



Review

# Education 4.0: Teaching the Basis of Motor Imagery Classification Algorithms for Brain-Computer Interfaces

David Balderas \* , Pedro Ponce , Diego Lopez-Bernal and Arturo Molina

Tecnologico de Monterrey National Department of Research, Puente 222, Del. Tlalpan, Mexico City 14380, Mexico; pedro.ponce@tec.mx (P.P.); lopezbernal.d@tec.mx (D.L.-B.); armolina@tec.mx (A.M.)  
\* Correspondence: dc.balderassilva@tec.mx

**Abstract:** Education 4.0 is looking to prepare future scientists and engineers not only by granting them with knowledge and skills but also by giving them the ability to apply them to solve real life problems through the implementation of disruptive technologies. As a consequence, there is a growing demand for educational material that introduces science and engineering students to technologies, such as Artificial Intelligence (AI) and Brain-Computer Interfaces (BCI). Thus, our contribution towards the development of this material is to create a test bench for BCI given the basis and analysis on how they can be discriminated against. This is shown using different AI methods: Fisher Linear Discriminant Analysis (LDA), Support Vector Machines (SVM), Artificial Neural Networks (ANN), Restricted Boltzmann Machines (RBM) and Self-Organizing Maps (SOM), allowing students to see how input changes alter their performance. These tests were done against a two-class Motor Image database. First, using a large frequency band and no filtering eye movement. Secondly, the band was reduced and the eye movement was filtered. The accuracy was analyzed obtaining values around 70~80% for all methods, excluding SVM and SOM mapping. Accuracy and mapping differentiability increased for some subjects for the second scenario 70~85%, meaning either their band with the most significant information is on that limited space or the contamination because of eye movement was better mitigated by the regression method. This can be translated to saying that these methods work better under limited spaces. The outcome of this work is useful to show future scientists and engineers how BCI experiments are conducted while teaching them the basics of some AI techniques that can be used in this and other several experiments that can be carried on the framework of Education 4.0.



**Citation:** Balderas, D.; Ponce, P.; Lopez-Bernal, D.; Molina, A. Education 4.0: Teaching the Basis of Motor Imagery Classification Algorithms for Brain-Computer Interfaces. *Future Internet* **2021**, *13*, 202. <https://doi.org/10.3390/fi13080202>

Academic Editor: Patrizia Marti

Received: 17 June 2021

Accepted: 28 July 2021

Published: 3 August 2021

**Publisher's Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Copyright:** © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

**Keywords:** Education 4.0; artificial intelligence; classification methods; signal processing; brain computer interfaces; testbench

## 1. Introduction

Nowadays, new technologies are evolving at an exponential pace, and the consequential technological advancements achieved through them are blurring the lines between the physical, digital and biological worlds [1]. These advancements constitute the basis of the fourth industrial revolution (also called Industry 4.0), which is principally constituted of progress in the areas of artificial intelligence (AI), robotics, nanotechnology, quantum computing, energy storage systems and the internet of the things (IoT) [2]. As Industry 4.0 continues changing the world, new challenges arise in different branches of society, one of them being education; thus, Education 4.0 comes into existence.

In general, science and engineering education needs, learning and teaching methods are continuously and rapidly changing in order to adapt to the incoming innovation challenges caused by the digital transformation of industries. Therefore, one of the main objectives of Education 4.0 is to generate updated curricula at the undergraduate level that allows students to develop technological progress and knowledge that, in a future, can be usable for the welfare of society.

As stated by [3], one of the main pillars for the creation of this new curricula is competency development. This approach will allow students to apply the acquired knowledge towards real-life situations rather than only memorizing and repeating data. In the context of Education 4.0, competencies are considered to be sets of skills, attributes and behaviors that allow the successful realization of a specific task [4]. According to [3], in the case of scientific and engineering education, new curricula must focus on driving the development of the following competencies: virtual collaboration, resilience, social intelligence, novel and adaptive thinking, load cognition management, sense-making, new media literacy, design mindset, transdisciplinary approach and computational skills. Nevertheless, in an effort to make the most out of these competencies, students must also learn how they can be used to acquire a deeper knowledge on disruptive technologies.

Among these technologies, there is the area of Human–Computer Interaction (HCI), which studies the interaction between the human body as the control and a computer as the acting device [5,6]. HCI can use either physiological movements (i.e., body motion detection [7,8], eye tracking [9,10], tongue movement, etc.), spoken word recognition [11,12] and/or electrical signals produced by the human body (i.e., muscular or brain activation [13–15]) as the control signals. HCI has been used for augmented reality [16–18], control of exoskeletons [19], rehabilitation robots [15,20], spelling devices [21], video games or daily devices [22–24] and many others [16,25,26].

Particularly, the use of brain signals as control patterns is known as Brain–Computer Interfaces (BCI). These types of interfaces have the advantage of not requiring any type of movement; therefore, they can be used by people with motion impairment, where, in many cases, the brain activity remains intact (i.e., people with Lock-in-Syndrome [27,28]). Although there has been great progress in the BCI field, it is still necessary to further improve to make them work better for different subjects and on a daily basis [14,28]. As a consequence, many experiments and new researchers are still needed; thus, the importance of applying the Education 4.0 paradigm towards science and engineering students.

Kuhn [29] said that the development of science consists of past research on scientific achievements that have plenty of support from the scientific community to create new and improved models or paradigms for further research. Hence, it is necessary to fully understand what was done and how it works before developing new technology. This becomes problematic since BCI is a multidisciplinary field, which requires knowledge from many areas, such as engineering (e.g., statistics, signal analysis and control theory), computer science (e.g., machine learning and software development), medicine (e.g., physiology, anatomy, neuroscience and psychology) and many others to create expertise. Otherwise, it would require a lot of time to identify problems and develop solutions.

Knowing all the theories is not enough, testing is also required before any real-life application of BCI can be fully developed. Specifically, medical experimentation requires careful processes to prove a hypothesis, especially since it is going to be used by humans and it has a direct connection with technology. Therefore, in medical experimentation, three main things need to be considered: first, the use of medical equipment is limited due to it being complex to use, expensive or with a busy schedule; second, faulty experimentation could be harmful or tiresome for the subjects; and third, medical experimentation requires many repetitions to guarantee that it works and will not damage the patient. Thus, experimentation needs to be carefully performed; otherwise, a bad design of the experiment translates into a waste of time and/or money or inflicted damage to the patient.

All of these problems can be solved by becoming an expert and applying the basics. Under normal circumstances, becoming an expert in all these areas and knowing all these tasks is complex and very time-consuming because of the transdisciplinary nature of BCI. For that reason, it is important for students to have a starting point in this field where experimentation and analysis can be performed without harming any subject while allowing them to develop competencies such as adaptive thinking, sense-making, design mindset, transdisciplinary approach and computational skills. Furthermore, the involvement of applied knowledge, logical interpretation, adoption of digital tools and construction of real

learning scenarios through practical projects are some of the pillars that constitute the basis of Education 4.0 [30]. Hence, a way to introduce this paradigm on future engineers and scientists is to use practical approaches that help them acquire new skills, learn how theory and practice are linked, understand how to correctly structure and test hypotheses, know how to develop problem-solving techniques or simply to understand how to work with new equipment and to gather, manipulate and/or interpret data [31].

One way to achieve this is to provide students the option to learn over a testing bench or workbench, in which they can try out different protocols and verify their correctness without trying them on a human. This would help them acquire knowledge about the development of experiments, as well as how to manipulate data and understand results. It is very important to remark that the learning process over a workbench must be carried out over a similar context to the real subject to learn [32] and that it must be done using technologies that are similar to the ones that would be used for a real-life application [33].

To develop this workbench, it is important to understand how science and technology are usually taught. In general, the aim of education in science and technology is to inform people who live in a world with high dependency on technology. It is important to notice that science cannot be taught disjointed from the world because of the many relationships between science and society, especially through the countless applications of science and technology [34,35]. Thus, it is of high importance for future scientists and engineers to learn science and technology based on their own experience and their knowledge about the world and their surroundings [36]. This translates into learning through practical approaches over things that are related to them as individuals.

Having said that, the main objective of this study is to serve as educational material for science and engineering students and teachers that are dabbling in the Education 4.0 paradigm. Ultimately, this will help students to acquire expertise on a disruptive and transdisciplinary technology, such as BCI, while developing computational skills, adaptive and sense-making thinking. In order to achieve this objective, this work first explains the basics behind Brain–Computer Interfaces and five different artificial intelligence algorithms: Kohonen Self-Organizing Maps (SOM), Artificial Neural Networks (ANN), Linear Discriminant Analysis (LDA), Supported Vector Machines (SVM) and Restricted Boltzmann Machines (RBM). Furthermore, for this work to be fully in line with the Education 4.0 paradigm, we present a test bench for students to learn the applicability of the previous algorithms towards BCI and how to interpret the outcomes of the given experimentation. The proposed test bench in this work consists of a two-class Motor Image database obtained by [37], which includes three different bipolar EEG recordings and three monopolar EOG recordings.

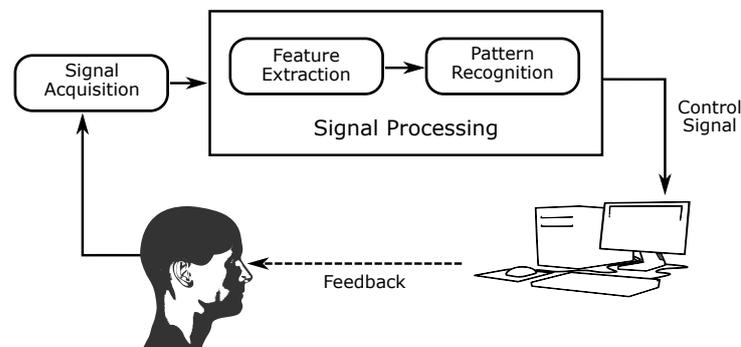
This article goes first through a review of Brain–Computer Interfaces with BCI control paradigms, signal processing, including signal acquisition and feature extraction, and Pattern Recognition methods. In the latter, an introduction to AI techniques is given, exploring two linear and two neural network classifiers and one more neural network that creates an internal representation of the signal. Furthermore, a bibliographic comparison is conducted to cover their corresponding advantages and disadvantages. Afterward, these algorithms are tested over a BCI database to show and compare their potential and performance.

## 2. Brain–Computer Interfaces

Among the main objectives of technological progress of Industry 4.0 is the intention of searching and implementing new ways of communication, interaction and remote control of devices. Thus, including BCI-AI teaching into the Education 4.0 paradigm is one way to introduce students to this type of technologies.

In general, BCI can be decomposed into four steps. The first one is signal acquisition, which requires an understanding of the intrinsic properties of the signals, what specific signals are to be recorded, where are they going to be captured and the sensors to be used (easy or hard attach). It is then followed by applying filtering and/or transformation

that unmask the intrinsic information within the signals and enhance their patterns and properties with some initial discrimination. Later, these signals are discriminated against to understand their intention, which is normally done using a machine learning algorithm. Finally, the resulting pattern is translated into control signals for device manipulation. These steps are shown in Figure 1 [38].



**Figure 1.** Basic components of a BCI. The image illustrates the map between the input and output through the translating algorithm. Signals are acquired by electrodes and then translated into a control signal for an external device (e.g., wheelchair, neuro-prosthesis or exoskeleton) using processing steps.

### 2.1. BCI Control Paradigms

The second component is feature extraction, which in the case of BCI, unfolds the brain signal characteristics from nonessential material and shows them in a more meaningful form, manageable for either humans or computers. Hence, it is important to show which commands would be used for control and the features that better represent the signal to be analyzed.

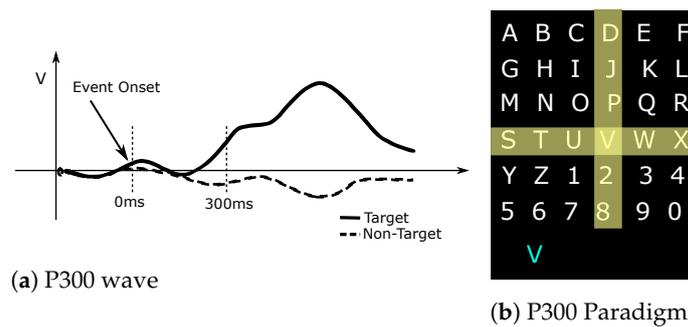
**BCI control paradigms** depend on choosing the feature and the type of signal used as a pattern for the BCI. There are mainly two types of EEG paradigms used on BCI: Evoked potentials (EP) [39] and changes in the spontaneous oscillatory EEG activity.

Generally, there are two types of brain paradigms: Evoked Potentials and changes in the spontaneous oscillatory EEG activity.

#### 2.1.1. Evoked Potentials

Evoked Potentials (EPs) are changes in the electrical potentials that are locked in time to certain events (i.e., visual or tactile). Normally, these brain signals are averaged over one second to be used as control signals. The main techniques are **P300 (P3) wave of visual evoked potentials** and **Steady State Visual Evoked Potential or SSVEP**. The P300 was first described by Sutton [40] as events that occur that have an alteration at 300 milliseconds after a visual event is presented (Figure 2a). The most used P300 was described by Farwell-Donchin [41], where a matrix of letters and numbers (or symbols) are presented on a six-by-six grid (Figure 2b) and flashed horizontally and vertically, and when a line that has the corresponding symbol blinks, the P300 stimulus appears. Further, P300 can be employed as a lie detector by presenting a visual stimulus related to the lie that occurs if the subject knows the stimuli [42,43].

Similarly, SSVEPs are brain responses to visual stimuli, such as flickering lights, that manifest frequency-locked signals with an increased amplitude of the stimulated frequency located over the occipital lobe. Due to that, they do not require eye movement, and they can be used for people that still have eye acuity but cannot move their eyes [44].



**Figure 2.** P300 wave and the classical P300 spelling paradigm described by Farwell-Donchin [41].

### 2.1.2. Oscillatory Activity Patterns

These kinds of signals are voluntarily induced by the user, such as hand movements that are associated with a power change or synchronization/desynchronization over certain rhythms. This effect also happens using imagination over body movement. In this case, the desynchronization and synchronization are known as event-related desynchronization (ERD) and event-related synchronization (ERS), respectively, [45].

Normally, they appear after the termination of the event. Unlike EP, these signals do not require locking to a stimulus; hence, they can be used at their own pace. The two most common are Motor Imaginary and Slow Cortical Potentials. The first signals are changes that occur with the imagination of motor movement [46–48]. Using imagination opens the path of using areas that are not normally used for the control of devices. Slow Cortical Potentials are generated slow voltage changes in some wave patterns over the cortex that can be produced by prolonged trained users to select words or pictograms from a computer [49].

## 3. Signal Processing

Decoding brain states is problematic since they have a poor signal-to-noise ratio, variability between trials (in different sessions or even on the same session), high dimensionality data, highly location-dependent data, etc. [50]. Thus, for correct decoding, the usage of brain signals requires several steps, starting from signal acquisition (e.g., EEG and ECoG recordings), feature extraction, pattern recognition and, finally, translation into control signals (Figure 1).

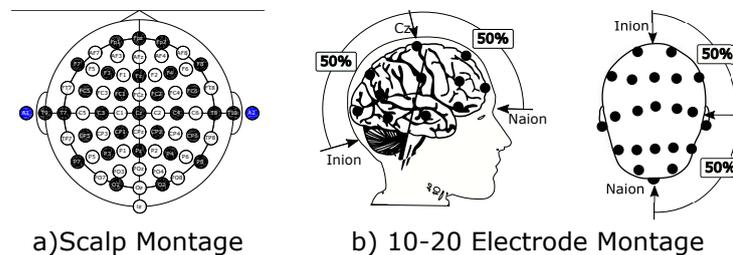
### 3.1. Signal Acquisition

The brain is composed of billions of neurons that communicate using electrical signals. These signals are produced at similar locations between individuals, yet it is not fully understood why they are emitted there and what their intentions are. However, it is still important to know when they are produced and their location, which reflects the normal or abnormal activity of the brain and user intentions.

Many techniques have been developed to record brain activity (e.g., EEG, ECoG, single-neuron recording, PET, fMRI, MEG and FNIR) [14,51]. Despite all of them being able to record brain activity, in this work, we will focus on EEG. The reason behind this is that the other enlisted techniques are either invasive, expensive or have high latency.

EEG recordings are done using electrodes attached to the surface of the scalp, and each electrode measures the potential difference from a reference electrode and itself [51]. Correspondingly, these potentials reflect activity within the brain, and to avoid unwanted signal noise due to poor connection, the electrodes must have good contact with the area of interest. Furthermore, for understanding and repeatability, it is of great interest to know exactly where the electrodes are commonly located. For that reason, international 10–20 electrode systems are used, which consist of making an arc grid over the scalp that starts at specific locations, where the Nasion and Inion are the longitudinal references and the right and left preauricular are the lateral references (see Figure 3). The corresponding name of

each arc crossing represents a location of the brain lobes, which is helpful for the spatial analysis of recorded signals.



**Figure 3.** EEG Electrode Montage.

Furthermore, since EEG signals go through several layers of muscle, skin and bone, having a correct measure of the brain signal requires a process of amplification and filtering to improve the signal quality.

First, the amplification helps increase the low-signal amplitude ( $\sim 10\text{--}20\ \mu\text{V}$ ), which is not easy to interpret using common displays, recorders or AC/DC converters. Notably, amplifiers must fulfill some requirements such as noise rejection and guarantee equipment and patient protection.

Then, filtering is done to reduce either the environmental noise (e.g., power lines and electrical and/or surrounding medical equipment) or the physiological noise (e.g., muscle activation, eye movement, and/or blinking) [52]. Dealing with environmental noise is usually easier than dealing with physiological noise. Environmental noise can be avoided by removing most of the sources of electromagnetic signals from the recording room and its vicinity. Furthermore, one of the most common techniques is to use a notch filter at 50 or 60 Hz that helps by removing the noise of the electric power lines' artifacts. For physiological noise, one of the most common approaches is to incorporate physiological signals in the recordings and subtract them from the EEG. Other methods include telling the subject to remain still, not blink and hold the gaze during the study; however, this is usually difficult and can introduce even more noise because of the voluntary attention needed to control those body actions.

### 3.2. Feature Extraction Methods

The second component is feature extraction. Once you have selected the correct control signal, it is necessary to find a way to better represent it. BCI mainly use four kinds of feature extraction methods to represent the signal: temporal methods (i.e., signal amplitude and auto-regressive), frequency methods (i.e., band power and power spectral densities), time-frequency methods (STFT and wavelets) and some others (e.g., coherency, phase synchronization, etc.). The selection of one of these methods will depend completely on the desired control command for classification. Thus, depending on the transformation, it is recommended that EEG recordings have a high sampling rate and more than a single electrode for a better signal recording.

## 4. Pattern Recognition

The third component is pattern recognition, which is the one that translates the feature into a control signal. The main problem in this step of the BCI is that the brain signals are highly variable and would be hard, if not impossible, to manually translate into control signals. Then, in the light of solving this problem, the use of Artificial Intelligence (AI) is highly beneficial. Given that there are many techniques and their applications in science are vast, it is required to understand the basics of AI techniques, what each technique can do and how they are developed. Thus, students must see what a real application of these techniques can do, especially in BCI.

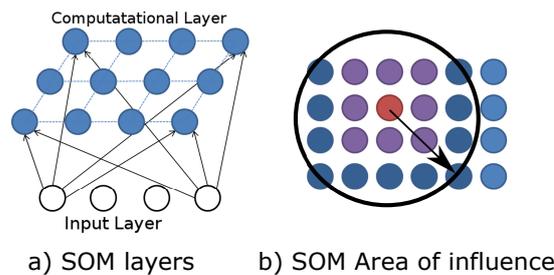
In particular, this work focuses on five AI algorithms: Kohonen Self-Organizing Maps (SOM), Artificial Neural Networks (ANN) trained by Backpropagation, Linear Discriminant Analysis (LDA), Supported Vector Machines (SVM) and Restricted Boltzmann Machines (RBM), as well as their applicability to BCI. These techniques were chosen because each one brings different properties that are interesting to be analyzed. In the case of SOM, as an unsupervised network that does not require labels, it is capable of creating an internal representation of the system. On the other hand, neural networks facilitate their training by using the error to correct its internal representation. Linear discriminant analysis is the most used technique used for BCI due to its simplicity and adaptability but with the limitation of working only for binary classification. Furthermore, Support Vector Machines is selected to be tested in this work since it is one of the most used classification techniques and has high separability capabilities. Finally, the Restricted Boltzmann Machine algorithm is analyzed as a different technique for BCI that explores and characterizes both the signals and their classes together, creating an internal map of them.

#### 4.1. Kohonen Self-Organizing Maps

A Self-Organizing Map (SOM) [53] is an unsupervised neural network that produces a discrete representation of an input space, which is referred to as a map. This algorithm is used as a clustering or dimensionality reduction method and consists of an input layer and a computational layer (Figure 4a) conformed to nodes or neurons. Each node has a topological position and has a number of weights equal to the number of inputs ( $V = [v_1, v_2, \dots, v_n]$ , with  $n$  as the number of inputs,  $W_m = [w_1, w_2, \dots, w_m]$ ;  $m =$  number of nodes, where  $n = m$ ). The SOM method calculates the Euclidean distance between an entry vector and the weights of each node:

$$dist = \sqrt{\sum_{i=0}^n (v_i - w_i)^2} \tag{1}$$

and chooses the lowest distance to one node as the best or winning node. This node is referred to as the Best Matching Unit or BMU.



**Figure 4.** (a) Two layers of a SOM, with the lower layer as the input and the upper one as the nodes that map the input. In (b), the BMU and the area of influence are illustrated, with the area of influence decaying as the nodes move from the BMU outwards.

Once the BMU is found, the nodes in the neighbor (i.e., influence area) of the BMU and the BMU itself are selected, and their weights are updated (Figure 4b). The BMU’s influence area is calculated as  $\sigma(t) = \sigma_0 e^{-\frac{t}{\tau_\sigma}}$ , with  $\sigma_0$  as the lattice width at the instant  $t_0$  and  $\tau_\sigma$  as the updating constant of  $\sigma$ . After the area is selected, the weights are updated using the equation below:

$$w(t + 1) = w(t) + \Theta(t)\eta(t)(V(t) - W(t)) \tag{2}$$

where  $\eta$  and  $\Theta$  represent the learning rate and influence rate at the instant of time  $t$ . As training time advances, the learning rate and influence rate diminish their effect by:

$$\eta(t) = \eta_0 e^{-\frac{t}{\tau_\eta}}; \quad \Theta(t) = e^{-\frac{dist^2}{2\sigma(t)}} \tag{3}$$

where  $\eta_0$  is the initial learning rate, and  $\tau_\eta$  is the update constant of  $\eta$ . With this learning technique, the inputs with similar characteristics will cluster together around a given node, while inputs with different characteristics will cluster apart in other different nodes. The steps of this method can be seen in Algorithm 1.

**Algorithm 1** : SOM Pseudo-code.

**Input network:**

Training set  $S = \{X_1, X_2, \dots, X_s\}$ ; Learning and influence rate  $\alpha$  &  $\theta = ((0 - 1])$

**Init network:**

Initialize the weights to a small random value

**Train network:**

Loop until  $w^{new!} = w^{old}$  &&  $iter < max \ iterations$ :

Select random input:  $(X_s)$

Compute the distances:  $dist = \sqrt{\sum_{i=0}^n (x_i - w_i)^2}$

Select BMU

Calculate area of influence:  $\sigma(t) = \sigma_0 e^{-\frac{t}{\tau_\sigma}}$

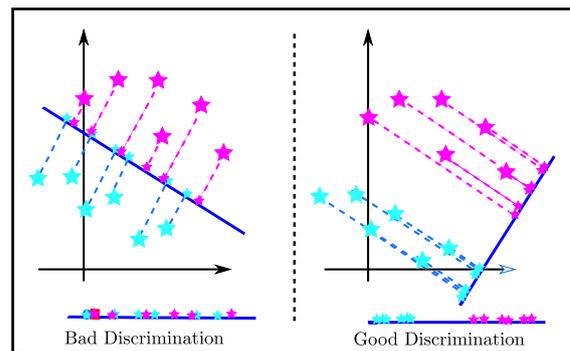
Update weights:  $w(t + 1) = w(t) + \theta(t)\alpha(t)(X(t) - W(t))$

Update learning and influence rate:  $\alpha(t) = \alpha_0 e^{-\frac{t}{\tau_\alpha}}; \quad \theta(t) = e^{-\frac{dist^2}{2\sigma(t)}}$

**Output network:** Weights  $w$

4.2. Fisher’s Linear Discriminant Analysis

Linear Discriminant Analysis (LDA), also known as Fisher’s LDA, uses a linear hyperplane to separate the data representing each of the two classes (see Figure 5).



**Figure 5.** Classes separation through the projection.

The hyperplane is obtained by projecting high-dimensional data onto a line. The objective of this projection is to maximize the distance between the means of the two classes while minimizing the variance within each class. This defines the Fisher criterion, which is maximized over all linear projections,  $w$ :

$$J(w) = \frac{(\tilde{\mu}_1 - \tilde{\mu}_2)^2}{\tilde{\sigma}_1^2 + \tilde{\sigma}_2^2} \tag{4}$$

where  $\tilde{\mu}_i$  represents the mean of the projections of classes 1 and 2 ( $\tilde{\mu}_i = w^T \mu_i$ ), and  $\tilde{\sigma}^2$  represents the variance of these projections ( $\tilde{\sigma}_i^2 = \sum_{y_i \in \text{Class}_i} (y_i - \tilde{\mu}_i)^2$ ), where  $y_i$  is the

projected samples  $y_i = w^T x_i$ . Based on these equalities, we can rewrite Fisher’s criterion as a function of  $w$  in the following way:

$$J(w) = \frac{w^T S_B w}{w^T S_W w} \tag{5}$$

where  $S_B$  and  $S_W$  measure the separation between means of both classes and the within-class scattering, respectively. Given the previous equation, we can find its maximum by solving the generalized eigenvalue problem as follows:

$$S_B w = \lambda S_W w \tag{6}$$

Solving this problem will result in a collection of eigenvectors  $w$  and their corresponding eigenvalues  $\lambda$ . Then, these eigenvectors must be sorted according to their eigenvalues from biggest to smallest, and finally, a set of  $k$  eigenvectors is chosen to create a weight matrix  $w$ , which is the representation of the new space in which the data are going to be projected. Algorithm 2 gives the basic steps for LDA.

---

**Algorithm 2** : LDA Pseudo-code.

---

**Input network:**

Training set  $S = \{(x_1, t_1), (x_2, t_2), \dots, (x_s, t_s)\}$

**Train network:**

Calculate the means  $\mu$

Calculate  $S_B$  and  $S_W$

Get the eigenvectors and values:  $(e_1, e_2, \dots, e_n), (\lambda_1, \lambda_2, \dots, \lambda_n)$

Obtain the matrix  $S_x$

Sort the eigenvectors and chose the  $k$  ones with the bigger eigenvalues

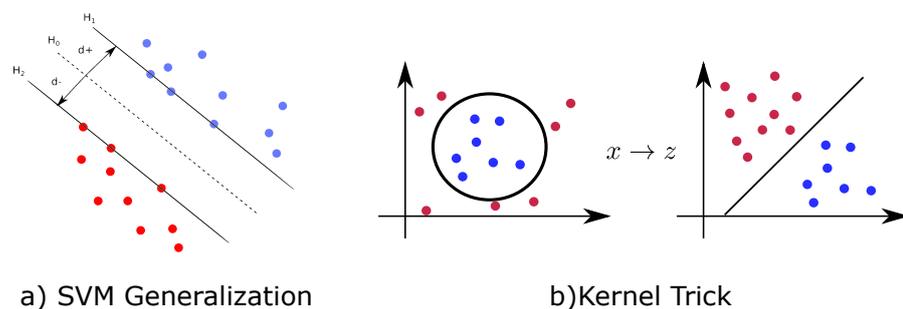
Form a matrix  $W(n \times k)$

**Output network:** Returns Matrix  $W$

---

4.3. Supported Vector Machine

Supported Vector Machines (SVM) are supervised models that search how to separate two classes using a discriminative hyperplane. SVM searches for a hyperplane that maximizes the separation margins of the system, i.e., the distances between the classes of the training points. In Figure 6a, these distances are shown as  $d_1$  and  $d_2$ .



**Figure 6.** Support Vector Machine. (a) SVM to get the optimal hyperplane for generalization. (b) The expansion of SVM capabilities using the kernel trick.

As it can be seen in the previous Figure, the hyperplanes divide the input data in two different regions, one considered to be positive ( $y_i = 1$  for  $H_1$ ) and the other to be negative ( $y_i = -1$  for  $H_2$ ). Thus, the hyperplanes, as shown in Figure 6a, are defined as  $H_1 : y_i(w^T x_i + b) \geq 1$  and  $H_2 : y_i(w^T x_i + b) \leq -1$ . Those conditions can be combined into  $y_i(w^T x_i + b) \geq 1$ . The main objective is that the classifier has a margin as big as possible,

i.e., maximize the distance between both hyperplanes, defined as  $d = \frac{2}{\|w\|}$ . This is the same as minimizing the function  $\frac{1}{2}w^T w$  constrained to the condition of  $y_i(w^T x_i + b) \geq 1$ .

Although SVM is a linear classifier, it can be extended to non-linear using the ‘kernel trick’, which maps the data into a different space (Figure 6b) where the data can be linearly separated. Furthermore, SVM is a binary classification but can be easily converted into a multi-class classifier using the technique of one vs. the rest, where a classifier is made for each class and discriminated against the rest of the classes. The winning class is the one with the higher final confidence value. The steps for SVM can be seen in Algorithm 3.

---

**Algorithm 3** : SVM Pseudo-code.

---

**Input network:**

Training set:  $S = \{(x_1, y_1), (x_2, y_2), \dots, (x_s, y_s)\}$

Regularization Parameter:  $C$

Tolerance and maximum number of iterations

**Init network:**

Initialize  $\alpha_i = 0, \forall_i$  and  $b = 0$

**Train network:**

Quadratic Programing such as SMO

**Output network:**

Lagrange Multipliers:  $\alpha \in \mathbb{R}^s$

Threshold:  $b \in \mathbb{R}$

---

4.4. Backpropagation

Backpropagation (BP) is a common technique for training Artificial Neural Networks (ANN), where, in few words, the error is propagated backwards so that the network can learn by itself and adapt depending on previous mistakes [54] (Figure 7). The main objective of the BP algorithm is to minimize the error function in a weight space using the gradient descent method. The combination of weights that minimize the error of the function is considered the solution to the learning problem. To use the gradient descent method, first, we must guarantee that the error function and the activation function are continuous and differentiable. One of the activation functions that is usually implemented for BP is the sigmoid function,  $S_c(x) = \frac{1}{1+e^{-x}}$ , of which a derivative exists and is continuous  $\frac{dS_c(x)}{dx} = \frac{e^x}{(1+e^{-x})^2} = S_c(x)(1 - S_c(x))$ . The activation function of a neuron calculates the sum of the inputs  $x_1, \dots, x_n$  times the weights  $w_1, \dots, w_n$  plus the bias of that particular neuron  $\theta$ . In the particular case of the sigmoid function:

$$o_m = \frac{1}{1 + e^{-\left(\sum_{i=1}^n w_i x_i - \theta_m\right)}} \tag{7}$$

where the output of the system is composed by the outputs of each neuron,  $(o_1, \dots, o_m)$ . The method of BP searches to minimize the error between the generated output  $o_i$  and the original output, i.e., the target  $t_i$ . This error can be represented through the mean sum squared loss function:

$$E = \frac{1}{2} \sum_{i=1}^P ||o_i - t_i||^2 \tag{8}$$

To minimize error, the weights need to be corrected using the gradient descent:

$$\nabla E = \left( \frac{\partial E}{\partial w_1}, \frac{\partial E}{\partial w_2}, \dots, \frac{\partial E}{\partial w_n} \right) \tag{9}$$

where each weight is updated using  $\Delta w_n = -\gamma \frac{\partial E}{\partial w_n}$ , with  $\gamma$  being the learning constant. The weights are updated iteratively until  $\nabla E \approx 0$  using:

$$w_n^{new} = w_n^{old} + \Delta w_n \tag{10}$$

With this, all weights are updated with the intent of error minimization. Once the error is minimized, the network can be used on unseen data to check out its performance.

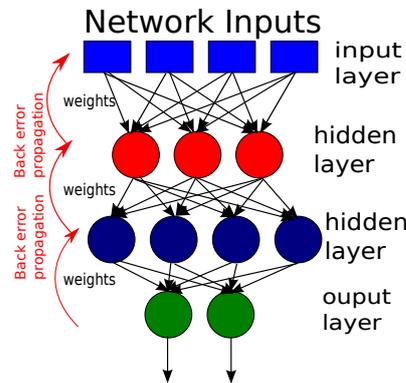


Figure 7. Backpropagation applied on a Neural Network.

The resulting method can be described in Algorithm 4.

---

**Algorithm 4 :** Artificial Neural Network trained with BackPropagation.

---

**Input network:**

Training set  $S = \{X_1, X_2, \dots, X_s\}$ ; Learning and influence rate  $\alpha$  and  $\theta = ((0 - 1])$

**Init network:**

Initialize the weights  $w_n$  and bias  $b$  to a small random value

**Train network:**

Loop until  $w^{new} = w^{old}$  &&  $iter < max \ iterations$ :

**repeat**

Chose random input  $X_i$

**Forward Propagation:**

**for** All MLP Layers **do**

Use 7 to each of the layers until the output layer

**end for**

**Backward Propagation:**

Calculate quadratic error according to 8

**for** All MLP Layers **do**

Calculate each of the deltas using  $\Delta w_n = -\gamma \frac{\partial E}{\partial w_n}$

**end for**

Update weights using  $w_n^{new} = w_n^{old} + \Delta w_n$

**until**  $w_n$  when it converges

**Output:**

Weights  $w_n$

Use trained network for classification

**Results:** Activity Labels A of the unlabeled data

---

4.5. Restricted Boltzmann Machines

A special type of ANN is the one developed by Hinton [55], known as Restricted Boltzmann Machines (RBM). RBMs are two-layer neural networks of stochastic units, divided into *visible* units  $\mathbf{v} = (v_1, \dots, v_i)$  and *hidden* units  $\mathbf{h} = (h_1, \dots, h_j)$ , which have symmetrical connected weights (see Figure 8). The visible units represent the data, while the hidden units are known as feature extractors. RBMs pretend to be model dependencies

over visible variables. The probability  $p(\mathbf{v}, \mathbf{h}; \Theta) \propto e^{-E(\mathbf{v}, \mathbf{h}; \Theta)}$  is known as the Boltzmann distribution, which has an energy function described as:

$$E(\mathbf{v}, \mathbf{h}; \Theta) = -\mathbf{h}^T \mathbf{W} \mathbf{v} - \mathbf{b}^T \mathbf{v} - \mathbf{c}^T \mathbf{h} \tag{11}$$

with  $\mathbf{W}$  as the symmetric weights,  $\mathbf{b}$  and  $\mathbf{c}$  as the bias of the visible and hidden units, respectively, and  $\Theta = (\mathbf{W}, \mathbf{b}, \mathbf{c})$ . The two conditional distributions over the variables, i.e., hidden given the visible and visible given the hidden, are given by:

$$p(h_j = 1 | \mathbf{v}) = \sigma_h(c_j + \sum_i v_i w_{ij}) \tag{12}$$

and

$$p(v_i = 1 | \mathbf{h}) = \sigma_v(b_i + \sum_j h_j w_{ij}) \tag{13}$$

where  $\sigma$  represents the activation function. Since the hidden variables cannot be observed, we need an algorithm that improves the RBM representation of the system. This algorithm is called Contrastive Divergence (CD) [56], which allows fitting the probability  $p(v)$  to a certain set of observations (e.g., EEG signals).

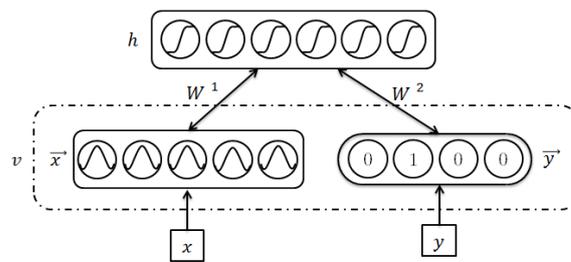


Figure 8. A representation of the RBM with connections symmetrically connected weights.

The pseudo code for RBM can be seen in Algorithm 5.

---

**Algorithm 5 :** RBM Pseudo-code using Contrastive Divergence-k.

---

```

% Notation: x ← b means x is set to value b
%           x ~ p means x is sampled from p
Input network:
  Training pair v = (xi, yi)
  Learning rate α
Init network:
Train network:
  v(0) ← v
  Loop Gibbs sampling t = 0, ... k - 1
    % Positive phase
    ĥ(t) ← σ(c + Wv(t))
    % Negative phase
    h(t) ~ p(h|v(t))
    v(t+1) ~ p(v|h(t))
  end Loop
  ĥ(k) ← σ(c + Wv(k))
  % Updates
  for θ ∈ Θ do
    θ ← -α ( ∂/∂θ E(v0, ĥ0) - ∂/∂θ E(vk, ĥk) )
  end for
Output network: Return weights and biases

```

---

#### 4.6. Advantages and Disadvantages of the Methods

It is intended to test the efficacy of the proposed methods for BCI, but, before doing that, it may be convenient to clarify the advantages and disadvantages of each method. Table 1 is a listing of the main characteristics of each of them.

**Table 1.** Advantages and disadvantages of the methods.

Method	Advantages	Disadvantages	Ref
SOM	<ul style="list-style-type: none"> <li>- Good for dimensional reduction</li> <li>- Does not require label</li> <li>- Outliers affect single portions of the map</li> </ul>	<ul style="list-style-type: none"> <li>- The maps are not the same (initialize randomly)</li> <li>- No direct classification</li> <li>- The size of the map is not obvious</li> </ul>	[57–59]
LDA	<ul style="list-style-type: none"> <li>- Fast Method</li> <li>- Easy to implement</li> </ul>	<ul style="list-style-type: none"> <li>- Requires normal Distribution</li> <li>- Decision boundaries are linear</li> <li>- Limited to two classes</li> </ul>	[60,61]
BP	<ul style="list-style-type: none"> <li>- Can detect complex relationships between variables</li> <li>- Require less formal statistical training to develop</li> </ul>	<ul style="list-style-type: none"> <li>- Are prone to overfitting</li> <li>- Computational Expensive</li> </ul>	[62,63]
SVM	<ul style="list-style-type: none"> <li>- Convex optimization problem (no local minima).</li> <li>- Can be used with a kernel so can model non-linear relationships</li> <li>- Regularization parameter helps to avoid over-fitting.</li> </ul>	<ul style="list-style-type: none"> <li>- Binary Classifier; for multiclass, it requires to use other methods.</li> <li>- Inefficient to train</li> <li>- Specific kernel is needed to have good separability</li> </ul>	[64–66]
RBM	<ul style="list-style-type: none"> <li>- Generative Model</li> <li>- Able to construct Deep artificial Neural Networks</li> </ul>	<ul style="list-style-type: none"> <li>- Computational Expensive</li> <li>- Time Consuming</li> </ul>	[67,68]

#### 4.7. Accuracy and Cross Validation

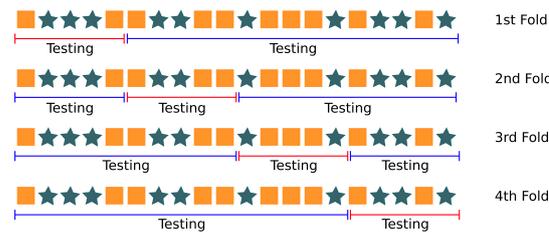
In general, it is important to know how well a given method performs over a specific task; thus, it is important to calculate its accuracy. Therefore, to calculate the accuracy, the Mean Square Error (*MSE*) was used, as shown in Equation (14), with  $t_i$  and  $y_i$  as the observed and predicted outputs.

$$Acc = 100 - MSE, \quad \text{with}$$

$$MSE = \frac{100\%}{n} \sum_{i=1}^n (t_i - y_i)^2 \quad (14)$$

Another important measure to be done is how the classification method is going to behave when dealing with independent data, i.e., how general the method is. It is important to check that the method does not overfit, which means that it obtains a perfect score when dealing with training data but has a poor performance when it is exposed to unseen data. One way to overcome this problem is to observe the performance of the classifier over a training dataset and then verify it using a test dataset; this is the basic idea behind a technique called *cross validation*. However, we still have the problem that the behavior of the system may depend heavily on which data points are used for training and which ones are used for testing. Thus, the algorithm may yield different results depending on how the data was divided into the training and testing datasets.

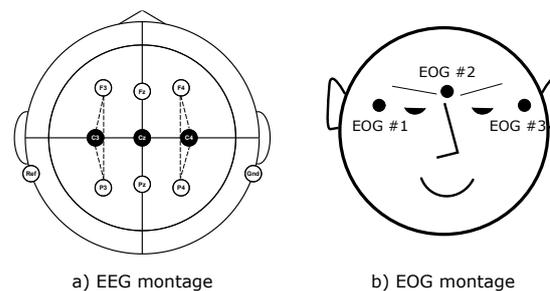
One of the most used methods to overcome this problem is known as K-fold Cross-Validation. This technique is based on splitting the data set into  $k$  smaller sets and repeat the training and testing  $k$  times. Each time the algorithm uses one different testing dataset, the other  $k - 1$  datasets are used for training. Then, the validation results are averaged to obtain the overall performance of the algorithm (see Figure 9). This testing procedure allows us to describe how well the classifier performs using different datasets.



**Figure 9.** One Iteration 4-Fold Cross Validation partition of the data set.

## 5. Method

To test the previously described techniques, we used the dataset from [37] (a link to this dataset can be found in Appendix C), consisting of EEG and EOG recordings from ten naive right-handed subjects (six male and four female) with an average age of  $24.7 \pm 3.3$  years. Furthermore, the participants had normal or corrected-to-normal vision during the experiments. The gathered data consisted of three bipolar EEG recordings (C3, Cz and C4) with a sampling frequency of 250 Hz and the electrode Fz as the EEG ground, as shown in Figure 10a. The recorded signals had a dynamic range of  $\pm 100 \mu\text{V}$ , which were analog bandpass filtered (0.5–100 Hz) and notch filtered (50 Hz). At the same time, EOG data were recorded using three monopolar electrodes (Figure 10b) with a dynamic voltage range of  $\pm 1 \text{ mV}$ .



**Figure 10.** EEG and EOG electrode montage.

Each subject participated in five sessions, two without feedback and three with feedback. At the beginning of each session, a 5-minute recording of continuous eye behavior was made to estimate the EOG artifact correction coefficients. These recordings were divided as follows: eyes open during 2 min, eyes closed during 1 min and eyes moving during 1 min (see Appendix A).

The sessions without feedback were done using a cue-based paradigm (Figure 11a), in which each subject had to perform motor imagery (MI) depending on the visual cue shown in the monitor. Each trial started with a fixated cross and an additional short warning tone. Then, after some seconds, the visual cue consisting of an arrow pointing either to the right or to the left appeared for 1.25 s. Afterward, the subject had to maintain the corresponding MI for a period of 4 s. In between trials, a short break of a random period between 1.5 and 2.5 s was given to avoid adaptation.

The three feedback sessions consisted of four runs with twenty trials for each type of motor imagery. These sessions were carried out using smiley feedback (see Figure 11b), which the initial state was centered and gray-colored. At the second two, a warning tone was emitted, which preceded a cue that lasted from second 3 to 7.5. According to the given cue, subjects had to move the smiley to the left or right by imagining hand movements towards those directions. The smiley changed color from gray to green or red and the curvature of the mouth from happy to sad if the direction was either correct or incorrect according to the cue, respectively.

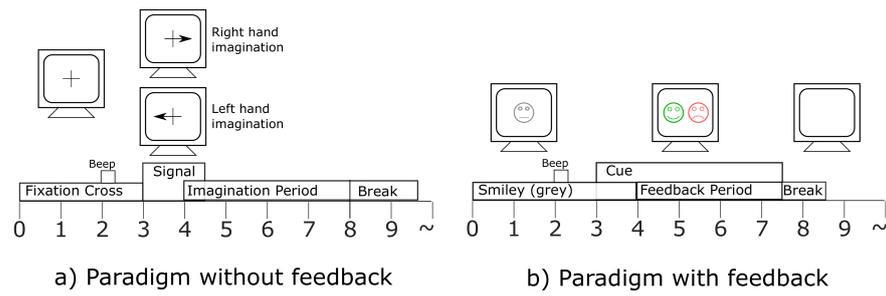


Figure 11. The paradigm of the recorded data.

Data Processing

Two different approaches were designed for the data processing step. The first one consisted of testing the performance of the classifiers with a large range of frequency bands and without any EOG removal. To do this, the data were transformed into a frequency domain between (8–30) Hz, which is the range in which changes in amplitude occur. Then, the second approach was to reduce the frequency bands into two different ranges, (8–12) Hz and (22–30) Hz, and the EOG was removed using the previously proposed regression.

A specific SOM was trained for over 50 epochs for each one of the nine subjects to obtain an internal representation of both classes in order to observe if they could be easily discriminated against. Each SOM had 100 units distributed in a 10 × 10 matrix (see Figure 12), an initial learning rate  $\eta_0 = 0.2$ , an initial lattice width of ( $\sigma_0 = 10$ ) and updating constants  $\tau_\eta = 100$  and  $\tau_\sigma = 4$ .

Being an unsupervised method, the training dataset was used to tune the weights of the SOM, and the testing dataset was used to observe the final internal representation that it could generate. In this case, the first three sessions were used as training data and the remaining two as testing data.

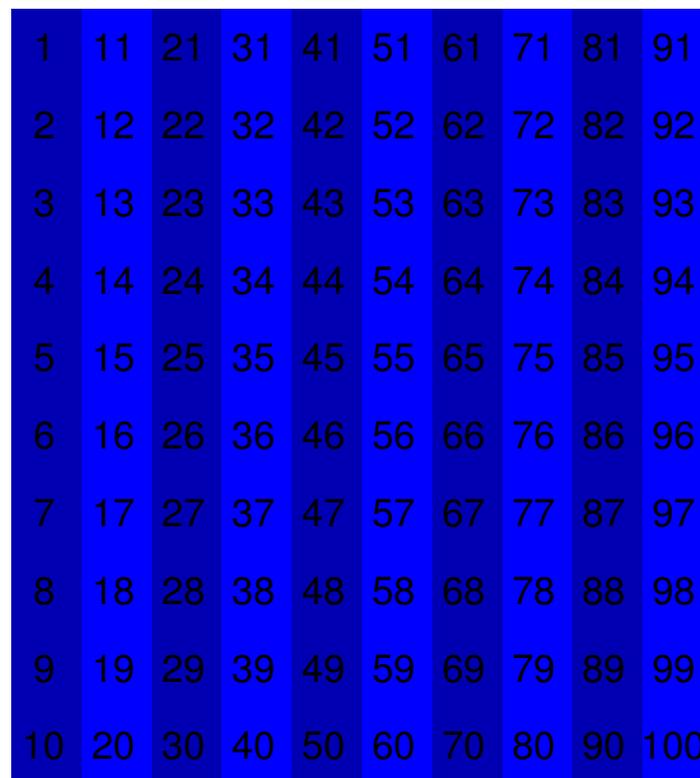


Figure 12. Neuron Distribution over the SOM.

The other techniques were supervised methods, where the testing dataset was used to check the final classification accuracy of each method. K-fold cross-validation was done to better evaluate the results of these algorithms, with  $k = 5$ . In other words, the data were split so that one of the recorded sessions was considered as testing data and the reminding sessions as training data. This process was repeated five times, then the accuracy was averaged.

As LDA calculates the mean and scatter matrices, it does not require any specific training parameter, so the process is as straightforward as shown in Section 4.2. For the SVMs, since this problem is a binary classification problem, there was no need to use any expansion method. However, the SVMs were trained on a radial basis using the kernel function  $K(x_i, x_j) = \exp(-(1/2\sigma^2)||x_i - x_j||^2)$ , which is one of the most common kernels used for BCI. The box constraint parameter  $C = 1e - 2$  was used since it gave the best overall results.

A 1000-neuron Artificial Neural Network was trained using backpropagation, which had a learning rate of 0.05 and a momentum of 0.01. The weights were initialized through a normal distribution  $N(0, 0.01^2)$ . The ANN has trained over 100 sweeps (or epochs) with batches of 100 randomly selected EEG trials.

Finally, the RBM initial training parameters (weights, biases and rates) were obtained from [56] and adapted after some preliminary analysis. The RBM was trained over 100 epochs, each comprising Contrastive Divergence updates derived from 10 Gibbs sampling iterations (CD10). The training datasets were composed of mini-batches of 100 randomly selected EEG trials. The weights were drawn from a normal distribution  $N(0, 0.1^2)$  for the Gaussian-Binary connections and  $N(0, 0.01^2)$  for the Softmax-Binary connections, with each bias initialized at zero. The weights and biases were updated with a learning rate of  $10^{-3}$  and a momentum of 0.5 with an increment of 0.1 at 40% and 80% of the learning process. The step-up value of 0.1 was selected because higher increments made the learning unstable. A cost value of  $2 \cdot 10^{-4}$  was selected since it facilitated the learning process of CD by increasing the mixing rate of the Markov chain.

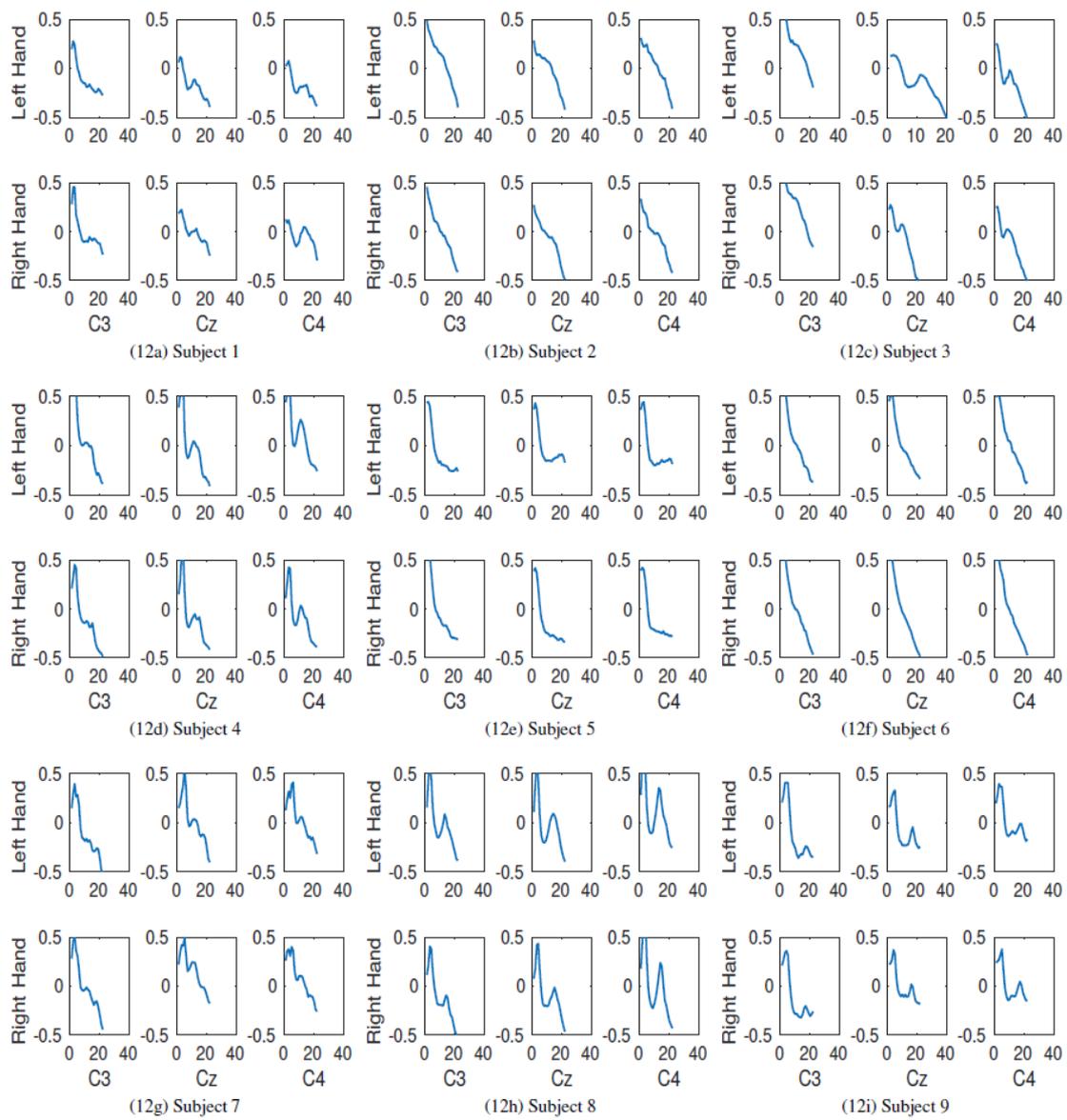
All the algorithms were implemented using Matlab™ on Windows™ 7 professional 64-bit operating system. The computer used to run the algorithms had an Intel® CPU E5-2618L v3 @ 2.30 Ghz with 16 cores and 24 GB RAM.

## 6. Results

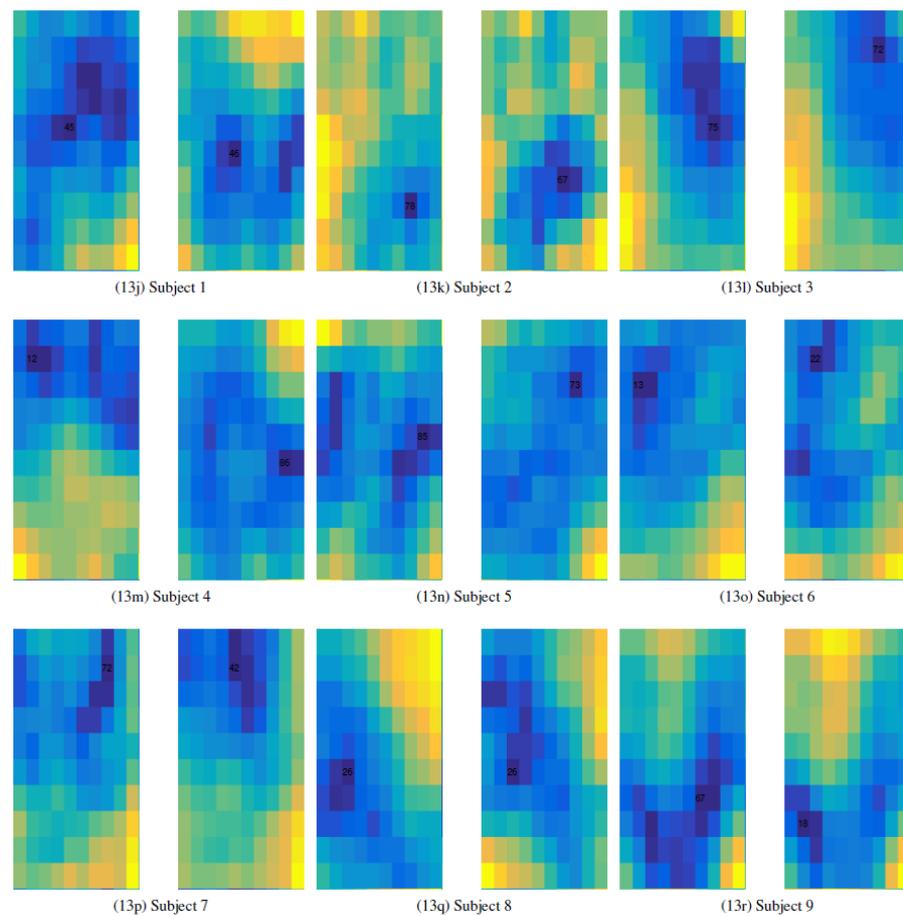
For the first part of the results, the classification accuracy (or mapping accuracy in the case of SOM) was obtained for each algorithm using a frequency band of (8–30) Hz and no regression filtering over the EOG. Then, the band was reduced for all subjects to (8–12) Hz and (22–30) Hz using an average of the frequencies obtained in [37], and the EOG artifacts were also reduced through the regression procedure.

### 6.1. Frequency Band of (8–30) Hz and no EOG Filtering

First, for a better comparison, the frequency response of every subject is shown in Figure 13. It is then followed by the results of SOM shown in Figure 14, which shows the final internal representation of the two classes inside the SOM network for each subject. Furthermore, Table 2 shows the winner neuron of the SOM for both classes for each subject. Table 3 shows the resultant training and testing accuracy of the four remaining methods.



**Figure 13.** The average frequency response over three bipolar recordings: C3, Cz and C4.



**Figure 14.** SOM representation of the different classes for the nine different subjects, including the winner neuron.

**Table 2.** The position of the the winner neuron for Class 1 and Class 2 of SOM with a frequency band of (8–30) Hz and no regression filtering over the EOG.

	1	2	3	4	5	6	7	8	9
Class 1	45	78	75	12	85	13	72	26	67
Class 2	46	67	72	86	73	22	42	26	18

**Table 3.** Results of the training and testing accuracy (in percentage) for nine different subjects using Linear Discriminant Analysis (LDA), Supported Vector Machines (SVM), ANN trained with Back Propagation (BP) and Restricted Boltzmann Machine (RBM) with a frequency band of [8–30] Hz and no regression filtering over the EOG.

		1	2	3	4	5	6	7	8	9
Train	LDA	65	57	57	78	68	58	67	70	72
	SVM	83	93	98	99	99	94	96	94	98
	BP	66	56	56	77	68	58	67	71	72
	RBM	67	57	56	77	68	57	68	70	73
Test	LDA	62	53	50	76	65	53	63	69	69
	SVM	51	50	50	50	49	51	49	52	50
	BP	63	52	53	76	66	55	63	68	70
	RBM	63	54	52	77	66	55	63	69	70

Figure 15 shows the training time (seconds) required by each of the five methods with the selected band (8–30) Hz and no EOG correction.

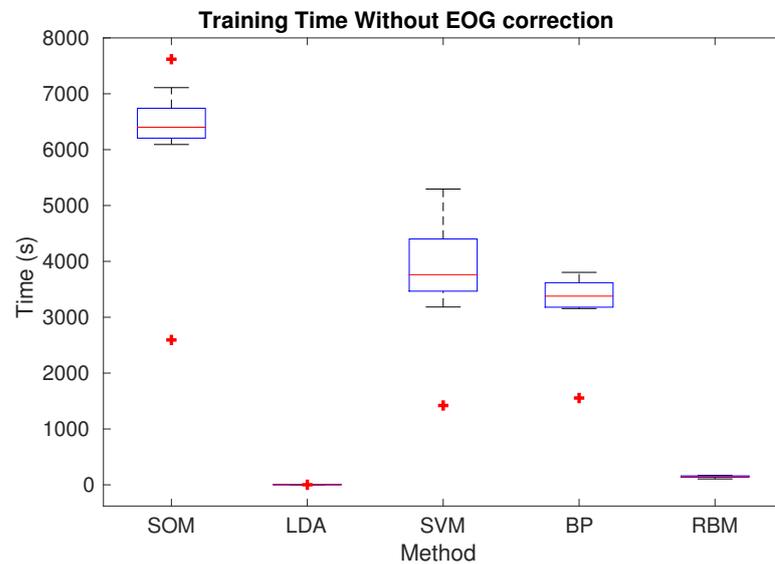


Figure 15. Times of the different methods without EOG correction.

### 6.2. Frequency Bands of [8–12] Hz and [22–30] Hz, with EOG Reduction

As in the previous section, first, the average frequency response of each subject is shown in Figure 16. Then, the final internal representations obtained through the SOM are shown in Figure 17, while Table 4 shows the position of the winner SOM neurons for both classes for each subject.

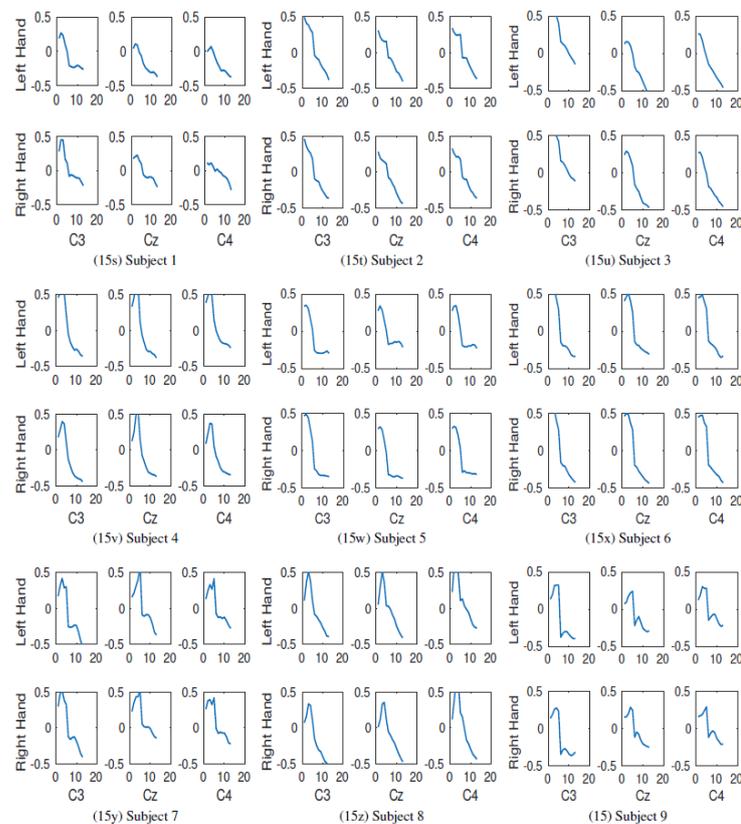
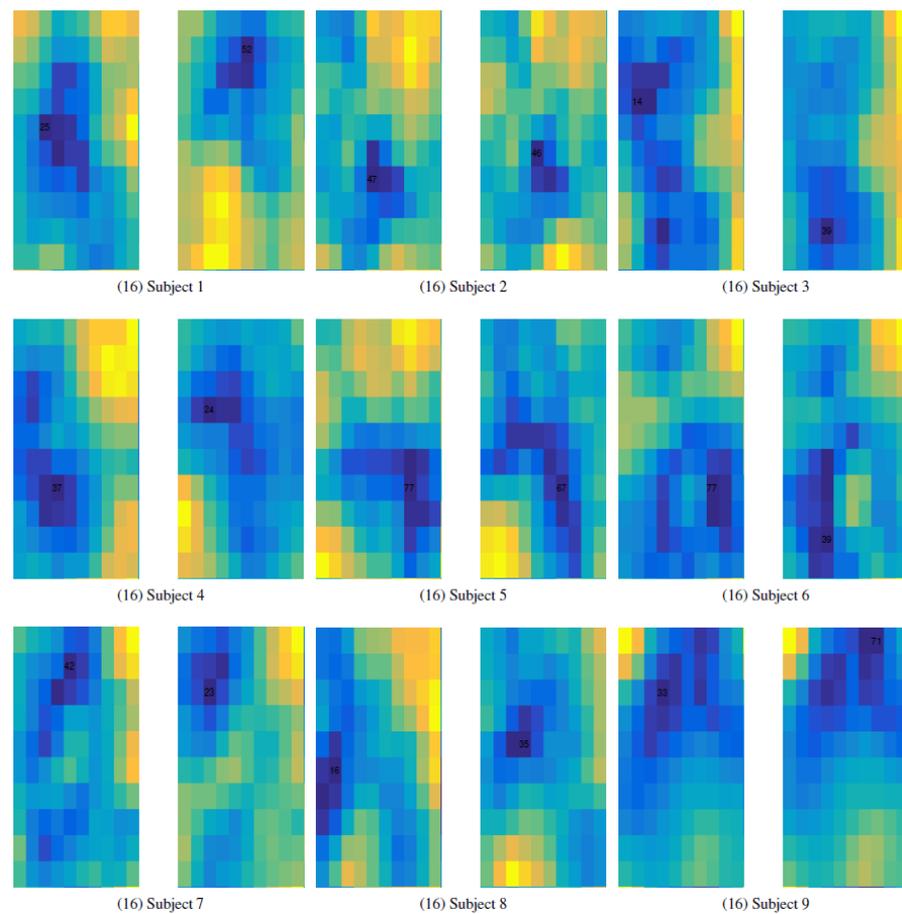


Figure 16. The average frequency response over the the three bipolar recordings: C3, Cz and C4 [37].



**Figure 17.** SOM representation, including the winner neuron, of the different classes for the nine different subjects with more limited frequency bands.

**Table 4.** The results of the position of the winner neuron for Class 1 and Class 2 of SOM with a frequency band of [8–12] Hz and [22–30] Hz and with regression filtering over the EOG.

	1	2	3	4	5	6	7	8	9
Class 1	25	47	14	37	77	77	42	16	33
Class 2	52	46	39	24	67	39	23	35	71

Table 5 enumerates the results for the training and testing accuracy using LDA, SVM, BP and RBM with the reduced frequency band and with the EOG reduction regression method. Finally, Figure 18 shows the training time (seconds) required by each of the five methods with the selected frequency bands ((8–12) Hz and (22–30) Hz) and EOG correction.

**Table 5.** The results of the training and testing accuracy (in percentage) for nine different subjects using Linear Discriminant Analysis (LDA), Supported Vector Machines (SVM), ANN trained with Back Propagation (BP) and Restricted Boltzmann Machine (RBM) for frequency bands of (8–12) Hz and (22–30) Hz and with regression filtering over the EOG.

		1	2	3	4	5	6	7	8	9
Train	LDA	68	56	55	81	68	57	65	73	64
	SVM	99	99	99	99	99	99	100	99	100
	BP	67	55	54	81	68	57	65	72	65
	RBM	69	57	57	81	68	57	68	73	73

Table 5. Cont.

		1	2	3	4	5	6	7	8	9
Test	LDA	64	52	49	79	66	54	62	71	62
	SVM	51	50	50	75	63	51	52	63	56
	BP	66	52	52	80	66	55	61	71	62
	RBM	68	53	52	80	66	55	63	71	70

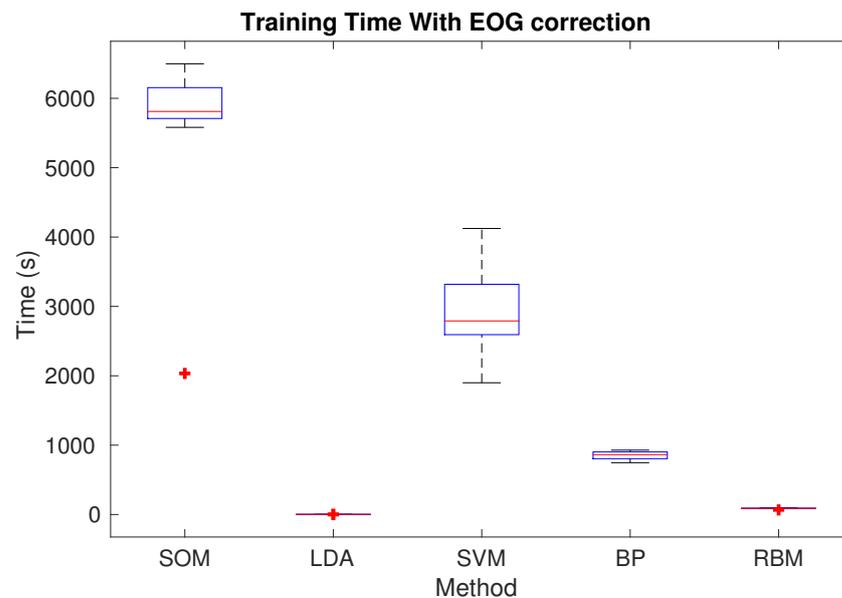


Figure 18. Times of the different methods with EOG correction.

## 7. Discussions and Conclusions

### 7.1. Discussions on the Results

**Frequency band of (8–12) Hz and no EOG filtering:** The results showed that for subject 4, 7 and 9, the SOM mapping has a better separation capacity (see Figure 14 (Subject 4), (Subject 7) and (Subject 9)) than for the rest of the subjects. Likewise, this effect is also represented on Table 2 (using Figure 12), where the winner neurons for the two classes are far apart from each other.

Correspondingly, a similar effect occurs for the classification methods (Table 3), where the same subjects, plus subjects 1, 5 and 8, showed training and testing accuracy higher than chance level  $\geq 60\sim 80\%$  for LDA, BP and RBM. However, for the other subjects, it is not easy to discriminate between both classes, which is illustrated in Figure 13, where there is no observable difference between the EEG frequency response among these subjects. It is important to notice that the SVM method did not show any discrimination capacity for any subject when applied to the testing dataset. This might be due to the kernel being too general and not working well on high dimension signals.

**Frequency bands of (8–12) Hz and (22–30) Hz and EOG reduction:** Using the band reduction and the EOG regression method, the results for SOM showed now that the best separation capacity was obtained for subjects 1, 3, 6, 8 and 9.

Although in the case of the classification techniques, there was a slight improvement for subjects 1, 4 and 8 in the accuracy percentage (Table 5), and there was no significant improvement over subjects 7 and 9, which already had good results on the (8–30) Hz band. In this case, it is assumed that there were some hidden attributes for subjects 1, 4 and 8 that were found due to the band reduction or filtering. However, these procedures did not necessarily help the other subjects, where the selected bands may not be optimal or the EOG contamination was not an important factor in their classification. In addition, there was no improvement over the accuracy for the subjects that already performed poorly. In

these cases, even the limited band and noise reduction technique could not help to uncover if there was any difference between the classes.

Lastly, for the SVM method, the accuracy was low for every subject except for subjects 4, 5 and 8. In general, the accuracy of SVM highly depends on finding the correct kernel to map the function, meaning that the initial parameters introduced for mapping into a higher dimension were not optimal for this database. Furthermore, SVM usually has problems discriminating when the same parameters are used in every individual, which means that it may require a specific setup for each subject.

In the case of processing times, the training time is reduced using the limited frequency bands and regression method. This is a natural consequence of having a reduced dimensionality of data. Furthermore, while comparing the processing times between the presented methods, it can be observed that SOM has the largest ones, followed by SVM. The former could be due to the process of inserting the data one by one to adjust the map, which could be improved using a batch method. On the other hand, SVM needs to solve a quadratic optimization problem with a large data set and a small box constraint, which limits the algorithm convergence speed. Moreover, it can be observed that the BP method is slower than the RBM method. The reason behind this is that the ANN had many more neurons than the RBM (1000 neurons for BP over 64 neurons for RBM). However, other numbers of neurons for the BP did not give as high accuracy as those obtained.

Finally, although LDA indeed is the fastest of all the presented algorithms by far, which is one of the reasons why its one of the most used methods for BCI, it has the problem of not being easily adaptable to a high number of classes, thus needing different methods for multi-class problems.

## 7.2. Conclusions

With the aim of driving the development of competencies for future engineers and scientists, schools require curricula that is in line with the technological progress and demands of Industry 4.0. Consequently, Education 4.0 is searching for new ways to introduce students to emerging technologies, such as artificial intelligence, and how they are applied on real-life situations.

Accordingly, Education 4.0 is responsible for helping future professionals begin being familiar with the area of artificial intelligence, and, at the same time, it must provide them with the opportunity of testing the acquired knowledge by applying it to real-life scenarios. Therefore, in the scope of the Education 4.0 framework, teachers and students are in need of updated educational material that helps them embrace their path towards teaching and learning more about the technologies that are being used in the incoming industrial revolution. Hence, the main objective of this work was to provide updated teaching/learning material that allows students to have an introduction to a cutting edge technology, such as BCI, which is used for several real-life applications, while providing them with the basic knowledge of five different AI techniques and how they can be applied over the basics of BCI experimentation.

These different AI techniques were presented through a brief review of the methods and their corresponding pseudo-codes while also presenting the results of their implementation on BCI so that students become aware of the problems and possible outcomes of these experiments. This implementation was done over a test bench that consists on EEG and EOG recordings obtained from [37]. From the obtained results, it is important for students to notice that the obtained behavior of each method made sense with the information presented in Table 1; however, the main problem in this work was that none of them was able to always discriminate or even discriminate similarly to all subjects.

Through the description of the AI techniques and the analysis of the results of applying them over the proposed BCI test bench, this work allows students to learn the basic theory of SOM, LDA, ANN-BP, SVM and RBM, as well as giving the guidelines for the application of those techniques over real-life BCI problems. Furthermore, teachers that are beginning to work under the Education 4.0 paradigm can use this work as introductory material

to BCI and artificial intelligence, and the proposed test bench can be used by them as a reinforcement exercise or project to test the understanding of students posterior to a BCI or AI lesson.

Notwithstanding of the contribution of this work to the development of updated curricula for the Education 4.0 framework, there is still a lot of work to do. To allow students to achieve a better comprehension of the presented methods, it is important to make an improved test bench with different band sizes as in [37] or use some other type of filter or dimensionality reduction method such as Common Spatial Patterns [69] (this can be seen in Appendix B), which is also part of the current state-of-the-art BCI. Additionally, more advanced artificial intelligence techniques for BCI classification can be explored to provide students with additional information about algorithms that are not only applied on BCI but also on other areas of Industry 4.0.

**Author Contributions:** Conceptualization, D.B., P.P. and A.M.; Data curation, D.B.; Formal analysis, D.B. and P.P.; Funding acquisition, A.M.; Investigation, D.B.; Methodology, D.B.; Project administration, P.P.; Software, D.B.; Supervision, P.P. and A.M.; Validation, D.B., P.P. and D.L.-B.; Visualization, D.B.; Writing—original draft, D.B., D.L.-B. and A.M.; Writing—review & editing, D.B., P.P. and D.L.-B. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research received no external funding.

**Data Availability Statement:** Not Applicable, the study does not report any data.

**Conflicts of Interest:** The authors declare no conflict of interest.

### Appendix A. Reduction of EOG Artifacts over EEG

One of the main sources of noise for EEG is EOG noise. EOG noise comes from retinal dipole and eyelid movements. Both of them create a potential shift in the surface of the scalp [70]. A normal procedure to remove these artifacts is the regression procedure explained in [71], which consists of using three EOG spatial components recordings (horizontal, vertical and radial) multiplied by a specifically weighted coefficient and subtracting them from the noisy signal. For this, it is supposed that the signal without artifacts is described by:

$$S = Y - N \cdot b \tag{A1}$$

with  $S$  as the non-contaminated EEG signal,  $Y$  is the recorded EEG channel at time  $t$ ,  $N$  are the noise sources ( $n_{ver,hor,rad}$  vertical, horizontal and radial recorded EOG channels), and  $b$  are the weighted coefficients ( $b_{ver,hor,rad}$ ) of the EOG artifacts at the EEG channel. To obtain the real signal  $S$ , the noise source has to be recorded, and the weighting coefficients must be obtainable. To calculate  $b$ , it is presumed that the noise source  $N$  (i.e., EOG) and the signal  $S$  (i.e., EEG) are independent, then:

$$\langle N^T S \rangle = \langle N^T Y \rangle - \langle N^T N \rangle b \tag{A2}$$

with  $\langle N^T S \rangle = 0$  we can calculate  $b$  as:

$$b = \langle N^T N \rangle^{-1} \langle N^T Y \rangle = C_{NN}^{-1} C_{NY} \tag{A3}$$

with  $C_{NN}$  as the auto-covariance matrix of the EOG channels and  $C_{NY}$  as the cross-covariance matrix between the EEG and EOG channels. In particular, the three monopolar electrodes for EOG are mounted over the face, as shown in Figure 10, where two bipolar EOG can be derived (i.e., horizontal and vertical EOG activity). To obtain a better approximation of weighted coefficients  $b$ , as a normal procedure, at the start of each BCI session, a specific type of EOG recordings are made, where the subjects are asked to perform eye blinks, roll the eyes clockwise and counterclockwise and move the eyes upwards and downwards (Table A1). These movements are done to cover the whole field of view without moving the head.

**Table A1.** Eye movement paradigm to ensure large EOG artifacts are recorded to estimate correction coefficients.

- 
- (1) Perform idling eye movements with eyes open and close for a minute each.
  - (2) Perform repeatedly eyes blinks for over 15 s.
  - (3) Perform eye movements (rolling, left/right and up/down) for over 15 s each. These movements should circumscribe the whole field of view without moving the head.
- 

### Appendix B. Common Spatial Patterns

Common Spatial Patterns (CSP) is used to learn spatial filters for brain signal analysis and introduced by Muller [72] for movement-related EEG and later proved to be useful for imaginary hand movement by H.Ramoser [69].

CSP's goal is to design a spatial filter that finds the optimal variance for discrimination, put differently, is to apply a linear transformation that maps the input to have maximum variance between two classes. Although CSP can only be applied to a binary problem, it has already been extended [73,74] by combining various binary spatial filters, which reduce the multi-class problem into various binary decisions. The algorithm of CSP can be seen in Algorithm A1.

---

#### Algorithm A1 : CSP Pseudo-code.

---

- 1: **Input network:**
  - 2:  $X_a$  and  $X_b$
  - 3: **Train network:**
  - 4: Calculate the covariance matrix  $R_a = \frac{X_a X_a^T}{\text{trace}(X_a X_a^T)}$
  - 5: Calculate the covariance matrix  $R_b = \frac{X_b X_b^T}{\text{trace}(X_b X_b^T)}$
  - 6: Calculate  $R = R_a + R_b$
  - 7: Obtain the eigenvalues  $\lambda = [\lambda_1, \lambda_2, \dots, \lambda_n]$  and eigen vectors  $Q = [q_1, q_2, \dots, q_n]$  of  $R$
  - 8: Calculate the whitening transformation matrix  $P = \lambda^{-1/2} Q^T$
  - 9: Transform the average covariance matrices  $S_a = P R_a P^T$   $S_b = P R_b P^T$
  - 10: Using  $S_a$  and  $S_b$  obtain the generalized eigenvector matrix  $B$
  - 11: Obtain the projection matrix  $W = B^T P$
  - 12: **Output network:** Returns Matrix  $W$
  - 13: Transform the samples using the projection  $Z_a = W X_a$  and  $Z_b = W X_b$
- 

### Appendix C. Database and Code

The database can be found in <http://www.bbci.de/competition/iv/#datasets> (accessed on 28 July 2021) under the Data set 2b.

The code to this work can be found under [https://github.com/DiegoLB96/BCI\\_MDPI/tree/main](https://github.com/DiegoLB96/BCI_MDPI/tree/main) (accessed on 28 July 2021).

### References

1. Hussin, A.A. Education 4.0 made simple: Ideas for teaching. *Int. J. Educ. Lit. Stud.* **2018**, *6*, 92–98. [CrossRef]
2. Diwan, P. Is Education 4.0 an imperative for success of 4th Industrial Revolution? 2017. Available online: <https://pdiwan.medium.com/is-education-4-0-an-imperative-for-success-of-4th-industrial-revolution-50c31451e8a4> (accessed on 28 July 2021).
3. Ramirez-Mendoza, R.A.; Morales-Menendez, R.; Iqbal, H.; Parra-Saldivar, R. Engineering Education 4.0:—Proposal for a new Curricula. In Proceedings of the 2018 IEEE Global Engineering Education Conference (EDUCON), Islands, Spain, 18–20 April 2018; pp. 1273–1282.
4. Prieto, M.D.; Sobrino, Á.F.; Soto, L.R.; Romero, D.; Biosca, P.F.; Martínez, L.R. Active learning based laboratory towards engineering education 4.0. In Proceedings of the 2019 24th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA), Zaragoza, Spain, 10–13 September 2019; pp. 776–783.
5. Prestopnik, N.; Zhang, P. Human–Computer Interaction (HCI): Interactivity, Immersion, and Invisibility as New Extensions. In *Wiley Encyclopedia of Management*; Wiley: Hoboken, NJ, USA, 2015; pp. 1–6.
6. Dix, A. Human–computer interaction, foundations and new paradigms. *J. Vis. Lang. Comput.* **2017**, *42*, 122–134. [CrossRef]

7. Suarez, J.; Murphy, R.R. Hand gesture recognition with depth images: A review. In Proceedings of the 2012 IEEE RO-MAN: The 21st IEEE International Symposium on Robot and Human Interactive Communication, Paris, France, 9–13 September 2012; pp. 411–417.
8. Liu, M.; Pu, X.; Jiang, C.; Liu, T.; Huang, X.; Chen, L.; Du, C.; Sun, J.; Hu, W.; Wang, Z.L. Large-area all-textile pressure sensors for monitoring human motion and physiological signals. *Adv. Mater.* **2017**, *29*, 1703700. [[CrossRef](#)]
9. Ianez, E.; Ubeda, A.; Azorin, J.M.; Perez-Vidal, C. Assistive robot application based on an {RFID} control architecture and a wireless {EOG} interface. *Robot. Auton. Syst.* **2012**, *60*, 1069–1077. [[CrossRef](#)]
10. Ghodrat Abadi, M.; Gestson, S.L.; Brown, S.; Hurwitz, D.S. Traffic signal phasing problem-solving rationales of professional engineers developed from eye-tracking and clinical interviews. *Transp. Res. Rec.* **2019**, *2673*, 685–696. [[CrossRef](#)]
11. Ghai, W.; Singh, N. Literature review on automatic speech recognition. *Int. J. Comput. Appl.* **2012**, *41*, 42–50. [[CrossRef](#)]
12. Griol, D.; Molina, J.M.; Callejas, Z. Incorporating android conversational agents in m-learning apps. *Expert Syst.* **2017**, *34*, e12156. [[CrossRef](#)]
13. Hochberg, L.R.; Bacher, D.; Jarosiewicz, B.; Masse, N.Y.; Simeral, J.D.; Vogel, J.; Haddadin, S.; Liu, J.; Cash, S.S.; van der Smagt, P.; et al. Reach and grasp by people with tetraplegia using a neurally controlled robotic arm. *Nature* **2012**, *485*, 372–375. [[CrossRef](#)]
14. Nicolas-Alonso, L.F.; Gomez-Gil, J. Brain computer interfaces, a review. *Sensors* **2012**, *12*, 1211–1279. [[CrossRef](#)]
15. Brooks, A. An HCI Approach in Contemporary Healthcare and (Re) habilitation. *Wiley Handb. Hum. Comput. Interact.* **2018**, *2*, 923–944.
16. Mugler, E.; Ruf, C.; Halder, S.; Bensch, M.; Kubler, A. Design and Implementation of a P300-Based Brain-Computer Interface for Controlling an Internet Browser. *Neural Syst. Rehabil. Eng. IEEE Trans.* **2010**, *18*, 599–609. [[CrossRef](#)] [[PubMed](#)]
17. Jimenez-Fabian, R.; Verlinden, O. Review of control algorithms for robotic ankle systems in lower-limb orthoses, prostheses, and exoskeletons. *Med. Eng. Phys.* **2012**, *34*, 397–408. [[CrossRef](#)]
18. Lotte, F.; Faller, J.; Guger, C.; Renard, Y.; Pfurtscheller, G.; Lecuyer, A.; Leeb, R. Combining BCI with Virtual Reality: Towards New Applications and Improved BCI. In *Towards Practical Brain-Computer Interfaces*; Allison, B.Z., Dunne, S., Leeb, R., Millan, J.D.R., Nijholt, A., Eds.; Biological and Medical Physics, Biomedical Engineering; Springer: Berlin/Heidelberg, Germany, 2013; pp. 197–220.
19. Raman, R.; Grant, L.; Seo, Y.; Cvetkovic, C.; Gapinske, M.; Palasz, A.; Dabbous, H.; Kong, H.; Pinera, P.P.; Bashir, R. Damage, healing, and remodeling in optogenetic skeletal muscle bioactuators. *Adv. Healthc. Mater.* **2017**, *6*, 1700030. [[CrossRef](#)]
20. Maciejasz, P.; Eschweiler, J.; Gerlach-Hahn, K.; Jansen-Troy, A.; Leonhardt, S. A survey on robotic devices for upper limb rehabilitation. *J. Neuroeng. Rehabil.* **2014**, *11*, 1. [[CrossRef](#)]
21. Donchin, E.; Spencer, K.M.; Wijesinghe, R. The Mental Prosthesis: Assessing the speed of a P300-Based Brain-Computer Interface. *IEEE Trans. Rehabil. Eng.* **2000**, *8*, 174–179. [[CrossRef](#)] [[PubMed](#)]
22. Rao, R.P.; Scherer, R. Brain-computer interfacing: More than the sum of its parts. *IEEE Signal Process. Mag.* **2010**, *27*, 150–152.
23. Han, J.; Shao, L.; Xu, D.; Shotton, J. Enhanced computer vision with microsoft kinect sensor. A review. *IEEE Trans. Cybern.* **2013**, *43*, 1318–1334.
24. Prahm, C.; Kayali, F.; Sturma, A.; Aszmann, O. Playbionic: Game-based interventions to encourage patient engagement and performance in prosthetic motor rehabilitation. *PM&R* **2018**, *10*, 1252–1260.
25. Jacko, J.A. *Human Computer Interaction Handbook: Fundamentals, Evolving Technologies, and Emerging Applications*; CRC Press: Boca Raton, FL, USA, 2012.
26. Lotte, F. Study of Electroencephalographic Signal Processing and Classification Techniques Towards the Use of Brain-Computer Interfaces in Virtual Reality Applications. Ph.D. Thesis, Intitute National des Sciences Appliquees de Rennes, Rennes, France, 2009.
27. Holz, E.M.; Botrel, L.; Kaufmann, T.; Kübler, A. Long-Term Independent Brain-Computer Interface Home Use Improves Quality of Life of a Patient in the Locked-In State: A Case Study. *Arch. Phys. Med. Rehabil.* **2015**, *96*, S16–S26. [[CrossRef](#)]
28. Le, D.N.; Van Le, C.; Tromp, J.G.; Nguyen, G.N. *Emerging Technologies for Health and Medicine: Virtual Reality, Augmented Reality, Artificial Intelligence, Internet of Things, Robotics, Industry 4.0*; John Wiley & Sons: Hoboken, NJ, USA, 2018.
29. Kuhn, T.S.; Hawkins, D. The structure of scientific revolutions. *Am. J. Phys.* **1963**, *31*, 554–555. [[CrossRef](#)]
30. Evans, J.; Jones, R.; Karvonen, A.; Millard, L.; Wendler, J. Living labs and co-production: University campuses as platforms for sustainability science. *Curr. Opin. Environ. Sustain.* **2015**, *16*, 1–6. [[CrossRef](#)]
31. Murray, J.K.; Studer, J.A.; Daly, S.R.; McKilligan, S.; Seifert, C.M. Design by taking perspectives: How engineers explore problems. *J. Eng. Educ.* **2019**, *108*, 248–275. [[CrossRef](#)]
32. Ross, S.M. Technology infusion in K-12 classrooms: A retrospective look at three decades of challenges and advancements in research and practice. In *Educational Technology Research and Development*; Springer: Berlin/Heidelberg, Germany, 2020; pp. 1–18.
33. Higgins, K.; Huscroft-D’Angelo, J.; Crawford, L. Effects of technology in mathematics on achievement, motivation, and attitude: A meta-analysis. *J. Educ. Comput. Res.* **2019**, *57*, 283–319. [[CrossRef](#)]
34. Saxena, A. An STSE (Science, Technology, Society, Environment) Approach for Teaching Ethics in Science—Case Narrative of an Undergrad Teacher. In *Ethics in Science*; Springer: Berlin/Heidelberg, Germany, 2019; pp. 165–183.
35. Prado, A.M.; Arce, R.; Lopez, L.E.; García, J.; Pearson, A.A. Simulations versus case studies: Effectively teaching the premises of sustainable development in the classroom. *J. Bus. Ethics* **2020**, *161*, 303–327. [[CrossRef](#)]

36. Schellinger, J.; Mendenhall, A.; Alemanne, N.; Southerland, S.A.; Sampson, V.; Marty, P. Using Technology-Enhanced Inquiry-Based Instruction to Foster the Development of Elementary Students' Views on the Nature of Science. *J. Sci. Educ. Technol.* **2019**, *28*, 341–352. [[CrossRef](#)]
37. Leeb, R.; Lee, F.; Keinrath, C.; Scherer, R.; Bischof, H.; Pfurtscheller, G. Brain Computer Communication: Motivation, Aim, and Impact of Exploring a Virtual Apartment. *Neural Syst. Rehabil. Eng. IEEE Trans.* **2007**, *15*, 473–482. [[CrossRef](#)]
38. Ponce, P.; Molina, A.; Balderas, D.C.; Grammatikou, D. Brain computer interfaces for cerebral palsy. In *Cerebral Palsy-Challenges for the Future*; IntechOpen: London, UK, 2014.
39. Müller, B.B.S.L.M.T.S.H.K.R. Single-trial analysis and classification of ERP components-A tutorial. *NeuroImage* **2011**, *56*, 814–825.
40. Sutton, S.; Braren, M.; Zubin, J.; John, E. Evoked-Potential Correlates of Stimulus Uncertainty. *Science* **1965**, *150*, 1187–1188. [[CrossRef](#)] [[PubMed](#)]
41. Farwell, L.A.; Donchin, E. Talking off the top of your head: Toward a mental prosthesis utilizing event-related brain potentials. *Electroencephalogr. Clin. Neurophysiol.* **1988**, *70*, 510–523. [[CrossRef](#)]
42. Farwell, L.; Smith, S.S. Using Brain MERMER Testing to Detect Knowledge Despite Efforts to Conceal. *J. Forensic Sci.* **2001**, *135*–143. [[CrossRef](#)]
43. Farwell, L.; Richardson, D.; Richardson, G. Brain fingerprinting field studies comparing P300-MERMER and P300 brainwave responses in the detection of concealed information. *Cogn. Neurodyn.* **2012**, 1–37. [[CrossRef](#)]
44. Allison, B.Z.; McFarland, D.J.; Schalk, G.; Zheng, S.D.; Jackson, M.M.; Wolpaw, J.R. Towards an independent brain-computer interface using steady state visual evoked potentials. *Clin. Neurophysiol.* **2008**, *119*, 399–408. [[CrossRef](#)] [[PubMed](#)]
45. Pfurtscheller, G.; Lopes da Silva, F. Event-related EEG/MEG synchronization and desynchronization: basic principles. *Clin. Neurophysiol.* **1999**, *110*, 1842–1857. [[CrossRef](#)]
46. Neuper, C.; Scherer, R.; Reiner, M.; Pfurtscheller, G. Imagery of motor actions: Differential effects of kinesthetic and visual-motor mode of imagery in single-trial EEG. *Cogn. Brain Res.* **2005**, *25*, 668–677. [[CrossRef](#)]
47. Li, M.; Xu, G.; Xie, J.; Chen, C. A review: Motor rehabilitation after stroke with control based on human intent. *Proc. Inst. Mech. Eng. Part H J. Eng. Med.* **2018**, *232*, 344–360. [[CrossRef](#)]
48. Rong, Y.; Wu, X.; Zhang, Y. Classification of motor imagery electroencephalography signals using continuous small convolutional neural network. *Int. J. Imaging Syst. Technol.* **2020**, *30*, 653–659. [[CrossRef](#)]
49. Birbaumer, N.; Kubler, A.; Ghanayim, N.; Hinterberger, T.; Perelmouter, J.; Kaiser, J.; Iversen, I.; Kotchoubey, B.; Neumann, N.; Flor, H. The thought translation device (TTD) for completely paralyzed patients. *Rehabil. Eng. IEEE Trans.* **2000**, *8*, 190–193. [[CrossRef](#)]
50. Müller, S.L.B.B.T.D.K.R. Introduction to machine learning for brain imaging. *NeuroImage* **2011**, *56*, 387–399.
51. Teplan, M.; others. Fundamentals of EEG measurement. *Meas. Sci. Rev.* **2002**, *2*, 1–11.
52. Chaudhary, U.; Mrachacz-Kersting, N.; Birbaumer, N. Neuropsychological and neurophysiological aspects of brain-computer-interface (BCI) control in paralysis. *J. Physiol.* **2020**, *599*, 2351–2359. [[CrossRef](#)] [[PubMed](#)]
53. Kohonen, T. *Self-Organizing Maps*; Springer: Berlin/Heidelberg, Germany, 2001.
54. Werbos, P. Beyond Regression: New Tools for Prediction and Analysis in the Behavioral Sciences. Ph.D. Thesis, Harvard University, Cambridge, MA, USA, 1974.
55. Hinton, G.E.; Sejnowski, T.J. Optimal Perceptual Interface. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Washington, DC, USA, June 1983. Available online: [https://web.stanford.edu/~ngoodman/Hinton\\_Sejnowski\\_OptimalPerceptualDifference\\_1983.pdf](https://web.stanford.edu/~ngoodman/Hinton_Sejnowski_OptimalPerceptualDifference_1983.pdf) (accessed on 28 July 2021).
56. Hinton, G.E. A Practical Guide to Training Restricted Boltzmann Machines. 2010. Available online: <http://www.cs.toronto.edu/~hinton/absps/guideTR.pdf> (accessed on 28 July 2021).
57. Giraudel, J.; Lek, S. A comparison of self-organizing map algorithm and some conventional statistical methods for ecological community ordination. *Ecol. Model.* **2001**, *146*, 329–339. [[CrossRef](#)]
58. Kaski, S. Data Exploration Using Self-Organizing Maps. 1997. Available online: <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.30.4343> (accessed on 28 July 2021).
59. Seiffert, U. *Self-Organizing Neural Networks: Recent Advances and Applications*; Physica: Berlin/Heidelberg, Germany, 2013; Volume 78,
60. James, G.; Witten, D.; Hastie, T.; Tibshirani, R. *An Introduction to Statistical Learning*; Springer: Berlin/Heidelberg, Germany, 2013; Volume 6.
61. Fernández-Delgado, M.; Cernadas, E.; Barro, S.; Amorim, D. Do we need hundreds of classifiers to solve real world classification problems. *J. Mach. Learn. Res.* **2014**, *15*, 3133–3181.
62. Tu, J.V. Advantages and disadvantages of using artificial neural networks versus logistic regression for predicting medical outcomes. *J. Clin. Epidemiol.* **1996**, *49*, 1225–1231. [[CrossRef](#)]
63. Basheer, I.; Hajmeer, M. Artificial neural networks: Fundamentals, computing, design, and application. *J. Microbiol. Methods* **2000**, *43*, 3–31. [[CrossRef](#)]
64. Anguita, D.; Ghio, A.; Greco, N.; Oneto, L.; Ridella, S. Model selection for support vector machines: Advantages and disadvantages of the machine learning theory. In Proceedings of the 2010 International Joint Conference on Neural Networks (IJCNN), Barcelona, Spain, 18–23 July 2010, pp. 1–8.

65. Karamizadeh, S.; Abdullah, S.M.; Halimi, M.; Shayan, J.; javad Rajabi, M. Advantage and drawback of support vector machine functionality. In Proceedings of the Computer, Communications, and Control Technology (I4CT), Langkawi Island, Kedah, Malaysia, 2–4 September 2014; pp. 63–65.
66. Cawley, G.C.; Talbot, N.L. On over-fitting in model selection and subsequent selection bias in performance evaluation. *J. Mach. Learn. Res.* **2010**, *11*, 2079–2107.
67. Larochelle, H.; Bengio, Y. Classification using discriminative restricted Boltzmann machines. In Proceedings of the 25th International Conference on Machine Learning, ACM, Pittsburgh, PA, USA, 25–29 June 2008; pp. 536–543.
68. Larochelle, H.; Mandel, M.; Pascanu, R.; Bengio, Y. Learning algorithms for the classification restricted Boltzmann machine. *J. Mach. Learn. Res.* **2012**, *13*, 643–669.
69. Ramoser, H.; Müller-Gerking, J.; Pfurtscheller, G. Optimal spatial filtering of single trial EEG during imagined hand movement. *IEEE Trans. Rehabil. Eng.* **2000**, *8*, 441–446. [[CrossRef](#)] [[PubMed](#)]
70. Croft, R.; Barry, R. Removal of ocular artifact from the EEG: A review. *Neurophysiol. Clin. Neurophysiol.* **2000**, *30*, 5–19. [[CrossRef](#)]
71. Schlogl, A.; Keinrath, C.; Zimmermann, D.; Scherer, R.; Leeb, R.; Pfurtscheller, G. A fully automated correction method of {EOG} artifacts in {EEG} recordings. *Clin. Neurophysiol.* **2007**, *118*, 98–104. [[CrossRef](#)] [[PubMed](#)]
72. Müller-Gerking, J.; Pfurtscheller, G.; Flyvbjerg, H. Designing optimal spatial filters for single-trial EEG classification in a movement task. *Clin. Neurophysiol.* **1999**, *110*, 787–798. [[CrossRef](#)]
73. Dornhege, G.; Blankertz, B.; Curio, G.; Müller, K.R. Boosting bit rates in noninvasive EEG single-trial classifications by feature combination and multiclass paradigms. *IEEE Trans. Biomed. Eng.* **2004**, *51*, 993–1002. [[CrossRef](#)] [[PubMed](#)]
74. Wu, W.; Gao, X.; Gao, S. One-versus-the-rest (OVR) algorithm: An extension of common spatial patterns (CSP) algorithm to multi-class case. In Proceedings of the Engineering in Medicine and Biology Society, 27th Annual International Conference of the IEEE, Shanghai, China, 1–4 September 2005; pp. 2387–2390.