A LabVIEW Instrument Aimed for the Research on Brain-Computer Interface by Enabling the Acquisition, Processing, and the Neural Networks based Classification of the Raw EEG Signal Detected by the Embedded NeuroSky Biosensor

Oana Andreea RUȘANU [1]

[1] Product Design, Mechatronics and Environment Department, *Transilvania* Univesity of Brasov, Brasov, Romania; oana.rusanu@unitbv.ro

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

Abstract: The Brain-Computer Interface (BCI) is a scientific field aimed at helping people with neuromotor disabilities. Among the current drawbacks of BCI research is the need for a cost-effective software instrument for simple integration with portable EEG headsets, the lack of a comparative assessment approach of various techniques underlying recognizing the most precise BCI control signal –voluntary eye-blinking, and the need for EEG datasets allowing the classification of multiple voluntary eye-blinks. The proposed BCI research-related virtual instrument accomplishes the data acquisition, processing, features extraction, and the ANN-based classification of the EEG signal detected by the NeuroSky embedded biosensor. The developed software application automatically generated fifty mixtures between selected EEG rhythms and statistical features. The EEG rhythms are related to the time and frequency domains of the raw, delta, theta, alpha, beta, and gamma. The extracted statistical features contain the mean, median, standard deviation, route mean square, Kurtosis coefficient, mode, sum, skewness, maximum, and range = maximum-minimum. The results include 100 EEG datasets to classify multiple voluntary eye-blinks: 50 datasets with 4000 recordings and 50 with 800 recordings. The LabVIEW application determined the optimal ANN models for classifying the EEG temporal sequences corresponding to detecting zero, one, two, or three voluntary eye-blinks.

Keywords: brain-computer interface, EEG signal, artificial neural networks, LabVIEW application, features extraction, eye-blinks detection, EEG portable 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]. Beyond its various applications in non-clinical fields (digital games, sleep avoidance, mental states monitoring, advertisement, and business), the fundamental 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 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 (BCI). BCI 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 intelligent 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 [16], processing, features extraction [17], and classification of signals related to brain patterns [18] triggered by the execution of specific cognitive tasks [19], [20], followed by the translation of the detected biopotentials and transmission of commands to the controlled applications.

Electroencephalography (EEG) is the most convenient method for acquiring signals across the scalp by using dry or wet electrodes that detect neuronal bio-potentials. It is non-invasive, portable, inexpensive, and advantageous, thanks to its high temporal resolution. Electronic producers released mobile EEG headsets to enable both research [21] or entertainment. Further, advanced methods process the EEG signal for noise reduction and filtering of the most significant ranges or frequencies. Then, the artificial neural networks (ANN) based techniques classify the EEG bio-potentials to result in 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 [22-23], mentally execute some arithmetic operations [24], focus his/her attention [25] on a single idea, relax or meditate, mentally counting and visualize the resizing of 3D figures. A specific EEG signals pattern characterizes each cognitive task, which is assigned the corresponding class. Machine learning techniques facilitate the classification process by employing neural networks, supported vector machines [26], or logistic regression. Every class is associated with a particular command transmitted to the target application, for example, a brain-controlled motorized wheelchair, a neuroprosthesis, a medical system aimed for rehabilitation [27], or the mind-controlled movement of the cursor on the computer screen aimed for people with neuromotor disabilities [28].

This paper proposes a novel BCI research-related virtual instrument to acquire, process, and classify the electroencephalographic (EEG) signals necessary to implement a Brain-Computer Interface system (BCI). Although the EEG signals classification process mainly focuses on detecting multiple voluntary eye-blinks, the implemented algorithm provides flexibility and robustness to recognize different signal patterns corresponding to other mental tasks related to the Brain-Computer Interface research field.

Regarding the main contribution brought by the BCI research related virtual instrument presented in this paper, a novel approach by leveraging the LabVIEW graphical programming environment of developing several original code sequences allows the users to acquire, analyze and classify the raw EEG signal acquired from the embedded bio-sensor of the portable EEG NeuroSky headset. The proposed unique and robust experimental paradigm involves various stages accomplished by the software system based on LabVIEW. Moreover, the current paper describes training or testing dataset preparation, an essential phase in the machine learning-based application.

The current research aims to provide an efficient, automated, unique solution that integrates all the following sections: data acquisition, processing, and classification of the EEG signals. The working principle underlying the LabVIEW application could also detect alternative brain patterns, for example, P300 and SSVEP EEG potentials [29-30] related to various cognitive tasks. The recognition of EEG patterns related to the classification of multiple voluntary eye-blinks constitutes the testing method of the developed system. The eye-blinks are artifacts across the raw EEG signal. According to scientific literature [31], eye-blinks were considered precise control signals in a brain-computer interface application.

According to BNCI HORIZON 2020 [32], across the World, there were established around 120 official Brain-Computer Interface Research Groups spreading in the following 28 countries: Australia, Austria, Bangladesh, Belgium, Brazil, Canada, China, Denmark, Finland, France, Germany, Ireland, India, Italia, Japan, Malaysia, Netherlands, New Zeeland, Pakistan, Portugal, Russia, Singapore, South Korea, Spain, Switzerland, Turkey, UK, and the USA.

Considering that currently there are recognized 195 countries in the World, it results in a highly reduced percentage of only 14.35% of all the worldwide states benefits from

optimal intellectual, financial, human, and timing resources to conduct high-quality research in the brain-computer interface scientific field. Even the professors from the University Centre, which are not part of those 14.35% privileged states, encounter several difficulties that negatively impact the design, implementation, and experimentation of a novel brain-computer interface system providing a real-life application for people with neuromotor disabilities. These external difficulties include too expensive EEG equipment, inexistent laboratory conditions for conducting invasive and non-invasive experiments, the unavailable partnership between multidisciplinary groups to enable the BCI research, and the absence of knowledge-sharing sessions with BCI experts. Unfortunately, all these issues determine the professors to manifest a lack of motivation to overcome the significant technical challenges addressed by the BCI research field.

Probably an awful consequence refers to the discouragement, uncertainty, frustration, and intrigue that young researchers may feel, such as passionate, enthusiastic, eager, creative, and ambitious undergraduates or doctoral students. Those professors from the countries, which do not benefit from financial support for BCI research, do not appreciate their student's endeavors and consider them joyful experiments. That is because they realize that their dream scientific project seems not to be achievable.

Therefore, it results in the utmost importance of the primary objective accomplished by this paper: the design, the development, and the implementation of a flexible, robust, simple to use, user-friendly and cost-effective virtual instrument aimed for BCI research. Thus, the young scientists benefit from a quick experimental platform enabling the fundamental processes: the acquisition, monitoring, processing, features extraction, and classification based on artificial neural networks of the biosignal detected with the help of an affordable portable EEG headset, such as NeuroSky Mindwave Mobile. Intelligent processing consists of automated solutions that the young researchers could customize according to the particular scientific purpose that they are pursuing.

The proposed research virtual instrument reveals a novel approach to designing a BCI versatile framework by allowing multiple selections between the EEG rhythms (both in time and frequency domains) and statistical features by delivering complete training and testing datasets and getting neural networks-based models aimed for the EEG data classification. The use of the BCI research-related virtual instrument revealed in this paper does not involve programming skills, signal processing abilities, or neuroscience knowledge. This advantage results in its universal usefulness and general purpose by attracting researchers from non-technical fields, such as psychology, social sciences, music, arts, business, and advertisement media.

Regarding the secondary objective of the current research, the most straightforward application tested with the proposed virtual instrument is the neural networks-based classification of the multiple voluntary eye-blinks used as precise control signals in a brain-computer interface application. The voluntary eye-blinking is an artifact across the raw EEG signal. It is easy to detect by its specific pattern showing an increase and a decrease of the EEG signal amplitude following the two states of closing and opening the eye. By capturing the EEG based eye-blinking pattern and counting its occurrences, there resulted in different commands that are easy to execute by people with neuromotor disabilities to control specific assistive devices: an electrical wheelchair, a mobile robot [33 - 34], a robot hand [35 - 36], a robot arm, home appliances, experimental prototypes [37] and communication systems [38 - 39].

The BCI scientific literature reports numerous papers focused on employing voluntary eye-blinking in developing a brain-computer interface system. The novice researchers preferred to call off-the-shelf functions to measure the strength of the eye-blinking necessary to set a threshold value [40 - 41], to apply statistical calculus to implement an algorithm for counting the voluntary eye-blinks [42 - 44] for the development of experimental BCI prototypes [45 – 46]. As for drawbacks, the thresholding-based method for voluntary eye-blink detection involves calibration sessions and user-customized amplitude threshold that could determine variable accuracy. Otherwise, the experienced researchers explored the advanced classification techniques based on neural networks [47 – 50], wavelet

design [51], and support vector machines [52] for getting high accuracy of discriminating between various types of eye-blinks (voluntary, involuntary, short, long, simple, and double) aimed for the achievement of highly performant brain-computer interfaces. Table 1 shows a summary including the acquisition, processing, extracted features, classifier of some of the papers aimed for voluntary eye-blinking that did not involve amplitude thresholding.

Table 1. A summary (acquisition, processing, extracted features, classifier) of the papers aimed for voluntary eye-blinking classification that did not involve amplitude thresholding

Reference	Acquisition	Processing	Features	Classifier
[47]	Raw EEG Signal	Butterworth Band Pass EEG signal was detrended, normalized, and segmented into windows with 480 samples each	Maximum amplitude Minimum amplitude Kurtosis Coefficient	Binary Classifier based on the Probabilistic Neural Network (RBF = Radial Basis Function)
[48]	Raw EEG Signal	Analog and digital processing, as well as an excellent signal-to-noise ratio provided by the commercial software	Kurtosis coefficient Maximum amplitude Minimum amplitude	FFBP (Feed-forward back-propagation CFBF (Cascade-forward Backpropagation)
[49]	Raw EEG Signal	A 2nd order Butterworth filter between 1 and 49 Hz and a DC correction	Eye-Blink Peak detection by using PeakUtils Matlab package	1D CNN (Convolutional Neural Network) with a shallow architecture (one convolutional layer, one dropout layer, and one pooling layer)
[50]	Raw EEG Signal	Not specified any information	Minimum value Maximum value Mean Median Mode Standard deviation	An artificial multi-layer neural network with backpropagation → input layer (48 neurons), three hidden layers, one output layer (1 neuron); Activation function – binary sigmoid; traincgb, traincgf, traincgp, trainrp, traingda reported low performance traingdx and trainlm reported high performance
[51]	Raw EEG Signal	EEG data analysis in the time domain A coefficients matrix based on the following variable parameters: the window size (number of samples) of the analyzed signal, thresholding the coefficients, and decomposition level A multi-resolution analysis Certain filters orders	The temporal location and the duration measurement of each eye blinking occurrence Certain thresholds and the decomposition level	Continuous wavelet transform (CWT) – designing an ad-hoc mother wavelet
[52]	Raw EEG Signal	A moving window in the time domain Correlation thresholds	Not applicable	Unsupervised novel approach

Eye-blink fingerprint

Both the novice and experienced researchers focused on a particular BCI application so that they established fixed, rigid, and restrictive thresholding, statistical and artificial intelligence-based research methods for the detection and counting of multiple voluntary eye-blinks. Currently, the BCI-related scientific literature does not prove any recent evidence or proposal of a flexible, versatile, customizable EEG research software solution aimed for further investigation, comparative analysis, and performance evaluation of the results obtained to classify the multiple voluntary eye-blinks.

Table 2 shows a summary, including the software, hardware, task, application, and the availability of a dataset related to the papers aimed for voluntary eye-blinking classification that did not involve amplitude thresholding.

Table 2. A summary (software, hardware, task, application, dataset) of the papers aimed for voluntary eye-blinking classification that did not involve amplitude thresholding

Reference	Software	Hardware	Task	Application	Dataset
[47]	Biomedical Monitor of BioRadio Pac Data Acquisition software of RMS EEG 32 Super System	BioRadio Pac	Execute intentional eye-blink	Eye-blinking detection	Not available
[48]		RMS EEG 32 Super System – 2 channels (FP1 and F3)	Execute voluntary eye-blink	Eye-blinking detection	Not available
[49]	Matlab Matlab R2018b PeakUtils package Keras with Tensorflow backend	OpenBCI Ultracortex Mark IV helmet - 2 channels (FP1 and FP2)	Execute voluntary eye-blinking	Discrimination between voluntary and involuntary eye-blinking	A dataset containing 1080 EEG epochs: 540 for voluntary eye-blinks and 540 for involuntary eye-blinks
[50]	Matlab	OpenBCI Cython	Performing five hand movements called: "cylindrical grip", "power hand-grip", "tripod pinch", "index point" and resting ("open grip")	Control five types of movements performed by a 3D printed robotic hand	Not available
[51]	Matlab Java	NeuroSky Mindwave Video camera	Execute voluntary eye-blink	Real-time detection of the voluntary eye-blinking	Experimental data obtained from the PhysioBank free database publicly available at www.physio-net.org
[52]	OpenViBE Muse Monitor Python	Muse Biopac 100c OpenBCI Platform	Watching a video Reading an article Execute voluntary eye-blinks	The voluntary and involuntary eye-blink detection algorithm	Annotated 2300+ eye blinks EEG dataset

BCI2000 software platform [53] proposed in 2000 as a free-of-charge solution enabled general-purpose brain-computer interface systems research. BCI2000 is a standalone development instrument that integrates neuronal biopotentials, signal analysis techniques, EEG data acquisition equipment, and communication protocols. Nevertheless, BCI2000 is not updated anymore to incorporate the necessary functionalities required by the newest NeuroSky EEG portable headset released in 2018. Therefore, currently, there is no evidence that BCI2000 is still applicable for the acquisition, processing, and classification of the raw EEG signal detected by the embedded sensor of the cost-effective NeuroSky Mindwave Mobile portable headset.

OpenViBE software platform [54] proposed in 2010 as an open-source development solution allowed the design, implementation, testing, and use of complex paradigms underlying a brain-computer interface system. OpenViBE proved to be a valuable research tool, especially for the experimentation of advanced BCI paradigms involving P300, SSVEP (Steady State Visually Evoked Potentials), Motor Imagery, and Virtual Reality based neurofeedback. These versatile features are provided only by the expensive EEG portable headsets: OpenBCI, OpenEEG, and Emotiv. Although the OpenViBE covers NeuroSky functionality, according to their forum, the most recently released version of NeuroSky Mindwave Mobile is possibly not compatible with their software platform anymore. Therefore, the customizable use of OpenViBE general-purpose software platform is not straightforward for enabling young researchers to integrate the functionality of microcontroller-based boards such as Arduino and Raspberry Pi that are frequently necessary to develop experimental BCI prototypes. Moreover, to benefit from all the outstanding features of OpenViBE, it is necessary to have solid programming skills.

EEGLAB Matlab-based software platform [55] created in 2000 as an open-source toolbox for analyzing the EEG signals can also contribute to developing a brain-computer interface system. Although EEGLAB provides advanced EEG processing techniques, it does not include machine learning-based methods for classifying the raw EEG signal to get commands for a BCI application.

The development of the proposed research virtual instrument 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 models based on the Artificial Neural Networks (ANN) through the EEG signal classification process by setting specific values for hyperparameters or searching the optimized ones;
- The testing phase of the trained model is based on Artificial Neural Networks (ANN) so that it will be possible to measure the accuracy and precision of the classification process.

Reviewing the current scientific literature shows that very few articles [49],[51-52] deliver EEG datasets consisting of the specific patterns corresponding to simple, double, or triple voluntary eye-blinks. Such EEG datasets covering eye-blinking are necessary for testing the performance of the EEG data processing-based algorithms to remove artifacts and evaluate the accuracy of the recognition of multiple voluntary eye-blinks used as control signals in brain-computer interface applications. Therefore, a significant contribution brought by this paper is consisting of delivering a large number of 100 EEG datasets aimed for the assessment of 50 different machine learning-based models accomplishing the classification of the following four states: no eye blink detected, one eye-blink detected, two eye-blinks detected and three eye-blinks detected.

Thus, 50 EEG training datasets (format .csv) contain 4000 recordings: 1000 – No Eye-Blink; 1000 – One Eye-Blink; 1000 – Two Eye-Blinks and 1000 – Three Eye-Blinks. The 50 EEG datasets contain 50 different mixtures representing 50 possible combinations between the selection of ten EEG rhythms and ten statistical features.

The ten EEG rhythms are the time and frequency domain of raw, delta, theta, alpha, beta, and gamma. The ten statistical features are: mean, median, route mean square, standard deviation, Kurtosis coefficient, mode, sum, skewness, maximum value, and range = maximum-minimum.

The 50 EEG datasets provided the training of the 50 classification models (format .json) based on neural networks delivered by the current research. In addition, there resulted in 50 EEG testing datasets (format .csv), each containing 800 recordings: 200 – No Eye-Blink; 200 – One Eye-Blink; 200 – Two Eye-Blink and 200 – Three Eye-Blinks. The EEG training and testing datasets have a similar content related to the values resulting from combining different EEG rhythms and statistical features.

Regarding the structure of the current paper, Section 2 provides some detailed insights about the materials and methods necessary to develop the LabVIEW application, Section 3 shows the obtained results, and Section 4 analyzes the outcomes by comparing them with previous similar achievements. Finally, Section 5 comprises some conclusions about the overall project work and highlights the future research directions.

2. Materials and Methods

2.1. LabVIEW graphical programming environment

This paper proposes a BCI research-related virtual instrument developed in the LabVIEW graphical programming environment [56], which provides efficient solutions for helping researchers, professors, students, and engineers in their projects and related activities. LabVIEW proved successful results regarding developing 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 valuable functions. Therefore, LabVIEW offers the benefit of allowing rapid and straightforward communication with several acquisition devices, hardware systems aimed for data processing and actuators. The programming language underlying LabVIEW is called 'G,' 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 the 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 of using data structures, for example, arrays and clusters. The source code implemented in LabVIEW consists of the Block Diagram, allowing various structures and functions to design compact, maintainable, scalable, and robust programming paradigms. The most frequently used structures are while loop, for loop, and switch case. Some useful functions are Bundle by Name, Build Array, and Search and replace string. The programming paradigms are State Machine, Producer/Consumer, Event Handler, or Master/Slave.

The independent use of the proposed virtual instrument aimed for BCI research on a different computer is possible either by installing the resulted standalone application or by running the executable file with the help of the LabVIEW Runtime Engine package.

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

The single embedded sensor of NeuroSky Mindwave Mobile (second edition, released in 2018) headset [57] enabled the detection of the EEG signal, thanks to advanced functionalities offered by the ThinkGear chipset. 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 placed on the forehead (FP1 location according to the 10-20 International System) or the frontal cerebral lobe of the user. Likewise, the headset contains a clip representing the reference or the ground necessary to close the electrical circuit and 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, the NeuroSky chipset increases the accuracy of the EEG signal due to its embedded advanced filters aimed for noise reduction. The technical features give further benefits: the sampling rate is equal to 512 Hz, the bandwidth range 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 [58-59], the NeuroSky Mindwave headset was a preferred choice 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 the proposal of a novel research approach on the development of a portable brain-computer interface system. An original LabVIEW application addresses this challenge by implementing several custom virtual instruments aimed to integrate the following three stages: 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 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 particular 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 to classify another EEG dataset that can be delivered by using a similar procedure as previously described regarding the proposed mixtures between EEG signals and statistical features.
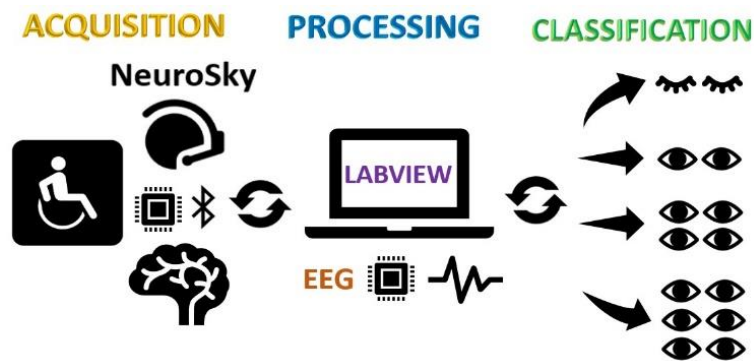
Figure 1. An overview of the proposed LabVIEW application aimed for the acquisition, processing, and classification of the electroencephalographic signal used in a brain-computer interface

The selection of the corresponding virtual button accomplishes each of the above functionalities by opening the tab or graphical windows consisting of customized settings and options. Moreover, each of these tabs comprises a button for returning to the main Configuration graphical window shown in Figure 2.
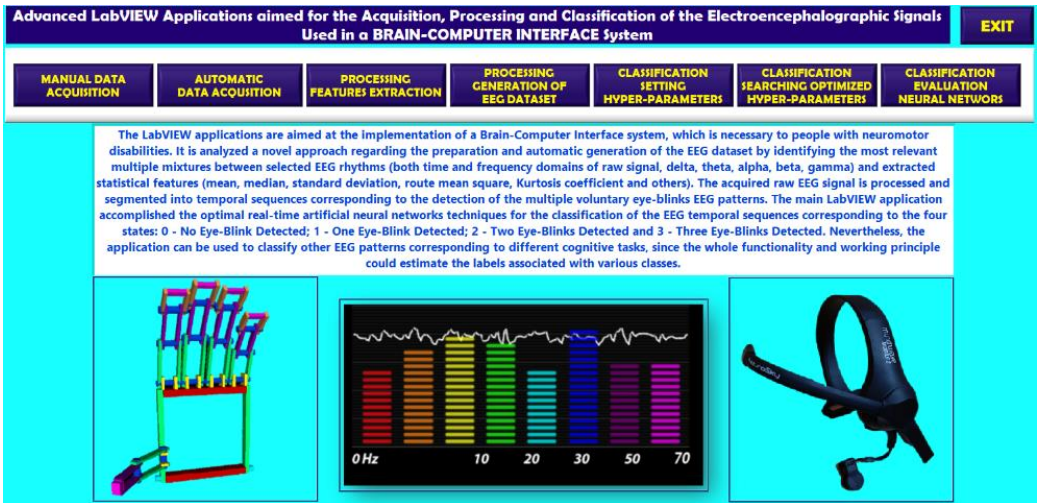
Figure 2. A sequence of the Front Panel showing the Configuration graphical window displaying the virtual buttons corresponding to all the functionalities based on a State Machine paradigm

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

A significant function, included by the LabVIEW NeuroSky toolkit and used to develop the application presented in this paper, is the 'ThinkGear Read – MultiSample Raw (EEG).' This function enables the raw EEG signal acquisition and returns an array containing a specific number of numerical values. The input parameter 'Samples to Read' should specify this number by assigning a numerical value 512, 256, 128, 64, or other. The 'Samples to Read' parameter does not have the same meaning as the 'Sampling Frequency.' According to technical specifications, in the NeuroSky chipset, the sampling frequency is a fixed value, established to 512 Hz, referring to the acquisition of 512 samples in one second. Therefore, setting 'Samples to read = 512' results in 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, a 1D array is a matrix with one dimension, meaning either one row and many columns or one column and many rows. Other functions that allow the communication between LabVIEW and NeuroSky headset were linked (Figure 3): 'Clear Connections,' 'Create Task,' 'Start Task,' 'Signal Quality,' 'Read MultiSample Raw,' and 'Clear Task.' 'Clear Connections' is used to reset all previous connections. 'Create Task' is used for the initial settings of serial data transfer: port name, baud rate, data format. Start Task'

is used to start the connection or to open the communication port. 'Signal Quality' is used to display the value characterizing the percentage of signal quality. 'Read MultiSample Raw' is used to acquire an array of samples of EEG raw signal. 'Clear Task' is used to close the connection or the communication port.
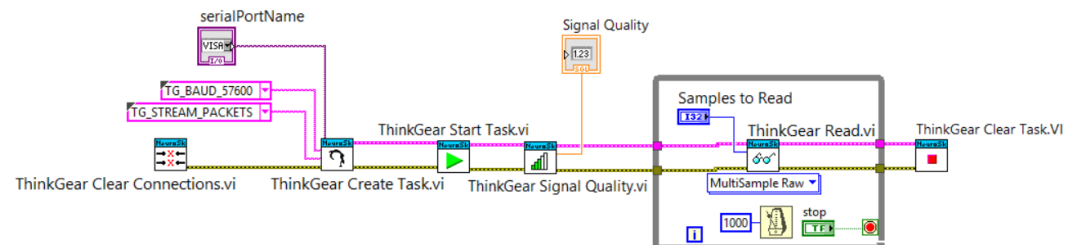
Figure 3. A sequence of the Block Diagram showing the set of the functions allowing the communication between LabVIEW application and NeuroSky headset.

Further, according to Figure 4, the output array of numerical values returned by the 'Read MultiSample Raw' function is used to input the 'Build Waveform' function. In addition, two parameters (t0 = current time in hours, minutes, seconds, and milliseconds and dt = time interval in seconds between data points) are necessary to obtain the appropriate data format to apply a filter for the extraction of a particular range of frequencies, which can be graphically displayed. The 'Get Date/Time' function facilitates calculating the 't0 = Current time' parameter. The 'dt = time interval' is given by the division of 1 to 512, taking into account 'Samples to Read = 512'.
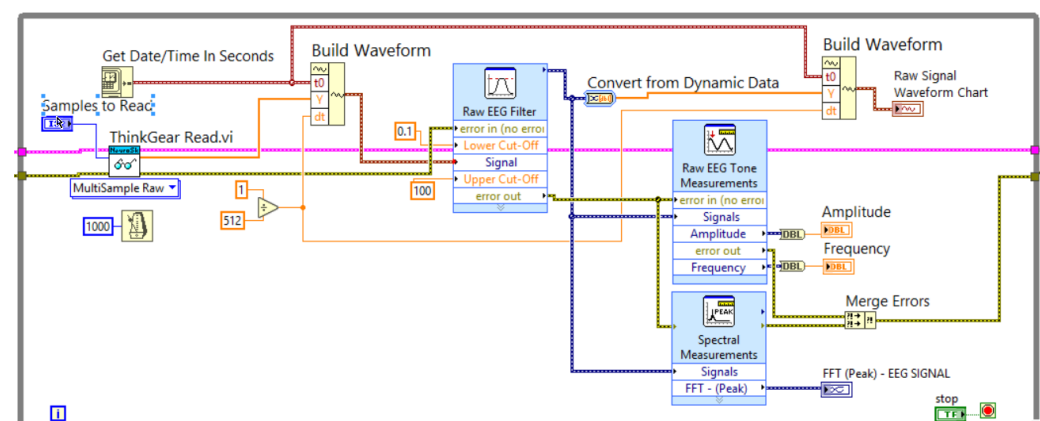
Figure 4. A sequence of the Block Diagram showing the set of the functions allowing the acquisition, processing, and graphical displaying of the raw EEG signal acquired from the NeuroSky

The output of the 'Build Waveform' function is passed through the 'Filter' function so that it results in 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, extracts the gamma (upper cut-off = 30) EEG rhythm. The output of the 'Filter' function is an array of samples or numerical values represented on a Waveform Chart.
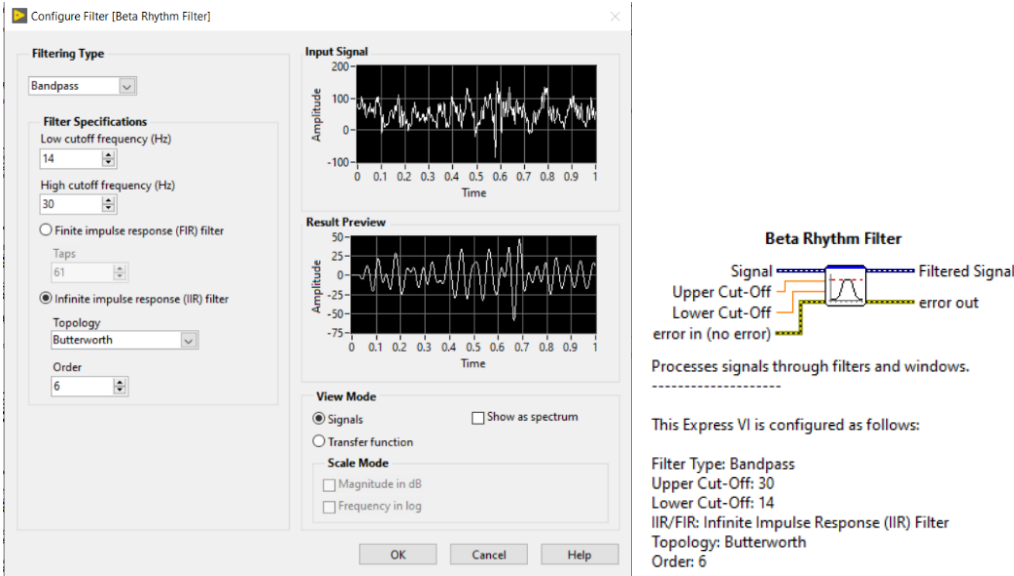
Figure 5. The settings and description corresponding to 'Filter Express VI' included by the 'Signal Analysis Express VIs' LabVIEW palette

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 'Tone Measurements' function (Figure 6) needs the output of the 'Filter' function to allow 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 the 'Tone Measurements' function is 'dynamic data.' Therefore, it is necessary to convert it to double-precision numerical data.
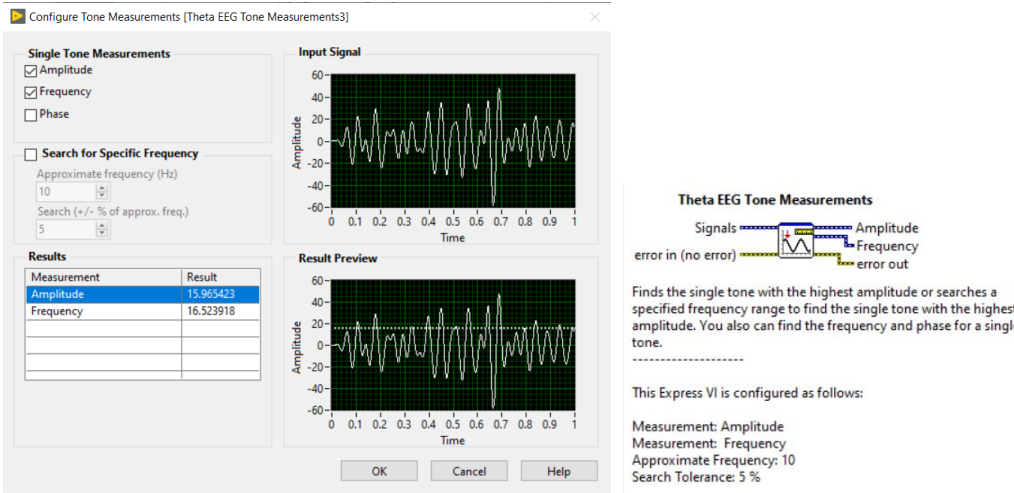
Figure 6. The settings and description corresponding to 'Tone Measurements Express VI' included by the 'Signal Analysis Express VIs' LabVIEW palette

Parallel to these programming sequences, the 'Spectral Measurements' function (Figure 7) follows after the 'Filter' function. Accordingly, a Waveform Graph displays the power spectrum of each EEG rhythm (delta, theta, alpha, beta, and gamma) extracted from the raw EEG signal by using the 'Filter' function. Likewise, the 'Spectral Measurements' LabVIEW function can perform the following features: averaged magnitude spectrum and phase spectrum on a specific signal.
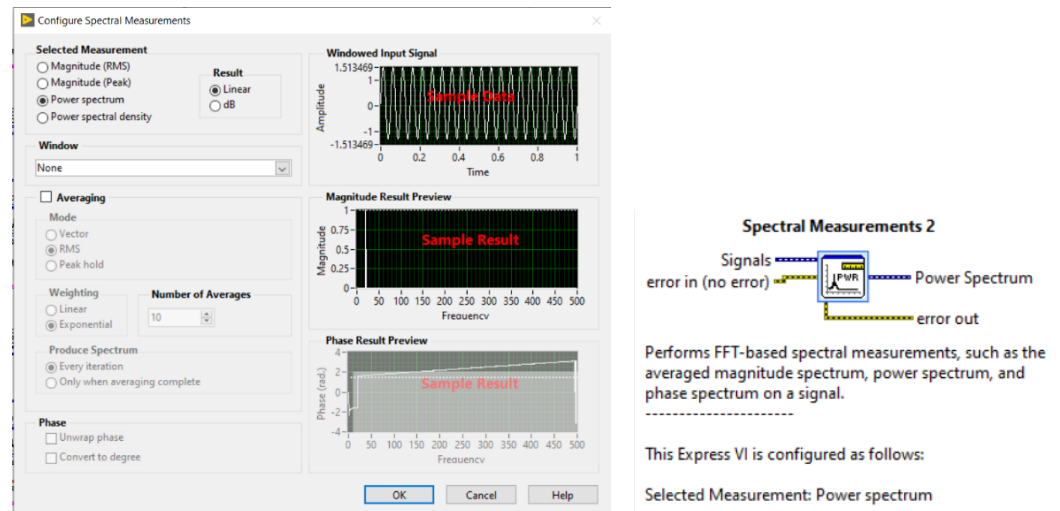
Figure 7. The settings and description corresponding to 'Spectral Measurements Express VI' included by the 'Signal Analysis Express VIs' LabVIEW palette

A case structure (Figure 8) encompasses all the previously mentioned functions: Filter – Tone Measurements – Spectral Measurements. The selector of the case structure is a button consisting of 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 represent the five EEG rhythms, which can be activated or deactivated by pressing those buttons. Therefore, the user can select specific EEG rhythms displayed on either the Waveform Chart corresponding to the time domain (that is, the output of the 'Filter' function) or the Waveform Graph, associated with the frequency domain (that is the output of the 'Spectral Measurements' function). A while loop includes all the five case structures. The while loop also contains the same network of functions previously described regarding displaying the raw EEG signal. Using the 'while loop,' the LabVIEW application runs in manual mode until occurring an exit condition.
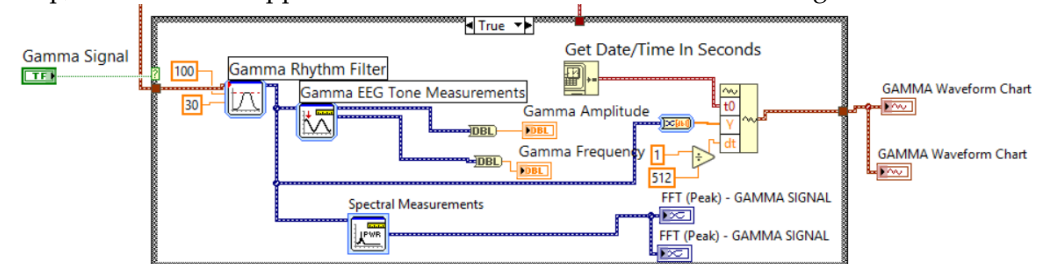
Figure 8. The settings and description corresponding to 'Spectral Measurements Express VI' included by the 'Signal Analysis Express VIs' LabVIEW palette

Considering that the State Machine design pattern is underlying the Block Diagram, the transition between states or different sequences should be quick and straightforward. Therefore, by pressing the 'Config' button, the exit condition is fulfilled so that the running of the manual mode of data acquisition (Figure 9) stops. The application continues to run in the 'Configuration' state, where the user can select another option: automatic mode of EEG data acquisition or features extraction or training neural networks model.
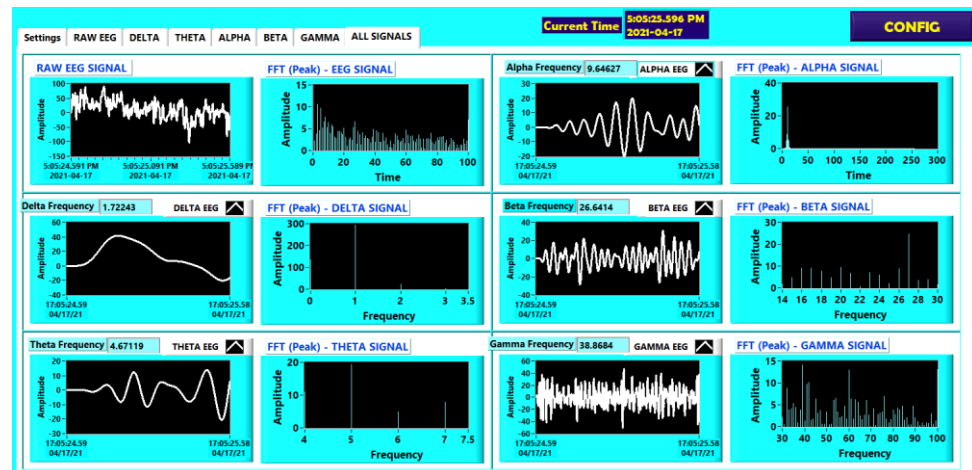
Figure 9. Monitoring EEG rhythms by graphical displaying their time variation (Waveform Charts – first and third columns) and its frequency variation (Waveform Graph – second and fourth columns)

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 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 implementing 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)', 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 a mental task because it is an artifact across the acquired EEG signal, and it can be precisely detected and categorized as a robust control signal.
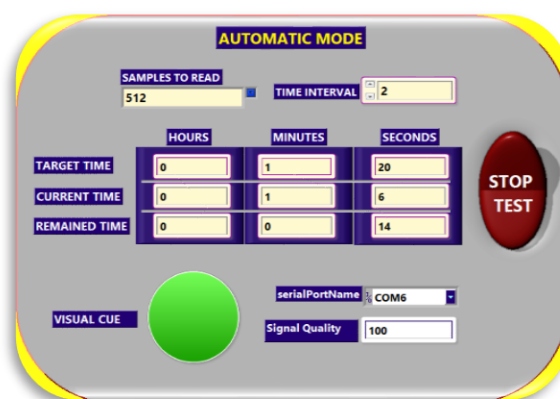
Figure 10. A sequence of the Front Panel showing the virtual chronometer aimed to calculate both the elapsed and remained time for the automatic acquisition of EEG signal

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 in 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. Each second will result in a 1D array

or a set of 512 samples for every one of the 12 EEG signals (both time and frequency domain).
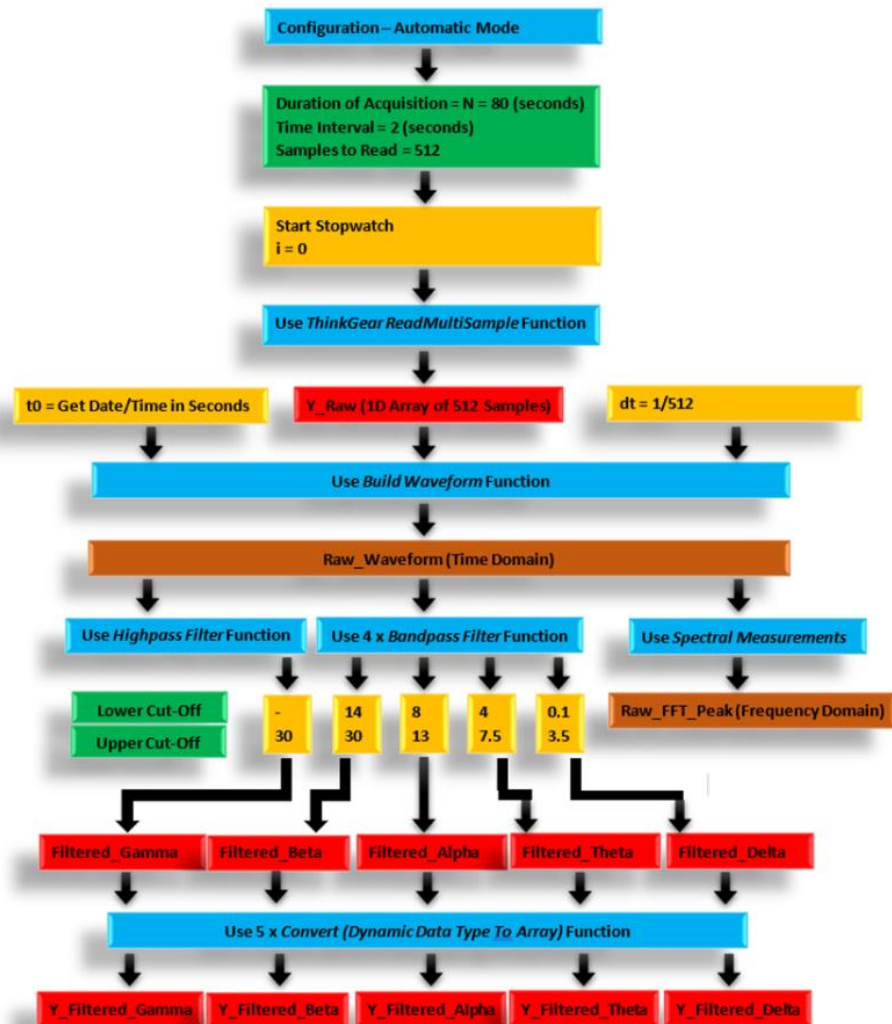
Figure 11. A diagram representing the block instructions underlying the implementation of Data Acquisition in the Automatic Mode – first view showing all the steps leading to obtaining the raw EEG signal (time and frequency domain) and extracting the EEG rhythms (gamma, beta, alpha, theta, and delta)

Accordingly, when the chronometer stops, indicating the finish of the EEG signal acquisition in automatic mode, 12 x 2D arrays will be returned. They consist of six types of EEG signals in the Time Domain (Figure 12) plus six types of EEG signals in Frequency Domain (FFT – Peak – Figure 13): raw, delta, theta, alpha, beta, and gamma. A 2D array is a matrix containing 80 rows (temporal sequences) and 512 columns (512 samples).
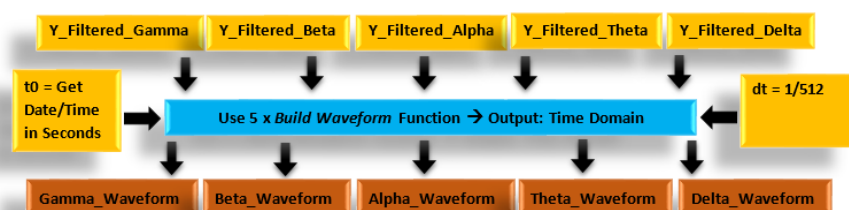
Figure 12. A diagram representing the block instructions underlying the implementation of EEG Data Acquisition in the Automatic Mode – the second view showing the five EEG rhythms obtained in the time domain.
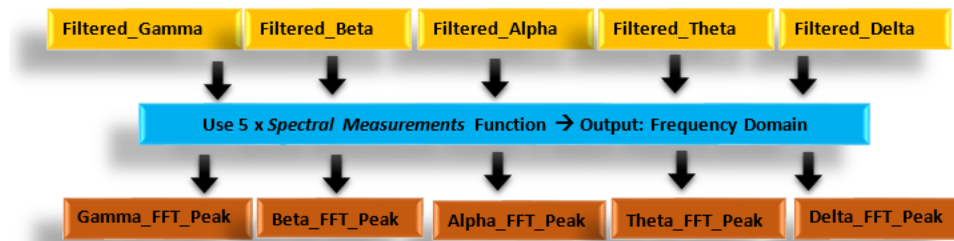
Figure 13. A diagram representing the block instructions underlying the implementation of EEG Data Acquisition in the Automatic Mode – the third view showing the five EEG rhythms obtained in the frequency domain.

2.6. The preparation of the EEG temporal sequences

Before applying the EEG acquired data preparation algorithm, 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. Table 3 shows every one of the 12 x 2D arrays (both time and frequency domain of raw, gamma, beta, alpha, theta, delta) before the preparation of the EEG data.

Table 3. 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. Table 4 shows every one of the 12 x 2D arrays (both time and frequency domain of raw, gamma, beta, alpha, theta, delta) after the preparation of the EEG data. Further, it results in the extraction or calculation of features (for example: mean, median, standard deviation) from every one of the 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 containing 40960 samples corresponding to the EEG rhythms previously saved .csv files. The second stage consists of implementing a customized VI aiming at converting each of the 12 x 2D arrays into 12 x 3D arrays, the third dimension results from the separate extraction of two rows or two sequences composed 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 by removing the third dimension because the previously extracted two rows or two sequences should form a single row or a single sequence and all the resulted rows/sequences determine a 2D array.

Table 4. 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 the graphical display in time and frequency domains

After the preparation of EEG data is finished, according to Figure 14, the user can manually set the label for each EEG temporal sequence by visually checking the graphical display 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 assess each EEG rhythm (delta, theta, alpha, beta, and gamma).
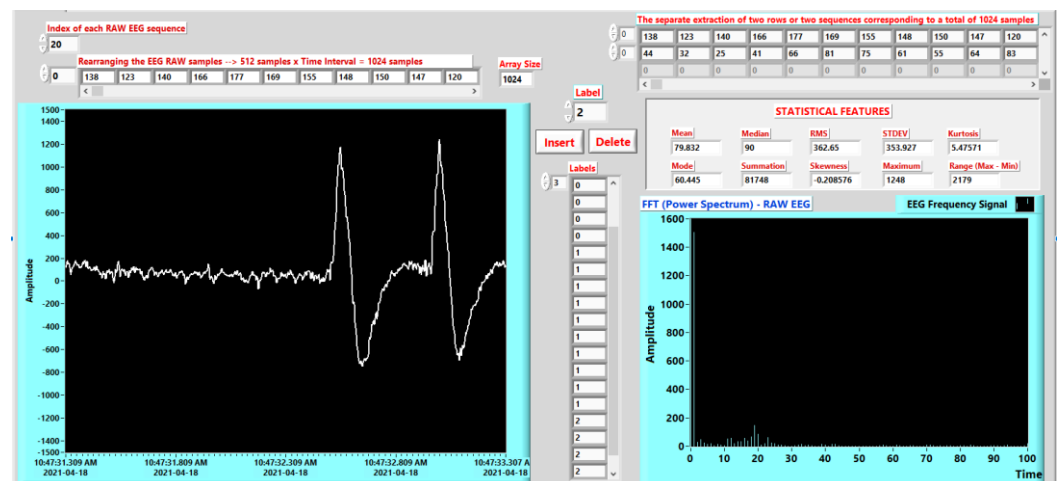
Figure 14. A sequence of the Front Panel – EEG Raw Signal - showing various options allowing the label assignment for each EEG temporal sequence by visually checking the graphical display in time and frequency domains

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

Further, each label saved in a numerical array generates the EEG datasets aimed for training and testing a neural network model. Using the 'Statistics Express VI' included by the 'Signal Analysis Express VIs' LabVIEW functions palette, it results in the calculation of the statistical features. They determine the content of the EEG datasets stored for each EEG temporal sequence. According to Figure 15, there result in the following statistical features: arithmetic mean, median, mode, the sum of values, root mean square (RMS), standard deviation, variance, Kurtosis, skewness, maximum, minimum, and range (maximum-minimum).
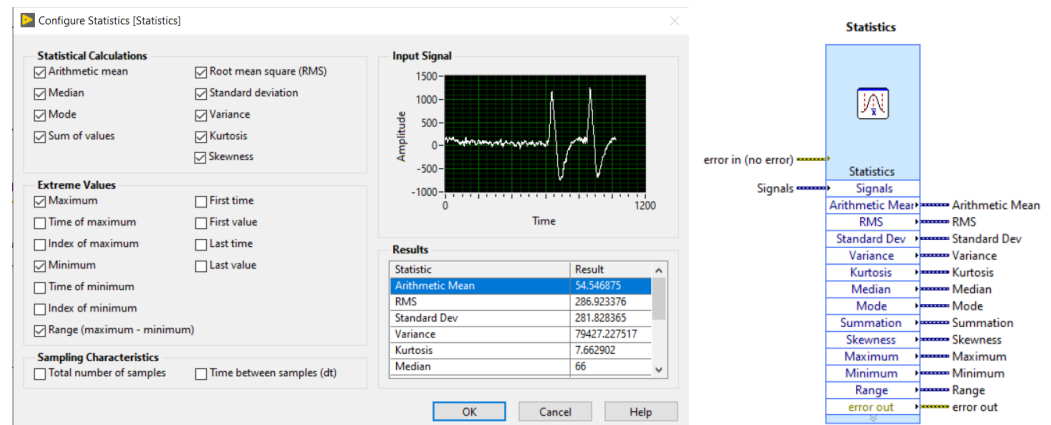
Figure 15. The settings corresponding to 'Statistics Express VI' included by the 'Signal Analysis Express VIs' LabVIEW palette

As previously stated, the working principle of the proposed LabVIEW application is applied to classify multiple voluntary eye-blinks, although it provides the benefit of processing and recognizing various EEG patterns associated with 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) displayed in Figure 16.
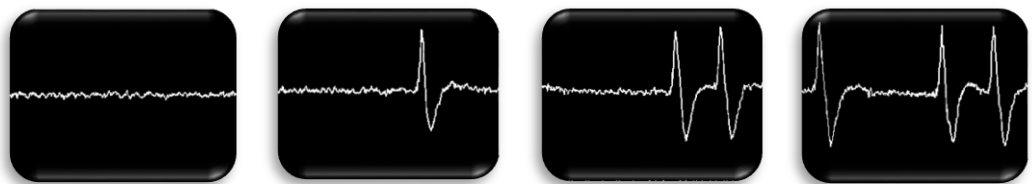
Figure 16. An example for the LabVIEW display of the EEG temporal sequences associated with each label: 0 – No Eye-Blink; 1 – One Eye-Blink; 2 – Two Eye-Blinks and 3 – Three Eye-Blinks.

2.8. The generation of training and testing EEG dataset

Multiple mixtures between the selected EEG signals and the extracted features are necessary to generate the training and testing EEG dataset. The boxes from the top side of Figure 17 represent 12 x 2D_arrays, 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 to insert 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 extract only those 2D arrays corresponding to the previously selected signals by using the checkboxes. Therefore, the resulting 3D array comprises pages equal to selected signals, 40 rows, and 1024 samples.

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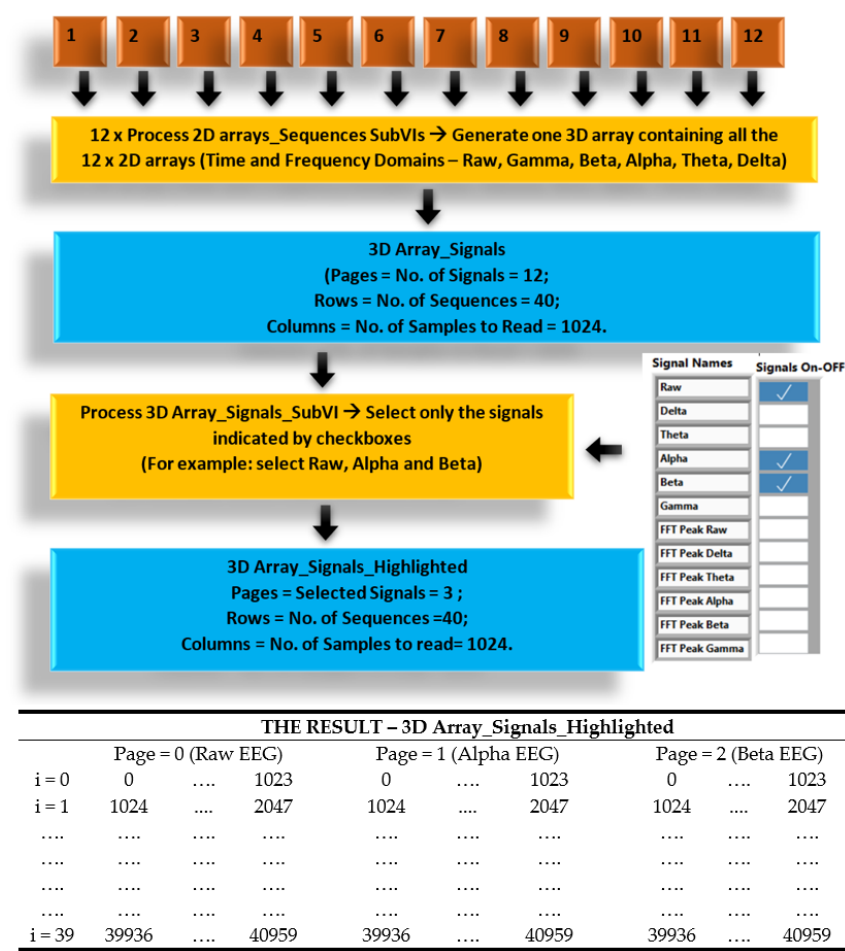
Figure 17. First view of the diagram representing the block instructions underlying the generation of training or testing dataset based on multiple mixtures between the selected EEG signals and the extracted features

The third customized subVI, called 'Process 3D Array_Signals_Highlighted_SubVI', was created to enable feature extraction by calculating the specified statistical measurements against the previously selected signals. Thus, by adding a new dimension, 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.
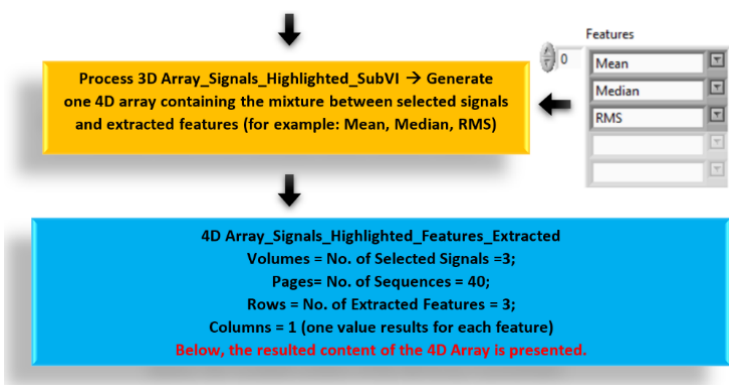
Figure 18. The second view of the diagram representing the block instructions underlying the generation of training or testing dataset based on multiple mixtures between the selected EEG signals and the extracted features

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

VOLUME = 2 (BETA EEG)

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

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 converted to String type. Then, a .csv file representing the training/testing dataset stores the 2D array_ Signals_Highlighted_ Features_Extracted (Figure 21) of string elements. 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 rows.
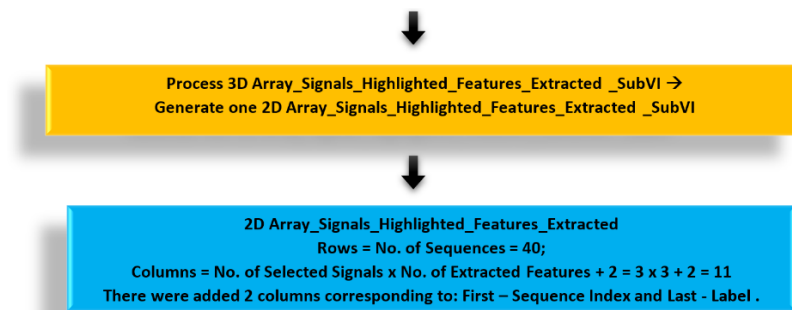
Figure 21. Fifth view of the diagram representing the block instructions underlying the generation of training or testing dataset based on multiple mixtures between the selected EEG signals and the extracted features

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

This phase involves applying the generated dataset to the classification process based on artificial neural networks (NN). Using the default subVIs included by the 'Analytics and Machine Learning' (AML) toolkit [60] results in the classification process.

'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.

'Normalize. vi' is used to normalize the training data with the 2-Score or Min-Max Method. Normalization is related to scaling each value of the 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 specific 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 AML LabVIEW toolkit [60], the 'hidden layer type' is related to the activation function applied to the neurons from the hidden layer type. Table 5 defines the available activation functions. According to Table 6, Sigmoid and Softmax are the two activation functions available in the neurons regarding the' output layer type.' Tables 7 shows the mathematical formulas for the supported cost functions type.

Tolerance or max iteration parameter value constitutes the criteria determining training stops or fitting the neural networks model. 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 5. 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	$f(x) = \dfrac{1}{1 + e^{-x}}$	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 6. 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	$f(x) = \dfrac{1}{1 + e^{-x}}$	x - the activation value of the output neuron
Softmax	$f(x) = \dfrac{e^{x_i}}{\sum_{j=1}^{n} e^{-x_j}}$	x - the activation value of the output neuron

Table 7. 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	$error = \dfrac{1}{l}(t - y)^2$	t – the target value, y – output value, l – number of training samples
Cross-entropy	$f(x) = -\dfrac{1}{l}\left(\sum_{j=1}^{n} t_j ln y_j\right)$	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 the 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 consists of one section and the training data comprises 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 'weighted metric configuration' type.

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.
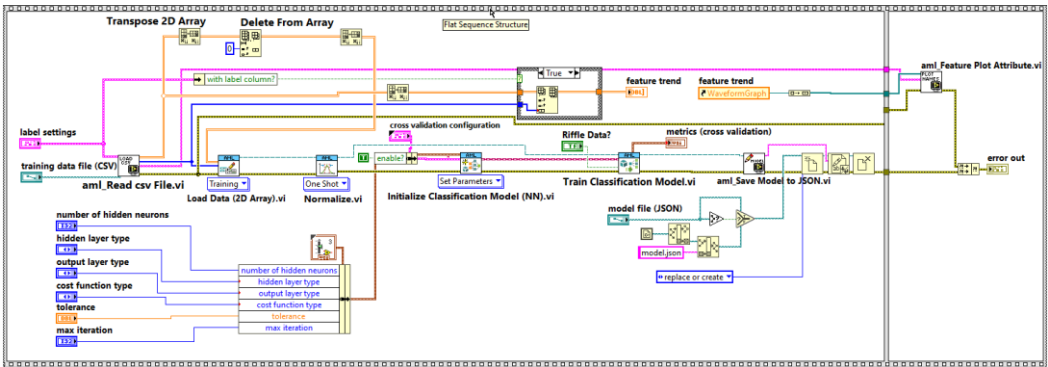
Figure 22. A sequence of the Block Diagram displaying the corresponding LabVIEW programming functions used to enable the setting of hyperparameters for training the neural networks model
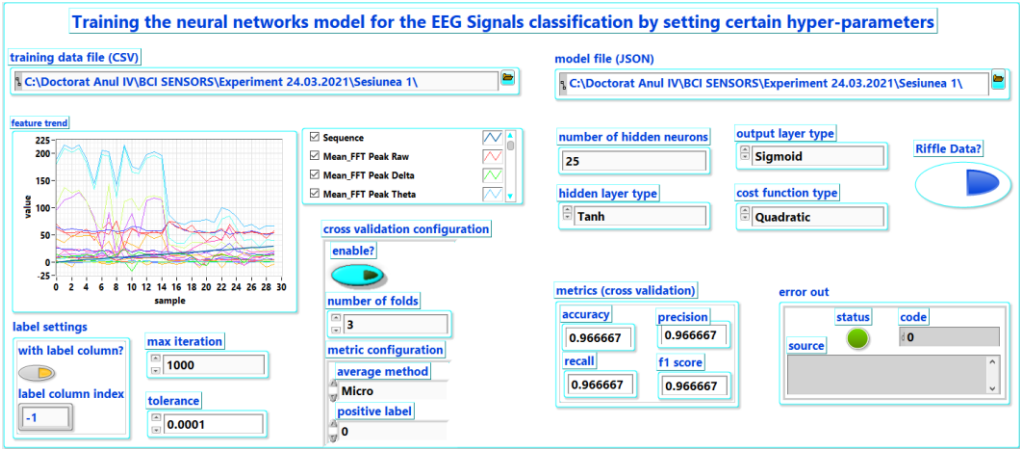
Figure 23. A sequence of the Front Panel showing the graphical window corresponding to training the neural networks model for the EEG signal classification by setting certain hyper-parameters

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 to the 'Train Classification Model. vi' could use a grid search to find the optimal set of parameters. This technique is underlying the training of the neural networks models from the current research paper because it is more reliable, efficient, and straightforward by enabling the option 'Exhaustive Search' to determine those metrics (accuracy, precision, recall, and F1 score) with all the possible mixtures between hyper-parameters. It will result in a mixture including the optimal hyper-parameters necessary to get the highest values for the metric specified in the 'Evaluation Metric' parameter. If the option 'Random Search' was enabled, the 'number of searchings' parameter indicates testing only some possible mixtures between hyper-parameters.

Moreover, the graphical user interface from 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 8.

The current research analyzed 50 generated artificial neural network-based models, and each of them needed a training time interval between 1 and 3 hours.
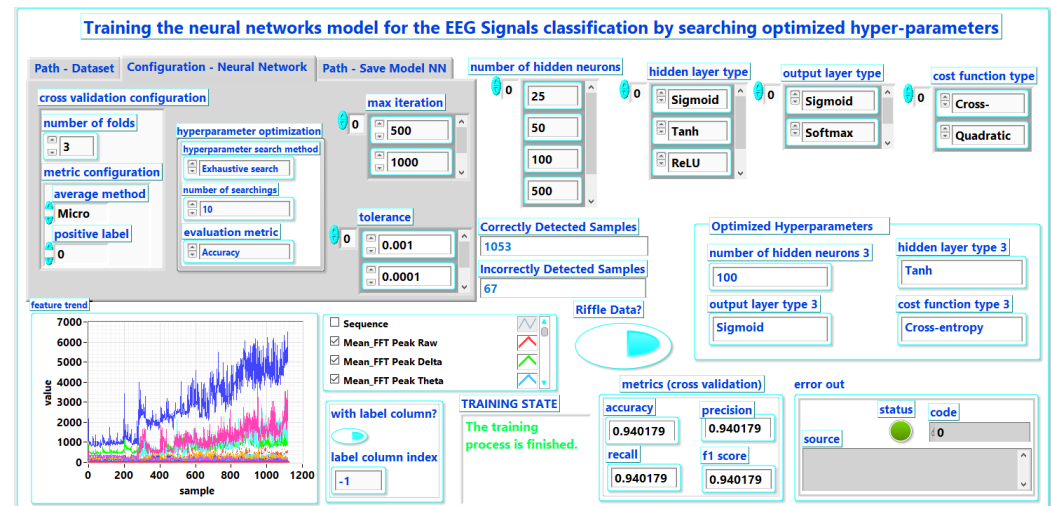
Figure 24. A sequence of the Front Panel showing the graphical window corresponding to training the NN model for the EEG signal classification by searching the optimized hyper-parameters

Table 8. The mathematical formulas for the evaluation metrics (accuracy, precision, f1 score, recall) described in the documentation of the AML LabVIEW toolkit

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

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

The following behavior is initially applicable: whenever the previous phases related to the classification process are running, it results in different values for the evaluation parameters of the NN model: accuracy, precision, recall, and F1 score. These parameters vary due to using 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 the 'Random Number' function. At the initialization of the model, it results in the configuration of the total number of subsets. One of these subsets is aimed for the pre-training phase, while the others are for the pre-testing phase. These phases are preceding the obtaining of the trained classification model. After running each training session, removing the 'Random Number' function will result in the same evaluation parameters. Keeping the exact configuration of the model should be accomplished to get identical results after running each training session.

The LabVIEW application presented in this paper provides flexibility by implementing a novel method that allows a button with two logical states, called 'Riffle Data,' to activate or deactivate the 'Random Number' function. Thus, the LabVIEW application can interactively enable or disable the random generation of the normalized EEG data. Therefore, it is necessary to implement some modifications in the subVIs provided by the 'Analytics and Machine Learning' toolkit. These changes are necessary to make the 'Riffle Data' button available in the main LabVIEW application, outside the subVI where it originally belonged. The updates are related to certain object-oriented programming concepts in LabVIEW explained in the below paragraphs.

The 'Analytics and Machine Learning. lvlib' should be accessed, which is the library of functions contained by the LabVIEW Toolkit. Then, the developer should open the following structure of folders: Classification/Data/2D array/Classes/AML 2D Array.lvclass, to find the 'AML 2D Array. ctl'. This element needs modification by adding a boolean control corresponding to the 'Riffle Data' button. This modification influences the 'aml_Read Data. vi' and 'aml_Write Data. vi' containing a typedef control called 'data.' The developer should add the 'data' typedef control to the 'Riffle Data' button. Further, the developer should open the following structure of folders (Figure 25): Classification/Data/Common/subVIs, to find out the 'Load Training Data (2D array).vi'. A 'Bundle by Name' function is necessary for this virtual instrument to add the 'Riffle Data' Boolean control (button) as a new input element to the 'data' cluster. This button should be available in the main application, as shown below, outside of the subVI implementing its functionality.
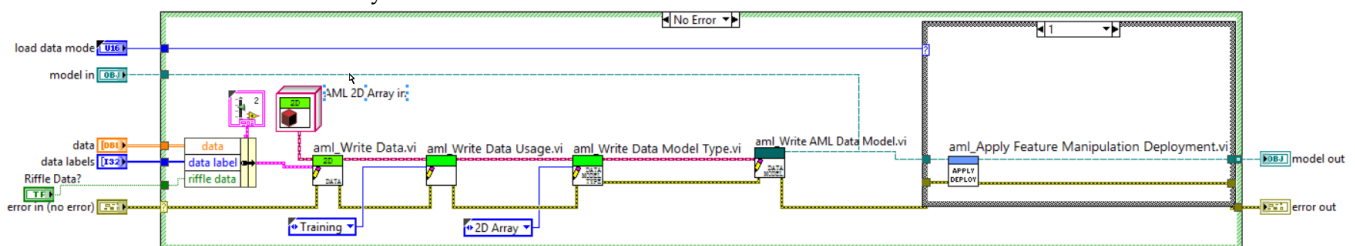
Figure 25. The Block Diagram of 'Load Training Data (2D array).vi' – modified by adding the 'Riffle Data' button

The following phases are necessary 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.

Implementation of the 'aml_ Riffle Training Data. vi' (Figure 26) focuses on re-organizing the training data in random order using a 'Random Number' function. The 'Select' function activates or deactivates the 'Random Number' function, based on the state of the 'Riffle Data' button, which is a Boolean control wired as an input terminal to the connector pane of the 'aml_Riffle Training Data.vi.' 'Select' function is corresponding to the 'if…else' statement from procedural programming.
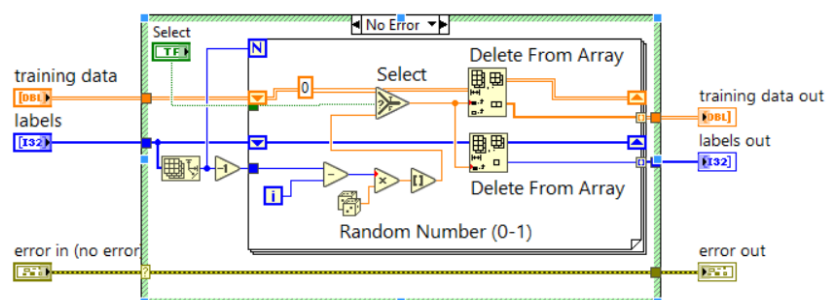
Figure 26. The Block Diagram of 'aml_Riffle Training Data. vi' – modified by adding the 'Select' Function to enable the possibility to activate or deactivate the 'Random Number' Function

- Modify the following virtual instruments: 'aml_Stratified K Folds. vi', 'aml_Cross Validation. vi', 'aml_Auto Tune. vi' and 'Train Classification Model. vi'. Then a Boolean

control should be added, representing a push- handles classification of a button, called 'Riffle Data.'

'Riffle Data' is a wired input terminal to the connector pane of the previously mentioned virtual instruments. Accordingly, the developer can select the 'Riffle Data' button from the Front Panel of the LabVIEW application. If the button is enabled, the 'Random Number' function is deactivated so that the training data is not randomized. If the button is disabled, the 'Random Number' function is activated to randomized the training data.

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

The LabVIEW programming sequence underlying this phase consists of specific virtual instruments presented in this section. First, 'aml_Read CSV File. vi' is used to open and read data and labels from the .csv file containing the testing EEG dataset. Secondly, the developer calls the 'aml_Read Model from JSON.vi' 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' aims to preprocess 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 essential role in assessing the classification model by comparing the predicted labels with initially set labels of the input data. Finally, it will result in evaluation metrics (accuracy, precision, recall, and F1 score) for the trained and tested neural networks model.

3. Results

The proposed research virtual instrument aims to acquire, process, and classify the EEG signals corresponding to neuronal patterns elicited by different cognitive tasks. The eye-blink is considered an artifact across the EEG signal, but it can also be considered a precise control signal in a brain-computer interface application. A simple spike pattern that increases and decreases the biopotential characterizes the voluntary eye-blink resulting from an ordinary effort. 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 classification of multiple voluntary eye-blinks is a testing method of the working principle underlying the proposed BCI research related virtual instrument 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, the sum of values, skewness, Kurtosis coefficient, maximum and range = maximum-minimum).

During the experiments conducted in the current research work, there resulted in 4000 temporal EEG sequences from a single subject (female, 29 years). The duration of each session of EEG data acquisition was 1 minute and 20 seconds. For example, during every period 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.

The results of the current research paper consist of 25 sessions of EEG data acquisition, each of them including 40 EEG temporal sequences. A session of EEG data acquisition set to 80 seconds corresponds to recording a series of 40 EEG temporal sequences. Therefore, there resulted in 25 x 40 = 1000 EEG temporal sequences 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 training dataset generation, the subject has been involved in 4 x 25 sessions of EEG data acquisition, enabling the recording of the 4 x 25 x 40 = 4000 EEG temporal sequences corresponding to the previously mentioned four classes.

Otherwise, for the generation of the testing dataset, the subject has been involved in 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 80 seconds.   Figure 27 shows the previously described general structure. Both training and testing datasets include a column that is assigned the labels.
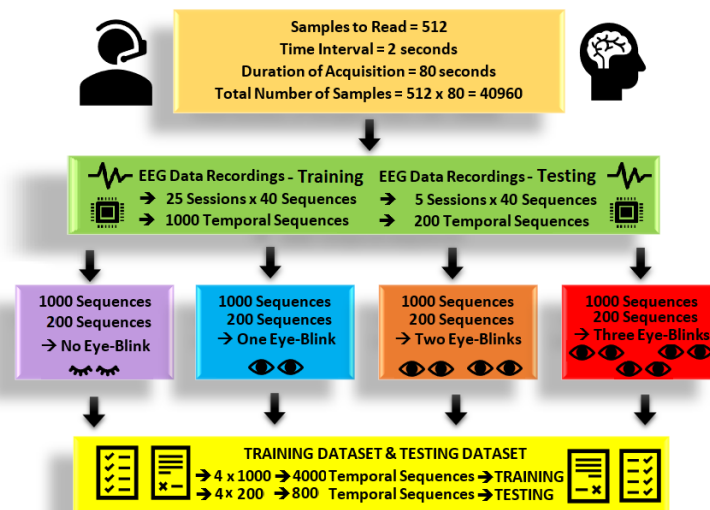
Figure 27. An overview of the proposed paradigm for performing the experimental sessions to get the datasets used for both training and testing of the neural networks model of voluntary multiple eye-blinks classification

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 with a specific class (for example, one eye-blink), it mistakenly resulted in only 35 correctly executed sequences. Other correctly five sequences acquired during another session kept as an alternative replaced the remained wrongly five sequences. Another original application achieved automatic replacement by implementing customized programming sequences in LabVIEW.

Firstly, each generated dataset allowed the initializing, configuring, and training of a specific machine learning (ML) based model to associate the labels with a series of features. Then, it should be able to classify a new input correctly.

Secondly, another generated dataset allowed the testing and validation of the previously obtained model deployed on testing data, including 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 in the following evaluation metrics: accuracy, precision, recall, and F1 score.

The following paragraph clearly describes all the steps necessary to get the results corresponding to the achievements mentioned above.

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

Note. The output consists 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 visual and auditory indicators, 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. 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 result in 50 multiple mixtures, and for every one of them, the EEG signals (Table 9) 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.
    Note. The path's name automatically incremented as follows: training_dataset_1; training_dataset_2; …...; training_dataset_50.
13. 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);
    Note. The path's name automatically incremented as follows: config_1; config_2; …… config_50.
14. Select the 'Processing' Button to generate the Training Dataset and save it with the configuration to .csv files. In the end, it should result in 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 10).
    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 two classes of signals that can be classified.

Table 9. Information regarding the selected EEG signals in the 50 training datasets

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

7	Alpha	All the ten statistical features
8	Beta	All the ten statistical features
9	Gamma	All the ten statistical features
10	FFT Peak Raw	All the ten statistical features
11	FFT Peak Delta	All the ten statistical features
12	FFT Peak Theta	All the ten statistical features
13	FFT Peak Alpha	All the ten statistical features
14	FFT Peak Beta	All the ten statistical features
15	FFT Peak Gamma	All the ten statistical features
16	Raw, delta, theta	All the ten statistical features
17	Raw, alpha, beta, gamma	All the ten statistical features
18	Raw, delta, theta, gamma	All the ten statistical features
19	Raw, beta, gamma	All the ten statistical features
20	Beta, gamma	All the ten statistical features
21	Alpha, beta, gamma	All the ten statistical features
22	Delta, theta, alpha	All the ten statistical features
23	Delta, theta	All the ten 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	Only the six signals in the time domain	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	Only the six signals in the time domain	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 six signals in the 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

15. Repeat steps 1 - 9.
16. Set 'First Index = 40' so that in the resulted .csv file, count the rows starting from 40.
17. Select Label Button so that the first row from the resulted .csv file should not contain the corresponding names or description of columns.
18. Repeat steps 12 - 14.
19. Select the 'Config' Button to return to the main window of the LabVIEW application, then select the 'Classification - Search optimal parameters' Button that enables the initialization, configuration, and training of a model based on neural networks.
20. Select the 'Riffle Data' Button to disable the 'Random Number' function applied to the normalized EEG Data acquired in Automatic Mode.
    Note. This setting is necessary to get fixed/consistent values for evaluation metrics (accuracy, precision, recall, F1 score) used to compare the efficiency of the multiple mixtures between selected signals and extracted features.
21. Insert the correct path to read every one of the 50 .csv files representing the 50 training datasets.
22. Insert a correct path to save each one of the obtained neural networks-based models. In the end, it will result in 50 .json files.
    Note. The path's name automatically incremented as follows: model_1; model_2; …… model_50.
23. Select the '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 Lab-VIEW 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 10 consists of complete results regarding the generation of hyper-parameters and evaluation metrics for every 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 resulted in also 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.

In the end, as shown in Table 11, it results in the evaluation metrics (accuracy, precision, recall, F1 score) corresponding to the deployment of every one of the 50 neural networks-based models.

Table 10. 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 11. Information regarding evaluation metrics after the deployment of the 50 neural networks-based models on the testing datasets

Dataset or Model Neural Networks	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 targets of analyzing the previously described results are the following:

- To identify which of the available ten EEG rhythms are associated with getting NN models with the highest accuracy in the training or testing phases;

- To identify which of the available ten statistical features are associated with getting NN models with the highest accuracy in the training or testing phases;

- To identify which hyper-parameters are associated with getting NN models with the highest accuracy in the training or testing phases;

- To identify how many correctly/incorrectly samples are detected by the NN models with the highest accuracy in the training or testing phases.

According to Table 10, the first highest accuracy of a neural networks-based model is 0.99 for uploading the 19th training dataset. According to Table 9, this training dataset comprises the three EEG rhythms (raw, beta, and gamma) and all the ten available statistical features. The hyper-parameters characterizing the NN model with the first highest accuracy of 0.99 are the following: number of hidden neurons = 20; hidden layer type = Tanh; output layer type = Softmax; cost function = Cross-entropy; number of input neurons = 30; average method = Weighted. Additionally, from the Total = 4000 samples, 3942 samples were correctly detected, and 58 were incorrectly detected.

According to Table 10, the second-highest accuracy of the neural networks-based models is 0.98 for uploading the 16 training datasets numbered as follows: the 1st, 2nd, 4th, 16th, 17th, 18th, 24th, 28th, 30th, 35th, 36th, 37th, 43th, 44th, 45th, and 46th. According to Table 9, these training datasets are composed of the following groups of selected EEG rhythms:

- 1st = all 12 signals in time and frequency domains;
- 2nd and 30th = only the six signals in the time domain;
- 4th = raw;
- 16th and 24th = raw, delta, theta;
- 17th, 28th, and 44th = raw, alpha, beta, gamma;
- 18th = raw, delta, theta, gamma;
- 35th and 43th = raw, beta, gamma;
- 18th, 36th, and 37th = raw, delta, theta, alpha.

According to Table 9, these training datasets are composed of the following groups of extracted statistical features:

- 1st, 2nd, 4th, 16th, 17th, 18th = all the ten statistical features;
- 24th, 28th = mean, median, RMS, standard deviation, Kurtosis Coefficient;
- 30th, 35th, and 36th = median, RMS, standard deviation, Kurtosis Coefficient;
- 37th, 43rd, and 44th = median, RMS, standard deviation, Kurtosis Coefficient, mode, sum, skewness;

The hyper-parameters characterizing the NN model with the second-highest accuracy of 0.98 are the following:

- The number of hidden neurons is between 5 and 500, as it follows for the below NN models/training datasets:
  - 2nd and 43th = 5;
  - 18th, 28th, 35th, and 44th = 10;
  - 1st, 16th, 30th, 37th and 46th = 20;
  - 17th and 45th = 50;
  - 24th and 36th = 200;

- 4th = 500;
- Hidden layer type:
    - 36th and 37th = ReLU;
    - 1st, 2nd, 16th, 18th, 28th, and 44th = Sigmoid;
    - 4th, 17th, 24th, 30th, 35th, 43th, 45th, and 46th = Tanh;
- Output layer type:
    - 1st, 2nd, 16th, 17th, 18th, 28th, 30th, 35th, 36th, 37th, 43th, 44th, 45th, and 46th = Sigmoid;
    - 4th and 24th = Softmax;
- Cost function: = Cross-entropy for all the NN models/training datasets: 1st, 2nd, 4th, 16th, 17th, 18th, 24th, 28th, 30th, 35th, 36th, 37th, 43th, 44th, 45th, and 46th.
- Number of input neurons is between 10 and 120, as it follows for the next NN models/training datasets: 4th = 10; 35th = 12; 24th = 15; 36th = 16; 28th = 20; 43th and 45th =21; 30th = 24; 44th and 46th =28; 17th and 18th = 40; 37th = 42; 2nd = 60; 1st = 120;
- Average method = Weighted for all the NN models/training datasets: 1st, 2nd, 4th, 16th, 17th, 18th, 24th, 28th, 30th, 35th, 36th, 37th, 43th, 44th, 45th, and 46th.

In addition, regarding the NN models that reported a training accuracy of 0.98, from Total = 4000 samples, the number of correctly/incorrectly detected samples is as follows for the below datasets:

- 35th = 3912 correctly detected samples and 88 incorrectly detected samples;
- 1st = 3916 correctly detected samples and 84 incorrectly detected samples;
- 2nd = 3924 correctly detected samples and 76 incorrectly detected samples;
- 16th, 18th, and 30th = 3925 correctly detected samples and 75 incorrectly detected samples;
- 28th and 37th = 3926 correctly detected samples and 74 incorrectly detected samples;
- 45th = 3927 correctly detected samples and 73 incorrectly detected samples;
- 36th and 44th = 3929 correctly detected samples and 71 incorrectly detected samples;
- 24th = 3930 correctly detected samples and 70 incorrectly detected samples;
- 17th = 3933 correctly detected samples and 67 incorrectly detected samples;
- 46th = 3935 correctly detected samples and 65 incorrectly detected samples;
- 4th and 43rd = 3937 correctly detected samples and 63 incorrectly detected samples;

According to Table 10, the third-highest accuracy of the neural networks-based models is 0.95 for uploading the three training datasets numbered as follows: the 3rd, 47th, and 50th. According to Table 9, these training datasets are composed of the following groups of selected EEG rhythms:

- 3rd and 47th = only the six signals in the frequency domain;
- 50th = Raw, Delta, Theta in the frequency domain.

According to Table 9, these training datasets are composed of the following groups of extracted statistical features:

- 3rd = all the ten statistical features;
- 47th and 50th = median, RMS, standard deviation, Kurtosis, Mode, Sum, Skewness;

The hyper-parameters characterizing the NN model with the third-highest accuracy of 0.95 are the following:

- The number of hidden neurons is between either 50 or 100, as it follows for the below NN models/training datasets:
    - 3rd and 50th = 50;
    - 47th = 100;
- Hidden layer type:
    - 3rd and 47th = ReLU;
    - 50th = Tanh;

- Output layer type = Sigmoid for all the NN models/training datasets: 3rd, 47th, and 50th;
- Cost function: = Cross-entropy for all the NN models/training datasets: 3rd, 47th and 50th;
- Number of input neurons is between 21 and 60, as it follows for the next NN models/training datasets: 3rd = 60; 47th = 42; 50th = 21;
- Average method = Weighted for all the NN models/training datasets: 3rd, 47th, and 50th.

In addition, regarding the NN models that reported a training accuracy of 0.95, from a Total = 4000 samples, the number of correctly/incorrectly detected samples is as follows for the below datasets:
- 3rd = 3803 correctly detected samples and 197 incorrectly detected samples;
- 47th = 3799 correctly detected samples and 201 incorrectly detected samples;
- 50th = 3790 correctly detected samples and 210 incorrectly detected samples;

According to Table 11, the first highest accuracy of a neural networks-based model is 0.97 for uploading the four testing datasets numbered as follows: the 1st, 2nd, 30th, and 36th. The groups of selected EEG rhythms and the extracted statistical features composing these testing datasets are described above to the 1st, 2nd, 30th, and 36th training datasets. The above paragraphs described the hyper-parameters characterizing the corresponding NN models.

In addition, regarding the NN models that reported a testing accuracy of 0.97, from a Total = 800 samples, the number of correctly/incorrectly detected samples is as follows for the below datasets:
- 1st = 778 correctly detected samples and 22 incorrectly detected samples;
- 2nd = 775 correctly detected samples and 25 incorrectly detected samples;
- 30th = 776 correctly detected samples and 24 incorrectly detected samples;
- 36th = 773 correctly detected samples and 27 incorrectly detected samples;

According to Table 11, it results that the second-highest accuracy of a neural networks-based model is 0.96 for uploading the five testing datasets numbered as follows: the 17th, 19th, 35th, 37th, and 45th. The groups of selected EEG rhythms and the extracted statistical features composing these testing datasets are described above to the 17th, 19th, 35th, 37th, and 45th training datasets. The above paragraphs described the hyper-parameters characterizing the corresponding NN models.

In addition, regarding the NN models that reported a testing accuracy of 0.96, from a Total = 800 samples, the number of correctly/incorrectly detected samples is as follows for the below datasets:
- 17th = 770 correctly detected samples and 30 incorrectly detected samples;
- 19th = 768 correctly detected samples and 32 incorrectly detected samples;
- 35th = 768 correctly detected samples and 32 incorrectly detected samples;
- 37th = 766 correctly detected samples and 34 incorrectly detected samples;
- 45th = 765 correctly detected samples and 35 incorrectly detected samples;

According to Table 11, it results that the third-highest accuracy of a neural networks-based model is 0.95 for uploading the six testing datasets numbered as follows: the 16th, 18th, 24th, 28th, 44th, and 46th. The groups of selected EEG rhythms and the groups of extracted statistical features composing these testing datasets are described above to the 16th, 18th, 24th, 28th, 44th, and 46th training datasets. The above paragraphs described the hyper-parameters characterizing the corresponding NN models.

In addition, regarding the NN models that reported a testing accuracy of 0.95, from a Total = 800 samples, the number of correctly/incorrectly detected samples is as follows for the below datasets:
- 16th = 763 correctly detected samples and 37 incorrectly detected samples;

- 18th = 759 correctly detected samples and 41 incorrectly detected samples;
- 24th = 761 correctly detected samples and 39 incorrectly detected samples;
- 28th = 763 correctly detected samples and 37 incorrectly detected samples;
- 44th = 761 correctly detected samples and 39 incorrectly detected samples;
- 46th = 762 correctly detected samples and 38 incorrectly detected samples;

Taking into account the previous analysis, the same 20 neural networks-based models reported almost similar highest accuracy values both in the training phase (0.99; 0.97; 0.95) by uploading datasets consisting of 4000 recordings each and in the testing phase (0.97; 0.96; 0.95) by uploading datasets consisting of 800 recordings each.

Table 11 summarizes the above discussion about results consisting of complete information related to the top 20 neural networks-based models. Therefore, the significance of the abbreviations corresponding to column headings is the following: A – NN model, 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 – training accuracy, I – testing accuracy, J – selected EEG signals, K – extracted statistical features.

Table 11. Summary content about the top 20 artificial neural networks-based models that reported the highest accuracy values both in the training and testing phase

A	B	C	D	E	F	G	H	I	J	K
1	20	Sigmoid	Sigmoid	Cross-entropy	120	Weighted	0.98	0.97	All 12 signals in the time and frequency domain	All the ten statistical features
2	5	Sigmoid	Sigmoid	Cross-entropy	60	Weighted	0.98	0.97	Only the six signals in the time domain	All the ten statistical features
3	50	ReLU	Sigmoid	Cross-entropy	60	Weighted	0.95	0.83	Only the six signals in the frequency domain	All the ten statistical features
4	500	Tanh	Softmax	Cross-entropy	10	Weighted	0.98	0.94	Raw	All the ten statistical features
16	20	Sigmoid	Sigmoid	Cross-entropy	30	Weighted	0.98	0.95	Raw, delta, theta	All the ten statistical features
17	50	Tanh	Sigmoid	Cross-entropy	40	Weighted	0.98	0.96	Raw, alpha, beta, gamma	All the ten statistical features
18	10	Sigmoid	Sigmoid	Cross-entropy	40	Weighted	0.98	0.95	Raw, delta, theta, gamma	All the ten statistical features
19	20	Tanh	Softmax	Cross-entropy	30	Weighted	0.99	0.96	Raw, beta, gamma	All the ten statistical features
24	200	Tanh	Softmax	Cross-entropy	15	Weighted	0.98	0.95	Raw, delta, theta	Mean, median, RMS, Standard Dev, Kurtosis
28	10	Sigmoid	Sigmoid	Cross-entropy	20	Weighted	0.98	0.95	Raw, alpha, beta, gamma	Mean, median, RMS, Standard Dev, Kurtosis
30	20	Tanh	Sigmoid	Cross-entropy	24	Weighted	0.98	0.97	Raw, delta, theta, alpha, beta, gamma	Median, RMS, Standard Dev, Kurtosis
35	10	Tanh	Sigmoid	Cross-entropy	12	Weighted	0.98	0.96	Raw, beta, gamma	Median, RMS, Standard Dev, Kurtosis
36	200	ReLU	Sigmoid	Cross-entropy	16	Weighted	0.98	0.97	Raw, delta, theta, alpha	Median, RMS, Standard Dev, Kurtosis
37	20	ReLU	Sigmoid	Cross-entropy	42	Weighted	0.98	0.96	Raw, delta, theta, alpha, beta, gamma	Median, RMS, Standard Dev, Kurtosis, Mode, Sum, Skewness
43	5	Tanh	Sigmoid	Cross-entropy	21	Weighted	0.98	0.94	Raw, beta, gamma	Median, RMS, Standard Dev, Kurtosis, Mode, Sum, Skewness

44	10	Sigmoid	Sigmoid	Cross-entropy	28	Weighted	0.98	0.95	Raw, alpha, beta, gamma	Median, RMS, Standard Dev, Kurtosis, Mode, Sum, Skewness
45	50	Tanh	Sigmoid	Cross-entropy	21	Weighted	0.98	0.96	Raw, delta, theta	Median, RMS, Standard Dev, Kurtosis, Mode, Sum, Skewness
46	20	Tanh	Sigmoid	Cross-entropy	28	Weighted	0.98	0.95	Raw, delta, theta, alpha	Median, RMS, Standard Dev, Kurtosis, Mode, Sum, Skewness
47	100	ReLU	Sigmoid	Cross-entropy	42	Weighted	0.95	0.83	Only the six signals in the frequency domain	Median, RMS, Standard Dev, Kurtosis, Mode, Sum, Skewness
50	50	Tanh	Sigmoid	Cross-entropy	21	Weighted	0.95	0.81	FFT Peak Raw, FFT Peak Delta, FFT Peak Theta	Median, RMS, Standard Dev, Kurtosis, Mode, Sum, Skewness

Further, to determine the first high-performance NN models from the top 20, the sum of the correctly detected samples obtained at the training phase and the correctly detected samples at the testing phase was calculated and displayed in Table 12. Then, taking into account that the maximum sum of correctly detected samples is 4800 (4000 samples for the training phase and 800 samples for the testing phase), it results in the overall accuracy of the top 20 NN models.

Thus, the NN model that reported the highest accuracy of 98.13% was trained and tested by uploading a dataset comprising all the possible mixtures between the raw, beta, and gamma EEG rhythms and all the ten statistical features. Accordingly, the maximum number of correctly detected samples was 4710 out of 4800.

The significance of the abbreviations corresponding to column headings from Table 12 is the following: A – NN model, B – selected EEG signals, C – extracted statistical features, D – correctly detected samples at training the NN model, E – correctly detected samples at testing the NN model, F – total number of correctly detected samples, G – accuracy.

Table 12. A summary content about the top 20 artificial neural networks-based models with the highest sum of correctly detected samples from the training and testing phase

A	B	C	D	E	F	G
19	Raw, beta, gamma	All the 10 statistical features	3942.00	768.00	4710.00	98.13
17	Raw, alpha, beta, gamma	All the 10 statistical features	3933.00	770.00	4703.00	97.98
36	Raw, delta, theta, alpha	Median, RMS, Standard Dev, Kurtosis	3929.00	773.00	4702.00	97.96
30	Raw, delta, theta, alpha, beta, gamma	Median, RMS, Standard Dev, Kurtosis	3925.00	776.00	4701.00	97.94
2	Only the 6 signals in time domain	All the 10 statistical features	3924.00	775.00	4699.00	97.90
46	Raw, delta, theta, alpha	Median, RMS, Standard Dev, Kurtosis, Mode, Sum, Skewness	3935.00	762.00	4697.00	97.85
1	All 12 signals in time and frequency domain	All the 10 statistical features	3916.00	778.00	4694.00	97.79
37	Raw, delta, theta, alpha, beta, gamma	Median, RMS, Standard Dev, Kurtosis, Mode, Sum, Skewness	3926.00	766.00	4692.00	97.75

45	Raw, delta, theta	Median, RMS, Standard Dev, Kurtosis, Mode, Sum, Skewness	3927.00	765.00	4692.00	97.75
24	Raw, delta, theta	Mean, median, RMS, Standard Dev, Kurtosis	3930.00	761.00	4691.00	97.73
43	Raw, beta, gamma	Median, RMS, Standard Dev, Kurtosis, Mode, Sum, Skewness	3937.00	754.00	4691.00	97.73
44	Raw, alpha, beta, gamma	Median, RMS, Standard Dev, Kurtosis, Mode, Sum, Skewness	3929.00	761.00	4690.00	97.71
28	Raw, alpha, beta, gamma	Mean, median, RMS, Standard Dev, Kurtosis	3926.00	763.00	4689.00	97.69
16	Raw, delta, theta	All the 10 statistical features	3925.00	763.00	4688.00	97.67
4	Raw	All the 10 statistical features	3937.00	749.00	4686.00	97.63
18	Raw, delta, theta, gamma	All the 10 statistical features	3925.00	759.00	4684.00	97.58
35	Raw, beta, gamma	Median, RMS, Standard Dev, Kurtosis	3912.00	768.00	4680.00	97.50
3	Only the 6 signals in frequency domain	All the 10 statistical features	3803.00	661.00	4464.00	93.00
47	Only the 6 signals in frequency domain	Median, RMS, Standard Dev, Kurtosis, Mode, Sum, Skewness	3799.00	664.00	4463.00	92.98
50	FFT Peak Raw, FFT Peak Delta, FFT Peak Theta	Median, RMS, Standard Dev, Kurtosis, Mode, Sum, Skewness	3790.00	647.00	4437.00	92.44

The results obtained by the current research confirm, bring complete information, reveal new insights, add improvements, and thoroughly explore the stages underlying the classification of voluntary eye-blinking also addressed in other scientific articles: [47], [48], and [50]. According to the paper [47], the extracted features were: maximum amplitude, minimum amplitude, and the Kurtosis coefficient. According to the paper [48], the extracted features were: Kurtosis coefficient, maximum amplitude, and minimum amplitude. According to the paper [50], the extracted features were: minimum value, maximum value, median, mode, and standard deviation.

The future versions of the presented LabVIEW-based virtual application for BCI research should also include additional features, such as average power spectral density, average spectral centroid, and average log energy entropy, addressed by [61]. These features were extracted from alpha and beta EEG rhythms, previously analyzed in time and frequency domains to classify three mental activities: quick math solving, do nothing (relax), and playing a game [61]. Moreover, an extensive comparative analysis could complete the current discussion about research regarding the values of statistical features and EEG rhythms associated with each of the four detected states: no eye-blink, one eye-blink, two eye-blink, and three eye-blink. Regarding the content of the 19th dataset, a comparison is possible between the values of Mean_Raw, Mean_Beta, and Mean_Gamma. A separate set of values is corresponding to each of the four previously described states. In the same way, there can be determined and compared the values of Median_Raw,

Median_Beta, and Median Gamma and all the other statistical features corresponding to the four different states of voluntary eye-blink classification.

In contrast with the current research that analyzes various mixtures between the selected EEG signals and the extracted statistical features, as well as different architectures of NN models, the papers [47], [48], and [50] focuses on the experimentation of a single method of features extraction and a specific classification technique that is considered optimal.

The paper [47] proposes a binary classifier based on Probabilistic Neural Network (RBF = Radial Basis Function). The EEG signal was acquired at 480 Hz sampling frequency, filtered by applying Butterworth Band Pass, detrended, normalized, and segmented into windows with 480 samples each.

The paper [48] reports the neural networks with the highest performance: R=0.8499 for FFBP (Feed-forward backpropagation) and R=0.90856 for CFBF (Cascade-forward Backpropagation).

Regarding the paper [50], the authors designed an artificial multi-layer neural network with backpropagation, taking into account the following structure: input layer containing 48 neurons based on the extraction of 6 features, three hidden layers, and one output layer including only one neuron. The used activation function is the binary sigmoid.

Moreover, the current paper reported higher values for accuracy than those from the previous scientific articles [47], [48], and [50]. Nevertheless, due to the execution of experiments during pandemic restrictions, both training and testing datasets from this paper were obtained based on the raw EEG data acquisition from a single subject, which could be a limitation that can influence the overall results. Therefore, the plan is to implement and assess an updated version of the presented BCI research-based virtual instrument by involving several healthy or disabled subjects of different categories, including age, profession, psychological traits, and intellectual background.

Otherwise, the current paper's advantage is the convenience of using the most affordable portable EEG headset with only one embedded biosensor for quick set-up and efficient neuronal biopotentials monitoring. The other scientific articles [47], [48] and [50] present experimental activities that involved the following EEG expensive devices: a wireless biomedical monitor, called BioRadio with four channels [47]; RMS EEG 32 Super System with two Ag-AgCl electrodes [48] and OpenBCI Cython with eight channels [50].

Nevertheless, the future version of the presented virtual instrument related to BCI research should be updated to handle more EEG channels to classify complex mental activities efficiently. Moreover, the aim is to use, as much as possible, the most inexpensive portable EEG headsets, for example, Muse and Emotiv Insight, to ensure the simple-to-use working principle and availability for the researchers with minimal experience related to the Brain-Computer Interface field. Although using a commercial portable headset seems to be the most straightforward solution there is still necessary to implement a customized communication protocol enabling the EEG data acquisition in a user-friendly software environment, such as LabVIEW.

Overall, the proposed solution aims for BCI research in the beginning stage by proving a standalone, simple to use virtual instrument with a user-friendly graphical user interface accomplishing all the necessary fundamental functions: EEG data acquisition, processing, features extraction, and classification.

5. Conclusion

This paper proposed a BCI research-related LabVIEW virtual instrument to acquire, process, and classify the EEG signal detected by the embedded sensor of NeuroSky Mindwave Mobile headset, second edition. The artificial neural network-based techniques facilitated the classification process by using the versatile functions included by the 'Analytics and Machine Learning' toolkit. Its functionality was customized to remove the randomization of EEG data.

The new approach described in this paper is consisting of original programming sequences implemented in LabVIEW. The LabVIEW application aims to recognize EEG signal patterns corresponding to different cognitive tasks efficiently. This paper presented the classification of multiple voluntary eye-blinks.

The application developed in the current research is consisting of different states. The first one allows the manual and automatic acquisition mode. The second one enables the processing of the EEG raw signal. The third one is related to preparing the 50 EEG training datasets with 4000 recordings each and 50 EEG testing datasets with 800 recordings each based on the generation of 50 multiple mixtures. Thus, there result in 50 multiple selections between ten EEG rhythms and ten statistical features. The selected EEG rhythms include time-domain: raw, delta, theta, alpha, beta, gamma, and frequency domain – Fast Fourier Transform with Peak parameter applied on the same signals. The extracted statistical features are the following: mean, median, route mean square, standard deviation, Kurtosis Coefficient, mode, summation, skewness, maximum, and range = maximum-minimum.

The use of the LabVIEW application developed in the presented work resulted in automatically identifying the most relevant multiple mixtures. Further, the training EEG datasets facilitated the initialization, configuration, and obtaining the 50 artificial neural networks (ANN) based classification models. After that, the trained ANN models are deployed on the testing EEG datasets to result in evaluation metrics, such as accuracy and precision. The final phase is a comparative assessment to determine the highest values of accuracy reported by the training of the top 20 ANN models (0.99; 0.97; 0.95) and the testing of the same 20 ANN models (0.07; 0.96; 0.95). Accordingly, it analyzed the corresponding mixtures between the selected EEG rhythms and the extracted statistical features underlying the datasets used for training and testing the 20 ANN models. Also, the hyperparameters were listed to generate the top 20 ANN models reporting the highest accuracy. Determining the maximum sum of the correctly detected samples in the training and testing phases resulted in the high-performance ANN model with an accuracy equal to 98.13% that can correctly recognize 4710 out of 4800 samples.

Future research directions are related to further improvements regarding the LabVIEW-based system's assessment by enabling different EEG signal patterns classification. 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 achieved by the execution of training processes based on Supported Vector Machines or Logistic Regression models. Therefore, a future BCI-related research project should also consider the 'Analytics and Machine Learning' LabVIEW toolkit, including these two methods.

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.   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.

17.   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.

18.   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.

19.   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.

20.   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.

21.   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.

22.   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

23.   Majidov, I.; Whangbo, T. Efficient Classification of Motor Imagery Electroencephalography Signals Using Deep Learning Methods. *Sensors* **2019**, *19*, 1736. DOI:10.3390/s19071736.

24. 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.

25. 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.

26. 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.

27. 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.

28. 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.

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. BNCI HORIZON 2020, http://bnci-horizon-2020.eu/community/research-groups, accessed on 4th August 2021.

33. O A Ruşanu et al 2020 IOP Conf. Ser.: Mater. Sci. Eng. 997 012059, DOI: https://doi.org/10.1088/1757-899X/997/1/012059.

34. O A Ruşanu et al 2018 IOP Conf. Ser.: Mater. Sci. Eng. 444 042014, DOI: https://doi.org/10.1088/1757-899X/444/4/042014.

35. O. A. Ruşanu, L. Cristea, M. C. Luculescu, and S. C. Zamfira, "Experimental Model of a Robotic Hand Controlled by Using NeuroSky Mindwave Mobile Headset," 2019 E-Health and Bioengineering Conference (EHB), 2019, pp. 1-4, DOI: 10.1109/EHB47216.2019.8970050.

36. O. A. Ruşanu, L. Cristea and M. C. Luculescu, "Simulation of a BCI System Based on the Control of a Robotic Hand by Using Eye-blinks Strength," 2019 E-Health and Bioengineering Conference (EHB), 2019, pp. 1-4, DOI: 10.1109/EHB47216.2019.8969941.

37. Oana A. Rusanu, Luciana Cristea and Marius C. Luculescu., "The development of a BCI prototype based on the integration between NeuroSky Mindwave Mobile EEG headset, Matlab software environment and Arduino Nano 33 IoT board for controlling the movement of an experimental motorcycle". 11th International Conference on Information Science and Information Literacy, Sciendo, 2021, pp. 290-297, https://doi.org/10.2478/9788395815065-033

38. O. A. Ruşanu, L. Cristea and M. C. Luculescu, "LabVIEW and Android BCI Chat App Controlled By Voluntary Eye-Blinks Using NeuroSky Mindwave Mobile EEG Headset," 2020 International Conference on e-Health and Bioengineering (EHB), 2020, pp. 1-4, DOI: 10.1109/EHB50910.2020.9280193.

39. O A Rusanu, et al 2019 IOP Conf. Ser.: Mater. Sci. Eng. 514 012020, DOI: http://dx.doi.org/10.1088/1757-899X/514/1/012020

40. Izabela Rejer, Łukasz Cieszyński, RVEB—An algorithm for recognizing voluntary eye blinks based on the signal recorded from prefrontal EEG channels, Biomedical Signal Processing and Control, Volume 59, 2020, 101876, ISSN 1746-8094, DOI: https://doi.org/10.1016/j.bspc.2020.101876.

41. Kamal Sharma, Neeraj Jain, Prabir K. Pal, Detection of eye closing/opening from EOG and its application in robotic arm control, Biocybernetics and Biomedical Engineering, Volume 40, Issue 1, 2020, Pages 173-186, ISSN 0208-5216, DOI: https://doi.org/10.1016/j.bbe.2019.10.004.

42. Sebastián Poveda Zavala, Sang Guun Yoo, David Edmigio Valdivieso Tituana, "Controlling a Wheelchair using a Brain Computer Interface based on User Controlled Eye Blinks", International Journal of Advanced Computer Science and Applications (IJACSA), Vol. 12, No. 6, 2021, DOI: https://dx.doi.org/10.14569/IJACSA.2021.0120607.

43. Yadav P., Sehgal M., Sharma P., Kashish K. (2019) Design of Low-Power EEG-Based Brain–Computer Interface. In: Singh S., Wen F., Jain M. (eds) Advances in System Optimization and Control. Lecture Notes in Electrical Engineering, vol 509. Springer, Singapore. DOI: https://doi.org/10.1007/978-981-13-0665-5_19.

44.   Prem S., Wilson J., Varghese S.M., Pradeep M. (2021) BCI Integrated Wheelchair Controlled via Eye Blinks and Brain Waves. In: Pawar P.M., Balasubramaniam R., Ronge B.P., Salunkhe S.B., Vibhute A.S., Melinamath B. (eds) Techno-Societal 2020. Springer, Cham. DOI: https://doi.org/10.1007/978-3-030-69921-5_32.

45.   William C Francis, C. Umayal, G. Kanimozh, "Brain-Computer Interfacing for Wheelchair Control by Detecting Voluntary Eye Blinks", Indonesian Journal of Electrical Engineering and Informatics (IJEEI), Vol. 9, No. 2, June 2021, pp. 521~537, ISSN: 2089-3272, DOI: http://dx.doi.org/10.52549/ijeei.v9i2.2749.

46.   P. K. Tiwari, A. Choudhary, S. Gupta, J. Dhar, and P. Chanak, "Sensitive Brain-Computer Interface to help manoeuvre a Miniature Wheelchair using Electroencephalography," 2020 IEEE International Students' Conference on Electrical, Electronics and Computer Science (SCEECS), 2020, pp. 1-6, DOI: https://doi.org/10.1109/SCEECS48394.2020.73.

47.   Rihana S., Damien P., Moujaess T. (2013) EEG-Eye Blink Detection System for Brain Computer Interface. In: Pons J., Torricelli D., Pajaro M. (eds) Converging Clinical and Engineering Research on Neurorehabilitation. Biosystems & Biorobotics, vol 1. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-34546-3_98.

48.   Chambayil, B., Singla, R., Jha, R.: EEG Eye Blink Classification using neural network. In: Proceedings of the World Congress on Engineering (2010).

49.   M. Lo Giudice et al., "1D Convolutional Neural Network approach to classify voluntary eye blinks in EEG signals for BCI applications," 2020 International Joint Conference on Neural Networks (IJCNN), 2020, pp. 1-7, DOI: 10.1109/IJCNN48605.2020.9207195.

50.   Saragih, A. S., Pamungkas, A., Zain, B. Y., and Ahmed, W., "Electroencephalogram (EEG) Signal Classification Using Artificial Neural Network to Control Electric Artificial Hand Movement", in *Materials Science and Engineering Conference Series*, 2020, vol. 938, no. 1, p. 012005. DOI: 10.1088/1757-899X/938/1/012005.

51.   Miranda, M., Salinas, R., Raff, U., & Magna, O. (2019). Wavelet Design for Automatic Real-Time Eye Blink Detection and Recognition in EEG Signals. Int. J. Comput. Commun. Control, 14, 375-387, DOI: https://doi.org/10.15837/ijccc.2019.3.3516.

52.   M. Agarwal and R. Sivakumar, "Blink: A Fully Automated Unsupervised Algorithm for Eye-Blink Detection in EEG Signals," 2019 57th Annual Allerton Conference on Communication, Control, and Computing (Allerton), 2019, pp. 1113-1121, DOI: 10.1109/ALLERTON.2019.8919795.

53.   G. Schalk, D. J. McFarland, T. Hinterberger, N. Birbaumer, and J. R. Wolpaw, "BCI2000: a general-purpose brain-computer interface (BCI) system," in IEEE Transactions on Biomedical Engineering, vol. 51, no. 6, pp. 1034-1043, June 2004, DOI: 10.1109/TBME.2004.827072.

54.   Yann Renard, Fabien Lotte, Guillaume Gibert, Marco Congedo, Emmanuel Maby, Vincent Delannoy, Olivier Bertrand, Anatole Lécuyer; OpenViBE: An Open-Source Software Platform to Design, Test, and Use Brain–Computer Interfaces in Real and Virtual Environments. Presence: Teleoperators and Virtual Environments 2010; 19 (1): 35–53. DOI: https://doi.org/10.1162/pres.19.1.35.

55.   Arnaud Delorme, Scott Makeig, EEGLAB: an open source toolbox for analysis of single-trial EEG dynamics including independent component analysis, Journal of Neuroscience Methods, Volume 134, Issue 1, 2004, Pages 9-21, ISSN 0165-0270, DOI: https://doi.org/10.1016/j.jneumeth.2003.10.009.

56.   National Instruments. What is LabVIEW? Available Online: https://www.ni.com/ro-ro/shop/labview.html (accessed on 15 September 2019).

57.   NeuroSky Mindwave Mobile 2. Available Online: https://store.neurosky.com/pages/mindwave (accessed on 15 September 2019).

58.   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.

59.   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.

60.   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).

61.   Rashid M. et al. (2020) Analysis of EEG Features for Brain Computer Interface Application. In: Kasruddin Nasir A.N. et al. (eds) InECCE2019. Lecture Notes in Electrical Engineering, vol 632. Springer, Singapore, DOI: https://doi.org/10.1007/978-981-15-2317-5_45