of the Block Diagram showing the set of the functions allowing the communication between LabVIEW application and NeuroSky headset.

Further, as it is shown in Figure 4, the output array of numerical values returned by 'Read MultiSample Raw' function is used as an input to the 'Build Waveform' function, along with other two parameters (t0 = current time in hours, minutes, seconds and milliseconds and dt = time interval in seconds between data points), in order to obtain the appropriate data format necessary to apply a filter for the extraction of a particular range of frequencies, which can be graphically displayed. The ‘t0 = Current time’ parameter is calculated by using the ‘Get Date/Time’ function, whereas the ‘dt = time interval’ is given by the division of 1 to 512, taking into account that ‘Samples to Read = 512’.

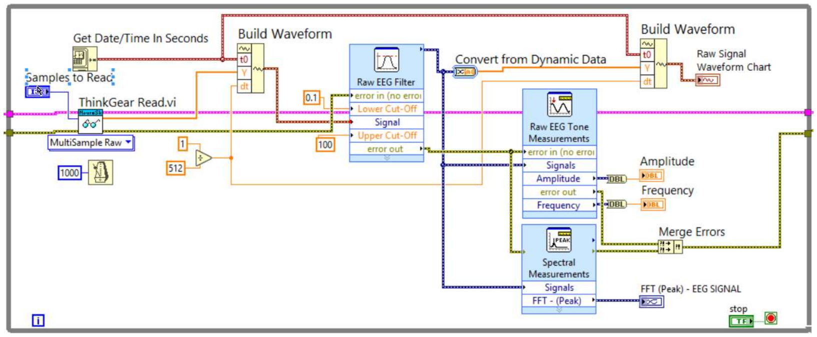
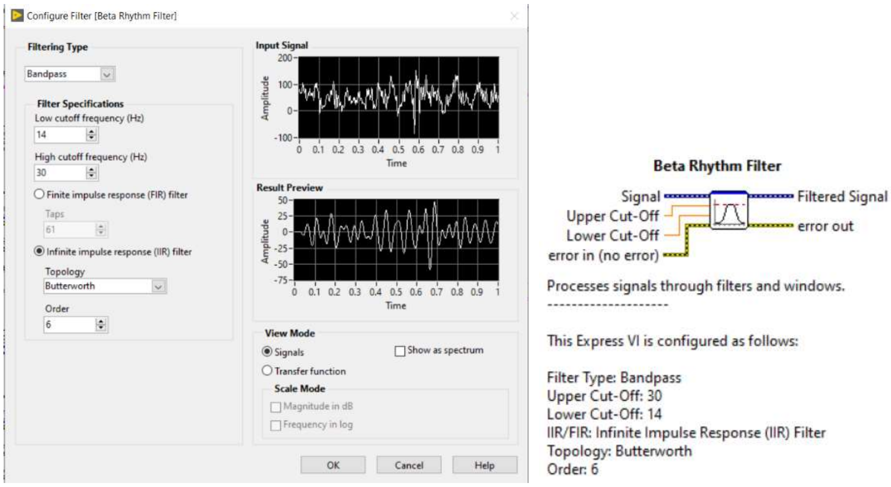


Figure 4. A sequence of the Block Diagram showing the set of the functions allowing the acquisition, processing and graphical displaying of the raw EEG signal acquired from NeuroSky headset

The output of ‘Build Waveform’ function is passed through the ‘Filter’ function so that it results a particular sequence of signal frequencies extracted from the entire range representing 0 – 512 Hz. According to Figure 5, the configuration of the ‘Filter’ is the following: filter type = Bandpass; lower cut-off = 14 Hz (for beta EEG rhythm); upper cut-off =30 Hz (for beta EEG rhythm); option = infinite impulse response (IIR); topology = Butterworth; order = 6. Two of the previously mentioned parameters – lower cut-off and upper cut-off – should be customized depending on the frequency range of the EEG rhythms: delta (0.1 – 3.5); theta (4 – 7.5); alpha (8 – 13) and beta (14 – 30). Another type of filter, called Highpass, is used to extract the gamma (upper cut-off = 30) EEG rhythm. The output of the ‘Filter’ function is an array of samples or numerical values which are represented on a Waveform Chart.


Configure Filter [Beta Rhythm Filter]

Filtering Type: Bandpass

Filter Specifications

Low cutoff frequency (Hz): 14

High cutoff frequency (Hz): 30

○ Finite impulse response (FIR) filter

Taps: 61

● Infinite impulse response (IIR) filter

Topology: Butterworth

Order: 6

View Mode

● Signals □ Show as spectrum

○ Transfer function

Scale Mode

□ Magnitude in dB

□ Frequency in log

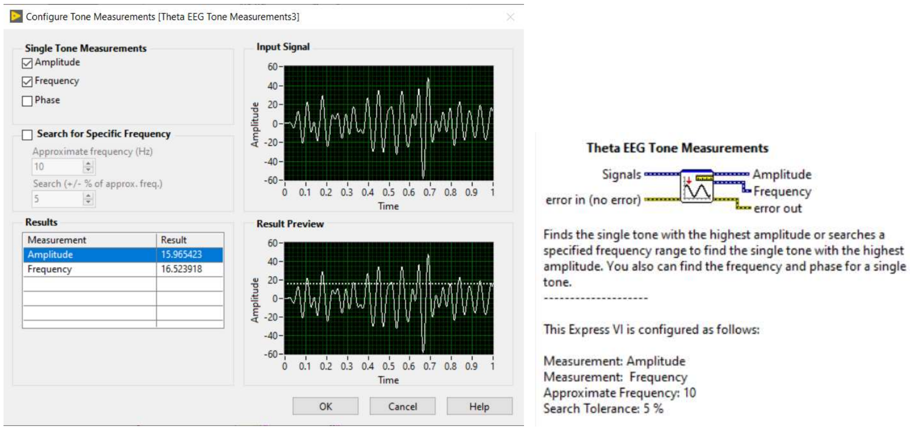
Processes signals through filters and windows.

This Express VI is configured as follows:

Filter Type: Bandpass
Upper Cut-Off: 30
Lower Cut-Off: 14
IIR/FIR: Infinite Impulse Response (IIR) Filter
Topology: Butterworth
Order: 6

Figure 5. The settings and description corresponding to ‘Filter Express VI’ included by the ‘Signal Analysis Express VIs’ LabVIEW palette

As mentioned before, it is necessary to use again the ‘Build Waveform’ function, by setting: current time (t0) = the output of ‘Get Date/Time’ function and time interval (dt) = 1 divided by 512. The output of ‘Filter’ function is also passed through the ‘Tone Measurements’ function (Figure 6), which allows the calculation of some parameters, including the highest amplitude and frequency of a single tone or a specified frequency range. The format of the output given by ‘Tone Measurements’ function is ‘dynamic data’. Therefore, it is necessary to convert it to double-precision numerical data.



Configure Tone Measurements [Theta EEG Tone Measurements3]

Single Tone Measurements

☑ Amplitude
☑ Frequency
□ Phase

□ Search for Specific Frequency

Approximate frequency (Hz): 10
Search (+/- % of approx. freq.): 5

Results

Measurement	Result
Amplitude	15.965423
Frequency	16.523918

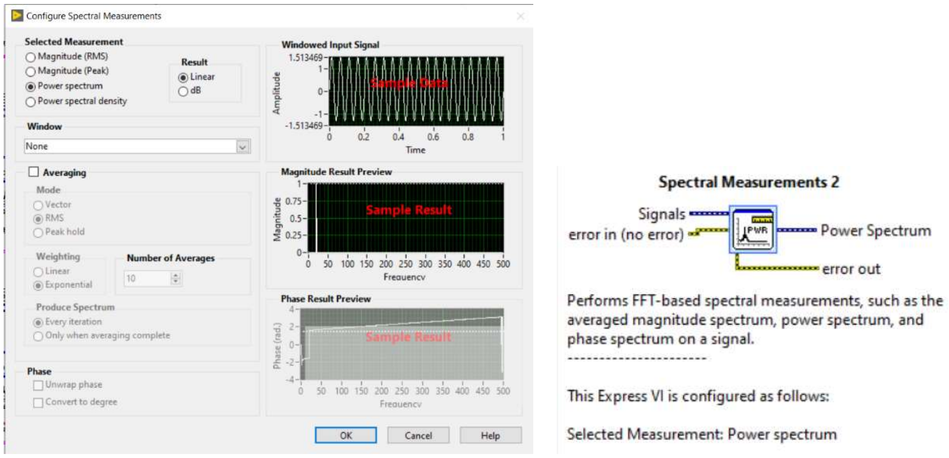
Finds the single tone with the highest amplitude or searches a specified frequency range to find the single tone with the highest amplitude. You also can find the frequency and phase for a single tone.

This Express VI is configured as follows:

Measurement: Amplitude
Measurement: Frequency
Approximate Frequency: 10
Search Tolerance: 5 %

Figure 6. The settings and description corresponding to 'Tone Measurements Express VI' included by the 'Signal Analysis Express VIs' LabVIEW palette

Parallel to these programming sequences, 'Spectral Measurements' function (Figure 7) is also applied to the output of the 'Filter' function. Accordingly, on a Waveform Graph, it is displayed the power spectrum of each EEG rhythm (delta, theta, alpha, beta and gamma) extracted from the raw EEG signal, by using 'Filter' function. Likewise, 'Spectral Measurements' LabVIEW function can perform the following features: averaged magnitude spectrum and phase spectrum on a specific signal.



Configure Spectral Measurements

Selected Measurement

○ Magnitude (RMS)
○ Magnitude (Peak)
● Power spectrum
○ Power spectral density

Result

● Linear
○ dB

Window: None

□ Averaging

Mode
○ Vector
● RMS
○ Peak hold

Weighting
○ Linear
● Exponential

Number of Averages: 10

Produce Spectrum
● Every iteration
○ Only when averaging complete

Phase
□ Unwrap phase
□ Convert to degree

Performs FFT-based spectral measurements, such as the averaged magnitude spectrum, power spectrum, and phase spectrum on a signal.

This Express VI is configured as follows:

Selected Measurement: Power spectrum

Figure 7. The settings and description corresponding to 'Spectral Measurements Express VI' included by the 'Signal Analysis Express VIs' LabVIEW palette

All the previously mentioned functions (Filter – Tone Measurements – Spectral Measurements) are grouped in a case structure (Figure 8), whose selector is a button, which is a Boolean control with two states: true and false. Those three functions are linked to each other to get output signals that are graphically displayed, depending on the state of the button corresponding to a certain EEG rhythm: delta, theta, alpha, beta, gamma or raw signal.

Overall, five case structures are representing the five EEG rhythms, which can be activated or deactivated by pressing those buttons. Therefore, the user can select specific EEG rhythms that should be displayed on either the Waveform Chart corresponding to the time domain (that is the output of the 'Filter' function) or the Waveform Graph, associated to the frequency domain (that is the output of the 'Spectral Measurements' function). All the five case structures are included in a while loop, which also contains the same network of functions that were previously described regarding the display of the

doi:10.20944/preprints202106.0016.v1

raw EEG signal. Because of using the ‘while loop’, the LabVIEW application is running in the manual mode, until an exit condition is fulfilled.

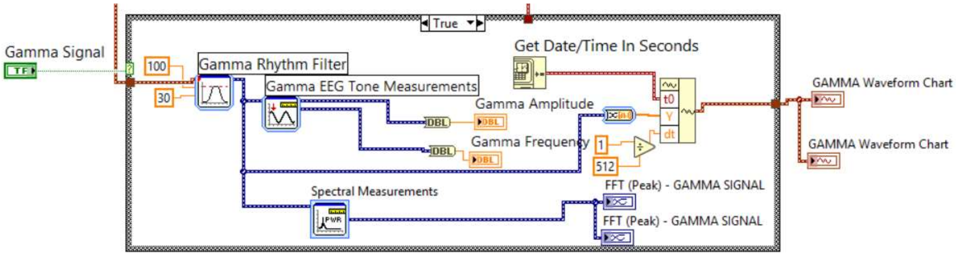
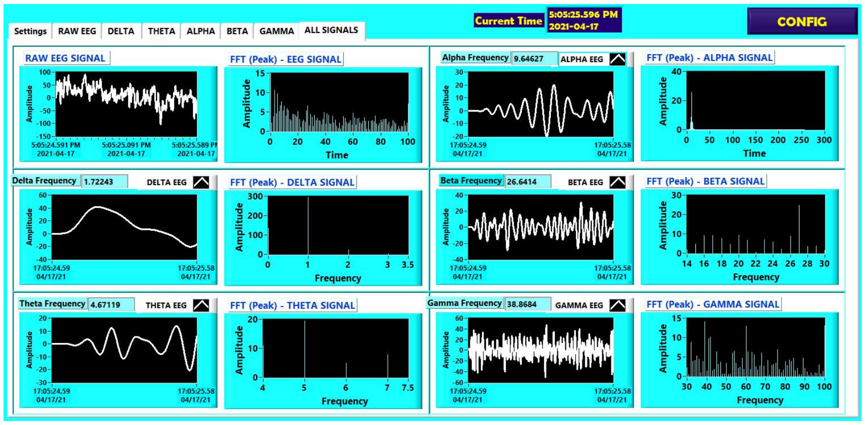


Figure 8. The settings and description corresponding to ‘Spectral Measurements Express VI’ included by the ‘Signal Analysis Express VIs’ LabVIEW palette

Taking into account that the Block Diagram is based on the State Machine design pattern, the transition between states or different sequences should be simple and quick. Therefore, by pressing the ‘Config’ button, the exit condition is fulfilled so that the running of the manual mode of data acquisition (Figure 9) is stopped. The application continues to run in the ‘Configuration’ state, where the user can select another option: automatic mode of data acquisition, features extraction, training neural networks model and others.


Current Time 5:05:25.596 PM 2021-04-17

Alpha Frequency 9.64627

Delta Frequency 1.72243

Beta Frequency 26.6414

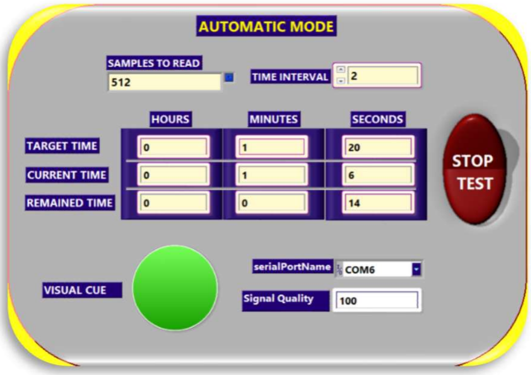
Theta Frequency 4.67119

Gamma Frequency 38.8684

Figure 9. Monitoring EEG rhythms by graphical displaying its time variation (Waveform Charts – 1st and 3rd columns) and its frequency variation (Waveform Graph – 2nd and 4th columns)

2.5. The automatic mode of the EEG signal acquisition

The LabVIEW programming sequence necessary to implement the automatic mode of EEG signal acquisition has a similar content to that needed for the manual mode EEG data recording, presented in the previous section. Nevertheless, there are some differences and an important particularity is related to the implementation of another state machine design pattern for the simulation of a virtual chronometer (Figure 10) able to calculate and display both the elapsed and remained time. Therefore, if the user selects ‘Automatic’ mode, then he/she should set the following initial parameters: ‘Target Time’ (hours, minutes and seconds) meaning the duration of EEG signal acquisition, ‘Time Interval’ and the number of samples – ‘Samples to read’. The time interval is associated with the frequency rate of triggering a warning sound indicating that the user should execute a specific mental task related to the research conducted in the Brain-Computer Interface scientific field. For instance, if ‘Time Interval = 2 (seconds)’, then the user will hear a warning sound from 2 to 2 seconds, which indicates him/her when it is the right moment to execute a voluntary eye-blink. In this case, performing eye-blinks is considered the mental task, because it is an artefact across the acquired EEG signal and it can be precisely detected and categorized as a robust control signal.



AUTOMATIC MODE

SAMPLES TO READ: 512

TIME INTERVAL: 2

	HOURS	MINUTES	SECONDS
TARGET TIME	0	1	20
CURRENT TIME	0	1	6
REMAINED TIME	0	0	14

serialPortName: COM6

Signal Quality: 100

Figure 10. A sequence of the Front Panel showing the virtual chronometer aimed to calculate both the elapsed and remained time for the automatic acquisition of EEG signal

Figure 11 shows the proposed experimental paradigm or the algorithm underlying the automatic mode of EEG data acquisition. If the 'Target Time = 80 (seconds)', 'Time Interval = 2 (seconds) and 'Samples to Read = 512', then it will result 80 temporal sequences of EEG signals, each of them having a length equal to 1 second of recording or it is equivalent with the acquisition of 512 samples. At each second, it will result a 1D array or a set of 512 samples for every one of the 12 EEG signals (both time and frequency domain).

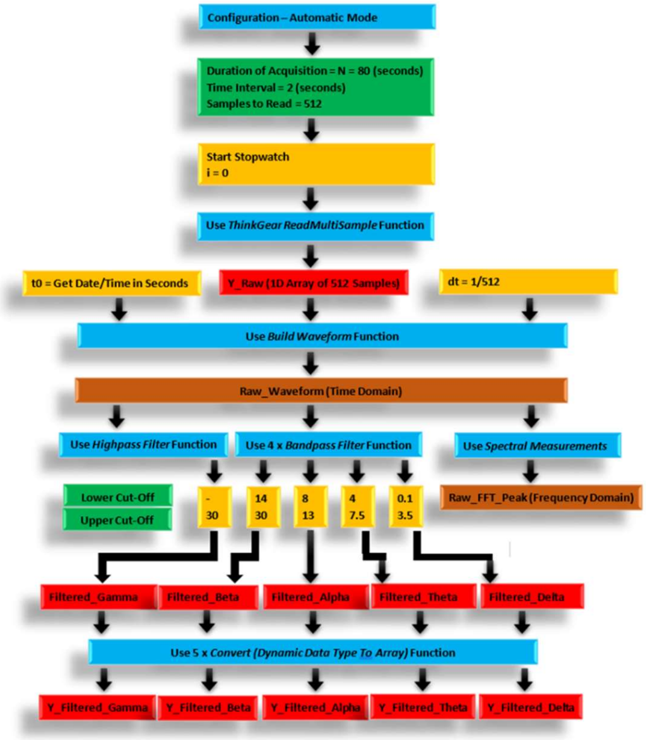

Configuration – Automatic Mode
Duration of Acquisition = N = 80 (seconds)
Time Interval = 2 (seconds)
Samples to Read = 512
Start Stopwatch
i = 0
Use ThinkGear ReadMultiSample Function
t0 = Get Date/Time in Seconds
Y_Raw (1D Array of 512 Samples)
dt = 1/512
Use Build Waveform Function
Raw_Waveform (Time Domain)
Use Highpass Filter Function
Use 4 x Bandpass Filter Function
Use Spectral Measurements
Lower Cut-Off
Upper Cut-Off
-
30
14
30
8
13
4
7.5
0.1
3.5
Raw_FFT_Peak (Frequency Domain)
Filtered_Gamma
Filtered_Beta
Filtered_Alpha
Filtered_Theta
Filtered_Delta
Use 5 x Convert (Dynamic Data Type To Array) Function
Y_Filtered_Gamma
Y_Filtered_Beta
Y_Filtered_Alpha
Y_Filtered_Theta
Y_Filtered_Delta

Figure 11. A diagram representing the block instructions underlying the implementation of Data Acquisition in the Automatic Mode – first view showing all the steps leading to obtaining the raw EEG signal (time and frequency domain) and extracting the EEG rhythms (gamma, beta, alpha, theta and delta)

Accordingly, when the chronometer stops, indicating that the EEG signal acquisition in automatic mode is finished, then 12 x 2D arrays will be returned. They are associated to 6 types of EEG signals in Time Domain (Figure 12) plus 6 types of EEG signals in Frequency Domain (FFT – Peak – Figure 13): raw, delta, theta, alpha, beta and gamma. A

2D array is represented by a matrix containing 80 rows (temporal sequences) and 512 columns (512 samples).

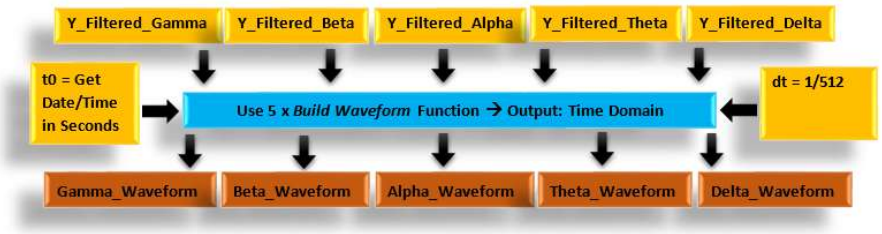


Figure 12. A diagram representing the block instructions underlying the implementation of EEG Data Acquisition in the Automatic Mode – the second view showing the five EEG rhythms obtained in the time domain.

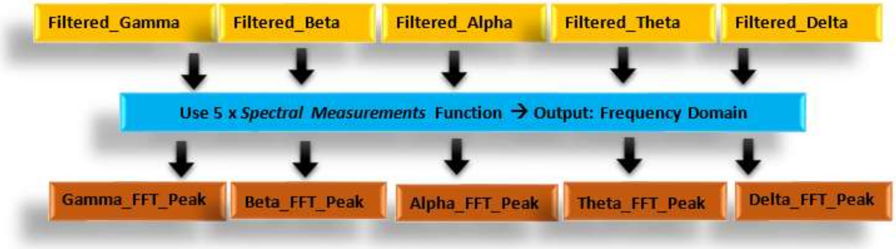


Figure 13. A diagram representing the block instructions underlying the implementation of EEG Data Acquisition in the Automatic Mode – the third view showing the five EEG rhythms obtained in the frequency domain.

2.6. The preparation of the EEG temporal sequences

Before applying the algorithm of preparation of the EEG acquired data, every one of 12 x 2D arrays contains: N rows and 'Samples to Read' Columns = 80 rows and 512 columns → 80 temporal sequences of 512 elements. Every one of the 12 x 2D arrays (both time and frequency domain of raw, gamma, beta, alpha, theta, delta) is organized as in Table 1.

Table 1. Structure of the 2D arrays (Time and Frequency Domain of all EEG Signals) before the preparation of the EEG acquired data

Temporal Sequence	Samples to Read – First Index	….	Samples to Read – The Last Index
i = 0	0	….	511
i = 1	512	….	1023
i = 2	1024	….	1535
i = 3	1536		2047
….	….	….	….
i = 38	19456	….	19967
i = 39	19968	….	20479
….	….	….	….
i = 78	39936	….	40447
i = 79	40448	….	40959

After applying the algorithm of preparation of the EEG acquired data, every one of 12 x 2D arrays contains: 'N divided by Time Interval' rows and 'Time Interval multiplied by Samples to Read' Columns = 40 rows and 2 x 512 columns = 40 rows and 1024 columns ◎ 40 sequences of 1024 elements. Every one of the 12 x 2D arrays (both time and frequency

doi:10.20944/preprints202106.0016.v1

domain of raw, gamma, beta, alpha, theta, delta) is organized as in Table 2. Further, the extraction or calculation of features (for example: mean, median, standard deviation) will be applied to every one of the resulted 40 sequences, each of them containing 1024 elements.

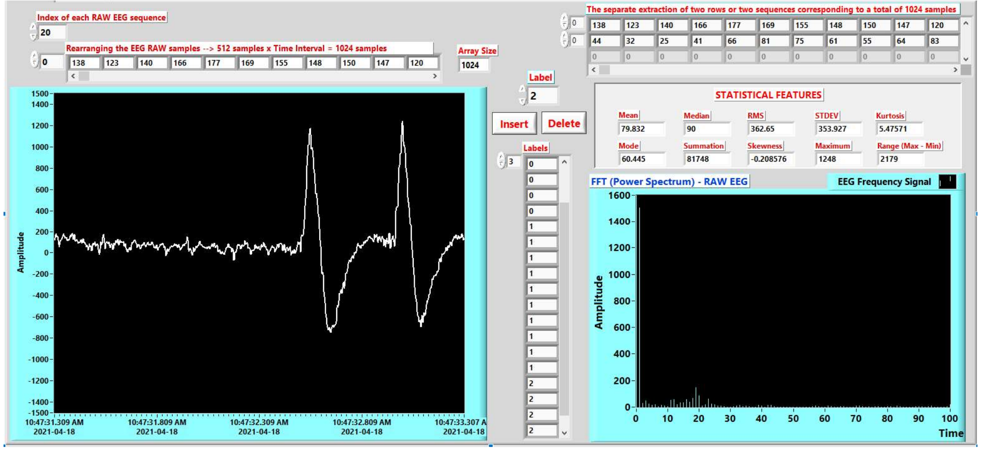
The algorithm of preparation of the acquired EEG data includes three stages. The first stage is related to using the predefined 'Read Delimited Spreadsheet VI' to read each of the 12 x 2D arrays, each of them containing 40960 samples corresponding to the EEG rhythms that were previously saved to .csv files. The second stage is consisting of the implementation of a customized VI aiming at the conversion of each of the 12 x 2D arrays into 12 x 3D arrays, the third dimension resulting from the separate extraction of two rows or two sequences that are composed of a total of 1024 samples. The third stage is related to the implementation of another customized VI to achieve the conversion of each of the 12 x 3D arrays into 12 x 2D arrays, so that the third dimension was removed because the previously extracted two rows or two sequences were merged together in a single row or a single sequence and all the resulted rows/sequences were stored into a 2D array.

Table 2. Structure of the 2D arrays (Time and Frequency Domain of all EEG Signals) after applying the preparation of the EEG acquired data

Temporal Sequence	Samples to Read – First Index	….	Samples to Read – The Last Index
i = 0	**0**	**….**	**1023**
i = 1	**1024**	**….**	**2047**
….	**….**	**….**	**….**
i = 19	**19456**	**….**	**20479**
….	**….**	**….**	**….**
i = 39	**39936**	**….**	**40959**

2.7. The label assignment for each EEG temporal sequence by visually checking its graphical displaying in time and frequency domains

After the preparation of EEG data is finished, according to Figure 14, the user has the possibility to manually set the label for each EEG temporal sequence by visually checking its graphical displaying in time and frequency domains. Figure 14 shows the options and settings related to checking the raw EEG signal. Other tabs / graphical windows with similar content were added to assess each of the EEG rhythms (delta, theta, alpha, beta and gamma).



Index of each RAW EEG sequence: 20

Rearranging the EEG RAW samples --> 512 samples x Time Interval = 1024 samples

0	138	123	140	166	177	169	155	148	150	147	120

Array Size: 1024

The separate extraction of two rows or two sequences corresponding to a total of 1024 samples

0	138	123	140	166	177	169	155	148	150	147	120
0	44	32	25	41	66	81	75	61	55	64	83
	0	0	0	0	0	0	0	0	0	0	0

Label: 2

Insert | Delete

Labels: 3

0
0
0
0
1
1
1
1
1
1
1
1
1
1
2
2
2
2

STATISTICAL FEATURES

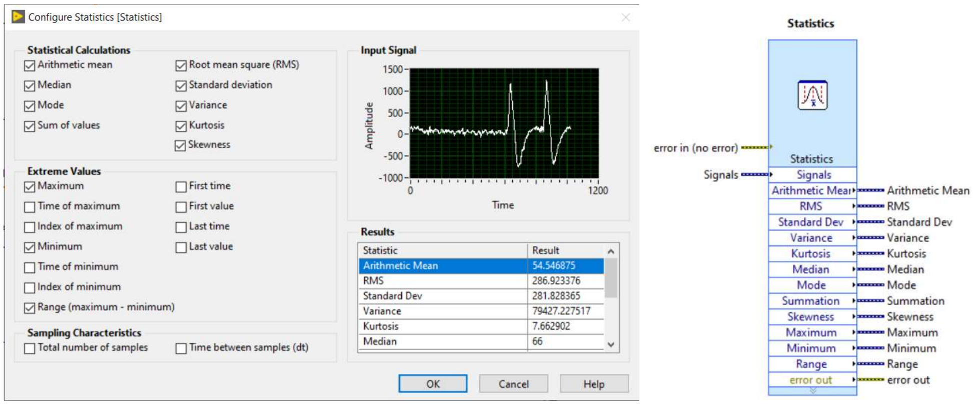
Mean	Median	RMS	STDEV	Kurtosis
79.832	90	362.65	353.927	5.47571
Mode	**Summation**	**Skewness**	**Maximum**	**Range (Max - Min)**
60.445	81748	-0.208576	1248	2179

Amplitude
1500
1400
1200
1000
800
600
400
200
0
-200
-400
-600
-800
-1000
-1200
-1400
-1500
10:47:31.309 AM 2021-04-18
10:47:31.809 AM 2021-04-18
10:47:32.309 AM 2021-04-18
10:47:32.809 AM 2021-04-18
10:47:33.307 A 2021-04-18
FFT (Power Spectrum) - RAW EEG
EEG Frequency Signal
Amplitude
1600
1400
1200
1000
800
600
400
200
0
0
10
20
30
40
50
60
70
80
90
100
Time

Figure 14. A sequence of the Front Panel – EEG Raw Signal - showing various options allowing the label assignment for each EEG temporal sequence by visually checking its graphical displaying in time and frequency domains

Therefore, by the implementation of original LabVIEW programming functions, a numerical index can be incremented or decremented to enable the switching between the EEG temporal sequences. Then, by the selection of the corresponding virtual button, the user is allowed to insert the label associated to the currently displayed EEG temporal sequence. To remove a wrong value, that was inserted by mistake, the user can select the button allowing the deletion of that label.

Each label is stored in a numerical array which is further used to the generation of the EEG datasets aimed for the training and testing of a neural networks model. By using the 'Statistics Express VI' that is included by the 'Signal Analysis Express VIs' LabVIEW functions palette, the statistical features determining the content of the EEG datasets are calculated and displayed for each EEG temporal sequence. According to Figure 15, there were selected the following statistical features: arithmetic mean, median, mode, sum of values, root mean square (RMS), standard deviation, variance, Kurtosis, skewness, maximum, minimum and range (maximum – minimum).



Configure Statistics [Statistics]

Statistical Calculations

☑ Arithmetic mean
☑ Median
☑ Mode
☑ Sum of values
☑ Root mean square (RMS)
☑ Standard deviation
☑ Variance
☑ Kurtosis
☑ Skewness

Extreme Values

☑ Maximum
□ Time of maximum
□ Index of maximum
☑ Minimum
□ Time of minimum
□ Index of minimum
☑ Range (maximum - minimum)
□ First time
□ First value
□ Last time
□ Last value

Sampling Characteristics

□ Total number of samples
□ Time between samples (dt)

Results

Statistic	Result
Arithmetic Mean	54.546875
RMS	286.923376
Standard Dev	281.828365
Variance	79427.227517
Kurtosis	7.662902
Median	66

Figure 15. The settings corresponding to 'Statistics Express VI' included by the 'Signal Analysis Express VIs' LabVIEW palette

As it was previously stated, the working principle of the proposed LabVIEW application is applied on the classification of multiple voluntary eye-blinks, although it provides the benefit of processing and recognition of various EEG patterns associated to other cognitive tasks. This way, the user should set the label of an EEG temporal sequence to one of the following values: 0 (if no eye-blink was detected), 1 (if one eye-blink was detected), 2 (if two eye-blinks were detected) and 3 (if three eye-blinks were detected), as they are displayed in Figure 16.

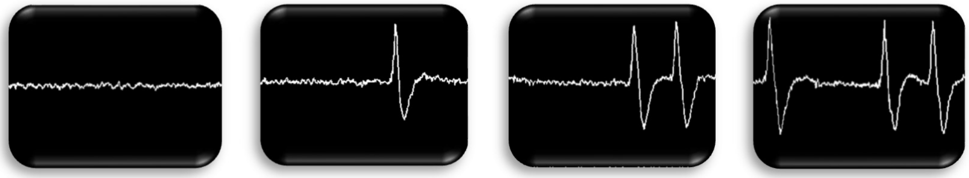


Figure 16. An example for the LabVIEW displaying of the EEG temporal sequences associated to each of the labels: 0 – No Eye-Blink; 1 – One Eye-Blink; 2 – Two Eye-Blinks and 3 – Three Eye-Blinks.

2.8. The generation of training and testing EEG dataset

It is based on choosing multiple mixtures between the selected EEG signals and the extracted features. The boxes from the top side of the Figure 17 represent 12 x 2D_arrays,

doi:10.20944/preprints202106.0016.v1

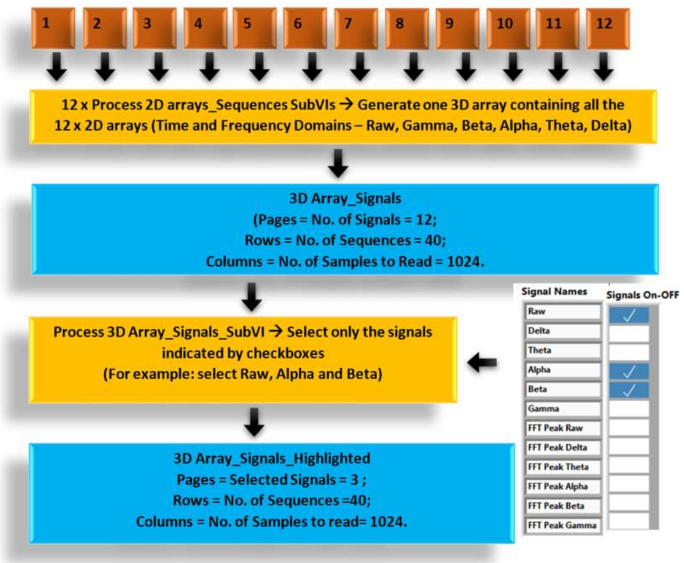
each of them containing 40 rows and 1024 columns or 40960 samples. Also, these boxes represent:

- 6 x 2D arrays for Time Domain – Waveform_Y_Data of Raw, Gamma, Beta, Alpha, Theta and Delta;
- 6 x 2D arrays for Frequency Domain – FFT_Peak of Raw, Gamma, Beta, Alpha, Theta and Delta.

Figures 17-19 show the graphical diagrams explaining the process of the EEG dataset generation.

The first customized subVI, called 'Process 2D arrays_Sequences', was developed for the purpose of inserting all the 12 x 2D_arrays into one 3D array, whose structure includes: 12 pages corresponding to the 12 EEG signals, 40 rows associated to the 40 temporal sequences and 1024 columns related to the 1024 samples.

The second customized subVI, called 'Process 3D_Array_Signals_SubVI', was implemented to achieve the extraction of only those 2D arrays corresponding to the signals that were previously selected by using the checkboxes. Therefore, the resulted 3D array is comprised of: a certain number of pages that is equal to the number of selected signals, 40 rows and 1024 samples.


1
2
3
4
5
6
7
8
9
10
11
12
12 x Process 2D arrays_Sequences SubVIs → Generate one 3D array containing all the 12 x 2D arrays (Time and Frequency Domains – Raw, Gamma, Beta, Alpha, Theta, Delta)
3D Array_Signals
(Pages = No. of Signals = 12;
Rows = No. of Sequences = 40;
Columns = No. of Samples to Read = 1024.
Process 3D Array_Signals_SubVI → Select only the signals indicated by checkboxes
(For example: select Raw, Alpha and Beta)
3D Array_Signals_Highlighted
Pages = Selected Signals = 3 ;
Rows = No. of Sequences =40;
Columns = No. of Samples to read= 1024.
Signal Names
Signals On-OFF
Raw ✓
Delta
Theta
Alpha ✓
Beta ✓
Gamma
FFT Peak Raw
FFT Peak Delta
FFT Peak Theta
FFT Peak Alpha
FFT Peak Beta
FFT Peak Gamma

THE RESULT – 3D Array_Signals_Highlighted

	Page = 0 (Raw EEG)			Page = 1 (Alpha EEG)			Page = 2 (Beta EEG)		
i = 0	0	….	1023	0	….	1023	0	….	1023
i = 1	1024	….	2047	1024	….	2047	1024	….	2047
….	….	….	….	….	….	….	….	….	….
….	….	….	….	….	….	….	….	….	….
….	….	….	….	….	….	….	….	….	….
….	….	….	….	….	….	….	….	….	….
i = 39	39936	….	40959	39936	….	40959	39936	….	40959

Figure 17. First view of the diagram representing the block instructions underlying the generation of training or testing dataset based on multiple mixtures between the selected EEG signals and the extracted features

The third customized subVI, called 'Process 3D Array_Signals_Highlighted_SubVI', was created to enable the process of feature extraction by the calculation of the specified statistical measurements against the previously selected signals. Thus, a new dimension is added so that the resulted 4D array is consisting of: a certain number of volumes equal to the number of selected signals, 40 pages corresponding to the 40 temporal sequences, a certain number of rows equal to the number of the extracted features and one column related to the single value of each feature.

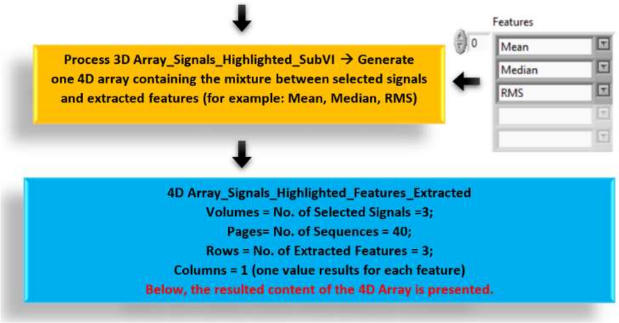
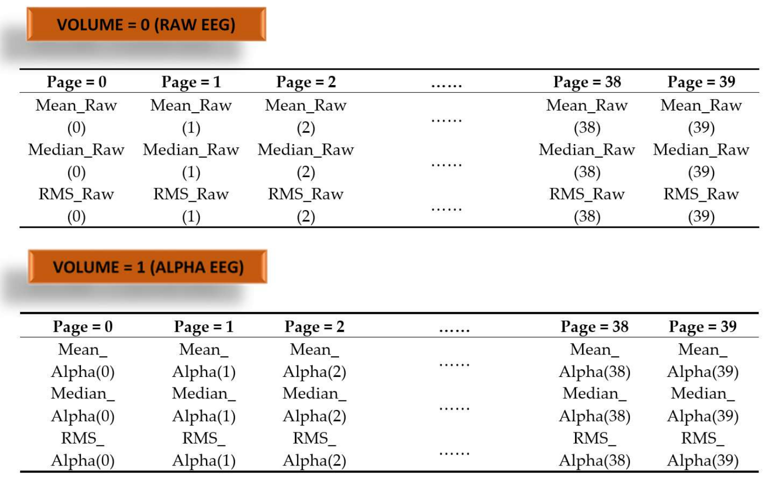

Features
0
Mean
Median
RMS
Process 3D Array_Signals_Highlighted_SubVI → Generate one 4D array containing the mixture between selected signals and extracted features (for example: Mean, Median, RMS)
4D Array_Signals_Highlighted_Features_Extracted
Volumes = No. of Selected Signals =3;
Pages= No. of Sequences = 40;
Rows = No. of Extracted Features = 3;
Columns = 1 (one value results for each feature)
Below, the resulted content of the 4D Array is presented.

Figure 18. The second view of the diagram representing the block instructions underlying the generation of training or testing dataset based on multiple mixtures between the selected EEG signals and the extracted features



VOLUME = 0 (RAW EEG)

Page = 0	Page = 1	Page = 2	......	Page = 38	Page = 39
Mean_Raw (0)	Mean_Raw (1)	Mean_Raw (2)	......	Mean_Raw (38)	Mean_Raw (39)
Median_Raw (0)	Median_Raw (1)	Median_Raw (2)	......	Median_Raw (38)	Median_Raw (39)
RMS_Raw (0)	RMS_Raw (1)	RMS_Raw (2)	......	RMS_Raw (38)	RMS_Raw (39)

VOLUME = 1 (ALPHA EEG)

Page = 0	Page = 1	Page = 2	......	Page = 38	Page = 39
Mean_ Alpha(0)	Mean_ Alpha(1)	Mean_ Alpha(2)	......	Mean_ Alpha(38)	Mean_ Alpha(39)
Median_ Alpha(0)	Median_ Alpha(1)	Median_ Alpha(2)	......	Median_ Alpha(38)	Median_ Alpha(39)
RMS_ Alpha(0)	RMS_ Alpha(1)	RMS_ Alpha(2)	......	RMS_ Alpha(38)	RMS_ Alpha(39)

Figure 19. The third view of the diagram representing the block instructions underlying the generation of training or testing dataset based on multiple mixtures between the selected EEG signals and the extracted features

The fourth subVI, called 'Process 4D Array_Signals_Highlighted_Features_Extracted' (Figure 20), was developed to re-organize and reduce the dimensionality of the obtained data, by the generation of one 3D array that is comprising of: 40 pages associated with the 40 temporal sequences, a certain number of rows equal to the number of the selected signals and a certain number of columns equal to the number of the extracted features.

The fifth subVI, called 'Process 3D Array_Signals_Highlighed_Features_Extracted', was implemented as a final stage of re-organization and reducing the dimensionality of the obtained data, by the generation of a 2D array that is consisting of: 40 rows corresponding to 40 temporal sequences, a certain number of columns involving the number of selected signals and the number of the extracted features. The generated 2D array included numerical elements that were converted to String type. Then, the 2D array_ Signals_Highlighted_ Features_Extracted (Figure 21) of string elements will be saved to a .csv file representing the training/testing dataset. The .csv file is consisting of a table with 11 columns (Sequence; Mean_Raw; Mean_Alpha, Mean_Beta; Median_Raw; Median_Alpha; Median_Beta; RMS_Raw; RMS_Alpha; RMS_Beta; Label) and 40.

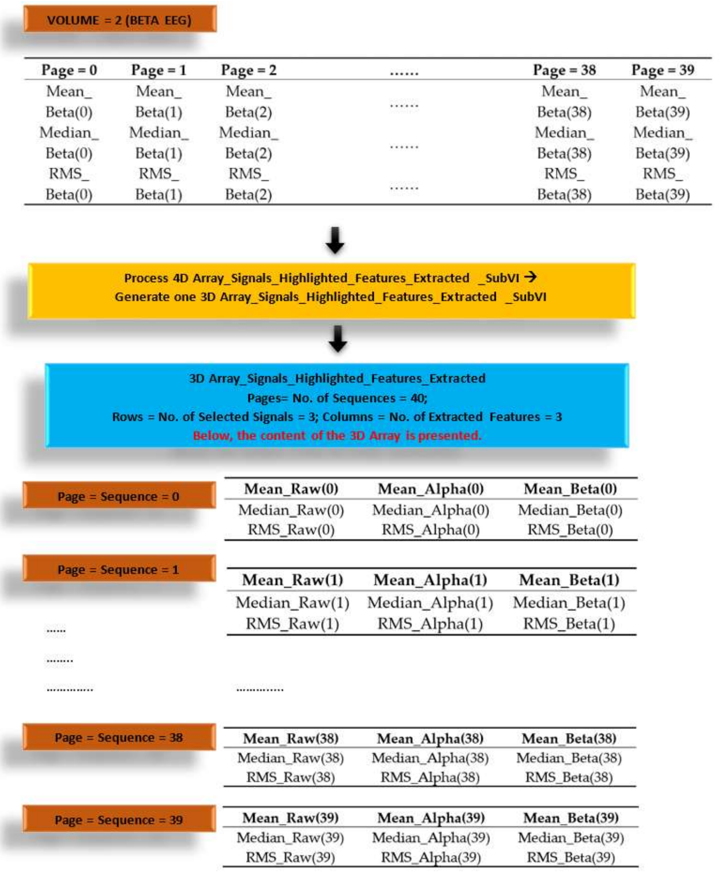

VOLUME = 2 (BETA EEG)
Page = 0
Page = 1
Page = 2
……
Page = 38
Page = 39
Mean_ Beta(0)
Median_ Beta(0)
RMS_ Beta(0)
Mean_ Beta(1)
Median_ Beta(1)
RMS_ Beta(1)
Mean_ Beta(2)
Median_ Beta(2)
RMS_ Beta(2)
Mean_ Beta(38)
Median_ Beta(38)
RMS_ Beta(38)
Mean_ Beta(39)
Median_ Beta(39)
RMS_ Beta(39)
Process 4D Array_Signals_Highlighted_Features_Extracted _SubVI →
Generate one 3D Array_Signals_Highlighted_Features_Extracted _SubVI
3D Array_Signals_Highlighted_Features_Extracted
Pages= No. of Sequences = 40;
Rows = No. of Selected Signals = 3; Columns = No. of Extracted Features = 3
Below, the content of the 3D Array is presented.
Page = Sequence = 0
Mean_Raw(0) Mean_Alpha(0) Mean_Beta(0)
Median_Raw(0) Median_Alpha(0) Median_Beta(0)
RMS_Raw(0) RMS_Alpha(0) RMS_Beta(0)
Page = Sequence = 1
Mean_Raw(1) Mean_Alpha(1) Mean_Beta(1)
Median_Raw(1) Median_Alpha(1) Median_Beta(1)
RMS_Raw(1) RMS_Alpha(1) RMS_Beta(1)
Page = Sequence = 38
Mean_Raw(38) Mean_Alpha(38) Mean_Beta(38)
Median_Raw(38) Median_Alpha(38) Median_Beta(38)
RMS_Raw(38) RMS_Alpha(38) RMS_Beta(38)
Page = Sequence = 39
Mean_Raw(39) Mean_Alpha(39) Mean_Beta(39)
Median_Raw(39) Median_Alpha(39) Median_Beta(39)
RMS_Raw(39) RMS_Alpha(39) RMS_Beta(39)

Figure 20. Forth view of the diagram representing the block instructions underlying the generation of training or testing dataset based on multiple mixtures between the selected EEG signals and the extracted features

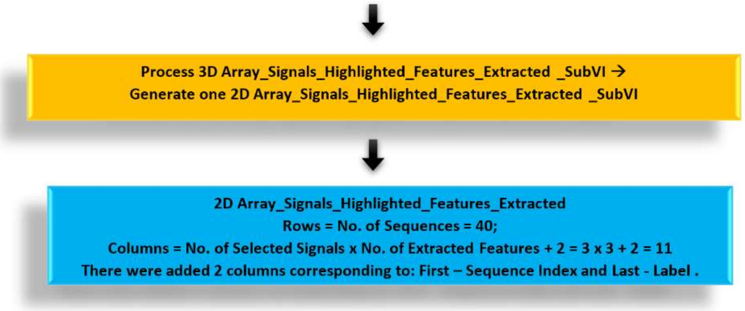

Process 3D Array_Signals_Highlighted_Features_Extracted _SubVI →
Generate one 2D Array_Signals_Highlighted_Features_Extracted _SubVI
2D Array_Signals_Highlighted_Features_Extracted
Rows = No. of Sequences = 40;
Columns = No. of Selected Signals x No. of Extracted Features + 2 = 3 x 3 + 2 = 11
There were added 2 columns corresponding to: First – Sequence Index and Last - Label .

Figure 21. Fifth view of the diagram representing the block instructions underlying the generation of training or testing dataset based on multiple mixtures between the selected EEG signals and the extracted features

2.9. Training a NN model for the EEG Signals classification by setting certain hyper-parameters

This phase is consisting of applying the generated dataset to the classification process based on artificial neural networks (NN). The classification process is facilitated by using the default subVIs included by the ‘Analytics and Machine Learning’ (AML) toolkit.

‘Aml_Read csv File.vi’ is used to open the csv file and read the training dataset. ‘Load Training Data (2D Array).vi’ is used to load the dataset for training the model.

doi:10.20944/preprints202106.0016.v1

'Normalize.vi' is used to normalize the training data with 2-Score or Min-Max Method. Normalization is related to the scaling each value of training dataset in the specified range. The 'Normalize.vi' has two parameters: one shot and batch.

'Initialize Classification Model (NN).vi' initializes the parameter of the classification algorithm: neural networks (NN). The user should set a certain value for every hyperparameter: the number of hidden neurons, the hidden layer type (Sigmoid, Tanh or Rectified Linear Unit functions), the output layer type (Sigmoid or Softmax function), the cost function type (Quadratic or Cross-Entropy function), tolerance and max iteration.

According to the documentation of AML LabVIEW toolkit, the 'hidden layer type' is related to the type of activation function applied to the neurons from the hidden layer type. The available activation functions are defined in Table 3. Regarding the 'output layer type', according to Table 4, Sigmoid and Softmax are the two activation functions available in the neurons. The mathematical formulas for the supported cost functions type are given in the Table 5.

The criteria determining the stopping of training or fitting the neural networks model is given by the value of either tolerance or max iteration parameter. The tolerance specifies the training error and the max iteration specifies the maximum number of optimization iterations. The default value for tolerance is 0.0001. The default value for max iteration is 1000.

Table 3. Information about the available activation functions determining the hidden layer type set as a hyper-parameter included by the 'Analytics and Machine Learning' LabVIEW toolkit

Funtion type	Definition	Description
Sigmoid	$f(x) = \frac{1}{1 + e^{-x}}$	x - the activation value of the hidden neuron
Tanh	$f(x) = \tanh(x)$	x - the activation value of the hidden neuron
ReLU	$f(x) = \max(0, x)$	x - the activation value of the hidden neuron

Table 4. Information about the available activation functions determining the output layer type set as a hyper-parameter included by the 'Analytics and Machine Learning' LabVIEW toolkit

Funtion type	Definition	Description
Sigmoid	$f(x) = \frac{1}{1 + e^{-x}}$	x - the activation value of the output neuron
Softmax	$f(x) = \frac{e^{x_i}}{\sum_{j=1}^{n} e^{-x_j}}$	x - the activation value of the output neuron

Table 5. Information about the available cost functions type set as a hyper-parameter included by the 'Analytics and Machine Learning' LabVIEW toolkit

Funtion type	Definition	Description
Quadratic	$error = \frac{1}{l}(t - y)^2$	t – the target value, y – output value, l – number of training samples
Cross-entropy	$f(x) = -\frac{1}{l}(\sum_{j=1}^{n} t_j lny_j)$	t – the target value, y – output value, l – number of training samples and n – number of classes

Likewise, the user can set the 'Cross Validation Configuration', which is an input cluster containing the following elements: a Boolean control called 'enable' (used to enable or disable cross validation in training model), number of folds (defining the number of sections that this VI divides the training data into) and metric configuration (average method: micro, macro, weighted or binary).

'Train Classification Model.vi' is used to train a classification model. 'Aml_save Model to JSON.vi' is used to save the model as a JSON file. It converts the trained model to a JSON string to save the trained model to a file.

According to the documentation of AML LabVIEW toolkit, by enabling the 'Cross Validation Configuration', confusion matrix and metrics are returned as output values of the 'Train Classification Model.vi'. The default number of folds is 3, meaning that the test data is consisting of one section and the training data is comprising of the remaining sections. The metric configuration parameter determines the evaluation metric in cross validation. The neural networks models trained by the proposed LabVIEW application involved the type called 'weighted metric configuration'.

Figure 22 shows the entire structure of the previously described AML functions used to enable the setting of hyperparameters for training the neural networks model. Figure 23 shows the graphical user interface of this LabVIEW programming sequence.

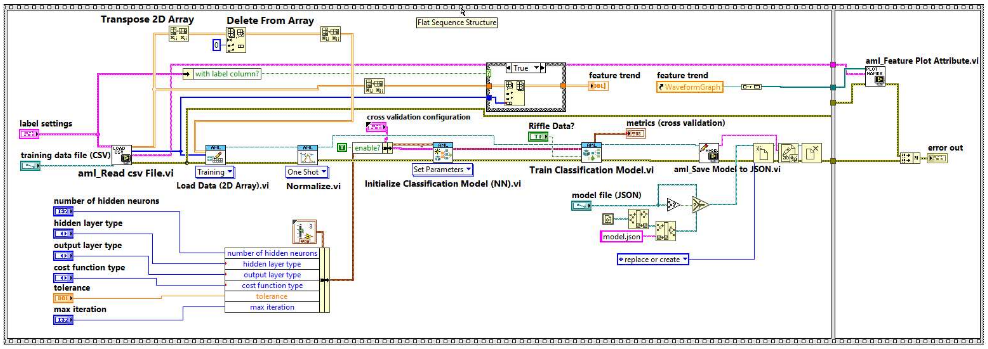
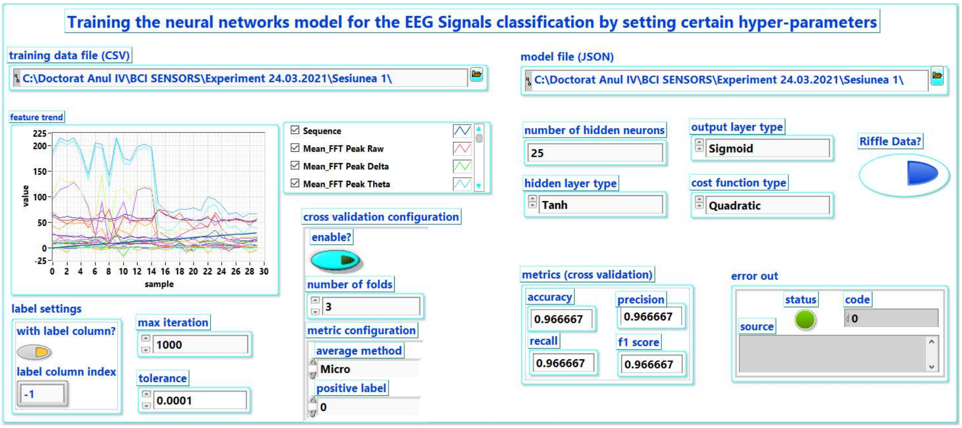


Figure 22. A sequence of the Block Diagram displaying the corresponding LabVIEW programming functions used to enable the setting of hyperparameters for training the neural networks model



Training the neural networks model for the EEG Signals classification by setting certain hyper-parameters

training data file (CSV): C:\Doctorat Anul IV\BCI SENSORS\Experiment 24.03.2021\Sesiunea 1\

model file (JSON): C:\Doctorat Anul IV\BCI SENSORS\Experiment 24.03.2021\Sesiunea 1\

number of hidden neurons: 25

output layer type: Sigmoid

Riffle Data?

hidden layer type: Tanh

cost function type: Quadratic

cross validation configuration
enable?
number of folds: 3
metric configuration
average method: Micro
positive label: 0

metrics (cross validation)
accuracy: 0.966667
precision: 0.966667
recall: 0.966667
f1 score: 0.966667

error out
status
code: 0
source

label settings
with label column?
label column index: -1

max iteration: 1000

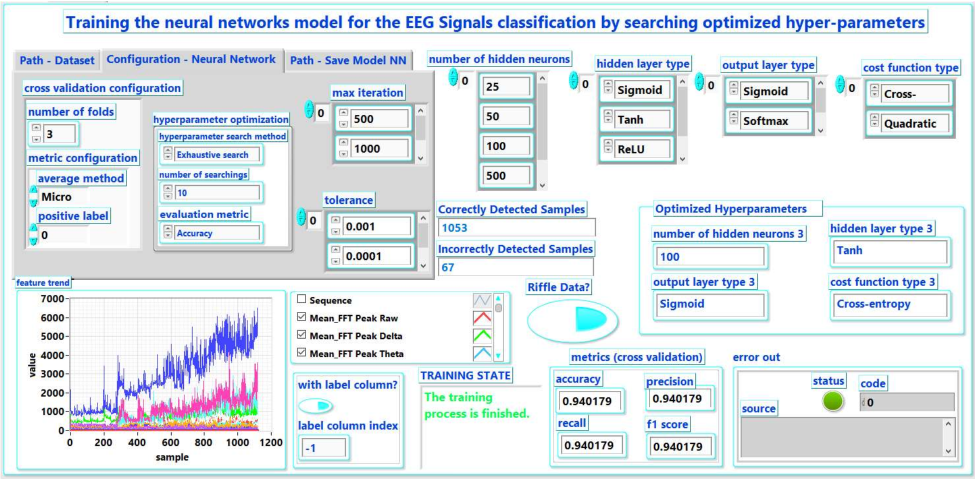
tolerance: 0.0001

Figure 23. A sequence of the Front Panel showing the graphical window corresponding to training the neural networks model for the EEG signal classification by setting certain hyper-parameters

2.10. Training the NN model for the EEG Signals classification by searching the optimal hyper-parameters

All the information presented in the above section – Classification by setting the parameters – are also applicable to the current section – Classification by searching the optimal parameters. Nevertheless, there is a single exception related to the 'Initialize Classification Model (NN).vi'. According to Figure 24, the user should specify multiple values for each hyper-parameter so that the 'Train Classification Model.vi' could use a grid search to find the optimal set of parameters. This technique was chosen for training the neural networks models from the current research paper because it is more reliable, simple and efficient by enabling the option 'Exhaustive Search' so that the metrics (accuracy, precision, recall and F1 score) are tested with the all the possible mixtures between hyper-parameters. It will result a mixture including the optimal hyper-parameters necessary to get the highest values for the metric that is specified in the 'Evaluation Metric' parameter. If the option 'Random Search' was enabled, then the 'number of searchings' parameter indicates that only some of the possible mixtures between hyper-parameters will be tested.

Moreover, the graphical user interface from the Figure 24 displays the number of correctly/incorrectly detected samples/temporal sequences, calculated by taking into account the mathematical formulas for the metrics described in Table 6.



Training the neural networks model for the EEG Signals classification by searching optimized hyper-parameters

Path - Dataset | Configuration - Neural Network | Path - Save Model NN

cross validation configuration
number of folds: 3
metric configuration
average method: Micro
positive label: 0

hyperparameter optimization
hyperparameter search method: Exhaustive search
number of searchings: 10
evaluation metric: Accuracy

max iteration: 0 | 500, 1000
tolerance: 0 | 0.001, 0.0001

number of hidden neurons: 0 | 25, 50, 100, 500
hidden layer type: 0 | Sigmoid, Tanh, ReLU
output layer type: 0 | Sigmoid, Softmax
cost function type: 0 | Cross-, Quadratic

Correctly Detected Samples: 1053
Incorrectly Detected Samples: 67
Riffle Data?

Optimized Hyperparameters
number of hidden neurons 3: 100
hidden layer type 3: Tanh
output layer type 3: Sigmoid
cost function type 3: Cross-entropy

☐ Sequence
☑ Mean_FFT Peak Raw
☑ Mean_FFT Peak Delta
☑ Mean_FFT Peak Theta

with label column?
label column index: -1

TRAINING STATE: The training process is finished.

metrics (cross validation)
accuracy: 0.940179
precision: 0.940179
recall: 0.940179
f1 score: 0.940179

error out
status
code: 0
source

Figure 24. A sequence of the Front Panel showing the graphical window corresponding to training the NN model for the EEG signal classification by searching the optimized hyper-parameters

Table 6. The mathematical formulas for the evaluation metrics (accuracy, precision, f1 score, recall) as they are described in the documentation of AML LabVIEW toolkit

Funtion type	Definition	Description
Accuracy	$Acc = \dfrac{TP + TN}{P + N}$	TP – number of true positive cases TN – number of true positive cases FP – number of false positive cases FN – number of false negative cases P – number of real positive cases N – number of real negative cases
Precision	$Prec = \dfrac{TP}{TP + FP}$	
F1 Score	$F1 = \dfrac{2TP}{2TP + FP + FN}$	
Recall	$Rec = \dfrac{TP}{TP + TN}$	

2.11. The flexibility of enabling/disabling the randomization of the normalized EEG data

It should be emphasized the following behaviour: whenever the previous phases related to the classification process are running, it results different values for the evaluation parameters of the NN model: accuracy, precision, recall and F1 score. These parameters are varying due to the use of a 'Random Number' function included in a subVI contained by the 'Train Classification Model.vi'. The training dataset is normalized and randomly distributed in subsets by using 'Random Number' function. The total number of subsets can be configured when the model is initialized. One of these subsets is aimed for the pre-training phase, while the others are aimed for the pre-testing phase. These phases are handled before the obtaining of the trained classification model. If the 'Random

doi:10.20944/preprints202106.0016.v1

Number' function is removed, then it will result the same evaluation parameters after running each session of training. It should be accomplished the condition of keeping the same configuration of the model to get identical results after running each training session.

The LabVIEW application presented in this paper provides more flexibility by the implementation of a novel method allowing the use of a button with two logical states, called 'Riffle Data', so that it can activate or deactivate the 'Random Number' function. Thus, the LabVIEW application offers the possibility to interactively enable or disable the random generation of the normalized EEG data. This possibility is achieved by the implementation of some changes in the subVIs provided by the 'Analytics and Machine Learning' toolkit. The changes are related to certain concepts of object-oriented programming in LabVIEW and they are explained in the below paragraphs. These changes are necessary so that the 'Riffle Data' button could be available in the main LabVIEW application, outside the subVI in which it has been added.

It should be accessed the 'Analytics and Machine Learning.lvlib', which is the library of functions contained by the LabVIEW Toolkit. Then, it should be opened the following structure of folders: Classification/Data/2D array/Classes/AML 2D Array.lvclass, where it is found the 'AML 2D Array.ctl'. This element should be modified by adding to it a boolean control corresponding to the 'Riffle Data' button. This modification will be propagated to the 'aml_Read Data.vi' and 'aml_Write Data.vi', both of them containing a typedef control called 'data', to which should be also added the 'Riffle Data' button. Further, it should be opened the following structure of folders (Figure 25): Classification/Data/Common/subVIs, where it is found the 'Load Training Data (2D array).vi'. In this virtual instrument, a 'Bundle by Name' function should be used to add the 'Riffle Data' Boolean control (button) as a new input element to the 'data' cluster. Now, this button should be available in the main application, outside of the subVI where its functionality is implemented, as it is shown in the below paragraphs.

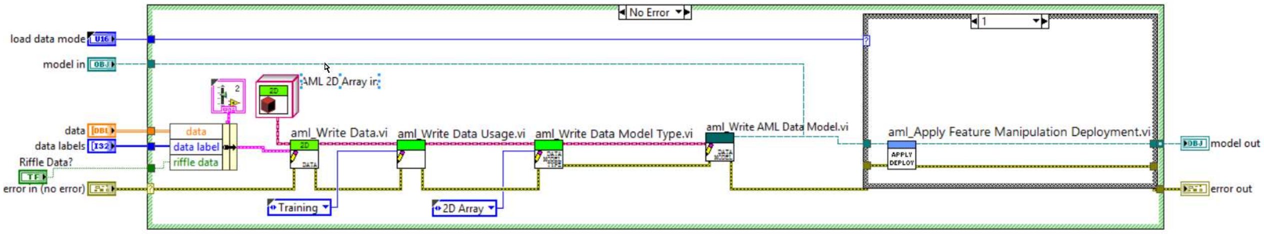


Figure 25. The Block Diagram of 'Load Training Data (2D array).vi' – modified by adding the 'Riffle Data' button

The following phases are executed to implement the possibility of removing the 'Random Number' Function applied to the training data.

- Open the Block Diagram (BL) of the 'Train Classification Model.vi'.
- Open the Block Diagram (BL) of the 'aml_Auto Tune.vi'.
- Choose 'AML Neural Network.lvclass: aml_Auto Tune.vi'.
- Open the Block Diagram of the 'AML Neural Network.lvclass: aml_Auto Tune.vi'.
- Open the Block Diagram of the 'aml_Cross Validation.vi'.
- Open the Block Diagram of the 'aml_Stratified K Folds.vi'.
- Open the Block Diagram of the 'aml_Riffle Training Data.vi' and modify it by adding a 'Select' Function.

'Select' function is corresponding to the 'if…else' statement from procedural programming. The implementation of the 'aml_ Riffle Training Data.vi' (Figure 26) is focused on re-organizing the training data in random order by using a 'Random Number' function. The 'Select' function is used to activate or deactivate the 'Random Number' function, based on the state of the 'Riffle Data' button, which is a Boolean control. It should be wired as an input terminal to the connector pane of the 'aml_Riffle Training Data.vi'

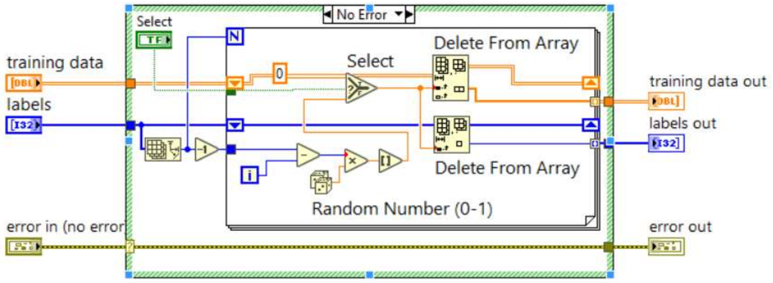


Figure 26. The Block Diagram of 'aml_Riffle Training Data.vi' – modified by adding the 'Select' Function to enable the possibility to activate or deactivate the 'Random Number' Function

- Modify the following virtual instruments: 'aml_Stratified K Folds.vi', 'aml_Cross Validation.vi', 'aml_Auto Tune.vi' and 'Train Classification Model.vi' so that it should be added a Boolean control, representing a push- handles the classification of a button, called 'Riffle Data'.

Then, it should be wired as an input terminal to the connector pane of the previously mentioned virtual instruments. Accordingly, the 'Riffle Data' button can be selected from the Front Panel of the LabVIEW application. If the button is enabled, then the 'Random Number' function is deactivated so that the training data is not randomized. If the button is disabled, then the 'Random Number' function is activated so that the training data is randomized.

2.12. The deployment/testing of the trained Neural Networks classification model

The LabVIEW programming sequence underlying this phase is consisting of certain virtual instruments which are presented in this section. First of all, 'aml_Read csv File.vi' is used to open and read data and labels from the .csv file containing the testing EEG dataset. Secondly, 'aml_Read Model from JSON.vi' is called to open and read the trained Neural Networks classification model. Thirdly, 'Load the test data file (2D Arrays).vi' is necessary to load or extract the data and labels in a format appropriate for the deployment of the model. Fourthly, 'Deploy Feature Manipulation Model.vi' is aimed for preprocessing the testing data by applying a trained feature manipulation model. Fifthly, 'Deploy Classification Model.vi' handles the classification of the testing dataset by applying the trained classification model and returning the predicted labels of input data. Sixthly, 'Evaluate Classification Model.vi' plays an important role in the assessment of the classification model by comparing the predicted labels with initially set labels of the input data. Finally, it will result evaluation metrics (accuracy, precision, recall and F1 score) for the trained and tested neural networks model.

2.13. The validation of the tested Neural Networks classification model by applying it on real-time EEG data

The research work presented in this paper includes the development of an independent LabVIEW application aimed to validate the trained and tested neural network classification model. Therefore, it is deployed on the EEG signals acquired in real-time. Thus, the working principle underlying this LabVIEW application is similar to the automatic mode of data acquisition, which was previously presented. First of all, the classification model should be opened and read by indicating the corresponding path of the location where the .json file was saved. Then, 'Samples to read' parameter should be set to the same value proposed in the automatic mode of data acquisition. Further, the mixture of selected EEG signals and extracted features should be also chosen according to the settings previously specified in the phase of training dataset preparation. All these initial configurations were saved in text files so that the user can open and remember them. After that, the real-time EEG signal acquisition can be started. The programming sequences underlying the implementation of the LabVIEW application are consisting of

doi:10.20944/preprints202106.0016.v1

similar processes (data acquisition, processing, features extraction and classification) with the phases described in the previous sections. Nevertheless, a significant difference is given by the dimensionality of the data because, at each time interval, only one temporal sequence (corresponding to a single row from the .csv file of the initial training dataset) is recorded in real-time. In the main application, the training data represented a total of 40 temporal sequences if the duration of the automatic acquisition was set to 1 minute and 20 seconds. The reduction of dimensionality of input data is associated with replacing the previously mentioned 4D arrays with 3D arrays. Accordingly, there were implemented the necessary modifications related to the structure of the Block Diagram of the independent LabVIEW application. Although it was used to detect voluntary eye-blinks across the EEG signal acquired in real-time, the developed LabVIEW instrument is also prepared to be tested on other EEG signal patterns corresponding to the execution of different cognitive tasks.

3. Results

The proposed LabVIEW application is aimed for the acquisition, processing and classification of the EEG signals corresponding to neuronal patterns elicited by different cognitive tasks. The eye-blink is considered an artefact across the EEG signal, but it can be also considered a precise control signal in a brain-computer interface application. The voluntary eye-blink, which resulted from the execution of a normal effort, is characterized by a simple pattern of a spike associated with the increasing and decreasing the biopotential. Therefore, if it does not require a higher amplitude or a strong effort, then the voluntary eye-blink is associated with a general pattern, that could be easy to detect even by visual checking of the EEG signal.

Thus, the working principle underlying the proposed LabVIEW application is tested on the classification of multiple voluntary eye-blinks based on the processing of the EEG temporal sequences consisting of multiple mixtures between several EEG rhythms (raw, delta, theta, alpha, beta, gamma) in Time and Frequency Domains and certain statistical features (mean, median, RMS, standard deviation, mode and others).

By the experiments conducted in the current research work, a total of 4000 temporal EEG sequences have been recorded from a single subject (female, 29 years) who is the first author of this paper. The duration of each session of EEG data acquisition was set to 1 minute and 20 seconds. For example, during every period of time equivalent to 80 seconds, at each time interval of 2 seconds, the subject had to accomplish one of the following four tasks: avoid the voluntary eye-blinks, execute one voluntary eye-blink, perform two voluntary eye-blinks and achieve three voluntary eye-blinks. A session of EEG data acquisition that was set to 80 seconds is corresponding to recording a series of 40 EEG temporal sequences. The results of the current research paper are based on 25 sessions of EEG data acquisition, each of them consisting of 40 EEG temporal sequences. Therefore, a total of 25 x 40 = 1000 EEG temporal sequences have been returned for each of the four classes: 0 – No Eye-Blink Detected; 1 – One Eye-Blink Detected; 2 – Two Eye-Blinks Detected and 3 – Three Eye-Blinks Detected. In fact, for the generation of the training dataset, the subject has been involved into 4 x 25 sessions of EEG data acquisition enabling the recording of the 4 x 1000 = 4000 EEG temporal sequences corresponding to the previously mentioned four classes. Otherwise, for the generation of the testing dataset, the subject has been involved into 4 x 5 sessions of EEG data acquisition enabling the recording of the 4 x 5 x 40 = 800 EEG temporal sequences corresponding to the previously mentioned four classes. The duration of each session of EEG data acquisition was set to 80 seconds. Figure 27 shows the previously described general structure. Both training and testing dataset include a column which is assigned the labels.

doi:10.20944/preprints202106.0016.v1

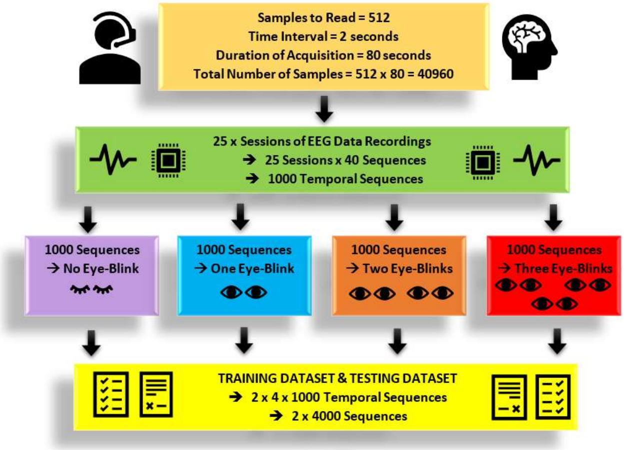

Samples to Read = 512
Time Interval = 2 seconds
Duration of Acquisition = 80 seconds
Total Number of Samples = 512 x 80 = 40960
25 x Sessions of EEG Data Recordings
→ 25 Sessions x 40 Sequences
→ 1000 Temporal Sequences
1000 Sequences
→ No Eye-Blink
1000 Sequences
→ One Eye-Blink
1000 Sequences
→ Two Eye-Blinks
1000 Sequences
→ Three Eye-Blinks
TRAINING DATASET & TESTING DATASET
→ 2 x 4 x 1000 Temporal Sequences
→ 2 x 4000 Sequences

Figure 27. An overview on the proposed paradigm for performing the experimental sessions to get the datasets used for both training and testing of the neural networks model of eye-blinks classification

Therefore, by visually checking the graphical representation of each EEG temporal sequence, the corresponding label is assigned. Moreover, if the initial aim was to get 40 sequences associated to a specific class (for example, one eye-blink), but it mistakenly resulted only 35 sequences correctly executed, the remained wrong 5 sequences can be automatically replaced with other correct 5 sequences, which were acquired during another session, that it is kept as an alternative. The automatic replacement was achieved by the implementation of customized programming sequences in LabVIEW resulting into another original application.

Firstly, each training dataset is generated and used to create a machine learning (ML) based model. This model is initialized, configured and trained so that it can associate the labels with a series of features. Then, it should be able to correctly classify a new input.

Secondly, each testing dataset is generated and used to validate the previously obtained model. Thus, this model is deployed on testing data, which include labels initially set. The evaluation of the model is related to the comparative analysis between the estimated labels (calculated by the model) and the initially set labels (included by the testing dataset). It will also result some evaluation metrics, for example: accuracy, precision, recall and F1 score.

Thirdly, each model is uploaded in the LabVIEW application aimed to run the classification process in real-time. Accordingly, the trained and validated model is tested on the processed EEG data acquired in real-time. The EEG data has been undergone the same processing methods related to the signal acquisition described in the section of automatic mode. Moreover, the EEG data should be organized as in the previously generated dataset, taking into account the mixture between selected EEG signals and extracted features. The model will automatically generate the labels.

In the following paragraph, there are clearly described all the steps necessary to get the results corresponding to the above-mentioned achievements.

1. Select Button that enables the EEG Data Acquisition in Automatic Mode.
2. Make the initial settings: Duration of Acquisition = 1 minute and 20 seconds; Time Interval = 2 seconds; Samples to Read = 512.

Note: It will result a .csv file representing the training dataset comprised of 40 temporal sequences, each of them containing 1024 samples. The feature extraction will be applied on the 12 x 2D arrays (12 x 40 rows x 1024 columns).

Note. The output is consisting of 12 x 2D arrays → 6 x 2D arrays are related to EEG Signals (raw, delta, theta, alpha, beta, gamma) in Time Domain and 6 x 2D arrays are related to EEG Signals in Frequency Domain - FFT Peak.

3. Start the EEG Data Acquisition in Automatic Mode.

4. According to both visual and auditory indicator, from 2 to 2 seconds, the user should execute one eye-blink.

Note: Thus, at the end of the acquisition, 40 temporal sequences will be returned, each of them including the EEG signal pattern of an eye-blink.

5. Wait until Duration of Acquisition = 1 minute and 20 seconds and the EEG Data Acquisition in Automatic Mode is over.
6. Select Config Button to return to the main window of the LabVIEW application, then select the Button that enables the extraction of features and the generation of the EEG dataset.
7. Select the Tab corresponding to the graphical displaying of each temporal sequence of the EEG Data acquired in the Automatic Mode. Look or visually analyze every one of the 40 EEG patterns and associate to it the appropriate Label - 1 for Eye-Blink Detected.
8. Select the Tab corresponding to the configuration of multiple mixtures between selected signals and extracted features to generate the EEG Training Dataset and save it to a .csv file.
9. There will be set 50 multiple mixtures and for every one of them, the EEG signals (Table 2) and the corresponding statistical features can be both manually or automatically selected.
10. Set 'First Index = 0' so that in the resulted .csv file, the rows can be counted starting from 0 (zero).
11. Deselect Label Button so that the first row from the resulted .csv file should contain the corresponding names or description of columns.
12. Set a correct path for saving the .csv file representing the Training Dataset.
13. Note. The name of the path is automatically incremented as it follows: training_dataset_1; training_dataset_2; …...; training_dataset_50.
14. Set a correct path for saving the .csv file containing the configuration (For example: Samples to Read = 512; Selected Signals = Alpha, Beta, Gamma; Extracted Features: Median, RMS, Standard Deviation);
15. Note. The name of the path is automatically incremented as it follows: config_1; config_2; …… config_50.
16. Select 'Processing' Button to generate the Training Dataset and save it along with the configuration to .csv files. In the end, it should result 50 .csv files containing the Training Datasets and 50 .csv files including the corresponding configuration of the 50 multiple mixtures of selected EEG signals and extracted features (Table 2).
17. Note. Currently, these 50 .csv files contain only 40 temporal sequences corresponding to Label - 1 for Eye-Blink Detected. Further, it is still necessary to acquire EEG Data in Automatic Mode (a record for 1 minute and 20 seconds) and obtain another 40 temporal sequences corresponding to Label - 0 for No Eye-Blink Detected. Thus, it is possible to generate the Training Dataset containing 2 classes of signals that can be classified.

Table 2. Information regarding the selected EEG signals in the 50 training datasets

Training Dataset	Selected EEG Signals	Extracted Statistical Features
1	All 12 signals in time and frequency domain	All the 10 statistical features
2	Only the 6 signals in time domain	All the 10 statistical features
3	Only the 6 signals in frequency domain	All the 10 statistical features
4	Raw	All the 10 statistical features
5	Delta	All the 10 statistical features
6	Theta	All the 10 statistical features
7	Alpha	All the 10 statistical features
8	Beta	All the 10 statistical features

9	Gamma	All the 10 statistical features
10	FFT Peak Raw	All the 10 statistical features
11	FFT Peak Delta	All the 10 statistical features
12	FFT Peak Theta	All the 10 statistical features
13	FFT Peak Alpha	All the 10 statistical features
14	FFT Peak Beta	All the 10 statistical features
15	FFT Peak Gamma	All the 10 statistical features
16	Raw, delta, theta	All the 10 statistical features
17	Raw, alpha, beta, gamma	All the 10 statistical features
18	Raw, delta, theta, gamma	All the 10 statistical features
19	Raw, beta, gamma	All the 10 statistical features
20	Beta, gamma	All the 10 statistical features
21	Alpha, beta, gamma	All the 10 statistical features
22	Delta, theta, alpha	All the 10 statistical features
23	Delta, theta	All the 10 statistical features
24	Raw, delta, theta	Mean, median, RMS, Standard Dev, Kurtosis
25	Delta, theta	Mean, median, RMS, Standard Dev, Kurtosis
26	Alpha, beta	Mean, median, RMS, Standard Dev, Kurtosis
27	Alpha, beta, gamma	Mean, median, RMS, Standard Dev, Kurtosis
28	Raw, alpha, beta, gamma	Mean, median, RMS, Standard Dev, Kurtosis
29	Delta, theta, alpha, beta, gamma	Mean, median, RMS, Standard Dev, Kurtosis
30	Raw, delta, theta, alpha, beta, gamma	Median, RMS, Standard Dev, Kurtosis
31	Delta, theta, alpha, beta, gamma	Median, RMS, Standard Dev, Kurtosis
32	Delta, theta	Median, RMS, Standard Dev, Kurtosis
33	Alpha, beta, gamma	Median, RMS, Standard Dev, Kurtosis
34	Beta, gamma	Median, RMS, Standard Dev, Kurtosis
35	Raw, beta, gamma	Median, RMS, Standard Dev, Kurtosis
36	Raw, delta, theta, alpha	Median, RMS, Standard Dev, Kurtosis
37	Raw, delta, theta, alpha, beta, gamma	Median, RMS, Standard Dev, Kurtosis, Mode, Sum, Skewness
38	Delta, theta, alpha, beta, gamma	Median, RMS, Standard Dev, Kurtosis, Mode, Sum, Skewness
39	Delta, theta	Median, RMS, Standard Dev, Kurtosis, Mode, Sum, Skewness
40	Delta, theta, alpha	Median, RMS, Standard Dev, Kurtosis, Mode, Sum, Skewness
41	Alpha, beta, gamma	Median, RMS, Standard Dev, Kurtosis, Mode, Sum, Skewness
42	Beta, gamma	Median, RMS, Standard Dev, Kurtosis, Mode, Sum, Skewness
43	Raw, beta, gamma	Median, RMS, Standard Dev, Kurtosis, Mode, Sum, Skewness
44	Raw, alpha, beta, gamma	Median, RMS, Standard Dev, Kurtosis, Mode, Sum, Skewness
45	Raw, delta, theta	Median, RMS, Standard Dev, Kurtosis, Mode, Sum, Skewness
46	Raw, delta, theta, alpha	Median, RMS, Standard Dev, Kurtosis, Mode, Sum, Skewness
47	Only the 6 signals in frequency domain	Median, RMS, Standard Dev, Kurtosis, Mode, Sum, Skewness

48	FFT Peak Delta, FFT Peak Theta	Median, RMS, Standard Dev, Kurtosis, Mode, Sum, Skewness
49	FFT Peak Alpha, FFT Peak Beta, FFT Peak Gamma	Median, RMS, Standard Dev, Kurtosis, Mode, Sum, Skewness
50	FFT Peak Raw, FFT Peak Delta, FFT Peak Theta	Median, RMS, Standard Dev, Kurtosis, Mode, Sum, Skewness

18. Repeat the steps 1 - 9.
19. Set 'First Index = 40' so that in the resulted .csv file, the rows can be counted starting from 40.
20. Select Label Button so that the first row from the resulted .csv file should not contain the corresponding names or description of columns.
21. Repeat the steps 12 - 14.
22. Select 'Config' Button to return to the main window of the LabVIEW application, then select 'Classification - Search optimal parameters' Button that enables the initialization, configuration and training of a model based on neural networks.
23. Select 'Riffle Data' Button to disable the 'Random Number' function applied to the normalized EEG Data acquired in Automatic Mode.
24. Note. This setting is necessary to get fixed/consistent values for evaluation metrics (accuracy, precision, recall, F1 score), which are used to compare the efficiency of the multiple mixtures between selected signals and extracted features.
25. Insert the correct path to read every one of the 50 .csv files representing the 50 training datasets.
26. Insert a correct path to save each one of the resulted neural networks-based models. In the end, it will result 50 .json files.
27. Note. The name of the path is automatically incremented as it follows: model_1; model_2; …… model_50.
28. Select 'Classification' Button to initialize, configure and train every one of the 50 neural networks-based models, obtained based on every one of the 50 training datasets.

Note. As mentioned previously, it was selected the tab corresponding to run the LabVIEW based functions from 'Analytics and Machine Learning' toolkit, that allow the searching of the optimal hyper-parameters (number of hidden neurons, hidden layer type, output layer type, cost function type) for every one of the 50 multiple mixtures.

Table 6 is consisting of complete results regarding the generation of hyper-parameters and evaluation metrics for every one of the 50 models based on neural networks. Therefore, the significance of the abbreviations corresponding to column headings is the following: A – training dataset, B – number of hidden neurons, C – hidden layer type, D – output layer type, E – cost function, F – number of input neurons, G – average method, H – accuracy, I – precision, J – recall, K – F1 score, L – correctly detected sample and M – incorrectly detected samples.

The above-presented steps are necessary to generate 50 .csv files corresponding to 50 training sets or 50 multiple mixtures of selected EEG signals and extracted features to initialize, configure and train 50 models (saved as .json files) based on artificial neural networks techniques. There were also generated 50 .csv files describing the content of the 50 training sets. Each training set is consisting of 4000 temporal sequences, based on the following assignments: 1000 sequences corresponding to Label - 0 (No Eye-Blink Detected), 1000 sequences associated with Label - 1 (One Eye-Blink Detected), 1000 sequences related to Label – 2 (Two Eye-Blinks Detected) and 1000 sequences linking to Label 3 (Three Eye-Blinks Detected).

Further, it is necessary to deploy the previously obtained 50 neural networks models (.json files) on different 50 testing datasets saved as 50 .csv files. The structure of the 50 testing datasets is the same as the structure of the 50 training datasets.

doi:10.20944/preprints202106.0016.v1

In the end, as shown in Table 7, it results the evaluation metrics (accuracy, precision, recall, F1 score) corresponding to the deployment of every one of the 50 neural networks-based model. The most convenient models will be used in the real-time LabVIEW app.

Table 6. Information regarding the generation of hyper-parameters and evaluation metrics after initialization, configuration and training of the 50 neural networks-based models

A	B	C	D	E	F	G	H	I	J	K	L	M
1	20	Sigmoid	Sigmoid	Cross-entropy	120	Weighted	0.98	0.98	0.98	0.98	3916	84
2	5	Sigmoid	Sigmoid	Cross-entropy	60	Weighted	0.98	0.98	0.98	0.98	3924	76
3	50	ReLU	Sigmoid	Cross-entropy	60	Weighted	0.95	0.95	0.95	0.95	3803	197
4	500	Tanh	Softmax	Cross-entropy	10	Weighted	0.98	0.98	0.98	0.98	3937	63
5	1000	ReLU	Sigmoid	Cross-entropy	10	Weighted	0.91	0.91	0.91	0.91	3659	341
6	200	ReLU	Sigmoid	Cross-entropy	10	Weighted	0.77	0.77	0.77	0.77	3089	911
7	100	ReLU	Softmax	Cross-entropy	10	Weighted	0.83	0.82	0.83	0.82	3305	695
8	50	ReLU	Sigmoid	Cross-entropy	10	Weighted	0.81	0.81	0.81	0.81	3226	774
9	10	ReLU	Softmax	Cross-entropy	10	Weighted	0.62	0.61	0.62	0.61	2462	1538
10	300	ReLU	Softmax	Cross-entropy	10	Weighted	0.92	0.92	0.92	0.92	3665	335
11	500	ReLU	Sigmoid	Cross-entropy	10	Weighted	0.83	0.83	0.83	0.83	3329	671
12	500	ReLU	Sigmoid	Quadratic	10	Weighted	0.78	0.77	0.78	0.78	3116	884
13	500	ReLU	Softmax	Cross-entropy	10	Weighted	0.80	0.80	0.80	0.80	3206	794
14	200	ReLU	Sigmoid	Cross-entropy	10	Weighted	0.74	0.73	0.74	0.74	2953	1047
15	300	ReLU	Softmax	Cross-entropy	10	Weighted	0.57	0.56	0.57	0.56	2268	1732
16	20	Sigmoid	Sigmoid	Cross-entropy	30	Weighted	0.98	0.98	0.98	0.98	3925	75
17	50	Tanh	Sigmoid	Cross-entropy	40	Weighted	0.98	0.98	0.98	0.98	3933	67
18	10	Sigmoid	Sigmoid	Cross-entropy	40	Weighted	0.98	0.98	0.98	0.98	3925	75
19	20	Tanh	Softmax	Cross-entropy	30	Weighted	0.99	0.99	0.99	0.99	3942	58
20	20	Tanh	Sigmoid	Cross-entropy	20	Weighted	0.82	0.82	0.82	0.82	3275	725
21	100	Tanh	Sigmoid	Cross-entropy	30	Weighted	0.91	0.91	0.91	0.91	3644	356
22	50	Tanh	Sigmoid	Cross-entropy	30	Weighted	0.92	0.92	0.92	0.91	3661	339
23	400	ReLU	Sigmoid	Cross-entropy	20	Weighted	0.91	0.91	0.91	0.91	3651	349
24	200	Tanh	Softmax	Cross-entropy	15	Weighted	0.98	0.98	0.98	0.98	3930	70
25	1000	ReLU	Sigmoid	Cross-entropy	10	Weighted	0.90	0.90	0.90	0.90	3619	381

26	300	Tanh	Softmax	Cross-entropy	10	Weighted	0.92	0.92	0.92	0.92	3664	336
27	50	ReLU	Sigmoid	Cross-entropy	15	Weighted	0.92	0.92	0.92	0.92	3665	335
28	10	Sigmoid	Sigmoid	Cross-entropy	20	Weighted	0.98	0.98	0.98	0.98	3926	74
29	20	ReLU	Sigmoid	Cross-entropy	25	Weighted	0.94	0.94	0.94	0.94	3757	243
30	20	Tanh	Sigmoid	Cross-entropy	24	Weighted	0.98	0.98	0.98	0.98	3925	75
31	50	Tanh	Sigmoid	Cross-entropy	20	Weighted	0.93	0.93	0.93	0.93	3739	261
32	500	ReLU	Softmax	Cross-entropy	8	Weighted	0.88	0.88	0.88	0.88	3526	474
33	20	Tanh	Sigmoid	Cross-entropy	12	Weighted	0.91	0.91	0.91	0.91	3655	345
34	50	Tanh	Sigmoid	Cross-entropy	8	Weighted	0.82	0.82	0.82	0.82	3262	738
35	10	Tanh	Sigmoid	Cross-entropy	12	Weighted	0.98	0.98	0.98	0.98	3912	88
36	200	ReLU	Sigmoid	Cross-entropy	16	Weighted	0.98	0.98	0.98	0.98	3929	71
37	20	ReLU	Sigmoid	Cross-entropy	42	Weighted	0.98	0.98	0.98	0.98	3926	74
38	10	Tanh	Sigmoid	Cross-entropy	35	Weighted	0.94	0.94	0.94	0.94	3763	237
39	50	ReLU	Sigmoid	Cross-entropy	14	Weighted	0.91	0.91	0.91	0.91	3658	342
40	50	ReLU	Sigmoid	Cross-entropy	21	Weighted	0.92	0.92	0.92	0.92	3668	332
41	20	Tanh	Softmax	Cross-entropy	21	Weighted	0.91	0.91	0.91	0.91	3639	361
42	20	ReLU	Sigmoid	Cross-entropy	14	Weighted	0.81	0.81	0.81	0.81	3254	746
43	5	Tanh	Sigmoid	Cross-entropy	21	Weighted	0.98	0.98	0.98	0.98	3937	63
44	10	Sigmoid	Sigmoid	Cross-entropy	28	Weighted	0.98	0.98	0.98	0.98	3929	71
45	50	Tanh	Sigmoid	Cross-entropy	21	Weighted	0.98	0.98	0.98	0.98	3927	73
46	20	Tanh	Sigmoid	Cross-entropy	28	Weighted	0.98	0.98	0.98	0.98	3935	65
47	100	ReLU	Sigmoid	Cross-entropy	42	Weighted	0.95	0.95	0.95	0.95	3799	201
48	50	Tanh	Sigmoid	Cross-entropy	14	Weighted	0.87	0.86	0.87	0.86	3461	539
49	100	Tanh	Sigmoid	Cross-entropy	21	Weighted	0.87	0.87	0.87	0.87	3492	508
50	50	Tanh	Sigmoid	Cross-entropy	21	Weighted	0.95	0.95	0.95	0.95	3790	210

Table 7. Information regarding evaluation metrics after the deployment of the 50 neural networks-based models on the testing datasets

Dataset or Model Neural Netwarks	Accuracy	Precision	Recall	F1 Score	Correctly Detected Samples	Incorrectly Detected Samples
1	0.97	0.97	0.97	0.97	778	22
2	0.97	0.97	0.97	0.97	775	25
3	0.83	0.83	0.83	0.83	661	139
4	0.94	0.94	0.94	0.94	749	51
5	0.73	0.77	0.73	0.73	583	217
6	0.68	0.69	0.68	0.68	545	255
7	0.68	0.68	0.68	0.68	546	254
8	0.75	0.75	0.75	0.75	603	197
9	0.58	0.59	0.58	0.58	463	337
10	0.84	0.84	0.84	0.83	668	132
11	0.75	0.75	0.75	0.75	597	203
12	0.65	0.66	0.65	0.65	520	280
13	0.69	0.69	0.69	0.69	552	248
14	0.68	0.67	0.68	0.67	540	260
15	0.52	0.52	0.52	0.51	416	384
16	0.95	0.95	0.95	0.95	763	37
17	0.96	0.96	0.96	0.96	770	30
18	0.95	0.95	0.95	0.95	759	41
19	0.96	0.96	0.96	0.96	768	32
20	0.77	0.77	0.77	0.77	618	182
21	0.84	0.83	0.84	0.83	668	132
22	0.83	0.86	0.83	0.83	661	139
23	0.85	0.87	0.85	0.85	683	117
24	0.95	0.95	0.95	0.95	761	39
25	0.85	0.87	0.85	0.85	680	120
26	0.83	0.84	0.83	0.83	663	137
27	0.84	0.85	0.84	0.84	669	131
28	0.95	0.95	0.95	0.95	763	37
29	0.90	0.91	0.90	0.90	717	83
30	0.97	0.97	0.97	0.97	776	24
31	0.88	0.90	0.88	0.88	703	97
32	0.80	0.82	0.80	0.80	636	164
33	0.84	0.85	0.84	0.84	668	132
34	0.76	0.77	0.76	0.76	610	190
35	0.96	0.96	0.96	0.96	768	32
36	0.97	0.97	0.97	0.97	773	27
37	0.96	0.96	0.96	0.96	766	34
38	0.88	0.90	0.88	0.88	706	94
39	0.85	0.86	0.85	0.85	683	117
40	0.87	0.88	0.87	0.87	694	106
41	0.83	0.84	0.83	0.83	664	136
42	0.77	0.76	0.77	0.76	612	188
43	0.94	0.94	0.94	0.94	754	46
44	0.95	0.95	0.95	0.95	761	39
45	0.96	0.96	0.96	0.96	765	35
46	0.95	0.95	0.95	0.95	762	38

47	0.83	0.83	0.83	0.83	664	136
48	0.79	0.80	0.79	0.79	633	167
49	0.77	0.78	0.77	0.77	619	181
50	0.81	0.82	0.81	0.81	647	153

4. Discussion

The development of the LabVIEW application presented in this paper is based on a novel approach regarding the processing of the acquired raw EEG signal, which is analyzed by applying a features extraction algorithm on the selected EEG rhythms. Thus, it is obtained the training/testing dataset which is classified by Artificial Neural Networks. Accordingly, four EEG signal patterns can be recognized, resulting the assignment of the following labels: 0 (No Eye-Blink Detected), 1 (One Eye-Blink Detected), 2 (Two Eye-Blinks Detected) and 3 (Three Eye-Blinks Detected). The following paragraphs describe the proposed approach related to the preparation of the training/testing dataset.

The initial settings are: duration of the acquisition (target time) = 80 (seconds); time interval = 2 (seconds) and (number of) samples to read = 512. It should be emphasized that the sampling rate is equal to 512 Hz, meaning that 512 samples are acquired during one second. It will result: 40 temporal sequences, each of them containing 2 arrays x 512 samples → 40 temporal sequences, each of them containing 1024 samples. Overall, it will result 40960 samples for both raw EEG signal (0.1 – 100 Hz) and each type of the EEG rhythms in time domain: delta (0.5 – 3.5 Hz); theta (4 – 7 Hz); alpha (8 – 12 Hz); beta (13 – 30 Hz) and gamma (>30 Hz). It will also result 40960 samples for each type of the EEG rhythms in the frequency domain (FFT): raw; delta; theta; alpha; beta and gamma.

Every one of the 40 temporal sequences, each of them containing 1024 numerical values, corresponding to a particular signal (mentioned above) has been undergone an analysis related to the measurement or the calculation of certain statistical meters, defining the following features: mean, median, standard deviation, RMS (route mean square) and others. Further, every one of the 40 temporal sequences is assigned to an appropriate label: ‘0’ for ‘No Eye-Blink Detected’ or ‘1’ for ‘One Eye-Blink Detected’ or ‘2’ for ‘Two Eye-Blinks Detected’ or ‘3’ for ‘Three Eye-Blink Detected’. Finally, taking into account that 25 sessions of EEG data acquisition were taken for each label, it will result: 25 x 40 temporal sub-sequences corresponding to each of the four previously mentioned class.

The advantages of the proposed approach are described in this section. It is simple to analyze and classify a temporal sequence of the raw EEG signal, represented by 2 arrays, each of them containing 512 samples (2 x 512 = 1024 samples) so that it can be distributed in one of the four categories: ‘No Eye-Blink Detected’, ‘One Eye-Blink Detected’, ‘Two Eye-Blinks Detected’ and ‘Three Eye-Blinks Detected’. The simplicity is related to the precise visual recognition of the waveform corresponding to the multiple voluntary eye-blinks pattern. This pattern is an artefact across raw EEG signal so that it is characterized by an increase followed by a decrease of the bio-potential. The labels (‘0’, ‘1’, ‘2’, ‘3’) are manually assigned, avoiding any mistake by the possibility of removing a wrong value. It facilitates the assignment of the labels, which is an important phase in the generation of the training/testing dataset.

The next phase is related to the generation of training or testing dataset. This is represented by a .csv file containing multiple mixtures between the selected EEG rhythms (for instance: alpha, beta and gamma) and the extracted features (for example: mean, median and standard deviation). The resulted .csv file, containing the training dataset, is necessary for the classification process based on the Artificial Neural Networks algorithm offered by the AML LabVIEW toolkit. It should be mentioned that a different .csv file, containing the testing dataset, is generated for the validation of the Artificial Neural Networks model. The final stage is consisting of the deployment the Artificial Neural Networks model by applying it to the raw EEG signal acquired in real-time. This task is achieved by running a different and novel LabVIEW application aimed for the real-time detection of the signals corresponding to the labels/categories: 'No Eye-Blink Detected', 'One Eye-Blink Detected', 'Two Eye-Blinks Detected' and 'Three Eye-Blink Detected'.

5. Conclusion

This paper proposed several LabVIEW based applications aimed for the acquisition, processing and classification of the EEG signal detected by the embedded sensor of NeuroSky Mindwave Mobile headset, second edition. The classification process is optimized by artificial intelligence techniques provided by versatile functions included by the 'Analytics and Machine Learning' toolkit. Its functionality was customized so that the randomization of EEG data was removed. The application developed in the current research is consisting of different states: manual and automatic acquisition mode, processing EEG raw signal and preparation of the EEG training dataset based on the generation of multiple mixtures between selected EEG rhythms (time-domain: delta, theta, alpha, beta, gamma and frequency domain – Fast Fourier Transform with Peak parameter applied on the same signals) and extracted statistical features (mean, median, route mean square, standard deviation, Kurtosis Coefficient, mode, summation, skewness, maximum, range = maximum – minimum). The most relevant multiple mixtures can be automatically identified by using the LabVIEW application developed in the presented work. Moreover, this instrument facilitates the quick and simple generation of some graphical representations related to those multiple mixtures. Further, an artificial neural network (ANN) based model of classification is initialized, configured and trained with previously mentioned EEG dataset. After that, the trained ANN model is deployed on a different EEG dataset so that it will result evaluation metrics, such as accuracy and precision. Finally, the trained and validated ANN model is used to classify the EEG signals acquired in real-time, by using another LabVIEW application, developed in the current research. The new approach described in this paper is based on original programming sequences implemented in LabVIEW. The main LabVIEW application aims to provide an efficient solution of recognition EEG signals patterns corresponding to different cognitive tasks. In this paper, it was presented the classification of multiple voluntary eye-blinks.

Future research directions are related to further improvements regarding the assessment of the LabVIEW based system by enabling the classification of different EEG signals patterns. Furthermore, the intention is to add more types of signals and more significant features. Likewise, other versatile applications of Brain-Computer Interface will surely need more flexibility that can be achieved by the execution of training processes based on Supported Vector Machines or Logistic Regression models. Therefore, these two methods that are also included by the 'Analytics and Machine Learning' LabVIEW toolkit will be taken into consideration for future research projects.

Supplementary Materials: A quick video demonstration of the working principle of the proposed LabVIEW applications is uploaded to this YouTube unlisted link: https://youtu.be/bmr04-QKJOg

Author Contributions: Conceptualization, O.A.R.; methodology, O.A.R.; software, O.A.R.; validation, O.A.R.; formal analysis, O.A.R.; investigation, O.A.R.; resources, O.A.R.; data curation, O.A.R.; writing—original draft preparation, O.A.R.; writing—review and editing, O.A.R.; visualization, O.A.R.; supervision, O.A.R.; project administration, O.A.R. The author has read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Conflicts of Interest: The author declares no conflict of interest.

References

1. Nguyen, T.-H.; Chung, W.-Y. Detection of Driver Braking Intention Using EEG Signals During Simulated Driving. *Sensors* **2019,** *19*, 2863. DOI:10.3390/s19132863.
2. Al-Hudhud, G.; Alqahtani, L.; Albaity, H.; Alsaeed, D.; Al-Turaiki, I. Analyzing Passive BCI Signals to Control Adaptive Automation Devices. *Sensors* **2019**, *19*, 3042. DOI:10.3390/s19143042.
3. Mamani, M.A.; Yanyachi, P.R. Design of computer-brain interface for flight control of unmanned air vehicle using cerebral signals through headset electroencephalograph. Proceedings of the IEEE International Conference on Aerospace and Signals (INCAS), Peru, 8 - 10 Nov. 2017. DOI: 10.1109/INCAS.2017.8123499.
4. López-Hernández, J.L.; González-Carrasco, I.; López-Cuadrado, J.L.; Ruiz-Mezcua, B. Towards the Recognition of the Emotions of People with Visual Disabilities through Brain–Computer Interfaces. *Sensors* **2019,** 19, 2620. DOI: 10.3390/s19112620.
5. Choudhari, A.M.; Porwal, P.; Jonnalagedda, V.; Mériaudeau, F. An Electrooculography based Human Machine Interface for wheelchair control. *Biocybernetics and Biomedical Engineering* **2019,** 39, 673-685. DOI: https://doi.org/10.1016/j.bbe.2019.04.002.
6. Choudhari, A.M.; Jonnalagedda, V. Bio-potentials for smart control applications. *Health and Technology* **2019**, 9, 1-25, DOI: 10.1007/s12553-019-00314-7.
7. Cheng, C.; Li, S.; Kadry, S. Mind-Wave Controlled Robot: An Arduino Robot Simulating the Wheelchair for Paralyzed Patients. *International Journal of Robotics and Control* **2018**, 1, 6-14, DOI:10.5430/ijrc.v1n1p6.
8. Dev, A.; Rahman, M.; Mamun, N. Design of an EEG-Based Brain Controlled Wheelchair for Quadriplegic Patients. Proceedings of the IEEE 3rd International Conference for Convergence in Technology (I2CT), India, 6 - 8 April 2018, DOI: 10.1109/I2CT.2018.8529751.
9. Đumić D.; Đug M.; Kevrić J. Brainiac's Arm—Robotic Arm Controlled by Human Brain. In: Hadžikadić M., Avdaković S. (eds) Advanced Technologies, Systems, and Applications II. IAT 2017. Proceedings of the International Symposium on Innovative and Interdisciplinary Applications of Advanced Technologies (IAT), Bosnia and Herzegovina, 25 – 28 May 2017, DOI: https://doi.org/10.1007/978-3-319-71321-2_73.
10. Bright, D.; Nair, A.; Salvekar, D.; Bhisikar, S. EEG-based brain controlled prosthetic arm. Proceedings of the Conference on Advances in Signal Processing (CASP), India, 9 – 11 June 2016. DOI:10.1109/CASP.2016.7746219.
11. Reust, A.; Desai, J.; Gomez, L. Extracting Motor Imagery Features to Control Two Robotic Hands. Proceedings of the 2018 IEEE International Symposium on Signal Processing and Information Technology (ISSPIT), Louisville, KY, USA, 6 – 8 Dec. 2018, DOI: 10.1109/ISSPIT.2018.8642627.
12. Katona, J.; Ujbanyi, T.; Sziladi, G.; Kovari, A. Speed control of Festo Robotino mobile robot using NeuroSky MindWave EEG headset based brain-computer interface. Proceedings of the 2016 7th IEEE International Conference on Cognitive Infocommunications (CogInfoCom), Poland, 16 – 18 Oct. 2016, DOI: 10.1109/CogInfoCom.2016.7804557.
13. Xiao, Y.; Jia, Y.; Cheng, X.; Yu, J.; Liang, Z.; Tian, Z. I Can See Your Brain: Investigating Home-Use Electroencephalography System Security. *IEEE Internet of Things Journal* **2019**, 6, 6681-6691. DOI:10.1109/JIOT.2019.2910115.
14. Jafri, S.; Hamid, T.; Mahmood, R.; Alam, M.; Rafi, T.; Ul Haque, M.; Munir, M. Wireless Brain Computer Interface for Smart Home and Medical System. *Wireless Personal Communications* **2019**, *106, 2163 – 2177.* DOI: https://doi.org/10.1007/s11277-018-5932-x.
15. López, A.; Ferrero, F.; Yangüela, D.; Álvarez, C.; Postolache, O. Development of a Computer Writing System Based on EOG. *Sensors* **2017**, *17*, 1505. DOI: 10.3390/s17071505.
16. Ko, L.-W.; Chang, Y.; Wu, P.-L.; Tzou, H.-A.; Chen, S.-F.; Tang, S.-C.; Yeh, C.-L.; Chen, Y.-J. Development of a Smart Helmet for Strategical BCI Applications. *Sensors* **2019**, *19*, 1867. DOI:10.3390/s19081867.
17. Padfield, N.; Zabalza, J.; Zhao, H.; Masero, V.; Ren, J. EEG-Based Brain-Computer Interfaces Using Motor-Imagery: Techniques and Challenges. *Sensors* **2019**, *19*, 1423. DOI:10.3390/s19061423
18. Majidov, I.; Whangbo, T. Efficient Classification of Motor Imagery Electroencephalography Signals Using Deep Learning Methods. *Sensors* **2019**, *19*, 1736. DOI:10.3390/s19071736.
19. Garcia A.; Gonzalez, J.M.; Palomino, A. Data Acquisition System for the Monitoring of Attention in People and Development of Interfaces for Commercial Devices. In: Agredo-Delgado V., Ruiz P. (eds) Human-Computer Interaction, Proceedings of the Iberoamerican Workshop on Human-Computer Interaction (HCI-COLLAB), Colombia, 23 – 27 April 2018, DOI: https://doi.org/10.1007/978-3-030-05270-6_7.
20. Li, M.; Liang, Z.; He, B.; Zhao, C.-G.; Yao, W.; Xu, G.; Xie, J.; Cui, L. Attention-Controlled Assistive Wrist Rehabilitation Using a Low-Cost EEG Sensor. *IEEE Sensors Journal* **2019**, *19*, 6497-6507. DOI: 10.1109/JSEN.2019.2910318.
21. Rashid, M.; Sulaiman, N.; Mustafa, M.; Khatun, S.; Bari, B.S. The Classification of EEG Signal Using Different Machine Learning Techniques for BCI Application. In: Kim JH., Myung H., Lee SM. (eds) Robot Intelligence Technology and Applications. RiTA

2018, Proceedings of the International Conference on Robot Intelligence Technology and Applications, Malaysia, 16 – 18 December 2018, DOI: https://doi.org/10.1007/978-981-13-7780-8_17.

22. Zavala, S.; Bayas, L.J.L.; Ulloa A.; Sulca, J.; López, M.J.L.; Yoo, S.G. Brain Computer Interface Application for People with Movement Disabilities, Proceedings of the HCC: International Conference on Human Centered Computing, Mexico, 5 – 7 December 2018, DOI: 10.1007/978-3-030-15127-0_4.
23. Wannajam S.; Thamviset, W. Brain Wave Pattern Recognition of Two-Task Imagination by Using Single-Electrode EEG. In: Unger H., Sodsee S., Meesad P. (eds) Recent Advances in Information and Communication Technology 2018, Proceedings of the International Conference on Computing and Information Technology, Thailand, 5 – 6 July 2018, DOI: 10.1007/978-3-319-93692-5_19.
24. Wadekar, R.S.; Kasambe, P.V.; Rathod, S.S. Development of LabVIEW platform for EEG signal analysis, Proceedings of the 2017 International Conference on Intelligent Computing and Control (I2C2), India, 23 – 24 June 2017, DOI: 10.1109/I2C2.2017.8321942.
25. Kumar, S.; Kumar, V.; Gupta, B. Feature extraction from EEG signal through one electrode device for medical application, Proceedings of the 2015 1st International Conference on Next Generation Computing Technologies (NGCT), India, 4 – 5 September 2015, DOI: 10.1109/NGCT.2015.7375181.
26. Mutasim, A.K.; Tipu, R.S.; Bashar, M.R.; Islam, M.K.; Amin, M.A. Computational Intelligence for Pattern Recognition in EEG Signals. In *Computational Intelligence for Pattern Recognition*; Pedrycz W., Chen S.M., Eds.; Studies in Computational Intelligence, vol. 777, 291 – 320, Springer, Cham.
27. Zhang J.; Huang W.; Zhao S.; Li Y.; Hu S. Recognition of Voluntary Blink and Bite Base on Single Forehead EMG. In: Liu D.; Xie S.; Li Y.; Zhao D.; El-Alfy, E.S (eds) Neural Information Processing. ICONIP 2017. Proceedings of the International Conference on Neural Information Processing, China, 14 – 18 November 2017, DOI: 10.1007/978-3-319-70096-0_77.
28. Harsono, M.; Liang, L.; Zheng, X.; Jesse, F.F., Cen, Y.; Jin, W. Classification of Imagined Digits via Brain-Computer Interface Based on Electroencephalogram. In: Wang Y., Huang Q., Peng Y. (eds) Image and Graphics Technologies and Applications. IGTA 2019. Proceedings of the Chinese Conference on Image and Graphics Technologies, China, 19 – 20 April 2019, DOI: 10.1007/978-981-13-9917-6_44.
29. Venuto, D.; Annese, V.; Mezzina, G. Towards P300-Based Mind-Control: A Non-invasive Quickly Trained BCI for Remote Car Driving, Proceedings of the International Conference on Sensor Systems and Software, France, 1 -2 December 2016, DOI: 10.1007/978-3-319-61563-9_2.
30. Kim, K.; Suk, H.; Lee, S. Commanding a Brain-Controlled Wheelchair Using Steady-State Somatosensory Evoked Potentials. *IEEE Transactions on Neural Systems and Rehabilitation Engineering* **2018**, *26*, 654-665, DOI: 10.1109/TNSRE.2016.2597854.
31. Varela, M. Raw EEG signal processing for BCI control based on voluntary eye blinks. Proceedings of the 2015 IEEE Thirty Fifth Central American and Panama Convention (CONCAPAN XXXV), Honduras, 11 – 13 Nov. 2015, DOI: 10.1109/CONCAPAN.2015.7428477.
32. National Instruments. What is LabVIEW? Available Online: https://www.ni.com/ro-ro/shop/labview.html (accessed on 15 September 2019)
33. NeuroSky Mindwave Mobile 2. Available Onlie: https://store.neurosky.com/pages/mindwave (accessed on 15 September 2019)
34. Rieiro, H.; Diaz-Piedra, C.; Morales, J.M.; Catena, A.; Romero, S.; Roca-Gonzalez, J.; Fuentes, L.J.; Di Stasi, L.L. Validation of Electroencephalographic Recordings Obtained with a Consumer-Grade, Single Dry Electrode, Low-Cost Device: A Comparative Study. *Sensors* **2019**, 19, 2808. DOI:10.3390/s19122808.
35. Bednář R.; Brozek J. Neural Interface: The Potential of Using Cheap EEG Devices for Scientific Purposes. In: Ntalianis K., Croitoru A. (eds) Applied Physics, System Science and Computers II. APSAC 2017. Proceedings of the International Conference on Applied Physics, System Science and Computers, Croatia, 27 – 29 September 2017, DOI: https://doi.org/10.1007/978-3-319-75605-9_18
36. National Instruments. Analytics and Machine Learning Toolkit. Available Online: http://zone.ni.com/reference/en-XX/help/377059B-01/lvamlconcepts/aml_overview/ (accessed on 15 September 2019)