

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

<https://doi.org/10.3991/ijoe.v19i05.37857>

Oana Andreea Ruşanu^(✉)

Product Design, Mechatronics and Environment Department, Transilvania University of
Brasov, Brasov, Romania
oana.rusan@unitbv.ro

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, root 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

1 Introduction

Brain-Computer Interface is a multidisciplinary research field, which comprises achievements in related scientific and technical areas: artificial intelligence, computer science, mechatronics, signal processing, neuroscience, and psychology. 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.

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 biosensor 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 proposed LabVIEW based solution consists in flexible, robust, simple to use, user-friendly and cost-effective virtual instruments aimed for BCI projects that can be also conducted by young researchers. Moreover, the proposed BCI versatile and customizable framework provides universal usefulness and general purpose by attracting researchers from non-technical fields. The previously developed BCI universal platforms – BCI2000, OpenViBE and EEGLAB – are not anymore up-dated to the newest versions of portable EEG headsets or require advanced programming experience. 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 [1–13], eye-blinks were considered precise control signals in a brain-computer interface application. Likewise, this paper provides a large number of EEG datasets including multiple mixtures between EEG rhythms and features necessary for classifying simple, double and triple eye-blinks. Across scientific literature, there is low evidence of papers delivering EEG datasets for classifying the multiple voluntary eye-blinks used as commands for BCI systems.

Regarding the structure of the current paper, Section 2 provides some detailed insights about the experimental software environment by developing 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 Experimental software environment

2.1 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.

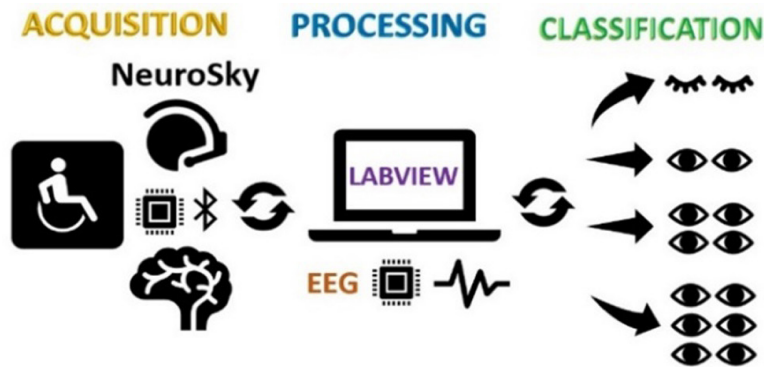


Fig. 1. An overview of the proposed LabVIEW application aimed for the acquisition, processing, and classification of the EEG signal used in a brain-computer interface

The proposed LabVIEW application is consisting of a State Machine paradigm accomplishing the following functionalities (see 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 models either by setting specific hyperparameters 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 other EEG dataset that can be delivered by using a similar procedure as previously described regarding the proposed mixtures between EEG signals and statistical features.

2.2 The EEG signal acquisition and processing

Figure 2 shows the proposed experimental paradigm or the algorithm underlying the automatic mode of EEG data acquisition.

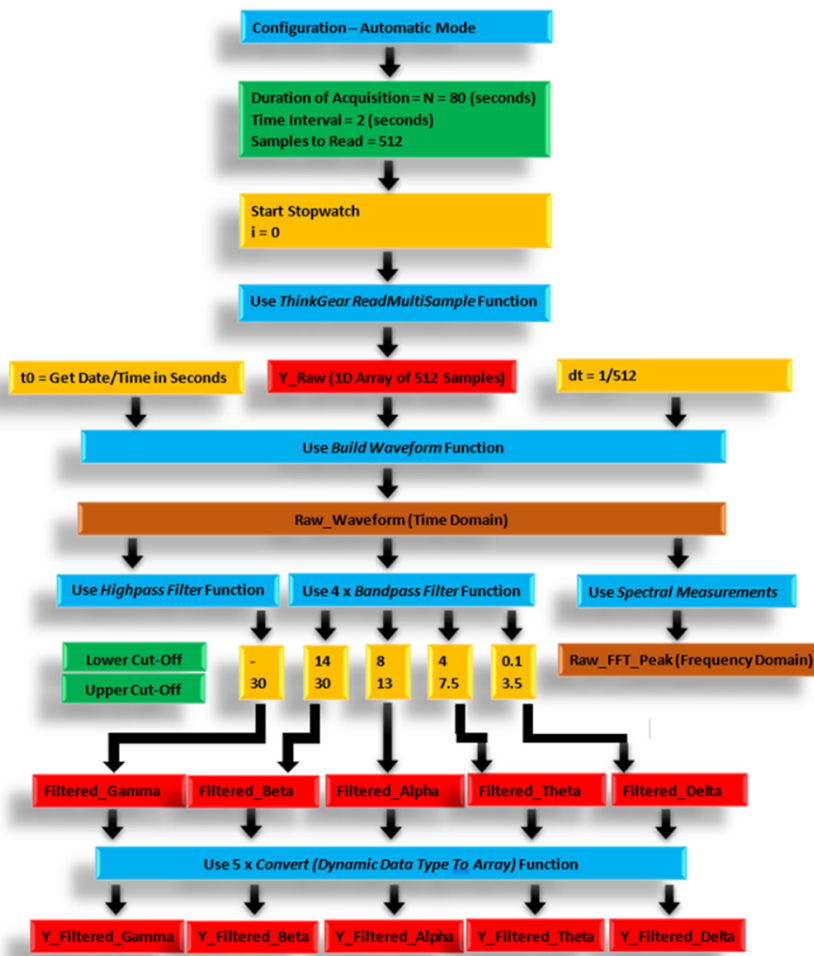


Fig. 2. A diagram representing the block instructions underlying the implementation of Data Acquisition in the Automatic Mode – first view showing all the steps leading to obtaining the raw EEG signal (time and frequency domain) and extracting the EEG rhythms (gamma, beta, alpha, theta, and delta)

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).

Accordingly, when the chronometer stops, indicating the finish of the EEG signal acquisition in automatic mode, $12 \times 2D$ arrays will be returned. They consist of six types of EEG signals in the Time Domain (see Figure 3) and six types of EEG signals in the Frequency Domain (FFT – Peak – see Figure 4): raw, delta, theta, alpha, beta, and gamma. A 2D array is a matrix containing 80 rows (temporal sequences) and 512 columns (512 samples).

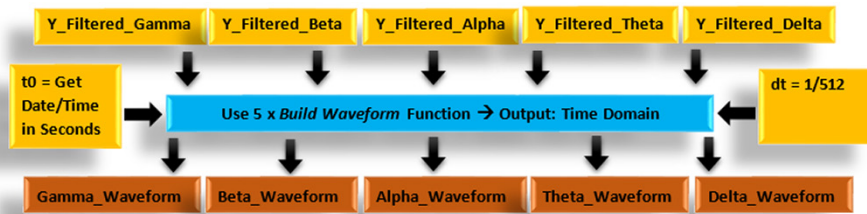


Fig. 3. A diagram representing the block instructions underlying the implementation of EEG Data Acquisition in the Automatic Mode – the second view showing the five EEG rhythms obtained in the time domain

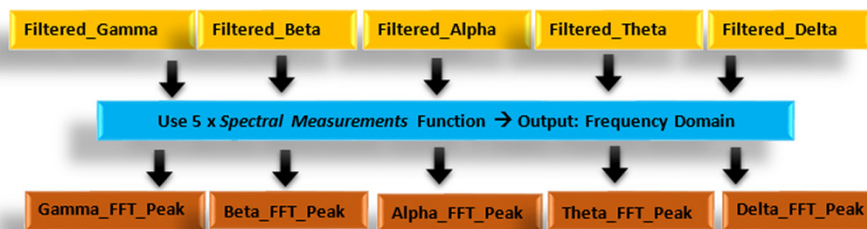


Fig. 4. A diagram representing the block instructions underlying the implementation of EEG Data Acquisition in the Automatic Mode – the third view showing the five EEG rhythms obtained in the frequency domain

2.3 The preparation of the EEG temporal sequences

Before applying the algorithm aimed at preparation the EEG acquired data, every one of $12 \times 2D$ arrays contains: N rows and ‘Samples to Read’ Columns = 80 rows and 512 columns \rightarrow 80 temporal sequences of 512 elements. Table 1 shows every one of the $12 \times 2D$ arrays (both time and frequency domain of raw, gamma, beta, alpha, theta, delta) before the preparation of the EEG data.

After applying the algorithm of preparation of the EEG acquired data, every one of $12 \times 2D$ arrays contains: ‘N divided by Time Interval’ rows and ‘Time Interval multiplied by Samples to Read’ Columns = 40 rows and 2×512 columns = 40 rows and 1024 columns \rightarrow 40 sequences of 1024 elements. Table 2 shows every one of the $12 \times 2D$ 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.

Table 1. Structure of the 2D arrays (time and frequency domain of all EEG signals) before the preparation of the EEG acquired data

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

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 $12 \times 2D$ 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 $12 \times 2D$ arrays into $12 \times 3D$ 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 $12 \times 3D$ arrays into $12 \times 2D$ 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 2. Structure of the 2D arrays (time and frequency domain of all EEG signals) after applying the preparation of the EEG acquired data

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

2.4 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 (see Figure 5), the user can manually set the label for each EEG temporal sequence by visually checking the graphical display in time and frequency domains. Figure 5 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).

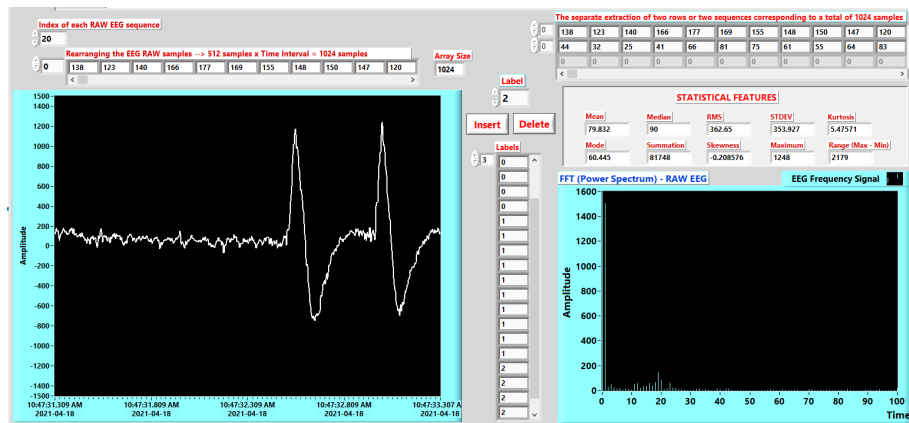


Fig. 5. A sequence of the front panel – EEG raw signal – showing various options allowing the label assignment for each EEG temporal sequence by visually checking the graphical display in time and frequency domains

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 6, 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).

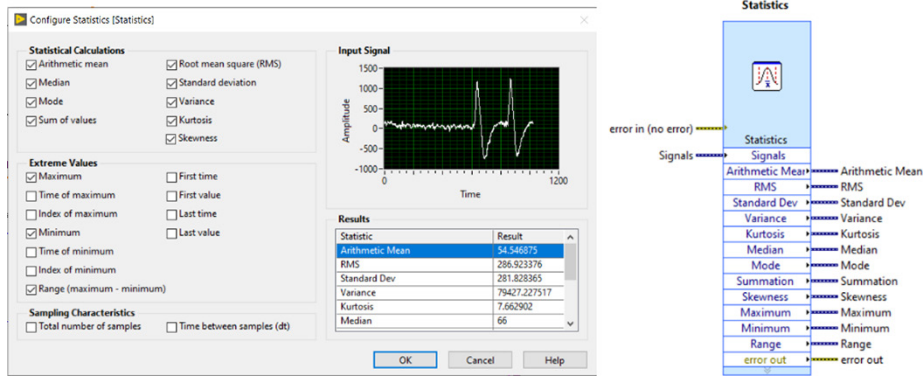


Fig. 6. The settings corresponding to 'Statistics Express VI' included by the 'Signal Analysis Express VIs' LabVIEW palette

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 7.

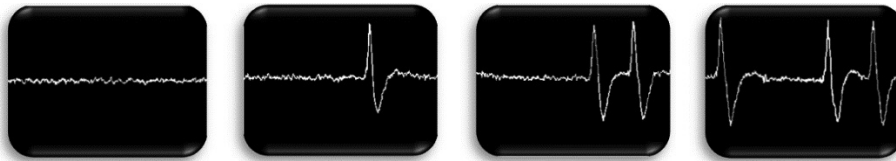


Fig. 7. An example for the LabVIEW display of the EEG temporal sequences associated with each label: 0 – no eye-blink; 1 – one eye-blink; 2 – two eye-blinks and 3 – three eye-blinks

2.5 The generation of training and testing EEG dataset

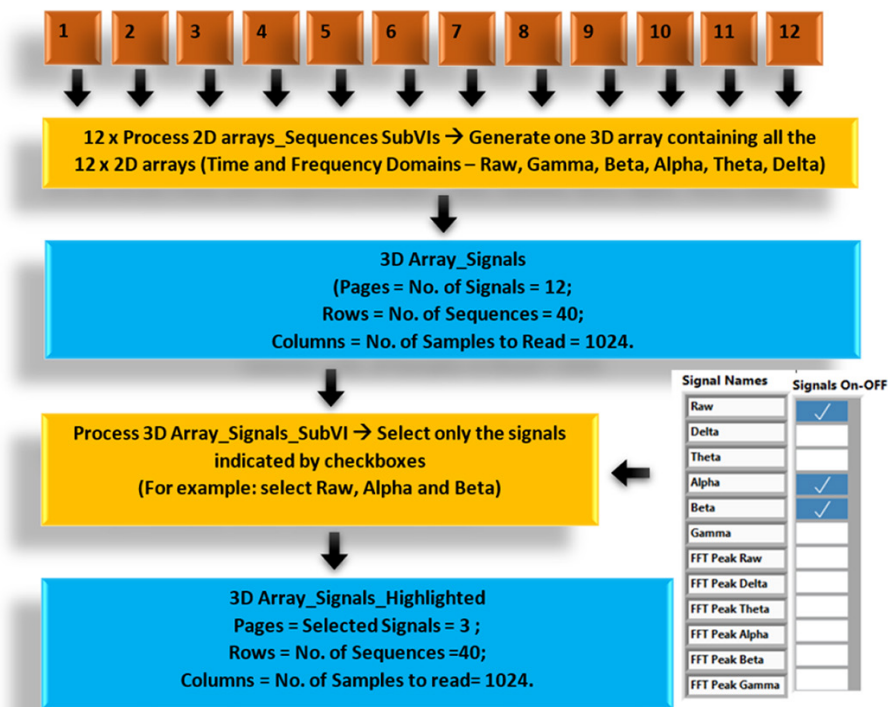
Multiple mixtures between the selected EEG signals and the extracted features are necessary to generate the training and testing EEG dataset. The boxes from the top side of Figure 8 represent $12 \times 2D$ arrays, each of them containing 40 rows and 1024 columns or 40960 samples. Also, these boxes represent:

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

Figures 8–10 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 $12 \times 2D_arrays$ 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

Fig. 8. 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.

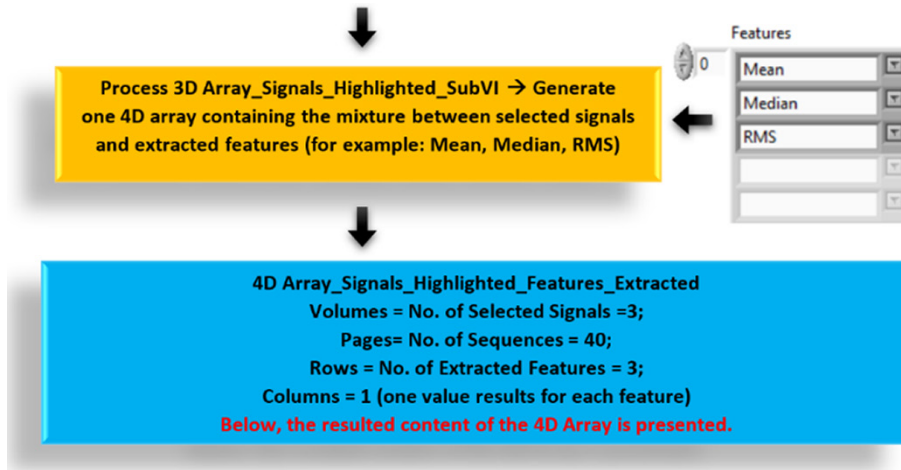


Fig. 9. The second view of the diagram representing the block instructions underlying the generation of training or testing dataset based on multiple mixtures between the selected EEG signals and the extracted features

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)

Fig. 10. 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

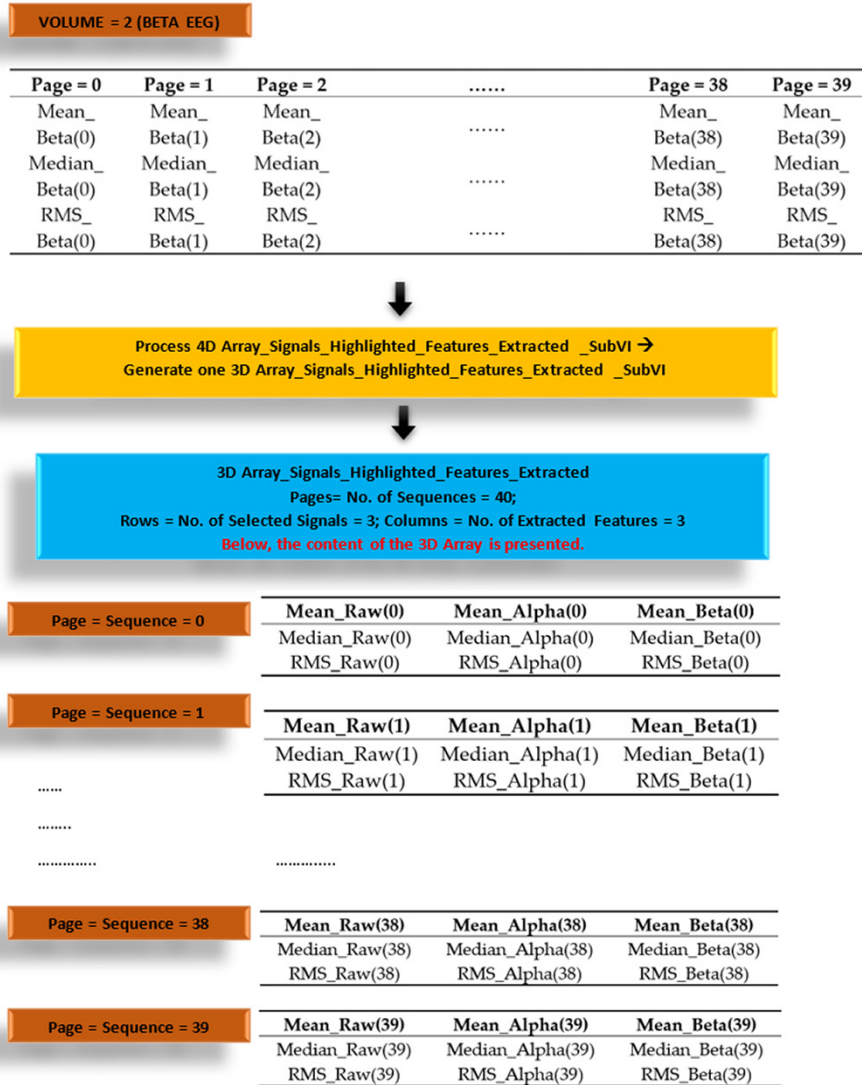


Fig. 11. Fourth view of the diagram representing the block instructions underlying the generation of training or testing dataset based on multiple mixtures between the selected EEG signals and the extracted features

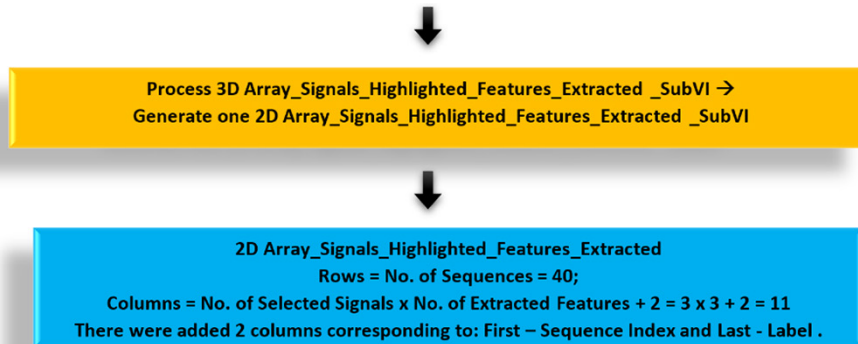


Fig. 12. Fifth view of the diagram representing the block instructions underlying the generation of training or testing dataset based on multiple mixtures between the selected EEG signals and the extracted features

The fourth subVI, called ‘Process 4D Array_Signals_Highlighted_Features_Extracted’ (see Figure 11), 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_Highlighted_Features_Extracted’, was implemented as a final stage of re-organization and reducing the dimensionality of the obtained data by the generation of a 2D array that is consisting of 40 rows corresponding to 40 temporal sequences, a certain number of columns involving the number of selected signals and the number of the extracted features. The generated 2D array included numerical elements that were converted into String type. Then, a .csv file representing the training/testing dataset stores the 2D array_Signals_Highlighted_Features_Extracted (see Figure 12) 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.

2.6 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 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 NN 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 maximum iteration.

According to the AML LabVIEW toolkit, the ‘hidden layer type’ is related to the activation function applied to the neurons from the hidden layer type. Table 3 defines the available activation functions. According to Table 4, Sigmoid and Softmax are the two activation functions available in the neurons regarding the ‘output layer type.’ Table 5 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 3. Information about the available activation functions determining the hidden layer type set as a hyper-parameter included by the ‘Analytics and Machine Learning’ toolkit

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

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

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

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

Function Type	Definition	Description
Quadratic	$error = \frac{1}{l}(t - y)^2$	t – the target value, y – output value, l – number of training samples
Cross-entropy	$f(x) = -\frac{1}{l} \left(\sum_{j=1}^n t_j \ln y_j \right)$	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 13 shows the entire structure of the previously described AML functions used to enable the setting of hyperparameters for training the neural networks model. Figure 14 shows the graphical user interface of this LabVIEW programming sequence.

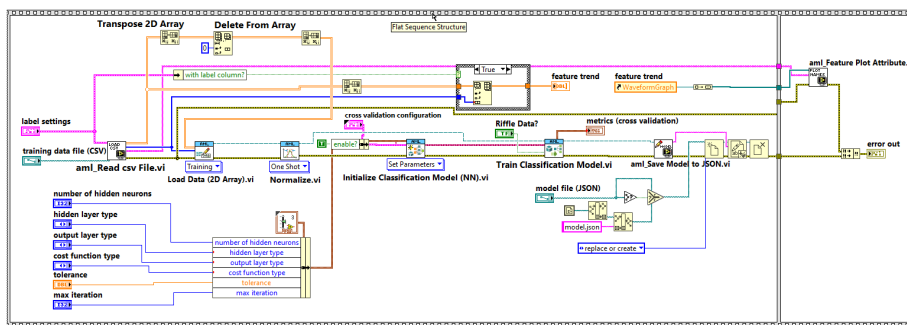


Fig. 13. A sequence of the block diagram displaying the corresponding LabVIEW programming functions used to enable the setting of hyperparameters for training the neural networks model

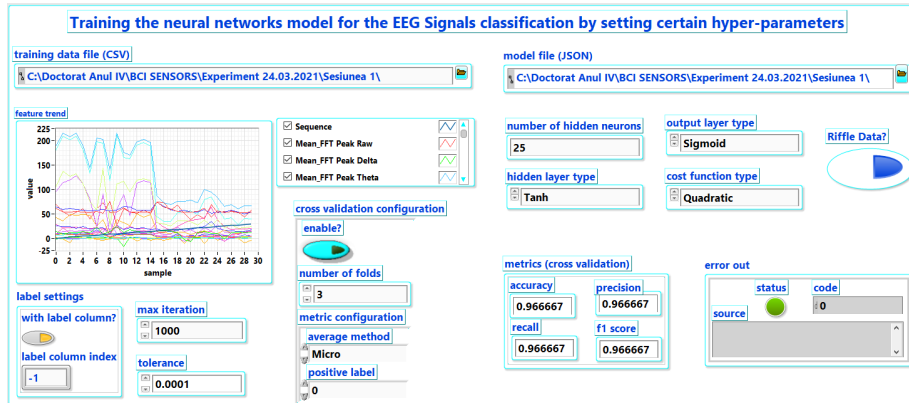


Fig. 14. A sequence of the front panel showing the graphical window corresponding to training the neural networks model for the EEG signal classification by setting certain hyper-parameters

2.7 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 15, the user should specify multiple values for each hyper-parameter so that the ‘Train Classification Model. vi’ could use a grid search to find the optimal set of parameters. This technique 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 15 displays the number of correctly/incorrectly detected samples/temporal sequences, calculated by taking into account the mathematical formulas for the metrics described in Table 6.

The current research analyzed 50 generated artificial neural networks-based models, and each of them needed a training time interval between 1 and 3 hours.

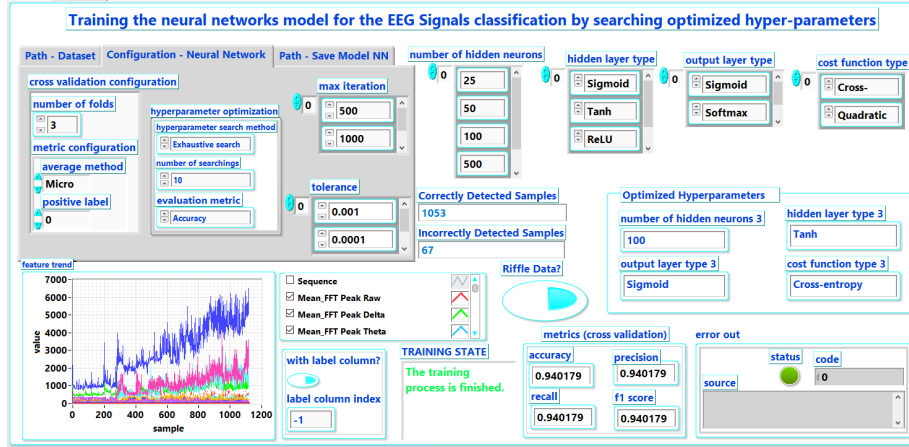


Fig. 15. A sequence of the front panel showing the graphical window corresponding to training the NN model for the EEG signal classification by searching the optimized hyper-parameters

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

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

2.8 The deployment/testing of the trained NN 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 $25 \times 40 = 1000$ 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 4×25 sessions of EEG data acquisition, enabling the recording of the $4 \times 25 \times 40 = 4000$ EEG temporal sequences corresponding to the previously mentioned four classes.

Otherwise, for the generation of the testing dataset, the subject has been also involved in 4×5 sessions of EEG data acquisition, enabling the recording of the $4 \times 5 \times 40 = 800$ EEG temporal sequences corresponding to the previously mentioned four classes. The duration of each session of EEG data acquisition was 80 seconds. Figure 16 shows the previously described general structure. Both training and testing datasets include a column that is assigned the labels.

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 could replace 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.

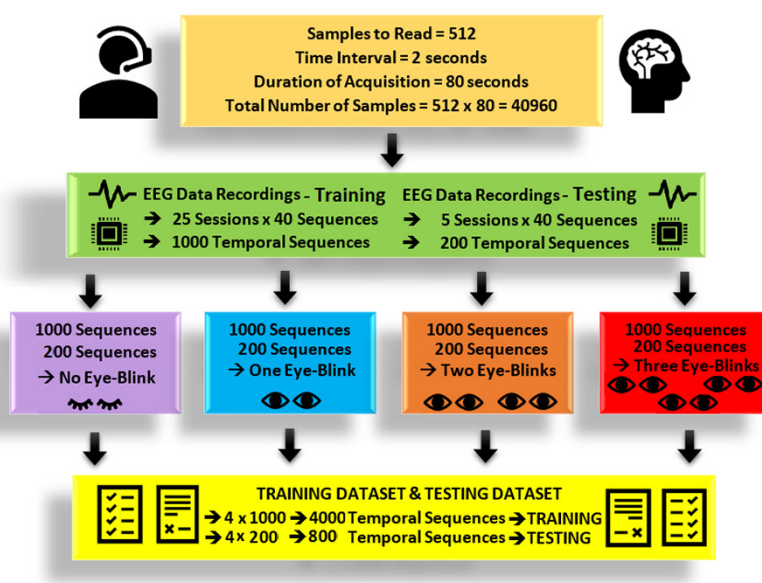


Fig. 16. An overview of the proposed paradigm for performing the experimental sessions to get the datasets used for both training and testing of the neural networks model of voluntary multiple eye-blinks classification

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, re-call, and F1 score.

There were generated 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.

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.

The first highest accuracy of a neural networks-based model is 0.99 for uploading a training dataset comprising 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.

The second-highest accuracy of the neural networks-based models is 0.98 for uploading the 16 training datasets composed of the following groups of selected EEG rhythms:

- all 12 signals in time and frequency domains;
- only the six signals in the time domain;
- raw;
- raw, delta, theta;
- raw, alpha, beta, gamma;
- raw, delta, theta, gamma;
- raw, beta, gamma;
- raw, delta, theta, alpha.

These training datasets are composed of the following groups of extracted statistical features:

- all the ten statistical features;
- mean, median, RMS, standard deviation, Kurtosis Coefficient;
- median, RMS, standard deviation, Kurtosis Coefficient;
- 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: 5; 10; 20; 50; 200; 500;
- Hidden layer type: ReLU; Sigmoid; Tanh;
- Output layer type: Sigmoid; Softmax;
- Cost function: = Cross-entropy for all the NN models/training datasets.
- Number of input neurons is between 10 and 120, as it follows: 10; 12; 15; 16; 20; 21; 24; 28; 40; 42; 60; 120;
- Average method = Weighted for all the NN models/training datasets.

In addition, regarding the NN models that reported a training accuracy of 0.98, from Total = 4000 samples, the pairs of correctly – incorrectly detected samples are as follows: 3912 – 88; 3916 – 84; 3924 – 76; 3925 – 75; 3926 – 74; 3927 – 73; 3929 – 71; 3930 – 70; 3933 – 67; 3935 – 65; 3937 – 63;

The third-highest accuracy of the neural networks-based models is 0.95 for uploading three training datasets composed of the following groups of selected EEG rhythms:

- only the six signals in the frequency domain;
- Raw, Delta, Theta in the frequency domain.

These training datasets are composed of the following groups of extracted statistical features:

- all the ten statistical features;
- 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

Hidden layer type: ReLU; Tanh;

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

In addition, regarding the NN models that reported a training accuracy of 0.95, from a Total = 4000 samples, the pairs of correctly/incorrectly detected samples are as follows: 3803 – 197; 3799 – 201; 3790 – 210.

The first highest accuracy of a neural networks-based model is 0.97 for uploading four testing datasets. The groups of selected EEG rhythms and the extracted statistical features composing these testing datasets were included above to the equivalent 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 pairs of correctly – incorrectly detected samples are as follows: 778 – 22; 775 – 25; 776 – 24; 773 – 27.

It results that the second-highest accuracy of a neural networks-based model is 0.96 for uploading five testing datasets. The groups of selected EEG rhythms and the extracted statistical features composing these testing datasets were included above to the equivalent 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 pairs of correctly – incorrectly detected samples are as follows: 770 – 30; 768 – 32; 768 – 32; 766 – 34; 765 – 35.

It results that the third-highest accuracy of a neural networks-based model is 0.95 for uploading six testing datasets. The groups of selected EEG rhythms and the groups of extracted statistical features composing these testing datasets were included above to the equivalent 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 pairs of correctly – incorrectly detected samples are as follows: 763 – 37; 759 – 41; 761 – 39; 763 – 37; 761 – 39; 762 – 38.

Considering 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.

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. Then, considering 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 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: [8], [9], and [10]. According to the paper [8], the extracted features were: maximum amplitude, minimum amplitude, and the Kurtosis coefficient. According to the paper [9], the extracted features were: Kurtosis coefficient, maximum amplitude, and minimum amplitude. According to the paper [10], 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 [14]. 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 [14]. 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 [8], [9], and [10] focuses on the experimentation of a single method of features extraction and a specific classification technique that is considered optimal.

The paper [8] 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 [9] 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 [10], the authors designed an artificial multi-layer neural network with backpropagation, considering 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 [8], [9], and [10]. 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 [8], [9] and [10] presented experimental activities that involved the following EEG expensive devices: a wireless biomedical monitor, called BioRadio with four channels [8]; RMS EEG 32 Super System with two Ag-AgCl electrodes [9] and OpenBCI Cython with eight channels [10].

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 or even manage hybrid BCI systems [15–17]. 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, it is still necessary to implement a customized communication protocol

enabling the universal 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 providing a standalone, simple to use virtual instrument with a user-friendly graphical interface accomplishing all the necessary fundamental functions: EEG data acquisition, processing, features extraction, and classification.

5 Conclusions

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, root 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.

6 Acknowledgements

A quick original video demonstration of the working principle of the proposed LabVIEW applications is uploaded to this YouTube unlisted link: <https://youtu.be/bmr04-QKJOG>. All the experimental data and the LabVIEW source code are available by request. This research received no external funding. The author is forever grateful for all the valuable ideas offered by a favorite professor.

7 References

- [1] 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*, Vol. 59, 2020, 101876, ISSN 1746-8094, <https://doi.org/10.1016/j.bspc.2020.101876>
- [2] 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*, Vol. 40, Issue 1, 2020, pp. 173–186, ISSN 0208-5216, <https://doi.org/10.1016/j.bbe.2019.10.004>
- [3] 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, Issue 6, 2021, <https://doi.org/10.14569/IJACSA.2021.0120607>
- [4] Yadav P., Sehgal M., Sharma P., and 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. https://doi.org/10.1007/978-981-13-0665-5_19
- [5] Prem S., Wilson J., Varghese S.M., and 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. https://doi.org/10.1007/978-3-030-69921-5_32
- [6] 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 (IJEI)*, Vol. 9, Issue 2, 2021, pp. 521–537, ISSN: 2089-3272, <https://doi.org/10.52549/ijeeci.v9i2.2749>
- [7] Tiwari P. K., Choudhary A., Gupta S., Dhar J., and Chanak P., “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, <https://doi.org/10.1109/SCEECS48394.2020.73>
- [8] Rihana S., Damien P., and 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

- [9] Chambayil, B., Singla, R., and Jha, R., EEG Eye Blink Classification using Neural Network. In: Proceedings of the World Congress on Engineering (2010).
- [10] Lo Giudice M. 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, <https://doi.org/10.1109/IJCNN48605.2020.9207195>
- [11] 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, Vol. 938, Issue 1, 2020, p. 012005, <https://doi.org/10.1088/1757-899X/938/1/012005>
- [12] Miranda, M., Salinas, R., Raff, U., and Magna, O., Wavelet Design for Automatic Real-Time Eye Blink Detection and Recognition in EEG Signals. *Int. J. Comput. Commun. Control*, Vol. 14, 2019, 375–387, <https://doi.org/10.15837/ijccc.2019.3.3516>
- [13] 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, <https://doi.org/10.1109/ALLERTON.2019.8919795>
- [14] 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, https://doi.org/10.1007/978-981-15-2317-5_45
- [15] Aljobouri, H.K., A Virtual EMG Signal Control and Analysis for Optimal Hardware Design. *International Journal of Online and Biomedical Engineering (iJOE)*, Vol. 18, Issue 02, 2022, pp. 154–166. <https://doi.org/10.3991/ijoe.v18i02.27047>
- [16] Osayande, E., Ayodele, K.P., and Komolafe, M.A., Development of a Robotic Hand Orthosis for Stroke Patient Rehabilitation. *International Journal of Online and Biomedical Engineering (iJOE)*, Vol. 16, Issue 13, 2020, pp. 142–149, <https://doi.org/10.3991/ijoe.v16i13.13407>
- [17] Najafi, T., Jaafar, R., Najafi, K., and Eslamdoust-Siahestalkhi, F., Brain Waves Characteristics in Individuals with Obsessive-Compulsive Disorder: A Preliminary Study. *International Journal of Online and Biomedical Engineering (iJOE)*, Vol. 18, Issue 01, 2022, pp. 96–110, <https://doi.org/10.3991/ijoe.v18i01.26805>

8 Author

Oana Andreea Ruşanu is a laboratory classes instructor at Faculty of Electrical Engineering and Computer Science from Transilvania University of Brasov, Romania. She also works as a Software Test Engineer at Preh Company – Brasov, Romania. In 2015, she graduated as valedictorian the Faculty of Product Design and Environment, Brasov with a Bachelor of Medical Engineering. In 2017, she completed her Master Degree in Mechatronics System for Medicine and Industry at the Faculty of Product Design and Environment. She is currently pursuing a Doctorate in Mechanical Engineering with the main research domain focused on Brain-Computer Interfaces. Being also passionate about digital technology, robotics, virtual reality, software development and computer science, she presented and published approximately 20 scientific papers to the International Conferences.

Article submitted 2022-12-03. Resubmitted 2023-01-25. Final acceptance 2023-01-25. Final version published as submitted by the authors.