

Article

Unsupervised Event Characterization and Detection in Multichannel Signals: An EEG application

Angel Mur ^{1,*}, Raquel Dormido ¹, Jesús Vega ², Natividad Duro ¹ and Sebastian Dormido-Canto ¹

¹ Department of Computer Sciences and Automatic Control, UNED, Juan del Rosal 16, 28040 Madrid, Spain; raquel@dia.uned.es (R.D.); nduro@dia.uned.es (N.D.); sebas@dia.uned.es (S.D.-C.)

² National Fusion Laboratory by Magnetic Confinement, CIEMAT, Complutense 40, 28040 Madrid, Spain; jesus.vega@ciemat.es

* Correspondence: a.r.m.g@outlook.fr; Tel.: +34-91-398-7192

Academic Editors: Steffen Leonhardt and Daniel Teichmann

Received: 4 March 2016; Accepted: 21 April 2016; Published: 23 April 2016

Abstract: In this paper, we propose a new unsupervised method to automatically characterize and detect events in multichannel signals. This method is used to identify artifacts in electroencephalogram (EEG) recordings of brain activity. The proposed algorithm has been evaluated and compared with a supervised method. To this end an example of the performance of the algorithm to detect artifacts is shown. The results show that although both methods obtain similar classification, the proposed method allows detecting events without training data and can also be applied in signals whose events are unknown *a priori*. Furthermore, the proposed method provides an optimal window whereby an optimal detection and characterization of events is found. The detection of events can be applied in real-time.

Keywords: artifacts; EEG; event characterization; event detection; unsupervised classification

1. Introduction

Electroencephalography (EEG) is a non-invasive method to record electrical activity of the brain using sensors distributed along the scalp. Signals detected by an EEG with no connection to a specific brain activity are called artifacts. These are usually classified as physiological and non-physiological artifacts. Physiological artifacts are generated by the patient and non-physiological artifacts can arise from outside of the body (*i.e.*, equipment, environment, *etc.*).

In many cases the information that is hidden behind the physiological artifacts is relevant to a proper diagnosis. Think, for instance, about early detection of mental fatigue or in monitoring stress levels. Others health applications need EEG signals without artifacts contamination in order to reduce the misinterpretation of the EEG and limit the potential for adverse clinical consequences. In this case brain computer interfaces need to filter artifacts in real time [1,2]. Consequently, it is necessary, especially for long recordings, to develop methods to detect and correctly identify artifacts either for analyzing or removing them. This is the main goal of this paper.

There are some approaches for detecting and removing artifacts using EEG recordings. Statistical methods as SCADS (Statistical Control of Artifacts in Dense Array Studies) [3] and FASTER (Fully Automated Statistical Thresholding for EEG Artifact Rejection) [4] detect and remove artifacts to analyze event-related potentials. The packages EEGLAB [5] and FieldTrip [6] include some routines to detect EEG artifacts but some threshold values need to be defined. Reference [7] is a practical example of how threshold values are used for detection of EEG artifacts in polysomnographic recordings.

DETECT [8] is a toolbox for detecting and identifying artifacts based on the supervised classifier support vector machine (SVM). Note that other supervised methods do not carry out an identification process [9–12]. An advantage of DETECT over other methods is that the user is not required to

manually define threshold values (with the exception of a certainty value applied to the classification outcome to reduce false positives). Moreover it captures the frequency and duration of artifacts. DETECT only finds artifacts included in the training process and the quality of this detection depends on the quality of the training data.

In [13], the authors present a practical example of how a SVM classifier is used to detect artifacts arising from head-movements. Reference [14] describes a novel application of one-class support vector machine novelty detection for detecting seizures in humans.

In this paper we first propose a new, simple, and effective unsupervised method (UMED) to characterize and detect events in multichannel signals. Then the proposed method is applied to detect artifacts in EEG recordings. We use the term “artifact detection” interchangeably with the term “event detection” although artifact is more closely associated with the EEG recordings.

Some advantages of the proposed method are the following: (i) it does not need to manually specify any threshold value; (ii) it allows the measurement of the frequency and duration of the events; (iii) as an unsupervised method it does not need training data; (iv) it can be applied to signals with *a priori* unknown events; (v) it provides an optimal detection technique to characterize the events by finding an optimal window; (vi) it can be used in real-time; and (vii) it provides the number of independent components (NIC) required to remove artifacts by means of an independent component analysis (ICA) [2,15,16].

This paper is organized as follows. Section 2 presents some terms and theoretical background used to implement UMED. The proposed unsupervised method to characterize and detect events in multichannel signals is presented in Section 3. In Section 4, we test this method using an EEG recording with artifacts. Finally in Sections 5 and 6 a discussion and conclusions of the paper are respectively presented.

2. Background

2.1. Events and Multi-Channel Signals

In this section, we formally define the terms “event” and “multichannel signal”. Both are linked to the concept of temporal sequence. Temporal sequence refers to a sequence of happenings or events in a time interval. An event is something that happens at a time t_E and it reflects some kind of change in a temporal evolution. An important characteristic of temporal sequences is the order in which events take place. Another important characteristic is the duration d_E between t_E and the next event.

A temporal sequence is normally represented by a series of nominal symbols from a particular alphabet. Every event is characterized by a symbol. In this way, an event with symbol E is described by two elements (E, t_E) . The three elements (E, t_E, d_E) represent a state. A temporal sequence P made of u events E^i for $i = 1, \dots, u$ can be described by a u -tuple of states $P = \langle (E^1, t_E^1, d_E^1), (E^2, t_E^2, d_E^2), \dots, (E^u, t_E^u, d_E^u) \rangle$.

In general, the number and temporal locations of events can be unknown within a temporal sequence. This means that events have to be recognized and located inside the sequence. It occurs, for example, in the monitoring of a physical system with Q sensors for a period of time $[T_1, T_2]$. In this case, every sensor represents a channel and simultaneously provides a signal describing part of the whole information. Signals of this type that are generated by multiple sensors are called multichannel (MC) signals. In other words, a MC signal refers to a set of signals that show cross-channel similarity or correlation. A MC signal with Q channels is represented by a vector $X(t) = [CH_1(t) CH_2(t), \dots, CH_Q(t)]$ where $CH_q(t)$ is the signal of the channel q for $q = 1, 2, \dots, Q$. A MC signal $X(t)$ with events can also be described, in compact form, as a temporal sequence P .

2.2. The Hierarchical Clustering, the Cophenetic Correlation Coefficient, and the S_Dbw Validity Index

In this section we briefly describe three concepts used in the implementation of the proposed algorithm: the hierarchical clustering (HC) algorithm [17], the Cophenetic Correlation Coefficient (CCC) [18], and the S_Dbw Validity Index [19].

The HC groups data over a variety of scales by creating a cluster tree or dendrogram. It follows this general procedure: (1) find the similarity or dissimilarity between every pair of objects in the data set; (2) group the objects into a binary, hierarchical cluster tree (linkage); (3) determine where to cut the hierarchical tree into clusters.

In a dendrogram, the cophenetic distance between two objects is represented by the height of the link at which those two objects are first joined. That height is the distance between the two subclusters that are merged by that link. The CCC for a cluster tree is defined as the linear correlation coefficient between the cophenetic distances obtained from the tree, and the similarities (or dissimilarities) used to construct the tree. The CCC is a measure of how faithfully a dendrogram maintains the original pairwise distances. The magnitude of CCC should be very close to 1 to achieve a high-quality solution of the HC algorithm.

Given a dataset, if $x(O_i, O_j)$ is the distance between the i th and j th object, $t(O_i, O_j)$ is their cophenetic distance, \bar{x} is the average of the $x(O_i, O_j)$ and \bar{t} is the average of the $t(O_i, O_j)$, then the CCC is given by Equation (1):

$$CCC = \frac{\sum_{i < j} (x(O_i, O_j) - \bar{x}) (t(O_i, O_j) - \bar{t})}{\left(\left[\sum_{i < j} (x(O_i, O_j) - \bar{x})^2 \right] \left[\sum_{i < j} (t(O_i, O_j) - \bar{t})^2 \right] \right)^{1/2}} \quad (1)$$

The S_Dbw validity index is used for measuring “goodness” of a clustering result. Its definition is based on cluster compactness and separation but it also takes into consideration the density of the clusters. Lower index value indicates better clustering schema.

Given a set of k clusters G_i ($i = 1, \dots, k$) of the dataset DS , the S_Dbw index is defined in the following way:

$$S_Dbw(k) = Scat(k) + Dens_bw(k) \quad (2)$$

where $Scat$ is the intra-cluster variance that measures the average scattering of clusters and it is described by:

$$Scat(k) = \frac{1}{k} \sum_{i=1}^k \frac{\|\sigma(G_i)\|}{\|\sigma(DS)\|}$$

$\sigma(DS)$ is the variance of the dataset DS and $\sigma(G_i)$ is the variance of the cluster G_i .

The inter-cluster density $Dens_bw$ is defined as follows:

$$Densbw(k) = \frac{1}{k(k-1)} \sum_{i=1}^k \left(\sum_{j=1, j \neq i}^k \frac{density(G_i \cup G_j)}{\max[density(G_i), density(G_j)]} \right)$$

where $density(G) = \sum_{i=1}^{|G|} f(x_i, \mu)$, μ is the center of the cluster G , $|G|$ is the number of objects in G , and the function $f(x_i, \mu)$ is defined by:

$$f(x, \mu) = \begin{cases} 0 & \text{if } distance(x, \mu) > \frac{1}{k} \left(\sum_{i=1}^k \|\sigma(G_i)\| \right)^{1/2} \\ 1 & \text{otherwise} \end{cases}$$

If $G = G_i \cup G_j$, then μ is the middle point of the line segment that is defined by the μ_i and μ_j clusters centers.

3. Method for the Characterization and Detection of Events in Multichannel Signals

In this section, we propose an unsupervised method (UMED) to characterize and detect the events of a MC signal of duration $[T_1, T_2]$ whose events are a priori unknown. A small temporal window slides along the interval $[T_1, T_2]$. For each window displacement, it is obtained a feature vector of S variables which picks up the behavior of the MC signal. A feature vector unsupervised classification is carried out, which allows automatic characterization and detection of the events. The quality of this characterization (and the number of events detected) depends significantly on the size of the sliding window. This is why the algorithm searches for an optimal window size that provides the maximum compactness and separation of the content between the events.

The UMED detects and characterizes events by means of an optimal unsupervised classification. The S_dbw index and the HC algorithm are used to find the best window along with its unsupervised classification characterized by its optimal number of clusters.

Given a MC signal X for a period of time $[T_1, T_2]$ with unknown events to be characterized and detected, the steps of UMED can be summarized as follows:

Step 1: Specification of windows. The algorithm uses a sliding window approach. Figure 1 shows the details of the sliding window in a MC signal X along the interval $[T_1, T_2]$. If the window W_i of size L_w is defined by the interval $[t_1, t_2]$ then the window W_{i+1} is defined by $[t_1 + d, t_2 + d]$ with $d > 0$. As first step, a set of IN windows is selected for a specific L_w and d .

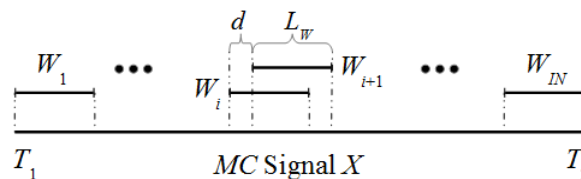


Figure 1. Different windows along the MC signal X .

Step 2: Feature matrix (FV) calculation. In this step a matrix FV of IN rows is generated. Each row i of FV represents a Q -feature vector of the window W_i . So first we calculate for each channel q and window W_i a feature vector of size s . Then, these features vectors are concatenated to form a unique Q -feature vector Fv_i (with $S = Q \times s$ features) for each W_i . At last, all the feature vectors Fv_i (for $i = 1, \dots, IN$) are saved in a matrix FV of size $IN \times S$ concatenating them vertically.

Two autoregressive coefficients are good candidates to form the feature vector for each channel q and window W . The coefficients a_1 and a_2 of an autoregressive model $AR(2)$ of a window $W(r)$ for a single channel can be written as:

$$W(r) = \sum_{r=1}^2 a_r W(r-i) + e(r) \quad (3)$$

where $e(r)$ is zero-mean white noise. The a_i coefficients can be estimated using Burg's method [20]. Normally the Q -feature vector with a $AR(2)$ (of length $Q \times 2$) offers superior or equal performance to detect artifacts in EEG recordings than a $AR(p)$ with $p > 2$ [8,21], and it is computationally less intensive.

Step 3: Selection of a window size L_w using the CCC. In this step, first the CCC of the IN rows of the matrix FV is calculated. If CCC reaches approximately a particular threshold U , it is selected the value of L_w . This value is named L_w^0 and its feature matrix FV^0 . In case CCC is much lower than U , we go back to the step 1 to choose a larger L_w , and then a new FV and CCC are tested.

The CCC threshold U is defined as 0.85. This value comes from experience and guarantees that the HC algorithm will find a significant classification. From a practical point of view, this threshold

is used to reduce the grid search for the window length in step 4. For small window sizes (some dozens of samples) the CCC value can be <0.85 . In this case the CCC is used to determine a minimum window size from which the step 4 will start. In step 4, the S_dbw takes the responsibility to calculate an optimum window size. A low value of S_dbw comes with a high CCC value. However, the CCC is not defined by compactness and separation of groups and it cannot be used to determine an optimal group number.

For each window size the processing time of CCC is low. So a possible strategy to determine L_w^0 is to select a small group of window sizes and check if the $CCC > 0.85$ is respected from a particular window size.

Step 4: Selection of an optimal window size (L_w^H) and its unsupervised classification. In this step, first the size of the window is increased N_L times by means of $L_w^m = L_w^0 + m \times D_L$ for $m = 0, \dots, N_L$ and the integer $D_L \geq 1$. Second, for each L_w^m , it is calculated its feature matrix FV^m . Third, the HC algorithm is applied N_G times to classify the rows of each matrix FV^m in different partitions *i.e.*, in 2, 3, ..., $N_G + 1$ groups. Fourth, for each partition, its $S_Dbw_g^m$ index (with $g = 2, \dots, N_G + 1$) is calculated. The minimum value of the index, named Min^m , determines the optimal number of groups No^m . The minimum value Min^H between the Min 's determines the optimal window size L_w^H , the optimal feature matrix FV^H and the optimal number of groups No^H .

The selected N_L value allows to check a significant number of window sizes. The MC signal X has an initial period of time (IPT) where there are no events. If IPT is known, then L_w^{NL} has to be $<IPT$. If IPT is not known, then it is possible to add at the start of X a synthetic portion of signal without events. When the size L_w^H is near to L_w^{NL} a bigger value of N_L is chosen (with the help of a synthetic IPT, if necessary) to ensure that L_w^H is well surrounded by other suboptimal sizes.

The selected N_G value allows to check a significant number of partitions. In case the optimal number No^m is near to N_G , a bigger value of N_G is chosen to ensure that No^m is well surrounded by other suboptimal group numbers.

Step 5: Event Detection by means of the optimal unsupervised classification. It is accomplished the unsupervised classification for the predetermined number of groups No^H . The condition $CCC > U$ and the selection of Min^H allow the HC algorithm to find a high-quality classification with the best group compactness and separation taking into consideration the density of the clusters.

Once all the windows of size L_w^H have been classified into No^H clusters, the different events can be detected along the MC signal X . After this classification, we will use, preferably, the term interval instead of the window to stress the window time interval.

The events are characterized by the group numbers of the optimal classification. These are placed at the beginning and the end of each group of consecutive intervals $[W_i, W_{i+1}, \dots, W_j]$ where $(1 \leq i \leq j \leq IN)$ with the same group number. The beginning and the end can be defined using the next formula [8]:

$$\left[(i-1) \times d + M \times L_w - \frac{d}{2}, (j-1) \times d + M \times L_w + \frac{d}{2} \right] \quad (4)$$

where i is the number of the first intervals W_i , j is the number of the last interval W_j , d is the slide width, and $M = 6/7$. The product $L_w \times 6/7$ is more accurate than using the midpoint with $M = 1/2$. Equation (4) uses only d samples per interval along the signal without overlapping. In this way, intervals between events are uniquely labeled regardless of whether events are placed in the intersection of two overlapping windows classified in a different way. Equation (4) is used to plot the events along the EEG signal. It finds the events' temporal locations along the EEG signal and, thus, it is possible to study their behavior.

4. Testing the Algorithm

4.1. EEG multichannel Signal with Artifacts

The proposed algorithm has been tested to detect artifacts in multichannel EEG signals. The test signal is an EEG recording of 8 min duration used in [8] to highlight the utility of the DETECT toolbox. The data are sampled at 256 Hz using a 64-channel Biosemi Active Two System (Amsterdam, Netherlands). Figure 2 shows the channel $CH_1(t)$ of the EEG recording. Our test uses all of the channels.

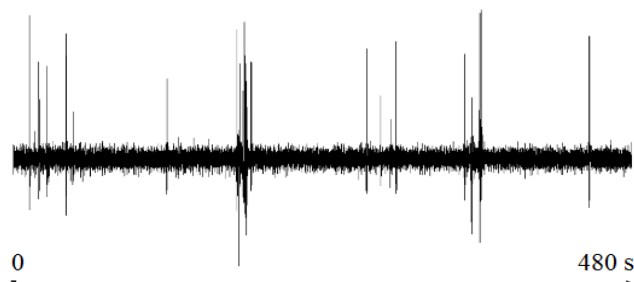


Figure 2. The channel $CH_1(t)$ of the 64-channel EEG.

The artifacts are categorized into six classes [8]: None (NN), Jaw Clench (JC), Jaw Movement (JM), Eye Blink (EB), Eye Left Movement (ELM), and Eye Up Movement (EUM). In reference [8] a SVM approach is used for detection of artifacts. This supervised method (SM) is trained with a balanced set of 20 trials for each type of artifact along with a window of $L_w = 128$ samples and a slide width of $d = 32$ samples. Each window of a channel is characterized by the first two autoregressive coefficients [21]. A number of 3837 windows (or intervals) are analyzed and classified. Furthermore, in [8] a certainty thresholding policy is applied to this classification to remove false positives in the data. The Table 1 summarizes the SM outcome after using a value of 0.5 to threshold the certainty.

Table 1. Number of intervals per group using the SM. The groups are None (NN), Jaw Clench (JC), Jaw Movement (JM), Eye Blink (EB), Eye Left Movement (ELM), and Eye Up Movement (EUM).

Group	Number of Intervals
NN	3715
ELM	12
EUM	21
EB	81
JM	5
JC	3

4.2. Detection and Characterization of Events Using EEG Recordings

In this section, we apply the proposed UMED method presented in Section 3 to detect and characterize the artifacts in the EEG recording described above. A comparison with the results summarized in Table 1 is also given. Both approaches can be compared because they use the same AR feature vectors and a similar sliding window approach.

The first two autoregressive coefficients have been used as feature vector per window and channel. For $L_w = 128$ samples, the CCC is 0.8510 and $N_o = 7$. The Table 2 shows (for the step 4) the CCC values, the Min values and the optimal number of clusters for different $L_w > 128$ samples starting at $L_w = 135$ and $D_L = 5$.

Table 2. Characteristic parameters for the selection of the optimal window (step 4): The window size L_w , the optimal number of clusters No , the Cophenetic Correlation Coefficient CCC, and the minimum value of the S_Dbw index Min .

L_w	135	140	145	150	155	160	165
CCC	0.8413	0.8564	0.8473	0.8475	0.8658	0.8616	0.8679
No	7	7	7	7	7	7	6
Min	0.97	0.80	0.85	0.90	0.77	0.82	1.05

From the data presented in Table 2 it is easy to find the minimum value of the index (Min) among the minimums, $Min^H = 0.77$; the optimal number of clusters $No^H = 7$ for $L_w^H = 155$ samples, and $CCC^H = 0.8658$. Then, an unsupervised classification (HC algorithm), as stated in step 5, allows finding of the optimal content of the seven groups.

We use a principal component analysis (PCA) [22] to display the clusters found on a 2D representation. PCA is a projection method that re-expresses a collection of correlated variables into a smaller number of variables called principal components, which maximizes the variance of the data.

If the original data formed by n_v vectors of l features are stored in a matrix $O_D(l \times n_v)$, then the PCA transform is:

$$T_D = A(O_D - m_{OD}) \quad (5)$$

where m_{OD} is the mean of the original data and A is a matrix whose rows are the eigenvectors of the covariance matrix $C_{OD} = (1/n_v)O_D O_D^T$.

Our PCA is carried out using the Equation (5) and $O_D = (FV^H)^T$. Then, it is selected the submatrix T_D^2 made up of the first two principal components (the first two rows of T_D). The column k of T_D^2 provides a new feature vector of only two elements to the window W_k .

Figure 3 shows the seven optimal clusters found. The interval z is represented by a point whose coordinates are $T_D^2(1, z)$ and $T_D^2(2, z)$. Each cluster is distinguished by a specific shape and color. Table 3 summarizes the UMED outcome for $L_w^H = 155$.

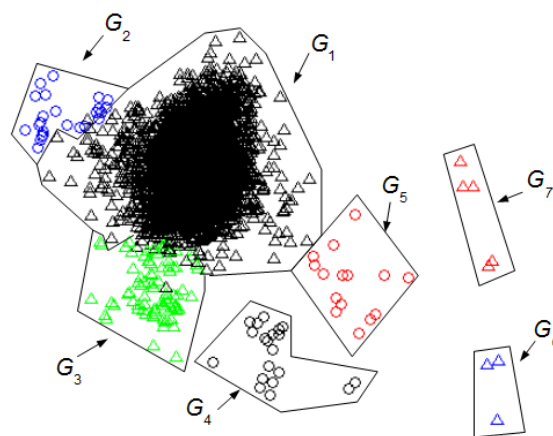


Figure 3. Groups of intervals found using the unsupervised classification (UMED) and a PCA. The L_w^H has 155 samples. The first two principal components contain 55% of the full information. The different clusters are characterized but not identified.

Table 3. Number of intervals per group using the UMED with $L_w^H = 155$ samples.

Group	Number of Intervals
G_1	3691
G_2	24
G_3	77
G_4	20
G_5	15
G_6	3
G_7	5

Figure 4 shows the seven clusters found using our UMED for $L_w = 128$. Table 4 summarizes the UMED outcome. This result is important to compare UMED with SM.

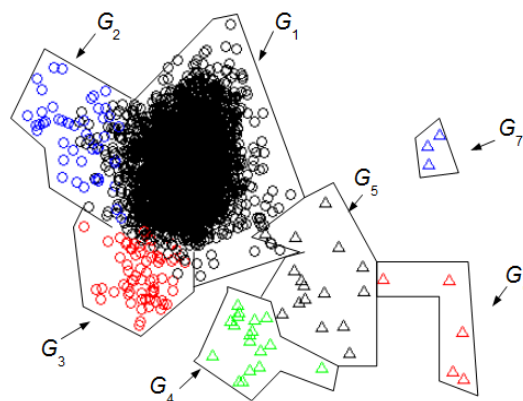


Figure 4. Groups of intervals found using the unsupervised classification (UMED) and a PCA. The L_w has 128 samples. The first two principal components contain 53% of the full information. The different clusters are characterized but not identified.

Table 4. Number of intervals per group using the UMED with $L_w = 128$ samples.

Group	Number of Intervals
G_1	3680
G_2	46
G_3	67
G_4	18
G_5	16
G_6	5
G_7	3

Figure 5 shows the 6 clusters found with DETECT [8] using a supervised classification (SM) for $L_w = 128$ and a value of 0.5 to threshold the certainty. The principal components are the same as those of Figure 4. Table 1 summarizes the SM outcome where the NN intervals are in G_{1+2} , the EB in G_3 , the EUM in G_4 , the ELM in G_5 , the JM in G_6 , and the JC in G_7 .

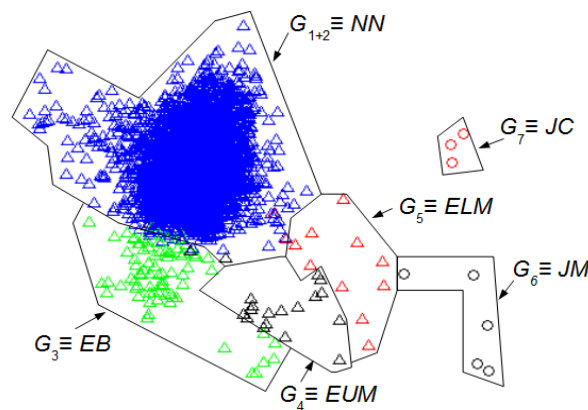


Figure 5. Groups of intervals found using the supervised classification (SM) [8] and the first two principal components of the Figure 4. The L_w has 128 samples. The first two principal components contain 53% of the full information. The different clusters are identified. Using the Table 1: NN intervals are in G_{1+2} , the EB in G_3 , the EUM in G_4 , the ELM in G_5 , the JM in G_6 , and the JC in G_7 .

Figures 4 and 5 allow the comparison of the UMED and the SM. The UMED finds a group G_2 that is not detected when using SM because the intervals of this group do not form part of the required training process in the SM. Consequently, these intervals are integrated into the G_{1+2} group that contain the NN intervals. It is observed that SM obtains groups more blurred than the UMED but, in general, the detection of intervals is similar. The only exception are around seven intervals between group G_3 and G_4 , and around six intervals between groups G_4 and G_5 . Moreover, the Normalized Mutual Information (NMI) index [23] allows to compare the intervals' group numbers for both methods. If, for the classification of UMED, the G_2 is included into the G_1 , the NMI between the UMED and SM result is 0.8317. This score is highly consistent since UMED and SM find a similar classification of intervals. A value of NMI close to 1 means that the results of both methods are similar.

The groups G_6 and G_7 have events that are consecutive in time. Figure 6 shows the effect on the classification of these groups using a $L_w^H = 155$, $L_w = 128$ and $L_w = 165$ (see also Table 2). By using Equation (4) the events can be temporally located along the EEG signal and, thus, it is possible to study their behavior.

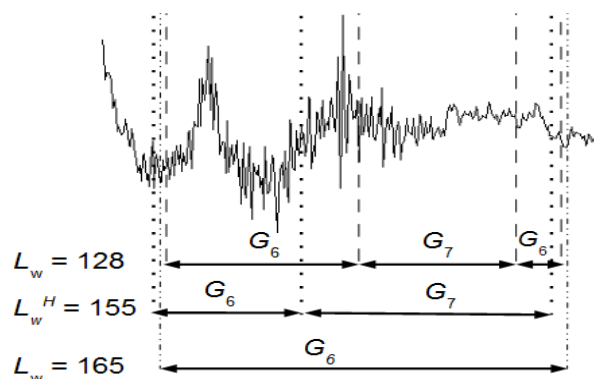


Figure 6. Events detected between the samples 46,389 and 46,653 using a $L_w^H = 155$, $L_w = 128$ and $L_w = 165$. The signal is a portion of an EEG channel.

The optimal L_w^H provides two groups G_6 (intervals 1447, 1448, 1449) and G_7 (intervals 1450, 1451, 1452, 1453, 1454) well separated and these groups are mathematically more significant than the $L_w = 128$ ones (intervals 1448, 1449, 1450, 1451, 1455 for G_6 and 1452, 1453, 1454 for G_7). The G_6 of L_w^H has one artifact and the one of $L_w = 128$ two artifacts. The $L_w = 165$ yields one group G_6 and this is why, in Table 2, its No is 6.

Figure 6 shows, graphically, how the quality of the results depends on the selected sliding window size.

5. Discussion

Artifacts can be either analyzed or filtered from an EEG recording. The UMED, as the SM approach, has been mainly developed to analyze artifacts, to study for instance the frequency and the duration of artifacts. The UMED also provides, in a unsupervised way, the number of different artifacts. For instance, in Figure 6 and for L_w^H , the number of artifacts is 2: JM and JC. The durations of JM and JC are 0.37 s and 0.62 s, respectively. Both appear once during the entire EEG recording.

In Section 4 the proposed UMED has been evaluated by comparing its classification result with the one obtained with the SM using the same L_w, d and EEG recording. It has been used with the PCA and the NMI to contrast, visually and quantitatively, both classifications. This evaluation is significant because, in addition to the fact that two different methods UMED and SM obtain similar results, the SM method was validated comparing “user labelings” with “automated labelings”. The SM method was validated by measuring the agreement (in seconds) between expert and SM labeling in some EEG recordings. This agreement is, in the worst case, over 80%. The best case reaches about 98%. In this way, we have checked that UMED with the same L_w, d , and EEG recording provides a similar result to the one given by an expert. Therefore, UMED could also be considered as an unsupervised expert system.

Unlike SM, UMED determines an optimal window size L_w^H which allows an optimal classification based on the S_Dbw index. With this index and the HC algorithm, UMED characterizes clusters that have different density and can provided new clusters that SM could not have considered during the training process. The HC computes the distance between two data points and the distance between two clusters (linkage) using, respectively, the Euclidean and the average distance (the average distance between each element in one cluster to every element in the other cluster).

The quality of the SM outcome depends on the quality of the training process but, as UMED, it also depends on the size of the sliding window selected. As said in [8]: “If the window size is too small, then the window may not contain sufficient information to model the feature. On the other hand, if the window is too large, SM will not be able to distinguish closely-spaced events, and temporal precision regarding the exact onset of an event is diminished”. This is also true for UMED. Unlike UMED, SM has no mathematical criteria to decide the sliding window size, except to check different sizes and compare the results with the opinion of an expert. This process seems slower and more tedious than the one of UMED.

We can compare the Figures 3 and 4 with Figure 5 to identify the clusters obtained in Figures 3 and 4. In this case, it is not necessary for an expert to identify the clusters found using UMED. In Figure 4, by means of UMED, the groups G_3 and G_4 are clearly distinguished. On the contrary, in Figure 5, SM does not differentiate well these groups because some Eye Up Movement (EUM) are considered as Eye Blink (EB). Some EUM do not follow the PCA representation (the lowest green triangles are in G_3 instead of G_4). Possibly, a better quality of the trials (used for the training process) would have improved the quality of the SM result. In Figure 3, due to the use of L_w^H , the compactness and separation of the clusters are better than in Figures 4 and 5. Likewise, in Figure 6, the intervals of Jaw Movement G_6 and Jaw Clench G_7 for L_w^H are better defined.

The SM can be used in real time to detect artifacts in an EEG recording. The classifier has been trained with a balanced set of trials for each type of artifact. Both, the number of artifacts and their trials have to be selected by an expert. Their size L_w is the same for all. The SM classifier uses a sliding window with the same size L_w . In real time, each new interval is classified during the acquisition of the next interval.

The UMED can also work in real time primarily in two ways. The first one (UMED₁) acts similarly to SM but now the training process of the SVM is different. The UMED automatically detects the number of artifacts and their trials from a significant EEG recording using an optimal window size

L_w^H . Then, an expert only needs to identify one portion of consecutive intervals of each cluster and select a significant number of intervals per cluster. This number has to be adequate to get a balanced set of trials. Once the SVM has been trained with these trials (the result of UMED), it is ready to work in real-time. Similarly to SM, each new interval (from other EEG recording) is classified during the acquisition of the next interval.

The second way (UMED₂) consists on using UMED as a supervised method without using a SVM. In this case the first step is to get a reference classification from a significant EEG recording similarly to UMED₁. Unlike UMED₁ that selects a set of intervals for a training process, UMED₂ characterizes a dataset DS_2 using the feature vectors of all the intervals (of size L_w^H). The UMED₂ selects DS_2 along with its optimal number of clusters No^H and its S_Dbw index named $IndN_o$. An expert only needs to identify one portion of consecutive intervals of each cluster.

Then, UMED₂ is ready to classify new sets of $Nint$ intervals from other EEG recording. Each set of intervals is added to DS_2 . The optimal number of clusters is calculated using the S_Dbw index and then it is compared the S_Dbw index for No^H (named $IndN_o$) and the one for $No^H + 1$ (named $IndN_1$). If the value $IndN_o < IndN_1$ the solution is the unsupervised classification using HC for No^H groups. If $IndN_o > IndN_1$ it is compared $IndN_1$ with $IndN_2$ and so on until $IndN_i < IndN_{i+1}$. In general, the solution is the classification using HC for No_i^H groups when $IndN_i < IndN_{i+1}$.

Usually with a significant DS_2 (containing all the types of artifacts) each of the $Nint$ intervals belongs to one of the No^H groups and $IndN_o < IndN_1$. The $Nint$ intervals are classified at the same time and, after that, their feature vectors are removed from the reference dataset. However, if any of the $Nint$ intervals are classified in a new cluster, then it can remain in the dataset. The new cluster will be a new type of artifact, but it is not identified. This is similar to a SVM that only identifies artifacts included in the training process. In real-time, every $Nint$ intervals are classified during the acquisition of the next $Nint$ intervals.

The SM method identifies the artifacts. The UMED method characterizes and detects the artifacts. The identification process is also possible by means of UMED₁ and UMED₂. Thanks to UMED the training process in UMED₁ and UMED₂ is simpler and faster than the one of SM. It is understood that UMED₁ and UMED₂ can also be used for offline applications.

The UMED does not reduce the feature vectors dimensionality by using for example PCA. The PCA is only used to visually contrast the SM and UMED outcomes. In addition to the loss of information, the use of PCA would be a disadvantage for UMED₁ and UMED₂.

Some real-time EEG applications use a small number of electrodes. The use of UMED is not limited by the number of channels. With a lower number, the computational effort decreases, though the quality of the result can change. Usually, with fewer channels, discrimination between artifacts decreases so different artifacts could be classified into the same group. If the goal is the correct identification of the artifacts, then, this rule applies: the more channels, the better the discrimination of artifacts.

Artifacts are superimposed on the real EEG signal. Once they have been detected, they can be directly removed by cutting the segments of the signal where there are overlaps. In this case there is a loss of information because those segments include a part of the real EEG signal. The UMED (that detects artifacts and the portions of signals without artifacts (PWA)), can also be used along with other algorithms, such as ICA [15,16], specialized to filter artifacts leaving the full EEG signal with useful information. The PWA can lose information after using ICA. By replacing the PWA after using ICA with the original PWA detected using UMED ensures the maximum of useful information along with the EEG signal without artifacts. UMED also provides the NIC used in an ICA that corresponds to the number of clusters. For the full EEG signal of Figure 1, $NIC = 7$. For a specific EEG segment, this number will depend on the artifacts detected by UMED in that portion of signal. In Figure 6, $NIC = 3$.

There is a large number of recent research papers that present artifact de-noising of EEG signals. Reference [24] presents an algorithm for removing peak and spike noise from EEG. This is based on filtering and thresholding the analytic signal envelope. Reference [25] presents an unsupervised

algorithm that uses modified multiscale sample entropy and Kurtosis to automatically identify the independent eye blink artefactual components and, subsequently, denoise these components using biorthogonal wavelet decomposition. This method neither requires manual identification for artefactual components nor an additional electrooculographic channel. The method *FORCe* [26] is an artifact removal method developed for use in brain-computer interfacing. It is based upon a combination of wavelet decomposition, independent component analysis, and thresholding. The method LAMIC [27] does not outperform *FORCe* but it is interesting because the artifact removal from the EEG has been performed by means of a clustering algorithm. The EEMD-ICA approach [28] removes artifacts preserving more information than other classical ICA methods.

The UMED has been used to study physiological artifacts without the presence of electrical interferences. Modern acquisition systems allow for reducing that kind of interference. However, for a specific application, it could be interesting to check the UMED behavior in the presence of non-physiological artifacts. Usually, the UMED will create new clusters if the electrical artifacts are strong and located in specific portions of the signal. In case the electrical interference is weak along the full EEG signal, there would be no new clusters related to that non-physiological artifact and the UMED performance would not change. When the electrical interferences are not the subject of study, the best choice is to avoid them or filter them during the acquisition process. That is imperative when the interference is strong and distributed continuously along the full EEG signal.

The UMED method has been presented using an EEG recording. The artifacts are very well-differentiated from the cognitive EEG part without artifacts. This cognitive part is only related to the thinking and its high variability is concentrated in a unique cluster that is the biggest. This cluster is not affected using different subjects since the *AR* coefficients are scale- and location-invariant [8,21]. Normally, there is also a unique cluster per artifact from multiple subjects. It may happen that a type of artifact, from a subject with very different morphology, could be grouped in a different cluster. In this case, we could find more than a cluster per artifact, although the distinction between the different artifacts remains. This possibility can diminish as the number of subjects increases.

The *AR*(2) coefficients are good candidates as feature vectors because they allow differentiation between artifacts in the EEG signal and are also scale- and location-invariant. Furthermore, with *AR*(2), the cognitive part without artifacts is well differentiated and is grouped in a single cluster.

The *AR* coefficients are useful to detect artifacts but they cannot detect cognitive events. In theory, once an EEG recording has been cleaned of artifacts, the UMED could be used along with other type of feature vectors that contain spatio-temporal information to detect cognitive events. The UMED outcome for these feature vectors could be evaluated by using a non-invasive optical technique as Near-infrared spectroscopy (NIRS) [29–32].

6. Conclusions

In this paper a new unsupervised method of classification of events in multichannel signals from a sensor system has been presented (UMED). This classification allows characterization and detection of the events. In particular, UMED has been applied to detect artifacts in EEG *MC* signals. It has also been evaluated and compared with a supervised method (SM) to detect artifacts.

The UMED allows to find an optimal classification of the events. This classification can be applied in real-time, either training a SVM classifier or operating as a supervised method. The UMED has been mainly developed to analyze artifacts. For example, to capture the frequency and duration of artifacts for psycho-physiological studies. The UMED also provides, in an unsupervised way, the *NIC* required to remove artifacts with the help of ICA.

Acknowledgments: This work was supported in part by the Spanish Ministry of Economy and Competitiveness under Project DPI2011-27818-C02-02, DPI2014-55932-C2-2-R, ENE2015-64914-C3-1-R, ENE2015-64914-C3-2-R and FEDER funds.

Author Contributions: A.M. researched the literatures, designed the algorithm, analyzed the results and wrote the manuscript; R.D. researched the literatures, wrote the paper, supervised the overall study; J.V., N.D and S.D.-C. participated in revising the manuscript. All authors have approved the final manuscript.

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

References

1. Lance, B.J.; Kerick, S.E.; Ries, A.J.; Oie, K.S.; McDowell, K. Brain-computer interface technologies in the coming decades. *Proc. IEEE* **2012**, *100*, 1585–1599. [[CrossRef](#)]
2. James, C.J.; Gibson, O.J. Temporally constrained ICA: An application to artifact rejection in electromagnetic brain signal analysis. *IEEE Trans. Biomed. Eng.* **2003**, *50*, 1108–1116. [[CrossRef](#)] [[PubMed](#)]
3. Junghöfer, M.; Elbert, T.; Tucker, D.M.; Rockstroh, B. Statistical control of artifacts in dense array EEG/MEG studies. *Psychophysiology* **2000**, *37*, 523–532. [[CrossRef](#)] [[PubMed](#)]
4. Nolan, H.; Whelan, R.; Reilly, R.B. FASTER: Fully automated statistical thresholding for EEG artifact rejection. *J. Neurosci. Methods* **2010**, *192*, 152–162. [[CrossRef](#)] [[PubMed](#)]
5. Delorme, A.; Makeig, S. EEGLAB: An open source toolbox for analysis of single-trial EEG dynamics including independent component analysis. *J. Neurosci. Methods* **2004**, *134*, 9–21. [[CrossRef](#)] [[PubMed](#)]
6. Oostenveld, R.; Fries, P.; Maris, E.; Schoffelen, J.-M. FieldTrip: Open source software for advanced analysis of MEG, EEG, and invasive electrophysiological data. *Comput. Intell. Neurosci.* **2011**, *2011*, 156869. [[CrossRef](#)] [[PubMed](#)]
7. Durka, P.J.; Klekowicz, H.; Blinowska, K.J.; Szelenberger, W.; Niemcewicz, S. A simple system for detection of eeg artifacts in polysomnographic recordings. *IEEE Trans. Biomed. Eng.* **2003**, *50*, 526–528. [[CrossRef](#)] [[PubMed](#)]
8. Lawhern, V.; Hairston, W.D.; Robbins, K. DETECT: A MATLAB toolbox for event detection and identification in time series, with applications to artifact detection in EEG signals. *PLoS ONE* **2013**, *8*, e62944. [[CrossRef](#)] [[PubMed](#)]
9. Hido, S.; Tsuboi, Y.; Kashima, H.; Sugiyama, M.; Kanamori, T. Statistical outlier detection using direct density ratio estimation. *Knowl. Inf. Syst.* **2010**, *26*, 309–336. [[CrossRef](#)]
10. Camci, F.; Chinnam, R.B. General support vector representation machine for one-class classification of non-stationary classes. *Pattern Recognit.* **2008**, *41*, 3021–3034. [[CrossRef](#)]
11. Sadik, S.; Gruenwald, L. DBOD-DS: Distance based outlier detection for data streams. In Proceedings of the 21st International Conference, DEXA 2010, Bilbao, Spain, 30 August–3 September 2010; pp. 122–136.
12. Takeuchi, J.; Yamanishi, K. A unifying framework for detecting outliers and change points from time series. *IEEE Trans. Knowl. Data Eng.* **2006**, *18*, 482–492. [[CrossRef](#)]
13. O'Regan, S.; Faul, S.; Marnane, W. Automatic detection of EEG artefacts arising from head movements using EEG and gyroscope signals. *Med. Eng. Phys.* **2013**, *35*, 867–874. [[CrossRef](#)] [[PubMed](#)]
14. Gardner, A.B.; Hall, H.; Krieger, A.M. One-class novelty detection for seizure analysis from intracranial EEG. *J. Mach. Learn. Res.* **2006**, *7*, 1025–1044.
15. Jung, T.P.; Makeig, S.; Humphries, C.; Lee, T.W.; McKeown, M.J.; Iragui, V.; Sejnowski, T.J. Removing electroencephalographic artifacts by blind source separation. *Psychophysiology* **2000**, *37*, 163–178. [[CrossRef](#)] [[PubMed](#)]
16. Mannan, M.M.N.; Kim, S.; Jeong, M.Y.; Kamran, M.A. Hybrid EEG—Eye tracker: Automatic identification and removal of eye movement and blink artifacts from electroencephalographic signal. *Sensors* **2016**, *16*. [[CrossRef](#)] [[PubMed](#)]
17. Rokach, L.; Maimon, O. Chapter 15—Clustering methods. In *Data Mining and Knowledge Discovery Handbook*; Springer: New York, NY, USA, 2010.
18. Sokal, R.R.; Rohlf, F.J. The comparison of dendrograms by objective methods. *Taxon* **1962**, *11*, 33–40. [[CrossRef](#)]
19. Halkidi, M.; Vazirgiannis, M. Clustering validity assessment: Finding the optimal partitioning of a data set. In Proceedings of the IEEE International Conference on Data Mining, San Jose, CA, USA, 2 December 2001; pp. 187–194.
20. Burg, J.P. Maximum entropy spectral analysis. In Proceedings of the 37th Annual International SEG Meeting, Soc. of Explor. Geophys, Oklahoma City, OK, USA, 31 October 1967.

21. Lawhern, V.; Hairston, W.D.; McDowell, M.; Westerfield, K.R. Detection and classification of subject-generated artifacts in EEG signals using autoregressive models. *Neurosci. Methods* **2012**, *208*, 181–189. [[CrossRef](#)] [[PubMed](#)]
22. Jolliffe, I.T. Principal Component analysis. In *Encyclopedia of Statistics in Behavioral Science*, 2nd ed.; Springer: New York, NY, USA, 2002; Volume 30, p. 487.
23. Strehl, A.; Ghosh, J. Cluster ensembles—A knowledge reuse framework for combining multiple partitions. *J. Mach. Learn. Res.* **2002**, *3*, 583–617.
24. Melia, U.; Clariá, F.; Vallverdú, M.; Caminal, P. Filtering and thresholding the analytic signal envelope in order to improve peak and spike noise reduction in EEG signals. *Med. Eng. Phys.* **2014**, *36*, 547–553. [[CrossRef](#)] [[PubMed](#)]
25. Mahajan, R.; Morshed, B. Unsupervised eye blink artifact denoising of EEG data with modified multiscale sample entropy, kurtosis and wavelet-ICA. *IEEE J. Biomed. Heal. Inform.* **2014**, *2194*, 1–8. [[CrossRef](#)] [[PubMed](#)]
26. Daly, I.; Scherer, R.; Billinger, M.; Muller-Putz, G. FORCe: Fully online and automated artifact removal for brain-computer interfacing. *IEEE Trans. Neural Syst. Rehabil. Eng.* **2014**, *4320*, 1–13. [[CrossRef](#)] [[PubMed](#)]
27. Nicolaou, N.; Nasuto, S.J. Automatic artefact removal from event-related potentials via clustering. *J. VLSI Signal Process. Syst. Signal Image Video Technol.* **2007**, *48*, 173–183. [[CrossRef](#)]
28. Zeng, K.; Chen, D.; Ouyang, G.; Wang, L.; Liu, X.; Li, X. An EEMD-ICA approach to enhancing artifact rejection for noisy multivariate neural data. *IEEE Trans. Neural Syst. Rehabil. Eng.* **2015**, *4320*. [[CrossRef](#)] [[PubMed](#)]
29. Naseer, N.; Hong, K.S. fNIRS-based brain-computer interfaces: A review. *Front. Hum. Neurosci.* **2015**, *9*. [[CrossRef](#)] [[PubMed](#)]
30. Coyle, S.M.; Ward, T.E.; Markham, C.M. Brain-computer interface using a simplified functional near-infrared spectroscopy system. *J. Neural Eng.* **2007**, *4*, 219–226.
31. Fazli, S.; Mehnert, J.; Steinbrink, J.; Curio, G.; Villringer, A.; Muller, K.R.; Blankertz, B. Enhanced performance by a hybrid NIRS-EEG brain computer interface. *NeuroImage* **2012**, *59*, 519–529.
32. Naseer, N.; Hong, K.S. Reduction of Delay in Detecting Initial Dips from Functional Near-Infrared Spectroscopy Signals Using Vector-Based Phase Analysis. *Inter. J. Neural Syst.* **2016**, *26*. [[CrossRef](#)]



© 2016 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 (<http://creativecommons.org/licenses/by/4.0/>).