






Article

Real-Time Localization of Epileptogenic Foci EEG Signals: An FPGA-Based Implementation

Jose V. Frances-Villora ^{1,*}, Manuel Bataller-Mompean ¹, Azeddine Mjahad ¹;
Alfredo Rosado-Muñoz ¹, Antonio Gutierrez Martin ², Vicent Teruel-Marti ³,
Vicente Villanueva ⁴, Kevin G. Hampel ⁴ and Juan F. Guerrero-Martinez ¹

¹ Processing and Digital Design Group, Department of Electronic Engineering, University of Valencia, 46100 Burjassot, Spain; Manuel.Bataller@uv.es (M.B.-M.); Mjahad.Azeddine@uv.es (A.M.); Alfredo.Rosado@uv.es (A.R.-M.); Juan.Guerrero@uv.es (J.F.G.-M.)

² Functional Neurosurgery Unit. La Fe Hospital, 46009 Valencia, Spain; antoniogutierrez100@gmail.com

³ Neural Circuits Laboratory, Department of Anatomy and Embryology, University of Valencia, 46010 Valencia, Spain; Vicent.Teruel@uv.es

⁴ Refractory Epilepsy Unit. La Fe Hospital, 46026 Valencia, Spain; vevillanuevah@yahoo.es (V.V.); hampel_kev@gva.es (K.G.H.)

* Correspondence: Jose.V.Frances@uv.es

Received: 11 December 2019; Accepted: 18 January 2020; Published: 23 January 2020



Abstract: The epileptogenic focus is a brain area that may be surgically removed to control of epileptic seizures. Locating it is an essential and crucial step prior to the surgical treatment. However, given the difficulty of determining the localization of this brain region responsible of the initial seizure discharge, many works have proposed machine learning methods for the automatic classification of focal and non-focal electroencephalographic (EEG) signals. These works use automatic classification as an analysis tool for helping neurosurgeons to identify focal areas off-line, out of surgery, during the processing of the huge amount of information collected during several days of patient monitoring. In turn, this paper proposes an automatic classification procedure capable of assisting neurosurgeons online, during the resective epilepsy surgery, to refine the localization of the epileptogenic area to be resected, if they have doubts. This goal requires a real-time implementation with as low a computational cost as possible. For that reason, this work proposes both a feature set and a classifier model that minimizes the computational load while preserving the classification accuracy at 95.5%, a level similar to previous works. In addition, the classification procedure has been implemented on a FPGA device to determine its resource needs and throughput. Thus, it can be concluded that such a device can embed the whole classification process, from accepting raw signals to the delivery of the classification results in a cost-effective Xilinx Spartan-6 FPGA device. This real-time implementation begins providing results after a 5 s latency, and later, can deliver floating-point classification results at 3.5 Hz rate, using overlapped time-windows.

Keywords: electroencephalogram; epileptic EEG signal classification; epilepsy; epileptogenic focus; real-time implementation; FPGA

1. Introduction

Epilepsy is a common neurological disorder usually described by seizures which are recurrent in nature. This disorder can be produced by different brain disorders, such as brain tumors, intracranial hemorrhages and brain malformations [1], and depending on the affected area, a disorder may generate, apart from epileptic seizures, malfunctions in motion and patient perception [2].

An epileptic seizure is a period of time where the patient experiences a set of symptoms with different levels of severity: uncontrolled shaking movements of the body with loss of consciousness

(generalized tonic-clonic seizure), shaking movements of a specific part of the body with different levels of consciousness (focal seizure), or short moments of focal seizures with impairment of awareness (absence seizure). Epileptic seizures can be originated by abnormal, synchronous, or even excessive brain neural activity, causing a temporary disruption to the way that the brain normally works. Anyway, what happens to someone during a seizure depends on the affected part of the brain and how far the seizure activity disseminates in the brain.

Epilepsy can be classified primarily into two types: generalized and partial (or focal) epilepsy [3]. Generalized onset seizures affect, at the same time, both sides of the brain or groups of cells on both sides of the brain. On the other hand, focal onset seizures (the term focal is used instead of partial to be more accurate when talking about where seizures begin) usually start in one area or group of cells on one side of the brain.

The activity of the brain is usually registered using either electroencephalography (EEG) or functional magnetic resonance imaging (fMRI). Although fMRI has better spatial resolution, the use of the multidimensional time series generated from electroencephalogram EEG is more popular, as it allows high precision time measurements, is functionally fast and is relatively cheap.

The EEG epileptogenic source's localization has been studied for decades [4,5]; however, the methods were not implemented in clinical practices until recently. Nowadays, the EEG is considered a noninvasive and useful test to assess whether a pharmaco-resistant patient can benefit from the resective epilepsy surgery [6]. As the resective surgery aims to remove surgically the brain sections involved in the focal onset epilepsy, it is important to distinguish precisely between "focal signals," those recorded in brain areas where first ictal signals are detected, and "non-focal signals," those registered from brain areas not related to the seizure onset [7]. Many patients with epilepsy may require EEG signals to be recorded from deep structures of the brain using intracranial electrodes.

Usually, the focus localization is essentially made using registers acquired monitoring the patient 24 h a day, during a stay of several days in an epilepsy monitoring unit (EMU). In these kind of units, apart from scalp EEG (and intracranial electrodes to record signals from deep structures of the brain, in many patients), the epilepsy patients are recorded on video, along with their speech and movements. Thus, all data are collected targeting the evaluation of its seizure disorder, seeking to gather data before a seizure starts, during one and during recovery. The evaluation of this information can be used to locate candidate areas for the epileptogenic focus, although in some cases it is not enough to locate the epileptogenic focus precisely prior to the surgery.

The visual analysis of the EEG recordings of seizures with intracranial electrodes can help in locating the seizure. However, visual inspection is a hard and time consuming process that can be affected by the clinician subjectivity. In addition, it is not easy to determine the seizure source by a direct visual inspection of the EEG signal recordings.

To select candidate areas from EEG signals could be helpful a computerized analysis of the EEG [8,9]. As with other pathologies [10–12], machine learning has been applied in epilepsy at many works [13–15] to classify EEG signals as normal versus epileptic or seizure versus inter-ictal. However, the most challenging classification problem is focal (F) versus non-focal (NF). The classification of seizure from normal and seizure-free signals has achieved a 100% classification accuracy. However, this goal has not been achieved to date for the classification of focal and non-focal EEG signals. Neurosurgeons have difficulties determining the brain region responsible of the initial seizure discharge, so this kind of classification may serve as a tool to help epileptologists to resect the epileptogenic area. Compared with signals of the epileptogenic areas, the signals from non-epileptogenic areas are more nonlinear, less random and more nonstationary.

Many machine learning systems have been developed to classify and detect the epileptogenic source signals. Sharma et al. [13] used entropies derived from the coefficients of the wavelet transform of the EEG signals to feed a least squares-support vector machine (LS-SVM) model to distinguish focal and non-focal EEG signals. In [16], Sharma et al. also used the LS-SVM classifier to feed the entropies derived from some subbands decomposed using tunable-Q wavelet transform (TQWT).

In [17], Sharma et al. utilized empirical mode decomposition (EMD) and entropy for the classification of focal and non-focal EEG signals. In this work, intrinsic mode functions (IMFs) from focal and non-focal EEG signals were extracted using EMD, and then the entropies were fed the input of a LS-SVM classifier. Das et al. [18] also used entropy-based features from the EMD, DWT (discrete wavelet transform) and EMD-DWT domains, along with a k-nearest neighbor (k-NN) classifier model. In turn, Zeng et al. [19] used features derived from euclidean measures obtained from the phase space reconstruction (PSR) of several IMFs, obtained using EMD as well. Bhattacharyya et al. [20] proposed the decomposition of the EEG signal into rhythms using the empirical wavelet transform (EWT), and then used some area measures from them as input for a LS-SVM classifier model, to recognize focal and non-focal EEG signals. Another work of Bhattacharyya et al. [15] also used multivariate subband entropy measures from TQWT along with multivariate fuzzy entropy in combination with a LS-SVM classifier model. Chatterjee et al. [21] also used SVM and k-NN classifiers fed by multifractal, detrended fluctuation analysis (MFDFA) based feature sets. Singh et al. [22] used features derived from DFT-based rhythms of the EEG to feed the LS-SVM classifier. Taran et al. [23] proposed the use of spectral moment based features extracted from the modes of the clustering VMD (CVMD) and extreme learning machine (ELM) classifiers. Deivasigamani et al. [24] utilized features extracted from the dual tree complex wavelet transform (DT-CWT) to feed an adaptive neurons fuzzy interference system (ANFIS).

However, these works tend to require a considerable computational load, especially the most recent ones. As an example, San-Segundo et al. [25] proposed a deep neural network (DNN) made up of two convolutional layers for feature extraction and three fully connected layers for classification. In this work, authors increased the classification accuracy a little at the expense of increasing, considerably, its computational needs. In turn, Daoud et al. [26] used both a deep convolutional autoencoder and an unsupervised learning scheme merging a deep convolutional variational autoencoder and a K-means algorithm.

This progressive increase of the computational demand could impede the jump of using the epileptogenic source localization during surgery. This application needs a real-time implementation, and could be used as a help-decision tool by neurosurgeons to refine the localization of an epileptogenic area during resective epilepsy surgery. Note that the recent technology is mature enough to implement machine learning processes in real-time [27,28].

Thus, the goals of this work were to assess the possibility of locating the epileptogenic focus in real-time, and study the simplification of the classification process to reduce its computational needs as much as possible while maintaining similar classification accuracy to previous works. We also studied the resource usage and performance of the real-time application on a recent Xilinx FPGA reconfigurable device.

The main contributions of this work are:

- The proposal of an automatic classification procedure optimized to classify in real-time the location of the epileptogenic focus from EEG inter-ictal signals. It is conceived to be used in a portable device as a decision-assisting tool by neurosurgeons during surgery.
- The proposed feature set and the classifier model have been selected to minimize both the number of features and the computational cost, while preserving the classification accuracy at a level similar to that in previous works.
- The classification procedure has been implemented using a reconfigurable logic FPGA device. This hardware implementation computes the whole procedure, accepts the EEG raw input signal and delivers the classification result. Two designs have been implemented, using single and double floating point precision following the IEEE 754 standard for floating-point arithmetic.
- The analysis of the resource usage of this kind of implementation, its accuracy with respect a Matlab implementation and how fast the device can deliver results (maximum frequency of operation).

The rest of the paper is organized as follows. Section 2 introduces the dataset, and the analyzed features and classifier models. The details of the hardware implementation and the proposed computational method are described in Section 3. Results of the analysis and discussion are presented in Sections 4 and 5, respectively. Finally, Section 6 concludes the paper.

2. Methodology, Materials and Methods

This section presents an overview of the dataset utilized, and introduces the analyzed features and classifier models.

2.1. Dataset

In this work, the publicly available Bern-Barcelona database [7] was used. This is an open source EEG dataset that has been used for a large number of epilepsy studies [16–20]. This dataset collects intracranial EEG recordings from five pharmacoresistant epilepsy patients, including two classes of EEG signals: focals and non-focals. *Focal* signals (F) are those captured from an epileptogenic area (acquired from those channels that detected ictal EEG signal changes first, as decided by at least two neurologists via visual inspection) and *non-focal* signals (NF) are captured in channels out of this area. Each class contains 3750 pairs of simultaneously acquired signals "x" and "y," all of them randomly selected, and consisting of 20 s windows of simultaneous recording, sampled at 512 Hz. Each focal pair consists of one of the focal EEG channels for the x signal, and one of this channel's neighboring focal channels for the y signal, both simultaneously acquired from the same patient. The non-focal pairs were selected from nonfocal EEG channels in the same way [7,29]. All EEG signal were band-pass filtered by an fourth order Butterworth (0.5 Hz–150 Hz). In addition, before being included into the database, signal pairs were visually inspected to discard prominent measurement artifacts.

Note that all recordings of seizure activity, and three hours after the last seizure, were excluded. Thus, this database contains neither ictal nor postictal stage activity.

2.2. Preprocessing

The 50 Hz of the EEG signals was filtered using a moving average of order 5. In addition, EEG signals were filtered using a Butterworth low pass IIR filter with $f = 80$ Hz and order $N = 6$.

2.3. Feature Extraction

This is one of the most important steps in classification problems. Table 1 lists the 39 features considered in this work for each segment of the signal dataset. All these have been used successfully as features in previous EEG seizure detection works [14,30–37].

The features considered are from different domains, such as time, frequency, information theory and entropy. But note that all of them are univariate and imply low or medium computational load to extract them.

Statistical parameters such as mean, variance, skewness and kurtosis have been used to extract information on changes in the distribution and amplitude of the EEG data. Those parameters have been considered on the first and second derivative too. Frequency parameters have been calculated by means of the DFT transform, as spectral power or relative energy between different bands. In addition, some nonlinear features have been calculated, such as fractal dimension, used to compare rhythms in the self-similarity present in the signals; entropy of the signal; and spectral entropy, to depict randomness of the EEG in the frequency domain.

However, note than other features having greater computational complexity were not considered, even if these had been used successfully in other works. The reason is obvious; this work aimed to select the set of features having the lowest computational load while providing similar classification accuracy to other works in the bibliography. Thus, the more computationally complex features were discarded from the beginning, such as the calculation of certain entropy measures from IMF signals calculated using EMD, wavelet transform or other time-frequency domain features.

Table 1. List of extracted features.

EEG Feature	Description
Log energy entropy (LogEn) [30]	Non-normalized energy based entropy
Median frequency (MDF) [31]	Division of the EEG power spectrum into two regions
Mean frequency (MNF) [31]	Mean normalized frequency of the power spectrum
Katz fractal dimension (KFD) [31]	Index characterizing the fractal pattern complexity
Lower quartile 1 (Q1) [31]	25% of the EEG signal
Upper quartile 3 (Q3) [31]	75% of the EEG signal
Inter quartile range (IQR) [31]	Difference between Q3 and Q1
Semi inter quartile deviation (SID) [31]	Measure of spread
Skewness (Sk) [32]	Measure of the degree of symmetry
Kurtosis (Kr) [32]	Measure of tailedness of the probability distribution
Root mean square (RMS) [33]	Root mean square of the EEG signal
Band power (PB) [33]	Average power of the EEG signal (0 to $f_s/2$)
Zero crossing (ZC) [33]	Number of times that the signal changes of sign
Complexity (Comp) [33]	Hjorth parameter
Mobility (Mob) [33]	Hjorth parameter
Activity (Act) [33]	Hjorth parameter
Spurious free dynamic range (SFDR) [34]	Length along a EEG signal
Curve length (CL) [34]	Length along a EEG signal
Teager energy (TE) [34]	Non linear energy
Variance (Var) [34]	Variance of the EEG signal
Standard deviation (Std) [34]	Standard deviation of the signal
Mean (Mean) [34]	Mean of the EEG signal
1st derivative variance (Var1) [34]	Variance of the first derivative
1st derivative standard deviation (Std1) [34]	Standard deviation of the first derivative
1st derivative mean (Mean1) [34]	Mean of the first derivative
2nd derivative variance (Var2) [34]	Variance of the second derivative
2nd derivative standard deviation (Std2) [34]	Standard deviation of the second derivative
2nd derivative mean (Mean2) [34]	Mean of the second derivative
Derivative variance ratio (RatioVar) [36]	Ratio of derivative respect absolute of derivative variances
Power (P) [35]	Power of the signal window
1st difference (1d) [35]	Feature extraction and selection for emotion recognition from eeg
Normalized 1st difference (N1d) [35]	Normalization of 1st difference
2nd difference (2d) [35]	Feature extraction and selection for emotion recognition from eeg
Normalized 2nd difference (N2d) [35]	Normalization of 2nd difference
Normalized Length Density (NLD) [35]	Quantifies self-similarities
Higher order crossings (HOC) [38]	Describes the oscillatory pattern of a signal
Band power (Pu) [35]	Spectral power in certain spectral band
Recursive Effective Efficiency (REE) [35]	Energy ratio of spectral bands
Relative Energy Ratio (rE) [37]	Relative energy between bands

2.4. Feature Reduction

Feature reduction reduces the computational complexity of the classifier and also avoids the possibility of redundancy. In this study, we obtained several discriminatory features for the two class classification process (Section 4.1). The number of features was reduced using the ReliefF feature selection technique [39], as is explained in Subsection 4.1.

2.4.1. Log Energy Entropy

Entropy is a concept handling predictability and randomness, with higher values of entropy always being related to a lesser system order and more randomness. The entropy of an EEG channel is a measure of uncertainty, where the EEG signals are considered a random variable. The log energy entropy of a x EEG signal is defined as [40]:

$$H_{LogEn}(x) = - \sum_{i=0}^{N-1} (\log_2(p_i(x)))^2, \quad (1)$$

$p(x)$ being the probability density function. With this entropy calculated regarding the signal power spectrum as a probability distribution, the log energy spectral entropy is obtained.

2.4.2. Skewness

It is a higher-order statistical attribute of a time series. Skewness is a measure of the asymmetry of the probability distribution (pdf) of a real-valued random variable around its mean.

$$Skew[x] = E\left[\left(\frac{x - \mu}{\sigma}\right)^3\right] = \frac{E[x^3] - 3\mu\sigma^2 - \mu^3}{\sigma^3}. \quad (2)$$

2.4.3. Root Mean Square

Is the square root of the mean square, the arithmetic mean of the squares of a set of numbers, also known as the quadratic mean:

$$RMS = \sqrt{\frac{1}{N} \sum_{i=1}^N x_i^2}. \quad (3)$$

2.4.4. Derivative Variance Ratio

This is the quotient between the variance of the derivative of the signal and the variance of the absolute value of said derivative. It is a derivative variance ratio (called RatioVar) [36]:

$$Ratiovar = \frac{E\left[\frac{dx^2}{dt}\right] - E\left[\frac{dx}{dt}\right]^2}{E\left[\frac{d|x|^2}{dt}\right] - E\left[\frac{d|x|}{dt}\right]^2}. \quad (4)$$

2.4.5. Relative Energy Ratio

It is used to observe the changes in EEG frequency bands due to the stressors. When stress occurs, the energy of Alpha band, HF, will reduce. Meanwhile, energy of lower bands will increase [37]:

$$rE = \frac{\text{Total Energy In HF}}{\text{Total Energy In LF}}. \quad (5)$$

2.5. Classifier Models

Several classifiers have been used in this analysis (Section 4.1), each one having its own specific strengths and weaknesses. All them are briefly outlined below.

2.5.1. Support Vector Machine (SVM)

It is a supervised classification technique that constructs a separating hyperplane maximizing the margin between the input data classes that are viewed in an n-dimensional space (n is the number of features used as inputs). Essentially, this involves orienting the separating hyperplane to be perpendicular to the shortest line separating the convex hulls of the training data for each class, and locating it midway along this line.

In addition to performing linear classification, SVMs can efficiently perform a non-linear classification using what is called the kernel trick, implicitly mapping their inputs into high-dimensional feature spaces.

2.5.2. K-Nearest Neighbor (KNN)

KNN is a supervised learning technique where a new instance is classified based on the closest training samples present in the feature space. It does not use any model to fit, and is only based on memory. When a test data is entered, it is assigned to the class that is most common amongst its *k* nearest neighbors.

2.5.3. Decision Tree

It is a method that creates a model that enables one to predict the target value of an item (represented in leaves) based on several input variables (represented as branches). In the case of using a classification tree analysis, the predicted outcome is the class (discrete) to which the data belongs.

2.5.4. Logistic Regression

It is a classification algorithm used to assign observations to a discrete set of classes. Unlike linear regression, which outputs continuous values, the logistic regression transform its output using the logistic sigmoid function, to return a probability value that can be mapped to two or more discrete classes.

2.5.5. Discriminant Analysis

Linear and quadratic discriminant analysis were used. Linear discriminant analysis (LDA) is a generalization of the Fisher's linear discriminant that finds a linear combination of features characterizing or separating two or more classes. In turn, quadratic discriminant analysis separates input features into two or more classes of objects by a quadratic surface, becoming a general version of the linear version.

2.5.6. Ensembles

Ensemble classification improves results by combining several models. It can be used with any learning method. Thus, this approach allows for better predictive performance compared to single models. The number of classifier components has a great impact on the classification accuracy. In this work, trees, discriminant and *k*-NN classifier components were used.

2.5.7. Neural Network Classifier (NN)

A neural network consists of a series of units (neurons) arranged in layers. This arrangement converts an input vector into some output. To do so, each neuron takes its inputs and calculates the output by applying a usually nonlinear function (the activation function), to later pass the output to the next layer. Generally, neural networks are defined as feed-forward: a unit feeds its output to all the units on the next layer, but there is no feedback to the previous layer. Signals are weighted when fed the input of a unit neuron. The weights are tuned in the training phase of the classifier.

Neural networks are considered to be good classifiers due to their inherent features, such as adaptive learning, robustness, self-organization and generalization capability.

2.6. Performance Analysis

In order to evaluate the performance of the proposed method, the performance of the classifiers are expressed in terms of classification accuracy (Acc), defined as follows [41]:

$$Accuracy(\%) = \frac{TP + TN}{TP + FN + TN + FP} \times 100 \quad (6)$$

where *TP*, *TN*, *FP* and *FN* denote true positives, true negatives, false positives and false negatives, respectively.

3. Hardware Implementation and Computational Method

This section discusses the implementation details. We will leave for Section 4 the reasons for and how this computational procedure was chosen. At this point, all that matters to know is that the classifier model used was a perceptron with 25 neurons in the single hidden layer, 5 neurons in the input layer and 2 neurons in the output layer (Algorithm 2). The 5 extracted features (Table 2) were computed following Algorithm 1. Details about feature selection and classifier model selection are left for Section 4.

The proposed hardware implementation was conceived as an intellectual property (IP) core using Xilinx Vivado HLS and the Xilinx Vivado Design Suite 2016.2 [42]. It provides a signal interface definition that enables it as a standalone module, being able to also be used as a peripheral of a more complex system on chip (SoC), embedded microprocessor, etc. Further, this approach offers the capability of customization for specific needs in many different hardware applications.

3.1. Working Modes

The IP core performs two different working modes:

- *Initialization mode.* The initialization of the IP core consists of the load of the matrices \mathbf{x}_{nmax} , \mathbf{x}_{nmin} , \mathbf{W}_h , \mathbf{b}_h , \mathbf{W}_o , \mathbf{b}_o , \mathbf{x}_{dmax} , \mathbf{x}_{dmin} , y_{min} , and y_{max} , using the external signal interface. These matrices are essentially weights and bias of the neural network, along with normalization and denormalization values and vectors. All together, these matrices allow the IP core to perform a proper classification. Note that the calculation of these matrices is achieved out of this IP core, and the results are transferred to it during this initialization process. Once the initialization is complete, the core can change to another working mode, never before.
- *Run mode or on-line mode.* In this mode, the input data \mathbf{x} is fed into the IP core. Then, several features are extracted, and in turn, fed into the neural network system. The output is computed according to the initialized network topology. The IP core, when running in this mode, computes and serves the corresponding output before accepting a new input.

3.2. IP Core Signal Interface

In Xilinx FPGAs, external core signal interfaces are used to follow proprietary protocol specifications, such as AXI4 [43,44], AXI4-Lite or AXI4-Stream [44,45]. In this work the AXI4 protocol was selected to permit specifying arrays as arguments. However, note that the protocol interface has almost no influence on performance when the core is running in the on-line mode. The reason is that the reported performance refers to the complete epileptic focus classification task, while the load of one input vector by iteration implies a negligible overhead. Therefore, for replication purposes, it can be expected to achieve similar results, for the running mode, regardless of the protocol specification implemented for the core signal interface (e.g., AXI4-Lite, AXI4 or AXI4-Stream).

Figure 1 outlines the external signal interface of the IP core, where signal lines are represented by thin black arrows and buses, and bunches of signals are represented by white thick arrows.

Signal START indicates when the core can start processing data, the READY signal indicates when the core is ready to accept new inputs, the IDLE signal indicates when the core is idle and the DONE signal indicates when the core operation has been completed. Altogether, these signals constitute the block-level interface, controlling the core independently of the port-level I/O protocol.

On the other hand, the input and output data ports implement a handshake data flow protocol. Lines A_TVALID, A_TREADY and the bus A_TDATA integrate the input data port, while B_TVALID, B_TREADY and the bus B_TDATA integrate the output data port. The TDATA bus is the payload, while TVALID and TREADY lines signal when the information passes across the interface. These signals integrate a two-way flow control mechanism that enables master and slave to control the data transmission rate across the interface.

Finally, the MODE signal is used as application-signaling. It requests the entering on initialization mode or on-line mode.

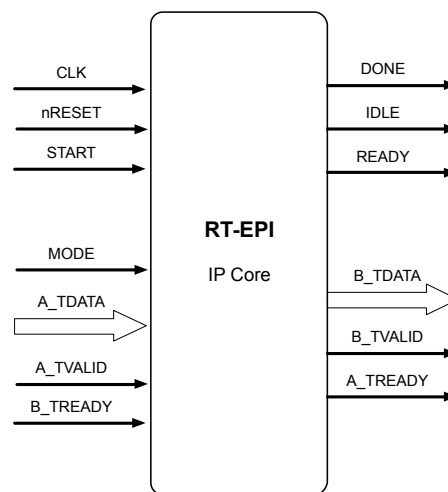


Figure 1. Interface signals used by the RT-EPI IP core. White arrows represents buses.

3.3. System Parameterization

The definition of the SLFN neural network topology was conceived parametrically to achieve a flexible design with minimal code modifications. The main parameters are:

- L: Is the length, in samples, of the input signal window.
- IN: Is the number of neurons in the input layer.
- \tilde{N} : Is the number of neurons in the hidden layer.
- ON: Is the number of neurons in the output layer.
- FT: Boolean determining the type of IEEE 754 floating-point arithmetic precision: single or double.

3.4. Computation of the RT-EPI Algorithm

The proposed IP hardware core implements a real-time epileptogenic focus classification of a different EEG input window signal each iteration.

3.4.1. Algorithm Description

Algorithm 1 shows the steps in which perform the feature computation. This computational procedure follows the expressions described in Table 2 to extract features from the input signal, and then, to compose the input data pattern to be used as input to the neural network.

Algorithm 2 shows the computational procedure implementing the classification. Note that it is mandatory to the previous IP core so that the implementation may be fully functional and begin accepting any input data. The initialized structures are the normalization and denormalization parameters ($x_{nmax}, x_{nmin}, x_{dmax}, x_{dmin}, y_{min}, y_{max}$), plus the weights and biases of the hidden and output layers (W_h, b_h, W_o, b_o). Further, the first step of the algorithm is the normalization of the data input pattern. Then, the outputs of the hidden and output layers are calculated. Although the hyperbolic tangent sigmoid function has been used as an activation function, the proposed algorithm allows the use of a wide range of activation functions (including piece-wise linear activation functions). The results of the output layer are then denormalized to obtain the final output.

The pseudocode in Algorithms 1 and 2 show all the matrices and vectors involved in each step of the computation, with their respective dimensions.

The computational procedure was implemented using a sequential architecture. This architecture minimizes both the usage of memory and arithmetic slices. And, although the throughput results can

be improved using a parallel architecture, the use of a sequential architecture enabled us to establish a standard machine to be used as a reference in subsequent works.

Algorithm 1 Feature extraction pseudocode.

Input: $\mathbf{x}_{(1 \times L)}$ \rightarrow Input signal.

Output: $\mathbf{x}'_{(1 \times 5)}$ \rightarrow Output features vector.

- 1: $\bar{\mathbf{x}}_{(1 \times 1)} = \frac{1}{L} \sum_{i=1}^L x_i$.
- 2: $\mathbf{x}_{(1 \times L)} = \mathbf{x} - \bar{\mathbf{x}}$.
- 3: $\mathbf{x}_{(1 \times L)} = 50\text{Hz_removal}(\mathbf{x})$.
- 4: $\mathbf{x}_{(1 \times L)} = \text{low_pass_filtering}(\mathbf{x})$ (*).
- 5: $\bar{\mathbf{x}}_{(1 \times 1)} = \frac{1}{L} \sum_{i=1}^L x_i$.
- 6: $\mathbf{x}_{(1 \times L)} = \mathbf{x} - \bar{\mathbf{x}}$.
- 7: $\text{RMS}_{(1 \times 1)} = \sqrt{\frac{1}{L} \sum_{i=1}^L x_i^2}$.
- 8: $\text{skew}_{(1 \times 1)} = \frac{\frac{1}{L} \sum_{i=1}^L (x_i - \bar{\mathbf{x}})^3}{\left(\frac{1}{L} \sum_{i=1}^L (x_i - \bar{\mathbf{x}})^2\right)^{\frac{3}{2}}}$.
- 9: $\mathbf{d}_{(1 \times L-1)} \mid d_{i-1} = x_i - x_{i-1} \text{ for } i = \{2, \dots, L\}$.
- 10: $\mathbf{a}_{(1 \times L-1)} \mid a_i = |d_i| \text{ for } i = \{1, \dots, L-1\}$.
- 11: $\text{RatioVar}_{(1 \times 1)} = \frac{\sigma_{\mathbf{d}}}{\sigma_{\mathbf{a}}}$.
- 12: $\mathbf{tmp}_{(1 \times L)} = |\text{FFT}(\mathbf{x})|$.
- 13: $\mathbf{p}_{(1 \times 1+L/2)} \mid p_i = k \cdot \text{tmp}_i, k = 2 \text{ for } i = \{2, \dots, L/2\} \text{ and } k = 1 \text{ for } i = \{1, 1 + L/2\}$.
- 14: $rE_{(1 \times 1)} = \frac{\sum_{i=49}^{321} p_i}{\sum_{j=5}^{48} p_j} \quad (p_i \in [12\text{Hz}, 80\text{Hz}], p_j \in [1\text{Hz}, 12\text{Hz}])$.
- 15: $\text{LogEn}_{(1 \times 1)} = \sum_i \log_2(p_i^2)$.
- 16: $\mathbf{x}'_{(1 \times 5)} = \left[rE \quad \text{LogEn} \quad \sigma \quad \text{RatioVar} \right]$.

(*) Filter coefficients: $B[0] = 0.0001; B[1] = 0.0006; B[2] = 0.0014; B[3] = 0.0019; B[4] = 0.0014; B[5] = 0.0006; B[6] = 0.0001; A[0] = 1.0000; A[1] = -4.1069; A[2] = 7.2450; A[3] = -6.9795; A[4] = 3.8569; A[5] = -1.1558; A[6] = 0.1464$.

Algorithm 2 Classification of an input pattern. pseudocode

Input: $\mathbf{x}'_{(1 \times IN)}$ \rightarrow Input data pattern.
 $\mathbf{x}_{nmax}_{(1 \times IN)}$, $\mathbf{x}_{nmin}_{(1 \times IN)}$ \rightarrow Input normalization vectors.
 $y_{min}_{(1 \times 1)}$, $y_{max}_{(1 \times 1)}$ \rightarrow Range normalizing input.
 $\mathbf{W}_h_{(IN \times \tilde{N})}$ \rightarrow Hidden layer weights matrix.
 $\mathbf{b}_h_{(1 \times \tilde{N})}$ \rightarrow Hidden layer bias.
 $\mathbf{W}_o_{(\tilde{N} \times ON)}$ \rightarrow Output weights matrix.
 $\mathbf{b}_o_{(1 \times ON)}$ \rightarrow Output layer bias.
 $\mathbf{x}_{dmax}_{(1 \times ON)}$, $\mathbf{x}_{dmin}_{(1 \times ON)}$ \rightarrow Output denormalization vectors.

Output: $\mathbf{y}_{(1 \times ON)}$.

- 1: $\mathbf{x}_{norm}_{(1 \times IN)} = \frac{(y_{max} - y_{min}) \cdot (\mathbf{x}' - \mathbf{x}_{nmin})}{\mathbf{x}_{nmax} - \mathbf{x}_{nmin}} + y_{min}$.
- 2: $\mathbf{tmp1}_{(1 \times \tilde{N})} = \text{Tansig}(\mathbf{x}_{norm} \cdot \mathbf{W}_h + \mathbf{b}_h)$.
- 3: $\mathbf{tmp2}_{(1 \times ON)} = \text{Tansig}(\mathbf{tmp1} \cdot \mathbf{W}_o + \mathbf{b}_o)$.
- 4: $\mathbf{y}_{(1 \times ON)} = \frac{(\mathbf{tmp1} - y_{min}) \cdot (\mathbf{x}_{dmax} - \mathbf{x}_{dmin})}{y_{max} - y_{min}} + \mathbf{x}_{dmin}$.

(\tilde{N} : number of hidden neurons, IN: number of input neurons, ON: number of output neurons.)

3.4.2. Design Considerations

The proposed design allows the definition of IEEE 754 floating point units using single or double data type precision. This selection together with the parametric definition of the SLFN neural network permits one to test the design in different conditions with few code modifications.

The activation function implemented in this design was the hyperbolic tangent sigmoid.

On the other hand, the proposed design uses pipelining. The main reasons for that is that pipelining alleviates the great latency involving the use of floating point operations, and the suitability of the algorithm for its use, since most of the steps of Algorithms 1 and 2 can be implemented with *for* loops or nested *for* loops.

The pipelining technique helps to optimize the initiation interval, defined as the number of clock cycles that must occur before a new input can be applied. Thus, the initiation interval becomes the parameter to optimize, and the effort must focus on approximating it to one as much closely as possible. To carry this out, we used the PIPELINE optimization pragma directive in each step of the computational procedure implemented using *for* loops. This generates a pipeline design with an initiation interval as low as possible, which dramatically reduces the total latency of the loop implementation.

In addition, the clock period target was set to 4 ns. This forces the compiler to obtain the fastest hardware implementation.

Take into account that both the above design considerations and the computational procedure described in Algorithms 1 and 2 must be followed to replicate the implementation results in this work.

4. Results

The analysis described in this work was carried out using the Bern-Barcelona dataset (Section 2.1). The first 50 focal and non-focal register pairs of the dataset were used. As all considered features are

univariate, and 40 s of EEG signals are available in each register pair ($20 \text{ s} \times 2$), a total of 4000 s of EEG signals were considered in this study.

In order to better compare our results with the results provided by previous works, we used five-fold cross validation in our experimental procedure. In this approach, the EEG signals are divided randomly into five equal portions. Four out five portions were considered for training and the rest, half for validation and half for testing.

All the feature extraction algorithms and classification models were implemented using Matlab R2018a and the "Statistics and Machine Learning Toolbox."

The first goal was to find the classification procedure with the lowest computing cost, because of the real-time implementation aim. Thus, we had to be find (1) a set of features with the minimum number of features and minimum computing cost, using (2) the simplest classifier model, and (3) the optimum segment length. A minimum classification accuracy of 95% was required, a value above the average of related works in the bibliography.

To do so, all considered features, Table 1, were ranked by discrimination capacity and by computational cost. Then, different classification procedures were checked in a loop. At each iteration of the loop, a combination of features were selected (taking into account the ranks); then, fed to 24 classification models, Table 3; and finally the classification accuracies for different segment lengths were calculated. The loop finished when the smallest set of features reached 95% of classification accuracy on at least one of the classification models.

Once the classification procedure was defined, it was coded in C using Xilinx Vivado HLS and the Xilinx Vivado Design Suite 2016.2 [42]. These tools were also used to carry out synthesis, simulations and cosimulations. The Xilinx Virtex-7 XC7VX1140T FPGA device was selected for synthesis and implementation, because it is a biggest Virtex-7 device that permits implement the application without resource restrictions.

The coded design is parameterizable, and follows a pipelined and sequential architecture that computes (Algorithms 1 and 2) all the classification procedure from accepting the raw signal to the delivery of the classifier output.

The reported analyses were conducted using two different arithmetic precisions: a 32-bit floating-point algorithm ("single" design), and 64-bit floating-point algorithm ("double" design). Both implementation designs used the IEEE 754 standard.

4.1. Set of Features and Classification Model

The ReliefF algorithm was used as feature extraction algorithm. ReliefF returns a rank of features and its weights to represent the discrimination capacity of these features. These ranks and weights were used to select the relevant set of features, along with the computational cost criteria, optimizing the real-time implementation of the application. The number of selected features was defined as the minimum set of features that allowed us to obtain a minimum threshold of 95% classification accuracy on at least one of the classification models (Table 3).

To determine the best classification model, 24 classification models were tested. Table 3 lists all these models along with their prediction speed and memory usage characteristics. Note that only those classification models with low or medium speed and memory usage were chosen, due to the importance of minimizing these parameters in the real-time implementation.

All classification models were trained with several sets of features, seeking to determine the feature set and classification model at the same time. However, the evaluation of the classification models depends on a third parameter: the window length. Figure 2 shows the accuracy results for the selected sets of features (Table 2), for all the classification models and the variations of the window length from 1 to 10 s. Note that this figure only shows the best score for the neural network classification model. Figure 3 details the classification accuracy obtained using different numbers of neurons for the hidden layer.

Thus, five features were finally selected, those in Table 2, to be used with a neural network as the classification model. As it can be seen, the computational cost of the five selected features is low (case of features derived from the temporal domain) or medium (in the case of frequency domain features, where it is necessary to compute the Fourier transform).

This minimum set of features achieves a classification accuracy of 95% using a Neural Network and five seconds of segment length. In addition, the optimum number of neurons in the hidden layer of the neural network is 25, as it can be shown in the analysis of Figure 3.

Thus, from that point we will assume that the implementation is done by a neural network of type perceptron with just one hidden layer of 25 hidden neurons and two output neurons in the output layer (provided the two classes of this classification problem). In addition, the neural network will have five input neurons, because five is the dimensionality of the selected number of features, which, in turn, would be computed from window lengths of 5fivesconds of the input signal.

Table 2. Set of selected features.

Features	
Root Mean Square	$RMS_x = \sqrt{\frac{1}{N} \sum_{i=1}^N x_i^2}$
Skewness	$s = \frac{E(x - \mu)^3}{(E(x - \mu)^2)^{\frac{3}{2}}}$
Derivative Variance Ratio	$RatioVar = \frac{\sigma_{\Delta}}{\sigma_{ \Delta }}$ with $\Delta(i) = x(i) - x(i-1)$
Relative Energy Ratio	$rE = \frac{\sum_{12-80Hz} P(x)}{\sum_{1-12Hz} P(x)}$ with $P(x)$ the Spectral Power
Log Energy Spectral Entropy	$LogEn = \sum \log_2(P(x)^2)$

Table 3. Acronyms, prediction speed and memory usage characteristics of all the classification models used in this work.

Acronym	Classifier	Prediction Speed	Memory Usage
[TreeCoarse]	Decision Tree (Coarse Tree)	Fast	Small
[TreeMedium]	Decision Tree (Medium Tree)	Fast	Small
[TreeFine]	Decision Tree (Fine Tree)	Fast	Small
[DiscrLin]	Discriminat Analysis (Linear Discriminant)	Fast	Small
[DiscrQuad]	Discriminat Analysis (Quadratic Discriminant)	Fast	Large
[RegrLog]	Logistic Resregion	Fast	Medium
[SVMLin]	Support Vector Machine (Linear SVM)	Fast ¹	Medium
[SVMQuad]	Support Vector Machine (Quadratic SVM)	Fast ²	Medium ³
[SVMCubic]	Support Vector Machine (Cubic SVM)	Fast ²	Medium ³
[SVMFineGaus]	Support Vector Machine (Fine Gaussian SVM)	Fast ²	Medium ³
[SVMMedGaus]	Support Vector Machine (Medium Gaussian SVM)	Fast ²	Medium ³
[SVMCoarseGaus]	Support Vector Machine (Coarse Gaussian SVM)	Fast ²	Medium ³
[KNNFine]	Nearest Neighbor (Fine KNN)	Medium	Medium
[KNNMedium]	Nearest Neighbor (Medium KNN)	Medium	Medium
[KNNCoarse]	Nearest Neighbor (Coarse KNN)	Medium	Medium
[KNNCosine]	Nearest Neighbor (Cosine KNN)	Medium	Medium
[KNNCubic]	Nearest Neighbor (Cubic KNN)	Slow	Medium
[KNNWeighted]	Nearest Neighbor (Weighted KNN)	Medium	Medium
[BoostTrees]	Ensemble (Boosted Trees)	Fast	Low
[BaggTrees]	Ensemble (Bagged Trees)	Medium	High
[SubspaceDiscr]	Ensemble (Subspace Discriminant)	Medium	Low
[SubspaceKNN]	Ensemble (Subspace KNN)	Medium	Medium
[RUSBoostTrees]	Ensemble (RUSBoost Trees)	Fast	Low
[NeuralNetwork]	Neural Network (3-Layer Perceptron)	Fast ⁶	Low ⁶

Fast for binary classification, medium for multiclass classification. Fast for binary classification, slow for multiclass classification. Medium for binary classification, high for multiclass classification. Slow for high-dimensional data. Medium for high-dimensional data. When a moderate number of neurons are used.

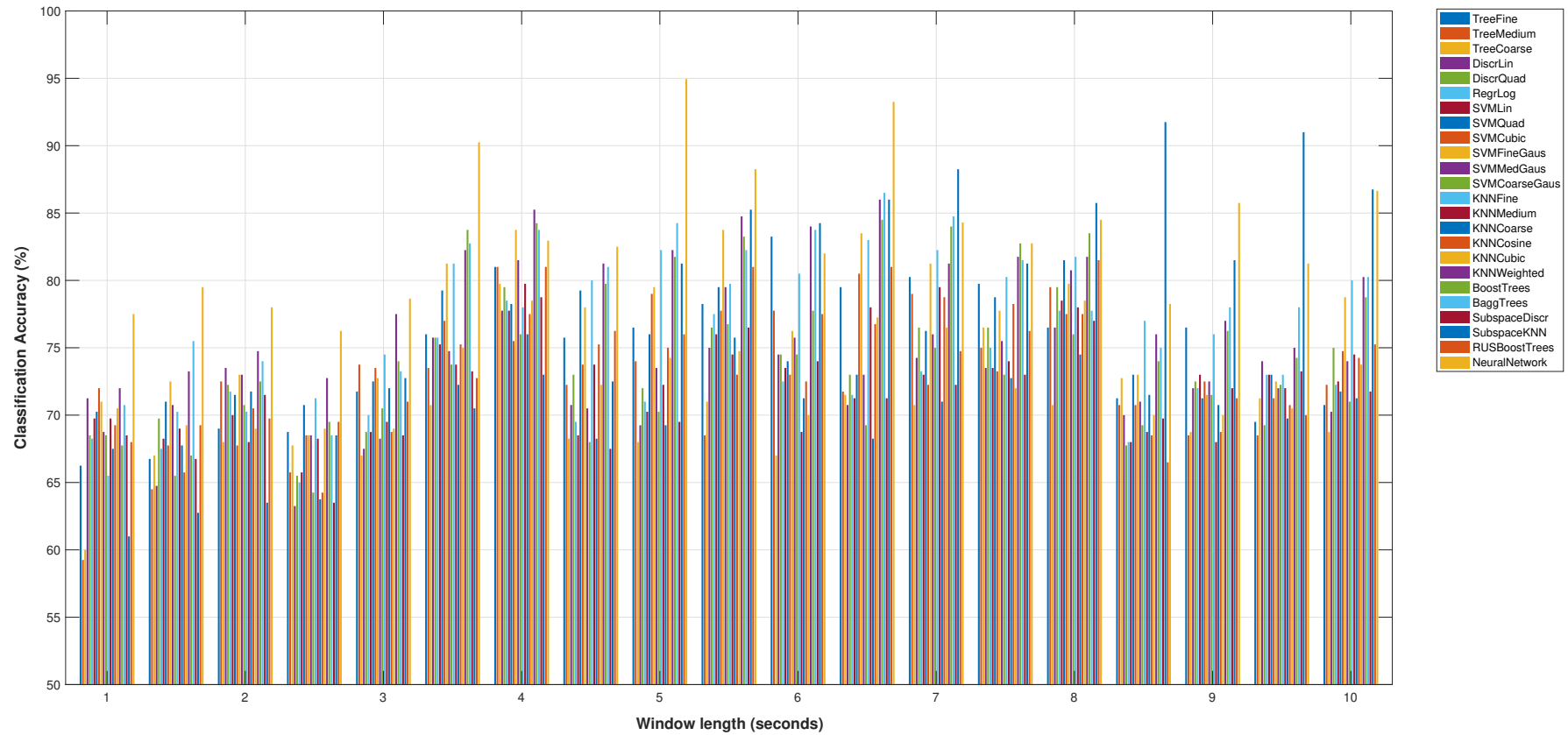


Figure 2. Classification accuracy for all the classification models (using the selected set of features) in front of the window length. As can be seen, the neural network model enabled us to achieve 95% accuracy for a window of five seconds.

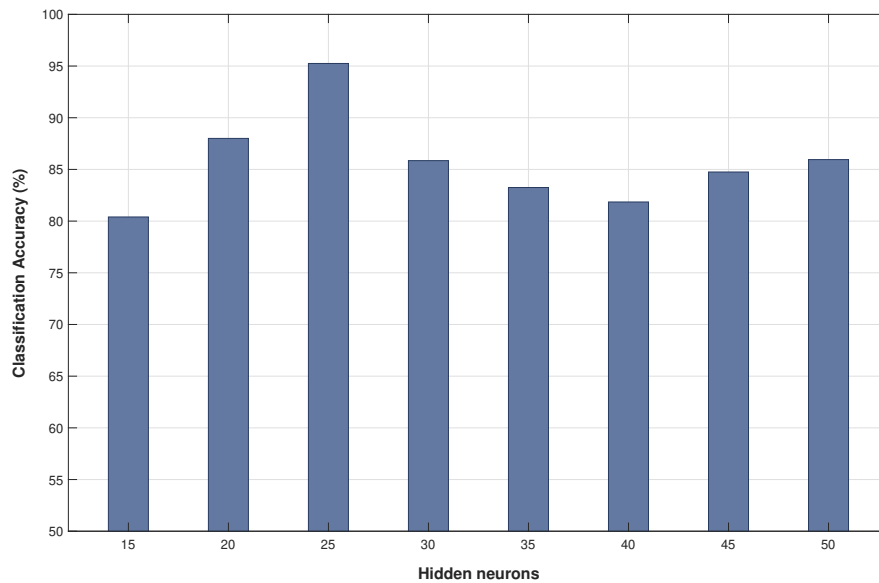


Figure 3. Classification accuracy of the neural network as a function of the number of hidden neurons in the hidden layer.

4.2. Hardware Resources Analysis

Table 4 gathers all the resource analysis results. It also shows resource usage as a percentage of occupation of the Xilinx XC7VX1140T FPGA, intending to provide an idea of the design occupancy in current FPGAs.

It can be seen that the design demands near the same DSP48E slices for both designs, 133 for single precision and 137 for double precision. Obviously, this slight variation has been achieved thanks to the pipelined design.

Table 4. Resource usage, performance and precision of the FPGA implementation as a function of the data type ("float" is for 32-bit floating-point arithmetic precision, and "double" for 64 bit floating-point arithmetic precision). Resource usage is indicated by the number of required slices and the percentage of occupation of a Xilinx XC7VX1140T FPGA.

Resources	Data Type	
	Double	Float
DSP48E	137	133
BRAM	185	95
FF	45,240	39,921
LUT	52,571	46,302
DSP48E	4.0%	4.1%
BRAM	4.9%	2.5%
FF	3.2%	2.8%
LUT	7.4%	6.5%
Clock Period (ns)	5.35 ± 0.5	5.21 ± 0.5
Max. Clock Frequency (MHz)	186.9	191.9
Clock cycles	73,999,901	53,996,354
Max. Classification Frequency (Hz)	2.53	3.55
Accuracy (MAE)	3.9428×10^{-15}	1.0005×10^{-6}

The required number of flip-flops (FF) and look up tables (LUT) were not high, presenting only small variations between designs.

On the other hand, as is natural, the amount of block RAM doubled for double arithmetic precision with respect to single arithmetic precision. That is due to the use of 64-bit representation of double precision and 32-bit of single precision. Then, block RAM requirement halves according the amount of memory needed for its representation.

4.3. Hardware Performance and Accuracy

Mean absolute error (MAE) was used to measure the accuracy of the results of each design with respect to the Matlab implementation. MAE measures the average magnitude of the errors without considering the direction of its deviation, taking into account the absolute differences between them.

The accuracy shown in Table 4 is the maximum of the accuracy obtained for the results of each output neuron. As it can be seen, the accuracy for the single design is very low, but the accuracy for the double design is so low that, in practice, it indicates that results in this case are similar to the results obtained in its Matlab implementation. It is natural, because both implementations use double precision data types.

In turn, the number of clock cycles shows a strong dependency of the data type (Table 4), and the numbers of minimum allowable cycles reported for both designs were similar.

The maximum frequency of operation is obtained from the minimum clock period and the required number of clock cycles. It is represented in Table 4 for both data type designs. As it can be seen, the computation of all the classification procedure, from accepting raw signals to the delivery of the classification output can be done at a rhythm of 2.53 Hz for the "double" design and 3.55 Hz for the "single" designs (3.55 classification outputs by second).

5. Discussion

The analysis described in this paper was carried out using the Bern-Barcelona dataset. Thus, all related works in the bibliography, selected for the sake of comparison, use the same database. That provides a more comparable framework, given that the use of works using other datasets may expose significant differences when performing the same classification method. As an example, San-Segundo et al. [25] shows that the focal-nonfocal (F-NF) classification accuracy may differ more than 20% when the same methods applies to the Bern-Barcelona dataset [7] and the Epileptic Seizure Recognition dataset [46]. In this example, the nature of the signals in the dataset, mainly the difference between signal lengths (only 1s for the latter), makes the difference. Thus, note that the Bern-Barcelona dataset is the logical selection when facing just the F-NF problem, provided its longer signal length (20 s) and its specialization in inter-ictal signals (recordings of seizure activity and three hours after the last seizure activity are excluded).

Table 5 compares our results with previous works for the task of classifying the focal and non-focal signals using the Bern-Barcelona EEG dataset. It details the obtained classification accuracy, summarizes the extracted features, and indicates the type of classifier used in each work.

Note that the purpose of this work was not to beat the accuracy results of previous works in the bibliography; our goal was to obtain, at the same time, the best classification procedure having the lowest possible computational load (for feature extraction and classification), aiming at its real-time implementation. Despite this, we obtained pretty good accuracy results. The 95.5% classification accuracy obtained in this work surpasses many other related works [13,15–18,20–22,47], while some other works [19,23,24,48] surpasses this result by a maximum of 1.5% classification accuracy ([19] achieved 97% classification accuracy). Thus, when not considering [25], the comparison with the other related works can be considered pretty good in light of the great simplification achieved for the feature extraction process. In turn, San-Segundo et al. [25] obtained to 98.6% classification accuracy (3.1% more than this work), but at the expense of using a computationally intensive tool, a deep neural

network, which is far away from the simplicity sought in this work; that is the reason to exclude this work from comparison from this point forward.

Table 5. Comparison of focal and non-focal classification results with previous works.

Authors	Classifier Type	Extracted Feature	Accuracy
Sharma et al. (2015) [17]	LS-SVM	EMD with IMFs	87.0%
Das et al. (2016) [18]	k-NN	EMD-DWT, log-energy entropy	89.4%
Chatterjee et al. (2017) [21]	SVM, k-NN	MF DFA	92.2%
Singh et al. (2017) [22]	LS-SVM	DFT based filter bank	89.7%
Sharma et al. (2017) [16]	LS-SVM	TQWT ¹	95.0%
Sharma et al. (2017) [13]	LS-SVM	Wavelet based entropies	94.3%
Bhattacharyya et al. (2017) [15]	LS-SVM	Fuzzy entropy of TQWT	84.7%
Bhattacharyya et al. (2018) [20]	LS-SVM	Rhythm separation from EWT	90.0%
Taran et al. (2018) [23]	ELM	Clustering VMD	96.0%
Deivasigamani et al. (2018) [24]	ANFIS	Extracted from CWT	96.0%
Zeng et al. (2019) [19]	RBF-NN	EMD and PSR	97.0%
San-Segundo et al. (2019) [25]	DNN	Fourier transform	95.5%
San-Segundo et al. (2019) [25]	DNN	Wavelet transform	97.4%
San-Segundo et al. (2019) [25]	DNN	Raw data	98.6%
San-Segundo et al. (2019) [25]	DNN	6 IMFs from EMD	98.9%
Rahman et al. (2019) [47]	Stacked SVM	VMD, DWT, and others ²	95.2%
Sharma et al. (2019) [48]	SVM	LSDA from bispectrum	96.2%
<i>This work</i>	<i>Neural Network</i>	<i>5 low computational complexity features extracted in real-time</i>	<i>95.5%</i>

¹ With different kernel function. ² Other features: refined composite multiscale dispersion entropy (RCMDE), refined composite multiscale fuzzy entropy (RCMFE) and autoregressive model coefficients (AR).

As it can be seen in Table 5, extracted features were used to proceed from computationally intensive processes, such as the decomposition of EEG signals using empirical mode decomposition (EMD) to extract intrinsic mode functions (IMFs). Thus, Sharma et al. [17] obtained 87.0% classification accuracy using five entropy features extracted from IMFs; Das et al. [18] also used entropy-based features from the EMD, DWT (discrete wavelet transform) and EMD-DWT domains, achieving 89.4% classification accuracy; and Zeng et al. [19] arrived to a 97% classification accuracy using features derived from Euclidean measures obtained from the phase space reconstructions (PSRs) of several IMFs obtained using EMD.

The wavelet transform has been also a computationally intensive process used in related works as the basis of the feature extraction process. Thus, Sharma et al. [16] obtained several entropy features from the tunable-Q wavelet transform (TQWT), reporting a 95.0% classification accuracy; Bhattacharyya et al. [20] obtained 90.0% classification accuracy using as features, projections of the reconstructed phase space (RPS) from the rhythm separation achieved using the empirical wavelet transform (EWT); Sharma et al. [13] obtained 94.25% classification accuracy from various wavelet based entropies; Bhattacharyya et al. [15] obtained 84.67% classification accuracy using TQWT based multivariate sub-band fuzzy entropy with LS-SVM classifiers; and Deivasigamani et al. [24] obtained 96.0% classification accuracy based on a set of features extracted from the dual tree complex wavelet transform (DT-CWT) and using an adaptive neuron fuzzy interference system (ANFIS).

Other works use variational mode decomposition as the basis of feature extraction, such as Rahman et al. [47], who obtained a 95.2% classification accuracy using features such as refined composite multi scale dispersion entropy (RCMSDE), refined composite multiscale fuzzy entropy (RCMSFE) and autoregressive model (AR) coefficients extracted from variational mode decomposition (VMD), DWT and VMD-DWT domains; or Taran et al. [23], who obtained 96.0% classification accuracy using spectral moment based features extracted from the modes of the clustering VMD (CVMD) and extreme learning machine classifiers.

Note that the computational cost of the feature extraction process in previous works is greater than the computational cost of the feature set proposed in our work. The only work in the bibliography having a computational cost comparable to that of our work is Singh et al. [22], that obtained a 89.7% of classification accuracy deriving features from DFT-based rhythms of the EEG. In the same way, we have to compute the DFT too. Nevertheless, we obtain a better classification accuracy (5.8% more).

Thus, despite its simplicity, a classification procedure that performs better than most of related works, or, in the worst case, got surpassed by a maximum of 1.5% of classification accuracy (not considering [25]) was achieved.

However, this work does not propose just an optimum feature set, but an optimum classification procedure, as a whole. Thus, the proposed feature set, Table 2, can be combined optimally with a neural network classifier model when five seconds of segment length are used (Section 4.1). Our analysis indicates that this, altogether, guarantees the best accuracy performance with a minimum computational cost. The proposed neural network is of perceptron type, with just one hidden layer of 25 hidden neurons, five input neurons (the dimensionality of the proposed feature set) and two output neurons in the output layer.

On the other hand, the FPGA real-time implementation of the classification procedure, following Algorithms 1 and 2, has been done using a sequential architecture. The benefits of using this architecture are the minimization of the memory usage and the number of arithmetic hardware blocks. Anyway, to improve the throughput results this computation can be easily parallelized.

The proposed hardware design allows the definition of floating point arithmetic units of single or double data type precision (following the IEEE 754 standard for floating-point arithmetic). As it was expected, both designs offer a great MAE accuracy. “Double” design achieves an accuracy similar to the Matlab environment implementation: 3.94×10^{-15} , while the “float” design offers a great accuracy: 1.00×10^{-6} . MAE accuracy was measured using the Matlab implementation as a reference.

The analysis demonstrates that the proposed hardware implementation does not use many resources. Both designs need no more than 137 DSP slices, while BRAM usage is 95 MB for the “single” design (doubling to 185 MB the requirements for the “double” design, provided that, obviously, the 64-bit double precision data types doubles the memory needs of the single 32-bit data type).

Note that the computational needs of the implementation in a Virtex 7 Xilinx FPGA device requires a reduced portion of its total resources; see Table 4. In fact, this application can be executed even on a small and cost-effective Xilinx XC6SLX100 Spartan-6 FPGA, assuring a low-cost of implementation.

Regarding the performance, it has been shown that the proposed implementation can perform all computation tasks at a maximum of 3.55 Hz when using the single data types, or 2.53 Hz when using double data types. That means that the single design can deliver outputs at a rhythm of 3.55 times by second.

However, note that this 3.55 Hz of classification frequency (2.53 Hz for double design) is only effective after the first 5 s of acquisition time due to the 5 s segmentation. This implies a minimum latency time to achieve the first result from the beginning of an acquisition without artifacts of the EEG signal. From this 5 s latency, the proposed implementation is capable of handling overlapped time windows, delivering results at the maximum classification frequency. Thus, for the single design, a result will be provided each $1/3.55 = 0.28$ seconds after the first 5 s window length.

Thus, we have shown that an adequate selection of the set of features, classifier model and length of the window segment, allows one to obtain good classification accuracy results (above the average of previous related works) while maintaining a low computational load for the whole classification procedure. It enables us to move the classification procedure to the real-time field, embedded in a logic-reconfigurable FPGA.

The proposed implementation can be carried out on a small portable device embedding a fast classification engine of epileptogenic focus. This device can serve as a help decision tool to assist neurosurgeons to refine the localization of the epileptogenic area during the resective epilepsy surgery in those cases where greater precision or confirmation were needed.

6. Conclusions

The locating of the epileptogenic focus using interictal EEG signals is generally a computerized analysis carried out off-line by neurosurgeons and epileptologists to determine the brain regions responsible for the initial seizure discharge. However, previous works tend to propose more computationally costly procedures the more recent they are.

This work shows that an adequate selection of the set of features, classifier model and length of the window segment, allows one to obtain good classification accuracy results (above the average) while maintaining a low computational load. It enables the real-time implementation of the whole classification procedure, on an FPGA reconfigurable device, from accepting the raw EEG signals to the delivery of the classification outputs at a rhythm of up to 3.55 Hz. It opens the door to the use of the automatic classification as a decision-assisting tool during surgery, enabling neurosurgeons to refine the localization of the epileptogenic area during the resective epilepsy surgery.

Concluding, it has been shown that the proposed hardware implementation of the epileptogenic foci locator can be embedded on a small portable device, embedding, thus, a fast classification engine of epileptogenic signals in epilepsy.

Author Contributions: Individual contributions: conceptualization, J.F.G.-M., A.G.M., V.V., K.G.H. and V.T.-M.; methodology, J.F.G.-M., J.V.F.-V. and M.B.-M.; software, J.F.G.-M., A.M. and J.V.F.V.; formal analysis, J.F.G.-M., J.V.F.-V., A.M., M.B.-M. and A.R.-M.; writing—original draft preparation, J.V.F.-V.; writing—review and editing, J.V.F.-V., A.R.-M. and M.B.-M. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

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

Abbreviations

The following abbreviations are used in this manuscript:

EMD	Empirical mode decomposition
IMF	Intrinsic mode function
DWT	Discrete wavelet transform
MF DFA	Multifractal detrended fluctuation analysis
DFT	Discrete Fourier transform
TQWT	Tuneable-Q wavelet transform
EWT	Empirical wavelet transform
ELM	Extreme learning machine
VMD	Variational mode decomposition
ANFIS	Adaptive neuro fuzzy interference system
RBF-NN	Radial basis function neural network
PSR	Phase state reconstruction
DNN	Deep neural network
LSDA	Locality sensitive discriminant analysis

References and Notes

1. Fisher, R.S.; Acevedo, C.; Arzimanoglou, A.; Bogacz, A.; Cross, J.H.; Elger, C.E.; Engel, J., Jr; Forsgren, L.; French, J.A.; Glynn, M.; et al. ILAE Official Report: A practical clinical definition of epilepsy. *Epilepsia* **2014**, *55*, 475–482. [[CrossRef](#)] [[PubMed](#)]
2. Falco-Walter, J.J.; Scheffer, I.E.; Fisher, R.S. The new definition and classification of seizures and epilepsy. *Epilepsy Res.* **2018**, *139*, 73–79. [[CrossRef](#)] [[PubMed](#)]
3. Daroff, R.B.; Jankovic, J.; Mazziotta, J.C.; Pomeroy, S.L. *Bradley's Neurology in Clinical Practice E-Book*; Elsevier Health Sciences, 2016; ISBN 9780323287838.
4. Ray, A.; Tao, J.X.; Hawes-Ebersole, S.M.; Ebersole, J.S. Localizing value of scalp EEG spikes: A simultaneous scalp and intracranial study. *Clin. Neurophysiol.* **2007**, *118*, 69–79. [[CrossRef](#)] [[PubMed](#)]

5. Bast, T.; Boppel, T.; Rupp, A.; Harting, I.; Hoechstetter, K.; Fauser, S.; Schulze-Bonhage, A.; Scherg, M. Noninvasive source localization of interictal EEG spikes: effects of signal-to-noise ratio and averaging. *J. Clin. Neurophysiol.* **2006**, *23*, 487–497. [[CrossRef](#)] [[PubMed](#)]
6. Dworetzky, B.A.; Reinsberger, C. The role of the interictal EEG in selecting candidates for resective epilepsy surgery. *Epilepsy Behav.* **2011**, *20*, 167–171. [[CrossRef](#)] [[PubMed](#)]
7. Andrzejak, R.G.; Schindler, K.; Rummel, C. Nonrandomness, nonlinear dependence, and nonstationarity of electroencephalographic recordings from epilepsy patients. *Phys. Rev.* **2012**, *86*, 046206. [[CrossRef](#)]
8. Wilson, S.B.; Turner, C.A.; Emerson, R.G.; Scheuer, M.L. Spike detection II: automatic, perception-based detection and clustering. *Clin. Neurophysiol.* **1999**, *110*, 404–411. [[CrossRef](#)]
9. Liu, D.; Wang, Q.; Zhang, Y.; Liu, X.; Lu, J.; Sun, J. FPGA-based real-time compressed sensing of multichannel EEG signals for wireless body area networks. *Biomed. Signal Process. Control.* **2019**, *49*, 221–230. [[CrossRef](#)]
10. Mjahad, A.; Rosado-Muñoz, A.; Bataller-Mompeán, M.; Francés-Villora, J.; Guerrero-Martínez, J. Ventricular Fibrillation and Tachycardia detection from surface ECG using time-frequency representation images as input dataset for machine learning. *Comput. Methods Programs Biomed.* **2017**, *141*, 119–127. [[CrossRef](#)]
11. Mjahad, A.; Rosado-Muñoz, A.; Guerrero-Martínez, J.F.; Bataller-Mompeán, M.; Francés-Villora, J.V.; Dutta, M.K. Detection of Ventricular Fibrillation Using the Image from Time-Frequency Representation and Combined Classifiers without Feature Extraction. *Appl. Sci.* **2018**, *8*. [[CrossRef](#)]
12. Mjahad, A.; Muñoz, A.R.; Mompeán, M.B.; Villora, J.V.F.; Martínez, J.F.G. Detección de Fibrilación Ventricular Mediante Tiempo-Frecuencia y Clasificador KNN sin Extracción de Parámetros. *Rev. Iberoam. Automática Informática Ind.* **2017**, *15*, 124–132. [[CrossRef](#)]
13. Sharma, M.; Dhere, A.; Pachori, R.B.; Acharya, U.R. An automatic detection of focal EEG signals using new class of time–frequency localized orthogonal wavelet filter banks. *Knowledge-Based Syst.* **2017**, *118*, 217–227. [[CrossRef](#)]
14. Sriraam, N.; Raghu, S. Classification of Focal and Non Focal Epileptic Seizures Using Multi-Features and SVM Classifier. *J. Med Syst.* **2017**, *41*, 160. [[CrossRef](#)]
15. Bhattacharyya, A.; Pachori, R.; Acharya, U. Tunable-Q wavelet transform based multivariate sub-band fuzzy entropy with application to focal EEG signal analysis. *Entropy* **2017**, *19*, 99. [[CrossRef](#)]
16. Sharma, R.; Kumar, M.; Pachori, R.B.; Acharya, U.R. Decision support system for focal EEG signals using tunable-Q wavelet transform. *J. Comput. Sci.* **2017**, *20*, 52–60. [[CrossRef](#)]
17. Sharma, R.; Pachori, R.; Acharya, U. Application of entropy measures on intrinsic mode functions for the automated identification of focal electroencephalogram signals. *Entropy* **2015**, *17*, 669–691. [[CrossRef](#)]
18. Das, A.B.; Bhuiyan, M.I.H. Discrimination and classification of focal and non-focal EEG signals using entropy-based features in the EMD-DWT domain. *Biomed. Signal Process. Control.* **2016**, *29*, 11–21. [[CrossRef](#)]
19. Zeng, W.; Li, M.; Yuan, C.; Wang, Q.; Liu, F.; Wang, Y. Classification of focal and non focal EEG signals using empirical mode decomposition (EMD), phase space reconstruction (PSR) and neural networks. *Artif. Intell. Rev.* **2019**, *52*, 625–647. [[CrossRef](#)]
20. Bhattacharyya, A.; Sharma, M.; Pachori, R.B.; Sircar, P.; Acharya, U.R. A novel approach for automated detection of focal EEG signals using empirical wavelet transform. *Neural Comput. Appl.* **2018**, *29*, 47–57. [[CrossRef](#)]
21. Chatterjee, S.; Pratiher, S.; Bose, R. Multifractal detrended fluctuation analysis based novel feature extraction technique for automated detection of focal and non-focal electroencephalogram signals. *IET Sci. Meas. Technol.* **2017**, *11*, 1014–1021. [[CrossRef](#)]
22. Singh, P.; Pachori, R.B. Classification of focal and nonfocal EEG signals using features derived from Fourier-based rhythms. *J. Mech. Med. Biol.* **2017**, *17*, 1740002. [[CrossRef](#)]
23. Taran, S.; Bajaj, V. Clustering variational mode decomposition for identification of focal EEG signals. *IEEE Sens. Lett.* **2018**, *2*, 1–4. [[CrossRef](#)]
24. Deivasigamani, S.; Senthilpari, C.; Yong, W.H. Computer Aided Automatic Detection and Classification of EEG Signals for Screening Epilepsy Disorder. *J. Inf. Sci. Eng.* **2018**, *34*, 687–700.
25. San-Segundo, R.; Gil-Martín, M.; D’Haro-Enríquez, L.F.; Pardo, J.M. Classification of epileptic EEG recordings using signal transforms and convolutional neural networks. *Comput. Biol. Med.* **2019**, *109*, 148–158. [[CrossRef](#)]
26. Daoud, H.; Bayoumi, M. Deep Learning Approach for Epileptic Focus Localization. *IEEE Trans. Biomed. Circuits Syst.* **2019**. [[CrossRef](#)]

27. Frances-Villora, J.; Rosado-Muñoz, A.; Bataller-Mompean, M.; Barrios-Aviles, J.; Guerrero-Martinez, J. Moving Learning Machine towards Fast Real-Time Applications: A High-Speed FPGA-Based Implementation of the OS-ELM Training Algorithm. *Electronics* **2018**, *7*. [[CrossRef](#)]
28. Abbaszadeh, A.; Iakymchuk, T.; Bataller-Mompeán, M.; Francés-Villora, J.V.; Rosado-Muñoz, A. An Scalable Matrix Computing Unit Architecture for FPGA, and SCUMO User Design Interface. *Electronics* **2019**, *8*. [[CrossRef](#)]
29. Fasil, O.K.; Rajesh, R.; Thasleema, T.M. Influence of differential features in focal and non-focal EEG signal classification. In Proceedings of the 2017 IEEE Region 10 Humanitarian Technology Conference (R10-HTC), Dhaka, Bangladesh, 21–23 December 2017; IEEE: Dhaka, Bangladesh, 2017; pp. 646–649. [[CrossRef](#)]
30. Raghu, S.; Sriraam, N.; Kumar, G.P. Classification of epileptic seizures using wavelet packet log energy and norm entropies with recurrent Elman neural network classifier. *Cogn. Neurodyn.* **2017**, *11*, 51–66. [[CrossRef](#)]
31. Aarabi, A.; Wallois, F.; Grebe, R. Automated neonatal seizure detection: A multistage classification system through feature selection based on relevance and redundancy analysis. *Clin. Neurophysiol.* **2006**, *117*, 328–340. [[CrossRef](#)]
32. Bedeuzzaman, M.; Farooq, O.; Khan, Y.U. Automatic seizure detection using higher order moments. In Proceedings of the 2010 International Conference on Recent Trends in Information, Telecommunication and Computing, Kerala, India, 12–13 March 2010; pp. 159–163.
33. Greene, B.; Faul, S.; Marnane, W.; Lightbody, G.; Korotchikova, I.; Boylan, G. A comparison of quantitative EEG features for neonatal seizure detection. *Clin. Neurophysiol.* **2008**, *119*, 1248–1261. [[CrossRef](#)]
34. Aarabi, A.; Grebe, R.; Wallois, F. A multistage knowledge-based system for EEG seizure detection in newborn infants. *Clin. Neurophysiol.* **2007**, *118*, 2781–2797. [[CrossRef](#)]
35. Hernández, D.; Trujillo, L.; Z-Flores, E.; Villanueva, O.; Romo-Fewell, O. Detecting Epilepsy in EEG Signals Using Time, Frequency and Time-Frequency Domain Features. In *Computer Science and Engineering—Theory and Applications*; Springer: Berlin/Heidelberg, Germany, 2018; pp. 167–182.
36. Rosado Muñoz, D.A. Desarrollo de Técnicas de Detección de Fibrilación Ventricular Basadas en Algoritmo Tiempo–Frecuencia. Ph.D. Thesis, Universidad de Valencia, Valencia, Spain, 2011.
37. Sharma, N.; Gedeon, T. Objective measures, sensors and computational techniques for stress recognition and classification: A survey. *Comput. Methods Programs Biomed.* **2012**, *108*, 1287–1301. [[CrossRef](#)]
38. Petrantonakis, P.C.; Hadjileontiadis, L.J. Emotion recognition from EEG using higher order crossings. *IEEE Trans. Inf. Technol. Biomed.* **2009**, *14*, 186–197. [[CrossRef](#)]
39. Urbanowicz, R.J.; Meeker, M.; La Cava, W.; Olson, R.S.; Moore, J.H. Relief-based feature selection: Introduction and review. *J. Biomed. Inform.* **2018**, *85*, 189–203. [[CrossRef](#)]
40. Aydın, S.; Saraoğlu, H.M.; Kara, S. Log energy entropy-based EEG classification with multilayer neural networks in seizure. *Ann. Biomed. Eng.* **2009**, *37*, 2626. [[CrossRef](#)]
41. Şen, B.; Peker, M.; Çavuşoğlu, A.; Çelebi, F.V. A comparative study on classification of sleep stage based on EEG signals using feature selection and classification algorithms. *J. Med. Syst.* **2014**, *38*, 18. [[CrossRef](#)]
42. Xilinx Vivado Design Suite. High-Level Synthesis User Guide, UG902 (v2016.1) 6 April. Xilinx. 2011.
43. ARM AMBA AXI and ACE Protocol Specification: AXI3, AXI4, and AXI4-Lite, ARM IHI 0022E (ID022613). Xilinx. 2011.
44. Xilinx. Vivado Design Suite. AXI Reference Guide, UG1037 v3.0, 24 June. Xilinx. 2015.
45. ARM. AMBA AXI4-Stream Protocol, ARM IHI 0051A (ID030510). Xilinx. 2010.
46. Andrzejak, R.G.; Lehnertz, K.; Mormann, F.; Rieke, C.; David, P.; Elger, C.E. Indications of nonlinear deterministic and finite-dimensional structures in time series of brain electrical activity: Dependence on recording region and brain state. *Phys. Rev.* **2001**, *64*, 061907. [[CrossRef](#)]
47. Rahman, M.M.; Bhuiyan, M.I.H.; Das, A.B. Classification of focal and non-focal EEG signals in VMD-DWT domain using ensemble stacking. *Biomed. Signal Process. Control.* **2019**, *50*, 72–82. [[CrossRef](#)]
48. Sharma, R.; Sircar, P.; Pachori, R.B. A New Technique for Classification of Focal and Nonfocal EEG Signals Using Higher-Order Spectra. *J. Mech. Med. Biol.* **2019**, *19*, 1940010. [[CrossRef](#)]

