III. Methodology

Visual examination of electroencephalographic or electrocorticographic recordings readily ascertain various types of seizures and its propagation within the brain. However, such a process is painstakingly slow and error-prone due to fatigue. Much effort has to be expended to automate the process of detecting seizures by probing through various dynamic properties of EEG waveforms. Effective monitoring and analysis of EEG signals are used in identifying epileptic seizure onset, which may reduce morbidity or mortality by expounding the best intervention. Large amounts of multi-channel EEG signals are visually analyzed by neurologists with a goal. In order to obtain fast and balanced EEG analysis, automation in detecting seizures is mandatory. Figure. 3.1 represent the functioning of the Automated Seizure Detection system.

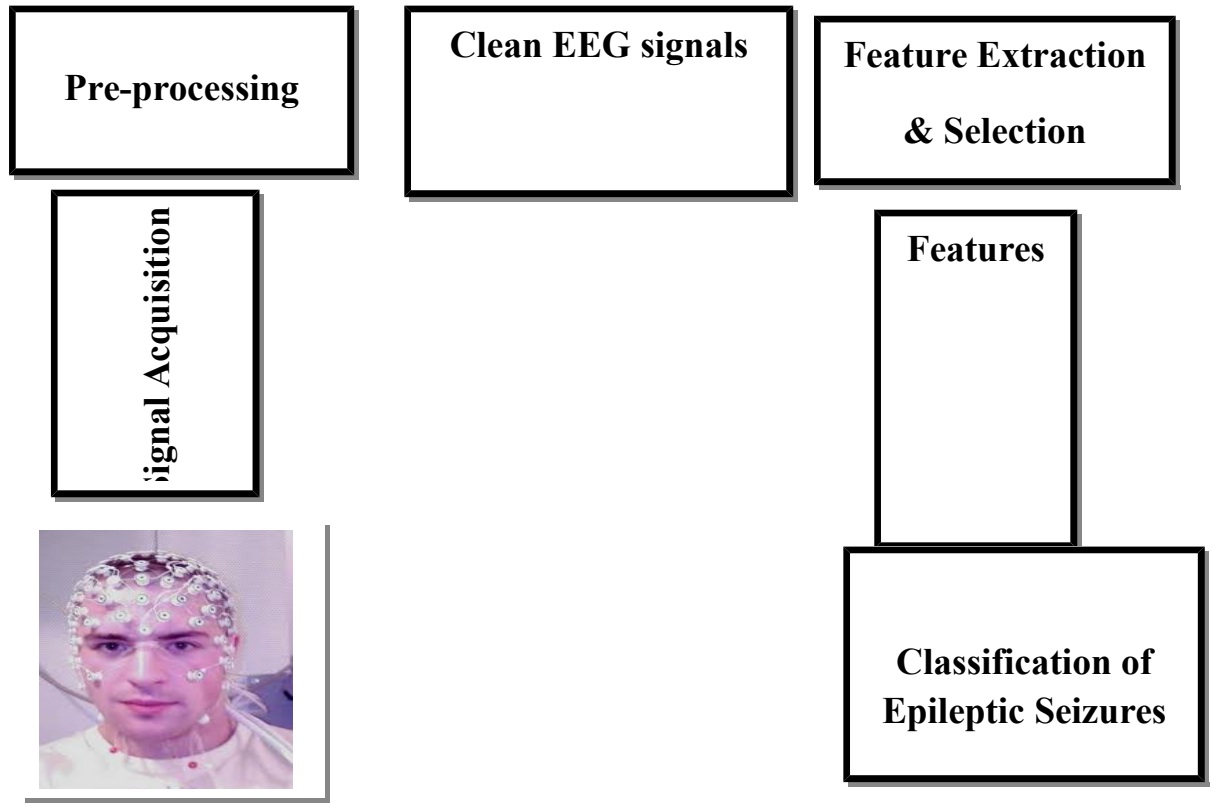


Figure. 3.1 Taxonomy of Automated Seizure Detection

3.1. Signal Acquisition

An evolution in the EEG amplifiers and recording instruments has been witnessed in the past 20 years. Paper recording has been supplanted by computer digitization and storage. Scalp EEG gives prima facie evidence of neuronal synchrony of underlying cortical activity. EEG sensors used for signal acquisition follows a 10/20 system, a globally recognized method of electrode placement. A relationship between the location of an electrode and the underlying area of the cerebral cortex is derived through this system. These signals are stored in a digital format. Signals are characterized by their amplitude and their frequency specification. The amplitude of EEGs recorded by intra cerebral or subdural electrodes is found to be larger than that of scalp electrodes. The amplitude of the order of 20μV to 100 μV is registered when using scalp electrodes and of the order 100 μV to 2 MV is registered when using depth electrodes. The spectral bandwidth of EEG ranges from 0.5 Hz to about 500 Hz.

It is mandatory to undergo some important considerations regarding amplifier and filter settings for EEG data acquisition in order to record data suitable for time-frequency analysis. One of the important concerns is to sample EEG signals at a faster rate to avoid frequency aliasing of the signal. The effects of under sampling are the main cause of aliasing which causes the misrepresentation of a high-frequency signal as a low frequency signal. The minimum sampling rate which is needed to avoid aliasing is known as the Nyquist rate. It is twice as fast as the highest frequency of interest, even though most of the EEG acquisition software inflicts a higher standard such as a sampling rate that is 4 times the highest frequency of interest.

The dataset used in this research work was acquired from Sri Ramakrishna and PSG Hospitals, Coimbatore with a size supporting 160 patients yielding a dataset of 65, 53,600 samples for 10 seconds. A conventional portable EEG device with the standard placement guide for electrodes was used to obtain EEG measurements. The 10-20 system, which is an internationally recognized system for placement of electrodes, was adopted. It describes the relationship between positioning of electrodes and the underlying cerebral cortex. Letters are used to identify the lobes and numbers are used to identify the hemisphere. The letters F, T, C, P and O refer to

Frontal, Temporal, Central, Parietal and Occipital lobes of the brain. Even numbers 2, 4, 6 and 8 are used to represent electrode positions on the right hemisphere, whereas odd numbers 1, 3, 5 and 7 are used to represent the positioning on the left hemisphere. The raw EEG signal consists of 2 sets of data for 160 patients; one corresponding to pathology and the other being normal. The dataset contains 16 channel recording for 160 patients and the length of the recording is for 10 seconds. The data were sampled at a rate of 256 samples per second. Thus, the total number of samples present in the 16 channel recording from a single data set is equal to 4096, which while considering 160 patients yields a dataset of 65,53,600 samples for 10 seconds. Figure. 3.2 represents the taxonomy of the signal acquisition system.

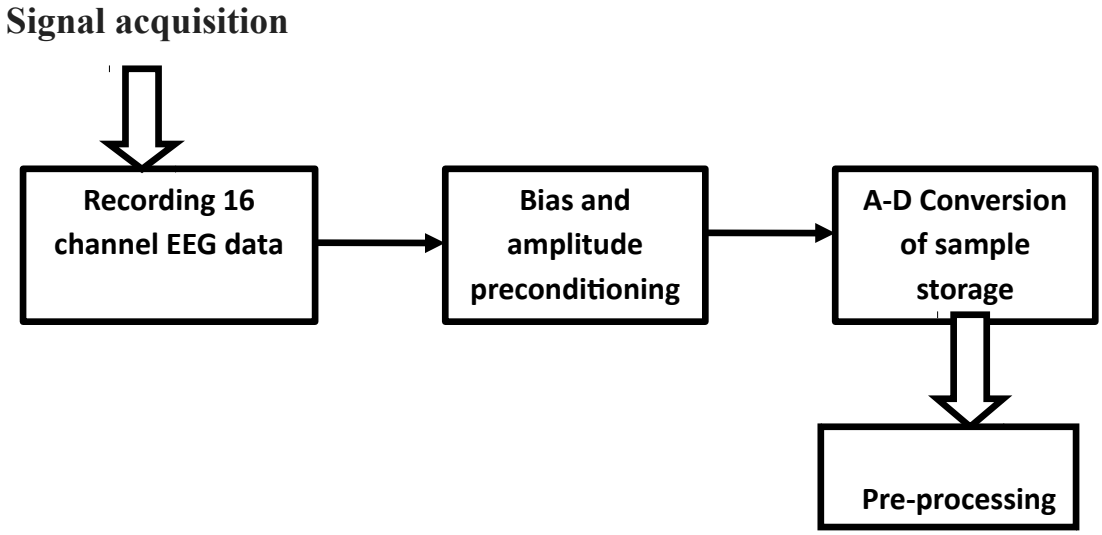


Figure 3.2 Taxonomy of Signal Acquisition

The signal acquisition system employs a bias and amplitude preconditioning, as they contribute greatly to the quality of the signals in terms of signal to noise ratio and strength of the neural signal by establishing predetermined voltage or current for the purpose of establishing proper operating conditions. It is important to review at this juncture that the noise generated by the state of art amplifiers is very small and negligible when compared to the EEG signals. Signals after amplification are stored in Sample and Hold (S&H) circuits and an analog multiplexer is used to scan the S&H outputs and the signals are converted sequentially to produce serial output signals using ADC.

3.1.1. Data Conversion

This research work used MATLAB 2010a for signal processing, development of algorithms and performance evaluation. The EEG signals acquired and stored as database are incompatible in format to work in MATLAB environment and hence needs a conversion. The signal files available from the EEG machine were of extension *.eeg , which provided a provision to export results to Microsoft Excel, making data convenient and user friendly to be exercised in MATLAB.

3.2. Proposed Methodology

To develop the proposed seizure detection system using EEG, a methodology comprising of various algorithms and techniques is categorized as follows.

1. Pre-processing - consists of various artifact removal techniques. Both internal as well as external artifacts are put into consideration.
2. Feature Extraction - comprises of methods that efficiently extract the spectral frequency bands and derives the feature based on power spectral density in each of these bands
3. Classification - encompasses algorithms and procedures to discriminate the seizure and non-seizure signals

Each of the above phases is dealt separately and the techniques that are used in each of these phases are refined for successful manoeuvre. It is also noteworthy that these phases are integrated such that the output of one phase feeds as an input to the second phase. The advantages obtained due to the artifact removal in the first phase have an impact on the final phase. The proposed ASDEEG framework has pre-processing as an optional argument, which can be switched on and off in order to emphasize the real need of artifact removal. The various methods and techniques used in the proposed system are presented in the Figure. 3.3.

It is substantially important to create a recording environment such as to minimize the potential of ambient artifacts. Steps must be taken to ensure the use of acoustically and electrically shielded cabin to support an optimal recording environment. Though steps are taken to reduce artifacts still the EEG signals are hampered by the presence of various artifacts that obscure the underlying authentic information necessary for

seizure classification. Hence it is essential to abdicate this artifactual information. Artifacts can be classified according to their region of origin and are broadly categorized as endogenous (physiological) and exogenous (extra-physiological) artifacts. When the artifacts have their origin as the subject's body it is termed as endogenous artifacts e.g. EOG, EMG, ECG etc. Exogenous artifacts have their origin from the surroundings i.e. Noise from the mains, spurious electrical noise, noise from elevators, engines, misplacement of electrodes etc. The techniques and methods used for the removal of artifact are referred to as "Pre-processing Techniques", which play a major role in appropriately determining seizures or its absence.

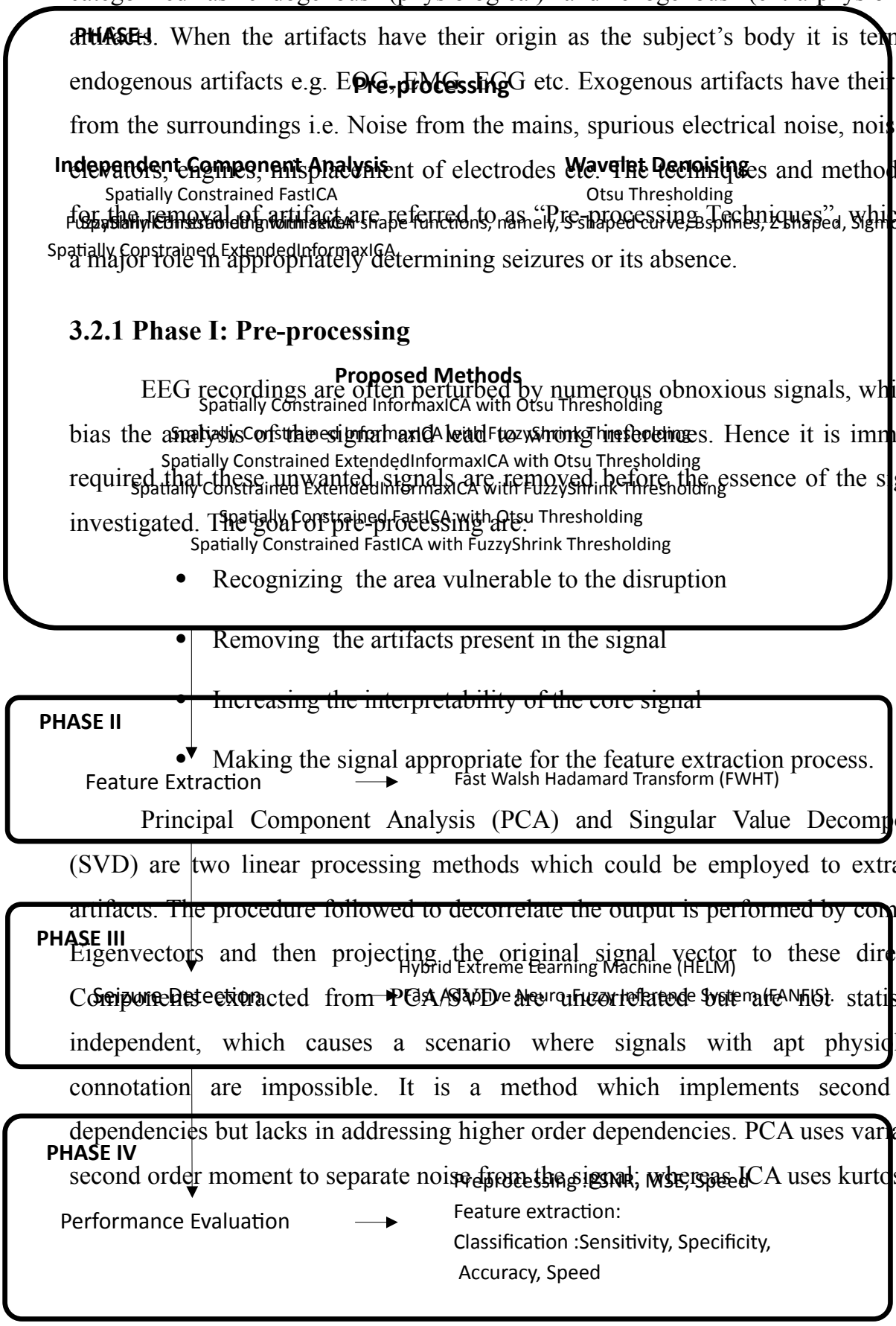


Figure. 3.3: Research Design

3.2.1 Phase I: Pre-processing

EEG recordings are often perturbed by numerous obnoxious signals, which can bias the analysis of the signal and lead to wrong inferences. Hence it is immensely required that these unwanted signals are removed before the essence of the signal is investigated. The goal of pre-processing are:

- Recognizing the area vulnerable to the disruption
- Removing the artifacts present in the signal
- Increasing the interpretability of the core signal
- Making the signal appropriate for the feature extraction process.

Principal Component Analysis (PCA) and Singular Value Decomposition (SVD) are two linear processing methods which could be employed to extract the artifacts. The procedure followed to decorrelate the output is performed by computing Eigenvectors and then projecting the original signal vector to these directions. Components extracted from PCA/SVD are uncorrelated but are not statistically independent, which causes a scenario where signals with apt physiological connotation are impossible. It is a method which implements second order dependencies but lacks in addressing higher order dependencies. PCA uses variance, a second order moment to separate noise from the signal; whereas ICA uses kurtosis, a

fourth order moment (Clifford, 2005). The above mentioned limitations are overcome with ICA and hence are being implemented in this study.

3.2.1.1. Independent Component Analysis (ICA)

ICA is an embodiment of Blind Source Separation (BSS) and is used for establishing independence between sources. It has gained popularity in the field of bio-medical signal processing (James et al., 2005). The Figure. 3.4 represents the taxonomy of ICA (Irene winker et al., 2011).

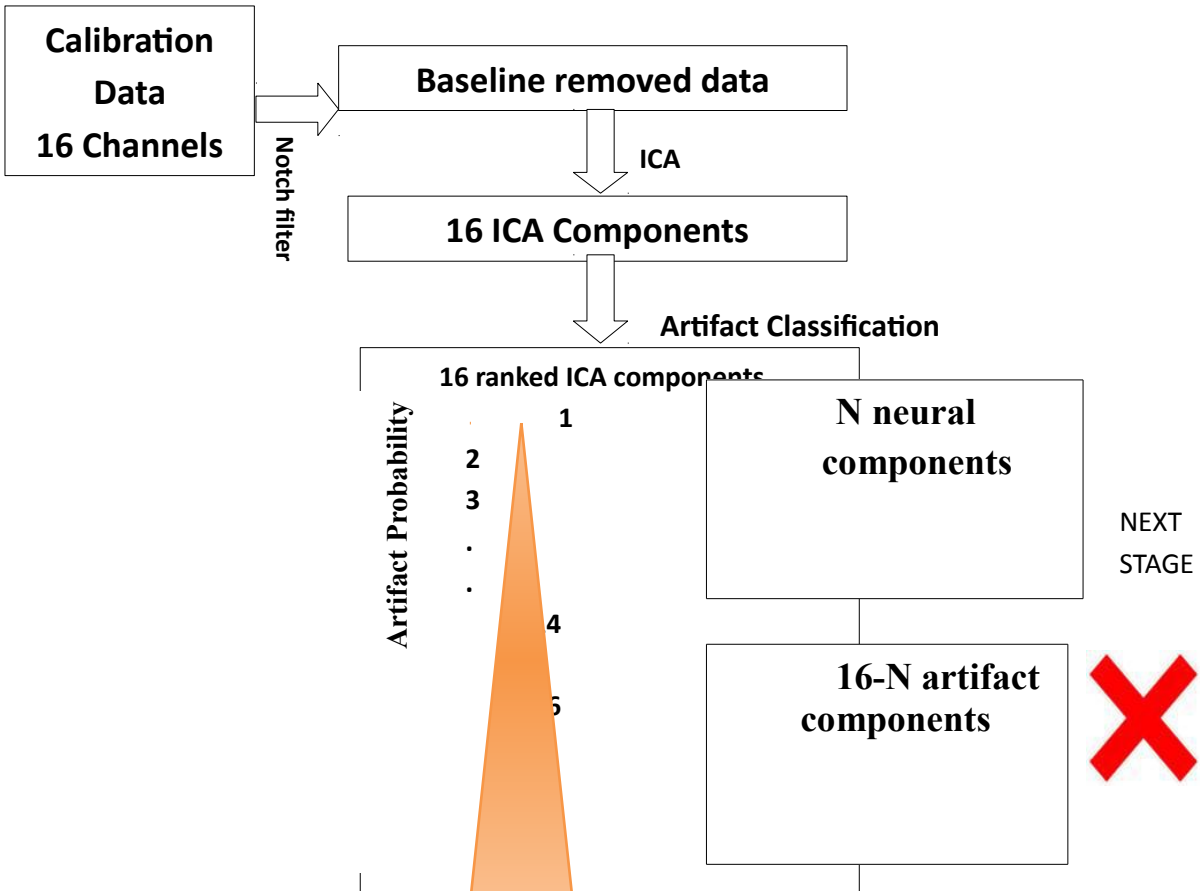


Fig 3.4 Taxonomy of ICA

It is used for separating multi-channel signals into their constituent underlying components. ICA model is highly associated with some important assumptions:

i) Statistical independence is assumed in case of underlying observed sources
ii) The observed linear mixture 'm' must be at least as large as the number of independent components 'n' i.e. $m \geq n$.
iii) The signal from each sensor has a different mixing ratio of the independent components

Hyvarinen et al. (2000) has defined ICA as "A statistical signal processing technique that models a set of observations, x, with an instantaneous linear mixing of independent latent variable's' ".

$$\mathbf{x(t) = As(t) + ns(t)} \qquad \mathbf{(1)}$$

where 'ns' represents additive noise over a time period 't'.

Let 'x' be the random vectors whose elements are the mixtures $x_1, ...,x_n$ and let 's' be the random vectors with the components $s_1,...,s_n$. Let 'A' be the matrix containing the elements 'a_{ij}'. The model can now be written as:

$$\mathbf{x=As \quad where \quad x=} \sum_{i=1}^{n} \mathbf{a_i S_i} \qquad \mathbf{(2)}$$

Here the problem is to determine both the matrix 'A' and the independent components 's' with the known value, the measured variables 'x'. The postulation undertaken is that component '$\mathbf{s_i}$'is independent and must have non-Gaussian distribution. The mixing matrix is a square matrix. As the ICA model is used to estimate sources s it can be rewritten in a simplified form as

$$\mathbf{s=Wx} \qquad \mathbf{(3)}$$

where 'W' represents the inverse of the estimated matrix 'A'.

There are several methods to measure the independence and each of these implicates different algorithms. There are two main families of ICA algorithms (Haykin, 2009). Some algorithms are entrenched in the minimization of mutual information, while others are imbedded in the maximization of non-Gaussianity. Choosing different algorithms result in different unmixing matrices. In this study, Non-Gaussianity is used to measure the independence. Based on the assumption that each underlying source is not normally distributed, one of the methods used to extract the components is by forcing each of them to be far from the normal distribution.

3.2.1.1.1 InfomaxICA

One of the foremost pervasive neural network based approach is the INFOMAX algorithmic rule proposed by Bell and Sejnowski (1995), Nadal and Parga (1994). This technique uses gradient-based algorithm and this learning rule is based on the principle of information maximization (infomax), which maximizes the output entropy of a neural network. A fundamental consequence of information theory is that a Gaussian variable has the largest entropy among all and hence indicates that entropy is a measure of non-Gaussianity(Cardoso,1997).

Negentropy (or differential entropy) is defined as the difference between the entropy of a Gaussian random variable and the entropy of the observed variable which have the same variance. Negentropy is zero when the observed random variable is also Gaussian and positive when the observed variable is non-Gaussian. Negentropy 'J' is defined as follows:

$$J_{(y)} = H_{(ygauss)} - H_{(y)} \quad (5)$$

'$H_{(y)}$' is defined as the entropy for the discrete random variable 'Y' and '$H_{(ygauss)}$' is the entropy of a Gaussian random variable of the same co-variance matrix as y. The advantage of using negentropy or differential entropy, as a measure of non-Gaussianity is that it is well justified by statistical theory. Based on the concept of differential entropy, the mutual information 'I' between 'm' (scalar) random variables, 'yi' , i = 1...m as follows

$$I(y_1, y_2, \ldots, y_m) = \sum_{i=1}^{m} H(y\,i) - H(y). \quad (6)$$

The extracted signals 'y' are obtained from signal mixtures 'x' by optimizing an unmixing matrix W. In Infomax the extracted signals are source signals if they are mutually independent. Since the independence of the signals cannot be measured, entropy can be measured. Entropy is related to independence and hence maximum entropy implies independent signals. The main objective of ICA is to find the unmixing matrix 'W' that maximizes the entropy in the extracted signals 'y'. The entropy of the signal mixtures 'x' is constant, but the change in entropy can be maximized by mapping the signals y = Wx to an alternate set of signals

$$Y = g(y) = g(Wx). \quad (7)$$

The change in entropy from x→Y can be maximized by optimizing the unmixing matrix 'W', and when entropy is maximized, the resulting signals are independent. The inverse $y = g^{-1}(Y)$ is then taken, resulting in extracting signals 'y' that are also independent. When the extracted sets of signals 'y' are independent, it indicates that they are the original source signals 's' .The approximation works well when it comes to recovering super-Gaussian components, but fails to extract the components having a sub-Gaussian distribution, if such components exist in the mixture of non-Gaussians.

3.2.1.1.2. Extended Infomax ICA

The original Infomax approach lacks in recovering signals that have a sub-Gaussian distribution (Lee et al., 1999). In order to alleviate this deficiency, the Extended Infomax approach was developed to extract the sub-Gaussian sources, which is an extension of the infomax algorithm of Bell and Sejnowski (1995) that has the capability of separating mixed signals with sub-Gaussian and super-Gaussian distributions. A simple learning rule derived by Girolami (1997) was adopted by choosing negentropy as a projection pursuit index. Parameterized probability distribution functions that have sub-Gaussian and super-Gaussian systems were used to derive a general learning rule that preserves the simple architecture proposed by Bell and Sejnowski (1995). It was then optimized using the natural gradient by Amari et al (1996), and the stability analysis was performed as suggested by Cardoso and Laheld (1996) to switch between sub and super-Gaussian regimes.

The purpose of the extended infomax algorithm is to provide a simple learning rule with a fixed nonlinearity that can separate sources for a variety of distributions. One way of generalizing the learning rule with either sub or super Gaussian distributions is to approximate the estimated probability distribution function with an Edgeworth expansion or Gram-Charlier expansion (Stuart & Ord, 1987), as proposed by Girolami and Fyfe (1997b). A parametric density estimate was derived by Girolami to project the same learning rule without making any approximations.
The learning rule for super Gaussian sources is

$$\Delta W \propto \left[I - \tanh(u) u^T - u u^T \right] W \quad (8)$$

The learning rule for sub Gaussian sources is

$$\Delta W \propto \left[I + \tanh(u) u^T - u u^T \right] W \quad (9)$$

The difference between the super-Gaussian and the sub-Gaussian learning rule is the sign before the tanh function and can be determined using a switching criterion introduced by Girolami (1997).

3.2.1.1.3 FastICA

The FastICA algorithm was introduced in 1997 by Aapo Hyvarinen and Erkki Oja. It is a computationally highly efficient method which takes into account a neural network learning rule and converts it into a fixed-point iteration. The result perceived is that it is 10-100 times faster than conventional gradient descent methods for ICA. This algorithm is very simple and does not depend upon any user-defined parameters. It rapidly converges to the most accurate solution. It attempts to separate underlying sources from the measurement set based on 'non-Gaussianity'. The simple principle behind FastICA is that the fast fixed-point iterative algorithm undertakes to find projections that maximize the non-Gaussianity of components by their kurtosis or the fourth-order cumulant. It is known that, kurtosis is identically zero for Gaussian distributed signals. The aim is to maximize the magnitude of the kurtosis to make the estimated sources as non-Gaussian, thereby making it as independent as possible. The kurtosis that is used to describe the peak of a distribution is defined as

$$\text{kurt}(x) = E\{x^4\} - 3(E\{x^2\})^2 \quad (10)$$

The kurtosis for a Gaussian random variable is zero and for a non-Gaussian random variable it can vary between -3 and +3.

3.2.1.1.4 Modifications over ICA

The implications of different algorithms used for deriving the independence of components in the current research work are Infomax ICA, Extended Infomax ICA and FastICA. Enhancements to the algorithms are exercised by spatial constraints on

the above specified algorithms which results in Spatially Constrained Infomax ICA, Spatially Constrained Extended Infomax ICA and Spatially Constrained FastICA and are depicted in the Figure. 3.5

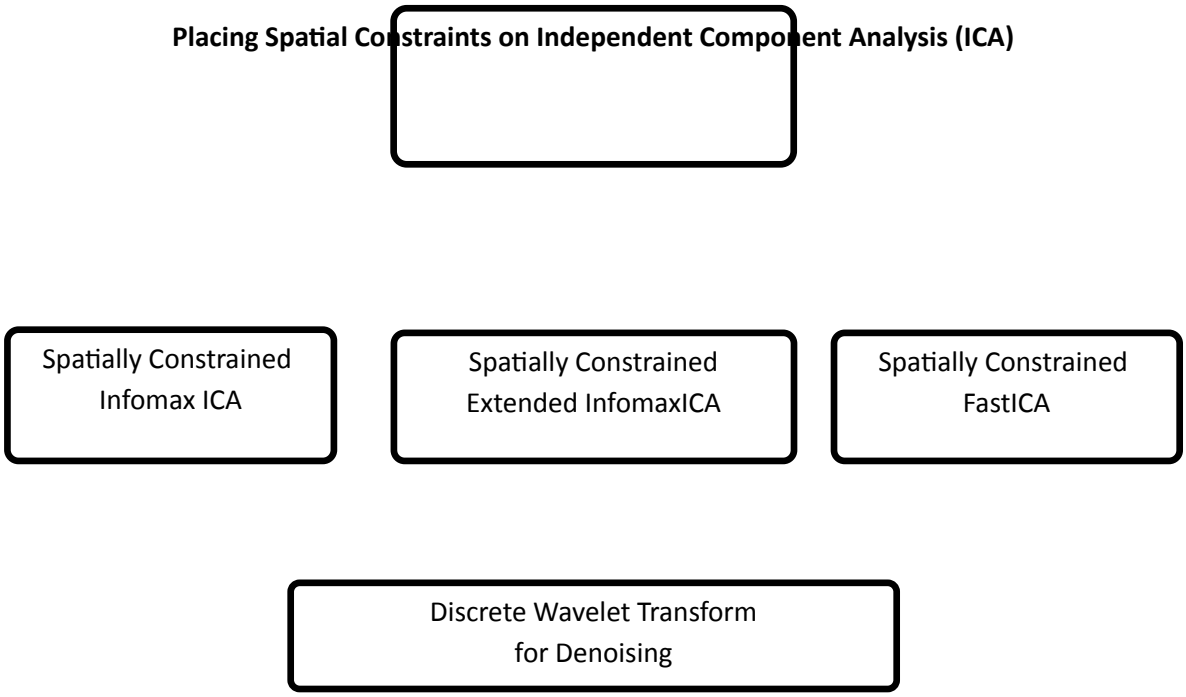


Figure. 3.5 Modifications over ICA

3.2.1.2 Discrete Wavelet transform

The Wavelet Transform (WT) is intended to achieve time frequency analysis of non-stationary signals. The signals represented by fixed building blocks are termed as wavelets and are derived from a single generating function called the mother wavelet by performing translation and scaling operations (Daubechies, 1990). Scaling summarizes the mother wavelet and translation shifts it along the time axis such that it has a varying window size which are broad at low frequencies and narrow at high frequencies, resulting with an optimal time-frequency resolution in all frequency ranges. Precisely wavelet is a mathematical means of performing time-scale analysis and is applicable to problems in which signals of differing spectral signatures are attributed to different parts of the waveform that are well localized in time and scale.

The WT can be broadly classified into continuous and discrete. The Discrete Wavelet Transform (DWT) is often used as CWT, involves substantial effort by

handling enormous amounts of data. The Discrete Wavelet Transform (DWT) has an established role in multi-scale processing of biomedical signals, such as EMG and EEG. The morphology of EEG signals is used for recognizing the functionality of the brain and it is indispensable to acquire the EEG parameters without artifacts in order to support appropriate clinical decision making. In order to address this issue, DWT plays a prominent role in noise reduction. EEG signals are converted into its appropriate DWT coefficients which hold back all pertinent information by eliminating the unwanted signals.

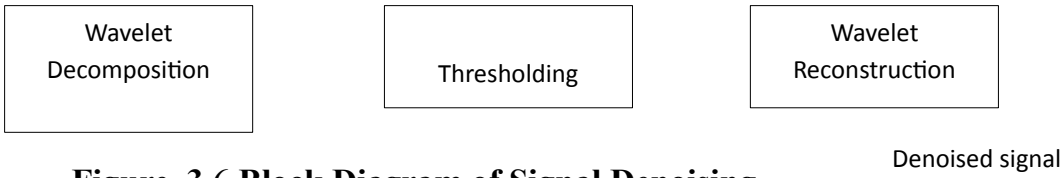


Figure. 3.6 Block Diagram of Signal Denoising

The process of signal denoising as depicted in Figure. 3.6 is explained in 3 steps as follows. i) DWT is applied to different physiological signals and the noise signals are decomposed. ii) An appropriate method such as hard or soft thresholding is adopted to remove or shrink the noisy wavelets. The above step is carried out by applying thresholding to each of the detail coefficient levels. iii) Reconstruction of the signal is next performed by using an inverse wavelet transform from the de-noised wavelet coefficient which is applied to high frequency wavelet sub-bands obtained as post-DWT. The coefficients that posses an absolute value greater than the threshold are considered to be a part of the information and those below the threshold are understood to be derived from noise. The noises coefficient are set to zero and a noise-free signal is reconstructed and used for signal detection. Several wavelet-based methods exist for unsupervised de-noising and detection of data with low signal-to-noise ratio. Each sub-band of wavelet processing, decorrelates successive noise related information from neural information using distinct time-frequency signatures. Some of the advantages of wavelets are discussed below:

a) It offers localization simultaneously both in time and frequency domain.
b) Computations are performed by wavelets rapidly.
c) They exhibit their exemplary performance by their ability to separate even

finer details in a signal. Large wavelets are used to identify coarser details and likewise the finer details by using small wavelets.

d) It is always possible to achieve a good approximation by merely using only fewer coefficients by using wavelets.

e) It has the capability in revealing certain aspects of the signals like trends, break down points and discontinuities present in higher derivatives , which could be missed by other signal analysis techniques.

3.2.1.3. Signal Denoising

It is noteworthy that noise corrupts the signals in a significant manner, and must be removed so as to proceed with further signal analysis. Signal denoising is termed as a process of noise removal from the signal. The task of denoising is based on two categories: denoising in the original signal domain (e.g., time or space) and denoising in the transform domain (e.g., Fourier or wavelet transforms) (Donoho et al. 1995). One of the approaches where the signal is compared with a threshold value and decided whether or not they could contribute to the part of the neural information is called thresholding. It is also worth-mentioning that thresholding is applied to the detailed part of the coefficient instead of the approximation part since the latter part usually contain low frequency components of the coefficient and contain significant information of the signal and are less affected by noise. By employing thresholding to the detail part, the coefficient below the threshold value are set to zero and is denoted by λ . The threshold wavelets are obtained using hard thresholding or soft thresholding.

Hard thresholding is otherwise referred to as 'wavelet thresholding', whereas soft thresholding is termed as shrinkage, since it shrinks the s of high amplitude towards zero. Hard thresholding follows the principle of keeping or kill.

- The coefficient which is less than or equal to the threshold value t_o are set to zero.

If (coeff [i] <= t_o)
Set coeff [i] = 0.0 (11)

Soft thresholding is based on the following rules:

- If the coefficient is less than or equal to the threshold value 't_o' then the coefficients are set to zero.
- If the coefficient is greater than the threshold value 't_o' then the threshold is subtracted from the coefficient

If (coeff [i] <= t_o)
Set coeff [i] = 0.0 (12)
else
Set coeff [i] = self [i] - t_o (13)

3.2.1.4 Hybrid model for pre-processing

In the first phase of the study, a hybrid model comprising of spatially constrained ICA with a wavelet-domain framework is proposed. The main intention of the study was to provide a scalable preprocessing algorithm so that it could adapt itself to remove any type of artifact. In order to facilitate the handling of voluminous matrix produced by ICA, and enabling data reduction, spatial constraints were implemented. Adequate corroborative and persuasive results remain unexplored in coalesce of BSS and WT and much affirmative results have not been reported on spatial constraints predominating BSS. The architecture of the proposed hybrid method for pre-processing of EEG data is presented in Figure. 3.7.

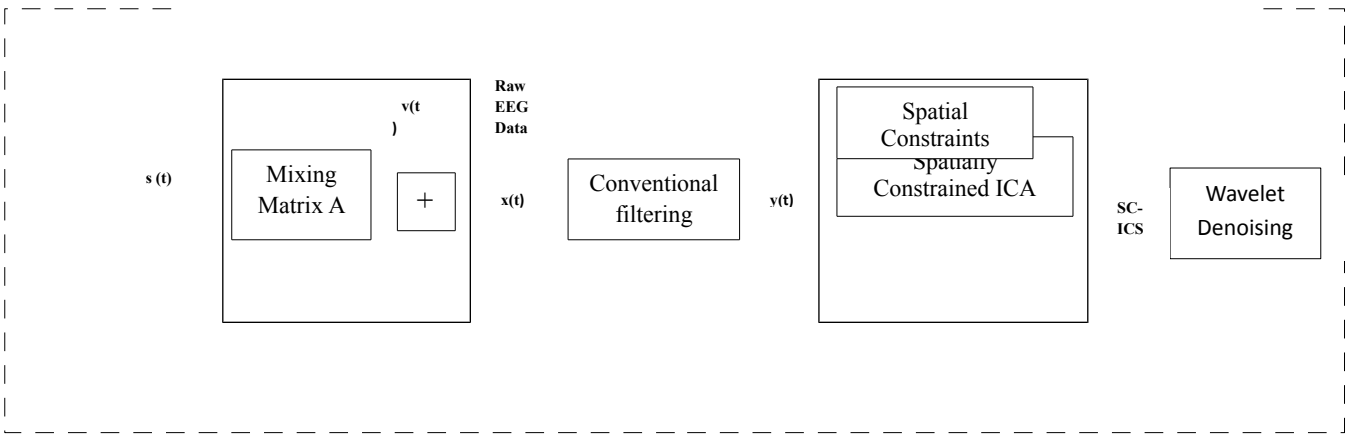


Figure. 3.7 Block Diagram of proposed Artifact removal algorithm

EEG data are implicated based on the ICA model as follows:

$$x(t) = As(t) + v(t) \tag{14}$$

where x (t) = [x_1 (t), x_2 (t), · · · , x_M (t)] ^{T} is the resultant signal based on a linear mixture of N sources s (t) = [s_1 (t), s_2 (t), · · · , s_n (t)] ^{T} , A is M×N mixing matrix, and v (t) = [v_1 (t), v_2 (t), · · · , VM (t)]^T is the additive noise at the EEG sensors. 'N' is the number of sources known, but the mixing matrix A and the waveforms s_i(t) are unknown. For the purpose of simplification a square mixing matrix is considered, where M = N. s_i (t) is constituted from signals arising from various regions of the brain which are embedded of artifacts. These artifacts encumber significant information on the brain, and are inappropriate for further examination and processing. The above requirements are satisfied by choosing the appropriate combination of algorithms. The proposed technique consists of following key processes:

Conventional filtering: The prime step in processing raw EEG data is by employing conventional filters. These filters are used to eliminate 50 Hz line noise, baseline values, sensor noise 'v (t)' and artifacts that dwell in very low frequency and high frequency zone. Interference of power line with a frequency of 50Hz causes spurious problems. High impedance of electrodes causes the wire running from them to function as antennae and has the possibility to pick up electrostatic noise. If possible, electrostatic noise should be reduced by shielding the power source. In order to satiate the removal of such interference a notch-filter with a block band at around 50Hz was implemented for this phase. A baseline correction is also performed so that the values of the signal are distributed around 0.

Spatially-Constrained ICA (SCICA): The accomplishments of ICA in medical signal processing with its due course the spatiotemporal data are becoming highly significant in EEG signal processing. The triumphant performance of spatially constrained ICA is due to a reduced size of the dataset. The performance of ICA deteriorates when the dataset becomes larger and results in an over complete ICA. There exist two different modalities by which ICA can be used to decompose the EEG's spatiotemporal data into a set of spatial or temporal ICs by using spatial ICA or temporal ICA. When prior information regarding the time-course of waveforms form

a reference, then such a reference acts as a basis for temporal constraint. Similarly when the prior information about the source sensor projections are used as a reference then it is termed as spatial constraint. Spatial ICA finds underlying independent spatial sources and the mixing matrix adopted here corresponds to a set of time sequences whereas the temporal ICA finds independent temporal sequences and the obtained mixing matrix corresponds to a set of spatial mode output. An enhancement in the performance of ICA is carried out by employing spatially constrained ICA. The motivation behind Spatially Constrained Independent Component Analysis (SCICA) is to provide a systematic and flexible method to incorporate more assumptions and prior information; so the ill-posed ICA, which has the compulsion to handle the enormous data, is converted to a better-posed problem which handles dimensionally reduced data. Incorporation of prior information avoids the problem of local minima and increases the quality of separation. The filtered and baseline corrected EEG data 'y(t)' is further processed with the application of SCICA Ille (2001). The idea behind the usage of spatial constraints SCICA is that it incorporates prior knowledge about spatial topographies and then extracts artifact-based independent components. Based on prior assumptions concerning the spatial topography of some source sensor projections Spatial Constraints (SC) are positioned on the matrix 'A'. The spatial constraints are deployed on the chosen columns of 'A' with reference to a set of predetermined constraint sensor projections, represented by 'A_c'. Thus, the spatially constrained mixing matrix consists of two types of columns

$$A = \left[\hat{A}_c, A_u \right] \tag{15}$$

where $\hat{A}_c \approx A_c$ are columns which are enforced with constraint, and 'A_u' are regarded as unconstrained columns. The predetermined sensor projections in this work are congregated by manual choice of sources extracted from a previous information segment with the help of existing ICA technique.

Based upon the accuracy of the constraint topographies 'A_c' and the level, to which constrained columns '$\hat{A}_c$' may diverge from reference 'A_c', it is possible to categorize the constraints as follows:

1) Hard constraints imposed to the columns are fixed.

2) Soft constraints are imposed within a small angular threshold α .It limits the divergence between constrained columns $\hat{A}_c$ and their corresponding topography which is represented in Figure. 3.8.

3) Weak constraints with a meagre initial approximation could be equated to an unconstrained estimation.

Soft Spatial constraints

Step 1: Consider two unit norm column vectors a_c and $\hat{a}_c$.

Step 2: Ensure that a_c and $\hat{a}_c$ subtends an absolute angle not greater than α, where α ranges between 0 to π /2, and is known to be the Euclidean space

Step 3: Compute the dot product of 2 Euclidean vectors which is the cosine of the angle between them.

Step 4: Based on the dot product compute the soft spatial constraints

If $\left|acos(a_c^T \hat{a})\right| \leq \alpha$ then $\hat{a}_c$ is not changed

else project $\hat{a}_c$ towards a_c such that $\left|acos(a_c^T \hat{a})\right| = \alpha$

$\widehat{H}_c$ symbolizes the constrained matrix and $\widehat{H}_u$ characterizes the unconstrained

matrix.

Step 2: Set $\widehat{H}_c = H_c$ and columns of $\widehat{H}_u$ to random unit norm vectors.

Step 3: Apply Gram-Schmidt orthonormalization to $\widehat{H}$ to preserve the columns of

$\widehat{H}_c$

Figure. 3.9 Algorithm for mixing matrix

There are two approaches to carry out spatially constrained ICA and they are alternating IC and SC updates and ensemble IC and SC update. In alternating IC and SC updates, the updation of matrices are performed one by one i.e. IC column update is performed followed by SC column update, on the other hand, in ensemble updates IC and SC are performed collectively and this ensemble of IC and SC was adopted in this research work. Algorithm for evaluating the mixing matrix is depicted in Figure 3.9. The ensemble updates are placed over Infomax ICA, Extended Infomax ICA and FastICA resulting into Spatially Constrained Infomax ICA (SCInfomaxICA), Spatially constrained Extended Infomax ICA (SCExtendedInfomaxICA) and Spatially Constrained FastICA (SCFastICA).

Wavelet Denoising (WD) of SC-ICs: The independent components (SC-ICs) determined by using SCInfomaxICA, SC Extended InfomaxICA and SCFastICA correspond to neural information. There is a possibility that some appendages of artifactual information still exist with the SC-ICs. Information from previous research works show that ICA and wavelets complement each other by removing the limitations of each. Spatially Constrained ICA performs well when datasets are larger, but suffers a trade-off between smaller dataset and performance. The difficulty found with wavelet is that it is hard to distinguish between noise and signals of nearly

similar or higher amplitude. A merger operation after performing SCICA enhances the performance of Wavelet Transform (WT). The reason for the choice of the sequencing SCICA followed by WT instead of vice-versa is that, after performing an SCICA, the components of higher magnitude depict artifactual information and those of lesser magnitude represents neural information i.e. ic1 (neural information), ic15 (artifact information). Noise which is of higher amplitude are apparent in the order ic15, i14 etc. Now the role of WT becomes very much easier, as the neural information is effortless traced in the approximate part likewise the artifactual information are easily trapped in the detail part. Wavelet decomposition was performed in this study using the Daubechies family db4 with the number of vanishing moments equal to 2. Implementation of Daubechies (1990) algorithm is conceptually more complex and has a slightly higher computational overhead when compared to the other wavelet algorithms. It has the advantage of picking up the detail which could be missed by any other wavelet family and also its smoothing features are more suitable to detect changes in EEG signal. Different cut off frequencies were used to analyze the signal at different scales. The signal was passed through a series of high pass filters to analyze the high frequencies, and it was passed through a series of low pass filters to analyze the low frequencies.

The resolution of the signal, which is a measure of the amount of detail information in the signal, is changed by the filtering operations, and the scale is changed by upsampling and downsampling (subsampling) operations. Subsampling a signal corresponds to reducing the sampling rate, or removing some of the samples of the signal. For example, subsampling by two refers to dropping every other sample of the signal. Subsampling by a factor n reduces the number of samples in the signal ‘n’ times. Upsampling a signal corresponds to increasing the sampling rate of a signal by adding new samples to the signal.

The transform progresses by examining the time-domain components into two parts: the approximation and the detail part. The obtained approximation domain is sequentially decomposed further into detail and approximation domains. The basic principle is that the decomposition of a noisy signal on a wavelet basis (by DWT) has

the property to “concentrate” the informative signal in few wavelet coefficients having large absolute values without altering the noise random distribution. The noise s found in the detail part has minimum values when compared to the informative signal. Denoising, is attained consequently by thresholding the wavelets using Otsu’s and Fuzzy shrinkage thresholding methods. Finally inverse DWT was performed to reconstruct the neural information.

3.2.1.4.1 Otsu’s Thresholding

Otsu's method is a bimodal algorithm, named after its inventor Nobuyuki Otsu (1979). This method iterates through all the possible threshold values and calculates a measure with two sides of the threshold, i.e. Peak or valley of the signal. The aim is to find the threshold value where the sum of peak and valley spreads at its minimum. The Otsu’s method selects the threshold by minimizing the within-class variance or maximizing the between- class variance

A signal consists of N values with levels from 1 to L. The number of values with level i is represented by f_i with a probability specified as

$$p_i = f_i / N \tag{16}$$

In bi-level thresholding, the values are separated into two classes, C_1 with levels [1… t] and C_2 with levels [t+1, …, L]. Then, the level probability distributions for the two classes are

$$C_1 : p_1 / \omega_1(t), \ldots, p_t / \omega_1(t) \wedge C_2 : p_{t+1} / \omega_2(t), p_{t+2} / \omega_2(t), \ldots, p_L / \omega_2(t),$$

where $$\omega_1(t) = \sum_{i=1}^{t} p_i \tag{17}$$

and $$\omega_2(t) = \sum_{i=t+1}^{L} p_i \tag{18}$$

Also, the means for classes C_1 and C_2 are

And $$\mu_1=\sum_{i=1}^{t} i\, p_i/\omega_1(t) \quad (19)$$

$$\mu_1=\sum_{i=t+1}^{L} i\, p_i/\omega_2(t) \quad (20)$$

Let μ_T be the mean intensity for the whole values. It is easy to show that

$$\omega_1\mu_1+\omega_2\mu_2=\mu_T \quad (21)$$

$$\omega_1+\omega_2=1 \quad (22)$$

Using discriminate analysis, between-class variance of the thresholded data, and is denoted as

$$\sigma_B^2=\omega_1(\mu_1-\mu_T)^2+\omega_2(\mu_2-\mu_T)^2 \quad (23)$$

For bi-level thresholding, the optimal threshold t* is selected so that the between-class variance σ_B^2 is maximized, which is denoted as

$$t^{*}=Arg\,Max\left\{\sigma_B^2(t)\right\}.1\le t<L \quad (24)$$

Algorithm for Otsu's thresholding is depicted in Figure.3.10

Step 1: Compute histogram and probabilities of each intensity level

Step 2: Set initial values for $\omega_i(0)$ and μ_i (0)

Step 3: Progress through all the possible values of threshold from 1 to maximum intensity

1. Update ω_i and μ_i
2. Compute σ_B^2 (t)

Step 4: Compute the maximum of σ_B^2 (t), which is the desired threshold.

Figure. 3.10 Algorithm for Otsu's Thresholding

3.2.1.4.2 Fuzzy Shrink Thresholding

Fuzzy systems

Fuzzy set theory introduced in the by Zadeh(1965) paved a way to capture the uncertainty and vagueness is often overlooked in complex systems. Fuzzy set theory is a generalization of the classical set theory. Fuzzy systems are knowledge-based or rule-based systems. The knowledge base is formulated based on fuzzy IF-THEN rules, which is the heart of a fuzzy system. A fuzzy set is an enumeration of objects with a continuum grade of membership function. It is characterized by a membership function, which assigns membership grade ranging between zero and one for each object.

The algorithm used in this study is based on the concept of windows with following considerations and used as a base for calculating the fuzzy feature:

- If the main value of the window ($w_{s,d}(i, j)$) and the average value of neighbour ($x_{s,d}(i, j)$) are both large enough then the main coefficients are signal components.
- If the average value of neighbor is higher than the main, then the main coefficients are still signal components.
- If the main and the average value of neighbour are both small enough then the coefficients are noisy components.
- 'S' and 'd' refers to scaling and orientation respectively

Fuzzy feature

Based on the concept that adjacent points are more similar in magnitude, larger weights are assigned to neighboring coefficients with similar magnitude, and smaller weights are assigned to neighboring coefficients with dissimilar magnitude. A fuzzy

function of magnitude similarity and spatial similarity represented as m(l,k), s(l,k) respectively is defined as

$$m(l,k)=\exp\left(-\left(\frac{w_{s,d}(i,j)-X_{s,d}(i+l,j+k)}{Thr}\right)^2\right) \quad (25)$$

$$s(l,k)=\exp\left(-\left(\frac{l^2+k^2}{N}\right)\right) \quad (26)$$

The adaptive weight w (l, k) can be calculated as

$$w(l,k) = m(l,k) \times s(l,k) \quad (27)$$

Using the adaptive weights the fuzzy feature for each coefficient is calculated which relies on the average value of the neighbour and is denoted as

$$f(i,j)=\frac{\sum_{I=-L}^{L}\sum_{K=-K}^{K} w(l,k)\times\left|x_{s,d}(i+l,j+k)\right|}{\sum_{I=-L}^{L}\sum_{K=-K}^{K} w(l,k)} \quad (28)$$

It is necessary that two thresholds 'T_1'and 'T_2' are required for building fuzzy membership function. It is observed that the threshold values are dependent on the estimated noise variance $\hat{\sigma}$ n. The values for obtaining different noise variances are calculated using the median estimator. It is observed that T_1 and T_2 have nonlinear relation with the $\hat{\sigma}$ n and they can be estimated using the equations mentioned below:

$$T_1 = k_1 \times \hat{\sigma} \text{ n} \quad (30)$$

$$T_2 = K_2 \times \hat{\sigma} \text{ n} \quad (31)$$

where k_1 and K_2 are constant values

Fuzzy Shrinkage rule

Problems which are formulated based on fuzzy sets have greater expressive power than their counterpart the crisp sets. The best utilization of fuzzy technology depends on the ability to construct membership functions that appropriately represent various concepts in different contexts. Once the fuzzy features are defined, the next step in wavelet denoising is shrinking the wavelet coefficient based on the principle that: if the coefficients contain noise primarily, then it should be reduced to negligible values and the coefficients containing a noise free component should be reduced less. A fuzzy rule based on the fuzzy feature is framed for shrinking the wavelet coefficients.

If the fuzzy feature f (i,j) is large then the shrinkage of wavelet coefficients w(l,k) is small. The outcome of a fuzzy process has value in the range of [0-1]. The value '0' corresponds to a coefficient, which is the signal of disinterest, while the value '1' corresponds to a coefficient, which is the signal of interest. The value between 0 and 1 indicates the degree of certainty of the coefficient of the signal of interest. A fuzzy set is characterized by properties which are normally fuzzy. e.g. Numbers close to zero are not a precise description. A membership function (MF) is a curve that defines how each point in the input space is mapped to a membership value ranging between 0 and 1, where the input space is referred to as the universe of discourse. Characterizing a fuzzy description with a membership function in effect defuzzifies the fuzzy description. In order to exploit the overall benefits of fuzzy technology, an efficient membership function is desirable with the following characteristics. It should attribute towards accuracy, computational affordability, ease and flexibility in using them.

The membership functions are selected by specifying the structure of the function and then fine tuning the parameters of the Membership Function (MF)

(Andres, 2002). There are 11 built-in membership function types. They are piecewise linear functions, triangular, trapezoidal, or smoother functions such as the symmetric

Gaussian function. Some of the curves could be obtained from a combination of the above curves i.e splines curves or inverse tangent curves. The simplest of the membership functions can be formed by using straight lines and the two types formed from straight lines are triangular and trapezoidal MFs. The function gaussmf () is used to define the Gaussian membership function. The generalized bell membership function is specified by the function name gbellmf ( ) and has one more parameter than the Gaussian membership function. Gaussian and bell membership functions are the most popular methods for specifying fuzzy sets. Though the aforementioned MFs achieve smoother, they are unable to specify asymmetric membership functions which are critical in certain applications. Sigmoid functions are either open left or right. Polynomial curves account for several of the membership functions and three of them are Z, S, and Pi curves, all named after their shapes. In this study seven different membership shape functions were analyzed namely S shaped curve, B splines, Z shaped, Sigmoid, Triangular, Bell curve. Algorithm for fuzzy shrink thresholding is depicted in Figure.3.11

Triangular M.F

$$\mu_F(x, a, b, c) = \begin{cases} 0, & \text{if } x < a \\ (x-a)/(b-a), & \text{if } a \le x \le b \\ (c-x)/(c-b), & \text{if } b \le x \le c \\ 0, & \text{if } c < x \end{cases} \quad \textbf{(32)}$$

Trapezoidal M.F

$$\mu_F(x, a, b, c, d) = \begin{cases} 0, & \text{if } x < a \\ (x-a)/(b-a), & \text{if } a \le x \le b \\ 1, & \text{if } b < x < c \\ (d-x)/(d-c), & \text{if } c \le x \le d \\ 0, & \text{if } d < x \end{cases} \quad \textbf{(33)}$$

Guassian M.F

$$\mu(x, a, b) = e^{\frac{-(x-b)^2}{2a^2}} \quad \textbf{(34)}$$

S-Shaped

$$\mu_Z(x, a, b, c) = \begin{cases} 1, & \text{for } x \le a \\ 1 - 2[(x-a)/(c-a)]^2, & \text{for } a \le x \le b \\ 2[(x-c)/(c-a)]^2, & \text{for } b < x \le c \\ 0, & \text{for } x \ge c \end{cases} \quad \textbf{(35)}$$

z-shaped

$$\mu_S(x, a, b, c) = \begin{cases} 0, & \text{for } x \le a \\ 2[(x-a)/(c-a)]^2, & \text{for } a \le x \le b \\ 1 - 2[(x-c)/(c-a)]^2, & \text{for } b < x \le c \\ 1, & \text{for } x \ge c \end{cases} \quad (36)$$

Spline Shaped

$$\mu_V(x, b, a) = \begin{cases} S(x, a-b, x-b/2, a), & \text{for } x \le a \\ 1 - S(x, a, x+b/2, a+b), & \text{for } x \ge a \end{cases} \quad (37)$$

Bell Shaped

$$\mu_b \ (x,a,b) = \frac{1}{1+¿\frac{x-b}{a}\vee¿^{2b}} \quad (38)$$

Step 1: The time series with its wavelet coefficients are given as input.

Step 2: The universe discourse, fuzzy sets with fuzzy features are defined.

Step 3: One of the fuzzy membership functions is selected.

Step 4: Each data element will be defuzzified based on selected membership function and the fuzzy set.

Step 5: The Steps 1 to step 4 are repeated for all the wavelet coefficients.

Step 6: Repeat step 2 to 5 for all the membership function.

Step 7: Stop

Fig 3.7 Algorithm for fuzzy Shrink thresholding

Figure. 3.11 Algorithm for fuzzy Shrink thresholding

3.2.2 Phase II: Feature Extraction

Feature Extraction is a procedure for determining a feature or a feature vector from a pattern vector. Features are condensed representations of patterns that contain only salient information. The feature vector is composed of a set of all features used to describe a pattern and has a major role in reducing the dimensional space needed to represent the pattern. Some significant advantages of feature extraction are a) It reduces the complexity of software and hardware. b) the cost of data processing and computation time are directly proportional to the degree of dimension. c) Classification becomes more accurate and is performed without any extra computation only when imperative features are considered. Feature Selection provides a means of choosing features which are best for classification. Fast Walsh Hadamard Transform accomplishes the above advantages required of a feature extractor and is implemented in this phase.

Spectral monitoring is very essential to determine the characteristics of EEG signals. Usually the time domain signal is broken into short epochs of a few seconds and a frequency analysis is performed over the segmented epoch. In this study, a Fast Walsh Hadamard Transform (FWHT) was performed on the data array. The data from the FWHT was binned into frequency ranges according to standard EEG definitions:

Delta δ (0-4 Hz) , Theta θ (4-8 Hz) , Alpha α (8-12 Hz), Beta β (12-16 Hz) and Gamma γ (16-50 Hz). In each of the frequency ranges the peak power value frequency and the cumulative power in the corresponding bin was calculated. The aforementioned procedure was executed for the consecutive EEG epochs.

In order to achieve an effective spectral monitoring procedure it was mandatory to record PSD based on time. Estimations of PSD on very short observations are acquired. Normally, in order to attain the above estimation two methods are possible. They are a fixed length sliding window and an exponentially growing window. The fixed length sliding window as the name suggests, considers only a finite number of data values. Exponentially growing window gives more importance to the recent data samples and attenuates the past ones. Though memory is a concern so as to remember all the best values, this study intends to use fixed length sliding window. Figure 3.12 represents the taxonomy of feature extraction.

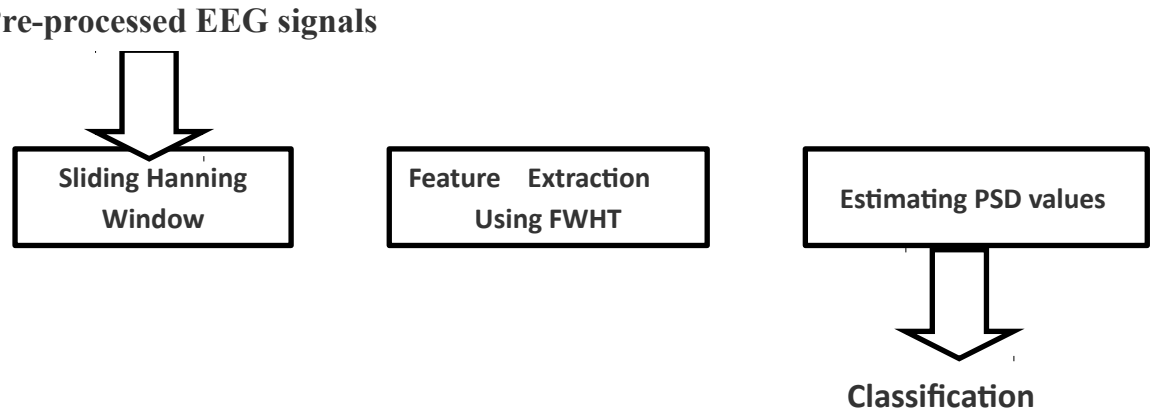


Figure. 3.12 Taxonomy of proposed EEG Feature Extraction

3.2.2.1 Necessity of Windowing

Digital Signal Analysers (DSA) measures the Frequency components like Fast Fourier Transform (FFT), Power Spectral Density (PSD), and Frequency Response Functions etc., which are computed from the digitized time data. Time data are digitized and sampled block by block. A block signifies fixed number of data points in

the digital time data. Most frequency functions are computed from one block of data at a time. This block of data is also referred to as time window.

The FWHT used for feature extraction in this study performs its functionality over the block of time data representing the frequency composition over the time data. When the FWH transformation is carried out on a non-periodic signal the resulting spectrum suffers from leakage, which results in energy smearing out over a wide frequency range which should be curtailed to narrow frequency range. The spread spectrum has a tedious effect in identifying the amplitude and frequency content, hence leakage is considered to have the most detrimental impact in signal processing.

Effects of leakage can only be curbed and not completely eliminated. The curtailed effect of leakage highly depends upon the periodic nature of the signals. In order to satisfy the periodicity requirement, time weighing functions called windows are used. These weighing functions attempt to weigh the beginning and end of the sample record to zero and the middle towards unity by concentrating on the middle and by ignoring the discontinuities at the end. In other words; it has a special shape in the middle, with exactly zero at the beginning and the end of the data block. As most of the signals are aperiodic in nature, it is quintessential that, the choice of an appropriate window function is made. Hence a simple yet effective windowing procedure adopted in this research work is derived from Hanning window.

3.2.2.2 Hanning Window

One of the practical issues imposed in spectral analysis is the issue of performing any transformation on an infinite length sequence signal. Hence it is mandatory to consider a small part of the sequence even in case of a finite length signal. The solution to limit the length of the sequence is to adopt windowing. Let 'y (n)' represent the infinite length sequence. In order to chunk the sequence into a series of finite length, the rectangular window of length 'n' can be represented as '$w_r(n)$'. The windowed sequence could be represented as 'x (n)'.

$$x(n)= y(n)\, w_r(n) \tag{39}$$

The desired properties of windowing are that a) the main lobe should be narrow b) side lobe should be low. Based on the width of the windowing function it is

determined that the signal has a good frequency resolution (frequency components close together can be separated) or good time resolution (the time at which frequencies change). The wider the window, better is the frequency resolution but resulting in poor time resolution. A narrow window results in good time resolution but poor frequency resolution. These transformations are called narrowband and wideband transforms, respectively (Harris, 1978).

Time frequency representations are easily depicted when using a time window which demonstrates a variation in the frequency. The time window is inversely proportional to the frequency. In other words, the window gets shorter as the frequency increases and gets longer when there is a decrease in frequency. Windowing has the advantage of temporal smoothing which decreases at higher frequencies. This experimentation was performed using Hanning Window. It is named after its inventor Julius von Hann and belongs to a family of trigonometric windows having the general form $\cos^{\alpha}(n)$. The exponent α is an integer which ranges from 1 to 4. When the value of $\alpha = 1$, it leads to a cosine tapered window. As the value of α increase the windows are smoothened, the side lobes fall off faster and the main lobe quickly widens. When the value of $\alpha = 2$ $\alpha = 2$ it represents the Hanning window, also called raised cosine and sine squared window .The window is represented as

$$w_{han}(n)=0.5 - 0.25 \exp \frac{(j2\pi n)}{N-1} - 0.25 \exp \frac{(j2\pi n)}{N-1}$$

(40)

$$w_{han}(n)=0.5 \left(1-\cos\frac{2\pi n}{N-1}\right) \text{ where } 0 \leq n \leq N-1$$

(41)

An attractive aspect of Hanning window is the simplicity achieved in smoothening of the frequency domain, which is accomplished using three convolution

terms $\theta_{o,}\theta_{\pm 1}$. Though it has some amplitude inaccuracy for sinusoidal signals, it contributes greater towards frequency resolution. The aforesaid factors are greatly attributed to the selection of the Hanning window in this study.

3.2.2.3 Fast Walsh Hadamard Transform

A general approach in signal processing is to carry out a transformation of the signal and then perform any operation over the transformed set of values rather than attempting to perform operations on original signals, by saving a substantial amount of computational time. The Fast Walsh Hadamard Transform (FWHT) is the transform of signals from time domain to the frequency domain. Discriminating features of signals are highlighted in frequency and spectral domain, and hence signals are classified or clustered with a greater speed and accuracy. Signals after transformation are non-sinusoids and this property reduces the complexity of signal processing by performing coefficient addition and subtraction instead of multiplication as those performed in Fast Fourier Transform (Monir,2011)

Walsh functions

Fourier series is used to create waveforms which are purely a sum of sine and cosine waves at selected frequencies. Similarly it is also possible to build up any wave form out of a sum of square waves as affirmed by the German mathematician H Rademacher in 1922 but with certain deficiencies in his system. Walsh presented his system in 1922 and was later asserted by a Polish mathematician in 1929 to include Rademacher system as a subset of Walsh orthonormal function. It was depicted that, using Walsh function any arbitrary periodic waveform can be built up by adding them together which was analogous to the sine wave summation of Fourier series.

Considering the Walsh functions WAL (0) through WAL (15), WAL (0) is merely a DC level and is usually ignored. WAL (1), WAL (3), WAL (7) and WAL (15) are analogous to square wave Rademacher functions. The functions are also labelled as CAL or SAL which represents Cosine Walsh and Sine Walsh respectively by analogy to Fourier analysis and a Walsh spectrum is also called as sequency spectrum

as opposed to Fourier frequency spectrum. All WAL (even n) are called CAL and all WAL (odd n) are called SAL. In order to simulate a waveform with a stair step approximation, the generated Walsh functions are added with appropriate magnitudes. Generation of Walsh functions consists of steps given in Figure 3.13 as follows:

Generate WAL(n) by writing number 'n' in Gray code which has one bit changing at a time.

Starting with LSB, assign a square wave Rademacher function to each bit i.e. WAL (1) to LSB.

Leave the Radamacher function whose bits are 0.

Perform Exclusive-Or operation of the Radamacher functions whose bit is 1 in order to derive the Walsh output .

3.13 Algorithm for Generating Walsh functions

The expression of a Walsh function representation as a summation analogous to Fourier analysis is:

$$\text{Arbitrary waveform } X(t) = A_o + \sum_{i=1}^{\infty} (A_i \, SAL(i) + B_i \, CAL(i)) \qquad (42)$$

where A_i and B_i represents weighting constants

They are function generating rectangular or square waveforms taking binary values +1 or -1. One of the important characteristics of this function is sequency which is determined from the number of zero crossings per unit time interval.

Hadamard matrix

The Walsh-Hadamard transform requires Hadamard matrix. Hadamard matrices are simple square matrices with elements ±1 and size m, whose distinct row vectors

are mutually orthogonal. Recursive construction can be used to construct a 2m×2m Hadamard matrix from a 2m−1 × 2m−1 Hadamard matrix. This structure derives a Fast Hadamard Transform algorithm, which reduces the computational cost from 2^{2m} additions and subtractions to m^{2m} additions and subtractions. Sylvester's original construction of Hadamard matrices is equivalent to finding Walsh functions which are the discrete analogues of Fourier series. The Walsh Hadamard transform is analogous to Discrete Fourier Transform. The basic functions are sampled Walsh functions, which are expressed in terms of Hadamard matrices. The representation of Hadamard matrix of order 3 bits i. e H_W (3) is depicted in Figure 3.14

$$H_w(3)= \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 \\ 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 1 & -1 & -1 & 1 & -1 & 1 & 1 & -1 \\ 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \end{bmatrix}$$

Figure 3.14 Representation of Hadamard matrix

To find the Walsh Hadamard transform of a vector with say eight components i.e. Eight data points, it is necessary to multiply with the data with H_w (3). A fast Walsh Hadamard transform is analogous to computing Fast Fourier transform (FFT). Similar to a Fourier transform, Wash Hadamard matrix would require a complexity of O $(n)^2$ .The complexity of FWHT could be derived to be as O ($N\log_2 N$). Now considering n=3 i.e. $N=2^n=8$; the Walsh Hadamard transform of a signal x [m] is defined as

$$\begin{bmatrix} ¿ \\ x[0] \\ . \\ x[4] \\ . \\ x[7] \\ ¿ \\ ¿ \end{bmatrix} = \begin{bmatrix} ¿ \\ x[0] \\ . \\ x[4] \\ . \\ x[7] \\ ¿ \\ \begin{bmatrix} H_2 & H_2 \\ H_2 & -H_2 \end{bmatrix} ¿ \end{bmatrix} \quad (43)$$

Fast Fourier Transform accomplishes the transformation by separating the vector into its odd and even components. Similarly a Fast Walsh Hadamard separates the signal into odd and even components and then it recombines the two halves as depicted below. Converting a WHT of size N = 8 into two WHT's of size N=N/2 which can be described as

$$\begin{bmatrix} x[0] \\ \vdots \\ x[4] \\ \vdots \\ x[7] \end{bmatrix} = \begin{bmatrix} H_1 & H_1 \\ H_1 & -H_1 \end{bmatrix} \begin{bmatrix} x_1[0] \\ x_1[2] \\ x_1[4] \\ x_1[6] \end{bmatrix} + \begin{bmatrix} H_1 & H_1 \\ H_1 & -H_1 \end{bmatrix} \begin{bmatrix} x_1[1] \\ x_1[3] \\ x_1[5] \\ x_1[7] \end{bmatrix} \quad (44)$$

This equation can be recursively divided into two halves

$$\begin{bmatrix} x_1[0] \\ x_1[2] \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} x_2[0] \\ x_2[2] \end{bmatrix} \quad (45)$$

The second half could be written as

$$\begin{bmatrix} x_1[1] \\ x_1[3] \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} x_2[1] \\ x_2[3] \end{bmatrix} \quad (46)$$

Finally from the above equation it could be summarized as

$$\begin{bmatrix} F_s & 0 & & & & & & \\ G_s & \ddots & \ddots & & & & & \\ 0 & \ddots & F_s & 0 & & & & \\ & \ddots & G_s & \ddots & \ddots & & & \\ & & 0 & F_s & G_s & \ddots & F_s & G_s \\ & & & \ddots & \ddots & \ddots & \ddots & \\ & & & & 0 & F_s & G_s & \\ & & & & & \ddots & \ddots & 0 \\ & & & & & & 0 & G_s \end{bmatrix}$$

$$H_N(i) = [\,], s = 2^{i-1},$$

$$x(0) = x_2(0) + x_1(0) \quad (47)$$

Repeating the same procedure for all the vectors a Fast Walsh Hadamard transforms can be obtained achieving the complexity as O (Nlog2N).

In the context of fast Walsh Hadamard transform, a butterfly diagram as depicted in Figure 3.15 is the pictorial representation of the computations carried out breaking larger Walsh Hadamard transform into modular portions of Walsh Hadamard transform.

IN	A1	A2	OUT
X0(0)	X1(0)	X2(0)	C (0)
X0(1)	X1(1)	X2(1)	C (1)
X0(2)	X1(2)	X2(2)	C (2)
X0(3)	X1(3)	X2(3)	C (3)
X0(4)	X1(4)	X2(4)	C (4)
X0(5)	X1(5)	X2(5)	C (5)
X0(6)	X1(6)	X2(6)	C (6)
X0(7)	X1(7))	X2(7)	C (7)

3.15Representation of Butterfly Diagram

3.2.2.4 Welch Periodogram

Spectral estimation is used to describe the distribution of the power contained in a Signal. There are various methods of spectral estimation and can be categorized as:

- Subspace methods
- Parametric methods
- Non-parametric methods

Subspace methods, also termed as high-resolution or super-resolution methods generate frequency component estimates based on an eigen analysis or eigen decomposition of the correlation matrix. Multiple Signal Classification (MUSIC) method and the eigenvector (EV) method are examples of Subspace methods. These are the methods best suited for detection of sinusoids dormant in noise; such a condition arises when the signal to noise ratios is low.

In parametric methods PSD is estimated by a higher-order mathematical equation, instead of a simple running average and involves in predictive and regression qualities. These methods typically assume some knowledge of the signal prior to the calculation of PSD. Yule-Walker Auto-Regressive (AR) method, Burg method are examples of parametric methods.

Using Non-parametric methods the PSD is estimated directly from the signal and does not make any assumptions about the data samples and work directly with DFT. These methods are used when little information is known about the signal ahead of time. They typically have less computational complexity than parametric models. The simplest of a non-parametric method is the periodogram method. In this method

the PSD is computed by dividing the set of N samples into p sets of M samples and computing the DFT of each set and squaring it and then computing the average of all of them. An enhanced version of the periodogram is Welch's method.

Welch's method, named after its inventor P.D. Welch is a Non-parametric method. It is used for estimating the power of a signal at different frequencies and therefore this method was adopted for the study. Welch method is an enhancement of the standard periodogram spectrum and Bartlett's method. The concept of windowing makes the Welch method a "modified" periodogram. It reduces the variance caused by the periodogram method. This method employs the concept of dividing the time series into overlapping sub sequences and consequently applying a window to each subsequence and then averaging the periodogram of each subsequence. The length of the applied window has an impact on the trade-off between bias and variance of the resulting power spectral density (PSD). The PSD of Welch estimate is given by

$$S_x^W(\omega_K) \triangleq \frac{1}{k}\sum_{m=0}^{K-1} P\,x_m, M(\omega_K) \qquad (48)$$

In the above equation $P\,x_m$ denotes the periodogram calculated using Welch estimate of the input signal of length m and ω_K is the window size. Figure 3.16 shows the algorithm for welch method

- Split the signal of Length M, and overlap by D points.
  If D=M/2 overlap is 50%
  If D=0 then overlap is 0%
- Window the segments using Hanning window

Figure 3.16 Algorithm for Welch method

The property of any window function is that, it has more influence to the center of the data set rather than at the edges, and this result in loss of information. To alleviate this loss the datasets are commonly overlapped in time.

The feature vectors due to high-dimension increase the computational complexity. Feature extraction and selection, play a significant role in reducing the dimensionality of the vectors. Statistical features along with the power levels of PSD were used in reducing the dimensionality in each band.

1) Maximum of the PSD of each EEG segment.
2) Minimum of the PSD of each EEG segment.
3) Mean of the PSD of each EEG segment.
4) Standard deviation of the PSD of each EEG segment.

3.2.3 Phase III: Classification

Classification is one of the most frequently encountered forecast or decision making task occurring in day to day human activity. The problem of classification transpires when an object needs to be assigned into a predefined group or a class. It examines the input feature vector and produces a suggestive hypothesis. This section presents two algorithms for detecting the presence or absence of seizures.

3.2.3.1 Adaptive Neuro Fuzzy Inference System (ANFIS)

3.2.3.2 Need for Neural Networks

The prominence of neural networks for classification proves it to be promising when compared to the traditional classification methods like discriminant analysis. Following are the advantages that make the neural network play an imperative role:

- They are data driven self-adaptive methods such that they can adjust themselves without any explicit specification of the functional or the distributional form of the underlying model.
- They are universal functional approximator and can approximate any function with arbitrary accuracy.
- They are nonlinear models and hence modelling them in any complex real world applications is highly flexible.

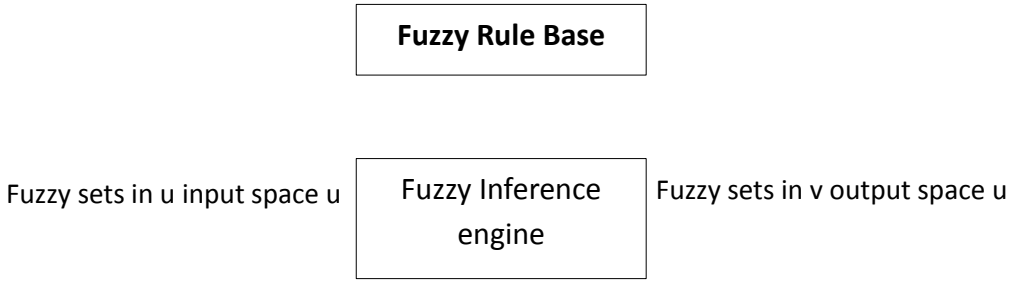
- They have the capability to estimate the posterior probabilities, thereby providing a foundation for establishing classification rules and performing statistical analysis.

Neural networks are routinely employed in signal classification algorithms including automatic seizure detection. Gotman et al. (1978), used template matching for the detection of epileptic seizures. Cheng-wenko et al. (1998) employed an Artificial Neural Network (ANN) for identifying seizures using radial basis function. As EEG signals are non-stationary, Proakis and Manolakis (1996) applied the conventional method of visual analysis but were not highly successful in diagnostic classification. Temporal patterns are processed by the approach of using Recurrent Neural Networks (RNNs) which have memory to encode past history.

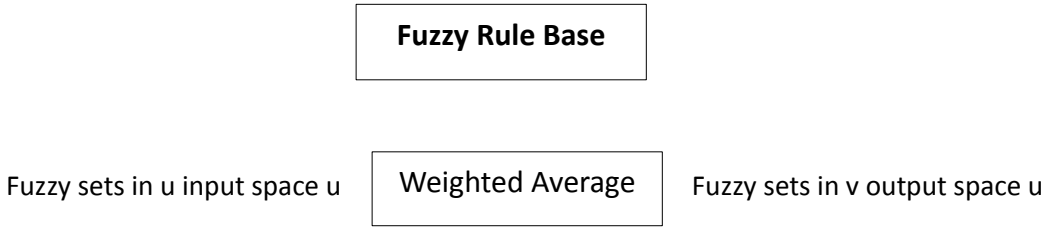
3.2.3.3 Need for Fuzzy Systems

As the real world is too convoluted for precise metaphors, so is a need for approximations or fuzziness in order to obtain a reasonable model. There are three types of fuzzy systems commonly used in the literature:(i) pure fuzzy systems, (ii) Takagi-Sugeno-Kang (TSK) fuzzy systems, and (iii) fuzzy systems with fuzzifier and defuzzifier.

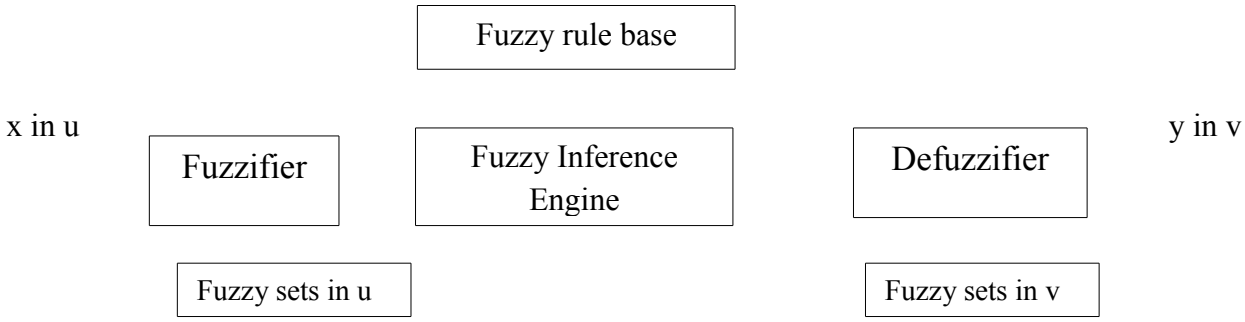
Fuzzy inference system is a method that interprets the values in the input vector and based on some set of rules, assign values to the output vector. Fuzzy rule-based models and especially Takagi-Sugeno (TS) fuzzy models have gained significant impetus due to their flexibility and computational efficiency (Takagi and Sugeno, 1985; Yager and Filev, 1994). The reason it owes is its ability to approximate nonlinear dynamics, multiple operating modes, significant parameter and structure variations (Takagi and Sugeno, 1985). Figure 3.16 represents the nomenclature of fuzzy models



a) Basic Configuration of fuzzy systems



b) Takagi and Sugeno fuzzy



c) Fuzzy system with Fuzzifier and Defuzzifier

Figure. 3.17 Nomenclature of different fuzzy models

Fuzzy set theory plays a vital role in dealing with uncertainty when making decisions on biomedical applications. Introduced by Zadeh, fuzzy logic and fuzzy set theory are employed to describe human thinking and reasoning in a mathematical framework. Fuzzy-rule based modelling is a qualitative modelling scheme where the system behaviour is described using a natural language. Fuzzy sets have attracted the growing attention and interest in modern information technology, production technique, decision making, pattern recognition, diagnostics, data analysis, etc. These intelligent computational methods offer real advantages over conventional modelling, especially when the underlying physical relationships are not fully understood.

3.2.3.4 Confluence of Neuro-Fuzzy Systems

In recent years, the integration of neural networks and fuzzy logic has emerged for solving complex problems. Neuro-fuzzy systems have the potential to capture the benefits of both these fields in a single framework. For building an FIS, fuzzy sets, fuzzy operators and knowledge base are required. For constructing an ANN the user needs to specify the architecture and learning algorithm. Drawbacks of each of these approaches complement each other and hence an integrated system combines the advantages of these systems.

3.2.3.4.1 Fast Adaptive Neuro Fuzzy Inference System (FANFIS) Classification Model

System modelling based on conventional mathematical tools is highly unsuitable for uncertain systems. A fuzzy inference system models the requirements posed by human knowledge by employing fuzzy rules. The perspective aim of this work is to propose a new architecture called FANFIS which adopts an effective method for tuning the membership functions so as to minimize the output error and to maximize the performance. Constructing an appropriate set of fuzzy if-then rules to tune membership functions for performing the stipulated input-output operation is also one of the requirements of FANFIS.

FANFIS is composed of five functional blocks. They are

- A rule - based block comprising of fuzzy if-then rules.
- A database block comprising of membership functions of fuzzy sets.
- A decision block, whose function is to perform the inference operation of the fuzzy rules.
- A fuzzification interface unit which does the job of transforming crisp input values into degrees of match with linguistic values.
- A defuzzification interface unit which does the reverse job of the previous unit. It transforms fuzzy results into crisp output values.

Architecture of FANFIS

FANFIS is an adaptive multilayer feed forward network which adapts the structure of ANFIS (Jyh-Shing et al., 1993). FANFIS architecture is depicted in the Figure. 3. 17. It consists of nodes and directional links through which the nodes are

connected. Each of the nodes performs a particular function and is adaptive in nature, such that their outputs depend on the parameters of the nodes. Square nodes are adaptive nodes which represent that the parameter sets in these nodes are adjustable, whereas, circle nodes are fixed nodes, which represents the parameter sets which are fixed in the system.

To explain the architecture of FANFIS, consider that the network has n layers and m^{th} layer has #(m) node i.e multiples of m. The i^{th} position of m^{th} layer is denoted as (m,i) and its node output is denoted as O_i^m . The node output depends upon the incoming signals and the premise parameter and could be represented as

$$\begin{matrix} O \\ (¿¿1^{m-1}\ldots O_{¿(m-1)}^{m-1}, a, b, c\ldots) \\ O_i^m = O_i^m ¿ \end{matrix}$$ a, b, c, … pertains to the parameters of this node and

O_i^m is used for representing the node output and node function. For simplifying the explanation procedure of the FANFIS, only two inputs x, y and one output f are being considered in the inference system. One degree of Sugeno's function is adapted to depict the fuzzy rule and hence the rule base will contain two fuzzy if–then rules as shown in equations (1) and (2):

Rule 1: if x is A_1 and y is B_1 then $f_1 = p_1x + q_1y + r_1$. (49)

Rule 2: if x is A_2 and y is B_2 then $f_2 = p_2x + q_2y + r_2$. (50)

where x and y are the inputs, A_i and B_i represent the fuzzy sets, f_i represents the output inside the fuzzy region represented by the fuzzy rule, p_i, q_i and r_i indicate the design parameters that are identified while performing training process.

The first layer has every node as adaptive nodes.

$$O_i^1 = \mu_{A_i}(x) i = 1,2 \quad (51)$$

x represents the input and A_i is the linguistic label associated with the input. O_i^1 represent the membership function of A_i and specifies the degree to which x satisfies

A_i. A bell shaped membership function is considered with values in the range 0-1 and accordingly $\mu_{A_i}(x)$ is represented by:

$$\mu_{A_i}(x)=\frac{1}{1+\left\{\left(\frac{x-c_i}{a_i}\right)\right\}^{b_i}} \tag{52}$$

where a_i, b_i and c_i represent the parameters of the membership function, controlling the bell shaped functions consequently .

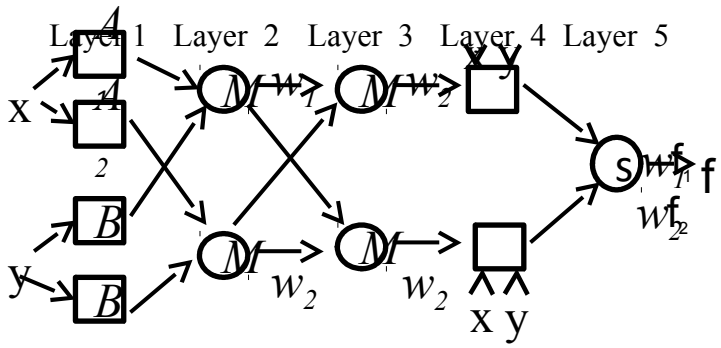


Figure. 3.18. Architecture of FANFIS

Layer 2 is a circle node (fixed node), which carries out the basic functionality of the multiplication of incoming signals. The outputs of this layer are called firing strengths of the rules and can be indicated by:

$$w_i=\mu_{A_i}(x)*\mu_{B_i}(y)i=1,2 \tag{53}$$

The nodes fixed in layer 3 are fixed nodes, which performs a normalization function by calculating the ratio of i^{th} rule's firing strength s, to the sum of all rules firing strength. The outputs of this layer can be indicated as:

$$\overline{w}_i=\frac{w_i}{w_1+w_2}i=1,2 \tag{54}$$

$\overline{w}_i$ is denoted as normalized firing strengths.

Layer 4 has adaptive nodes and the output of every node in this layer is merely the product of the normalized firing strength and a first order polynomial, which are indicated by:

$$O_i^4 = \overline{w}_i f_i = \overline{w}_i (p_i x + q_i y + r_i) i = 1,2 \quad (55)$$

Layer 5, is a single fixed node, which represents the sum of every incoming signal. Therefore, the overall output of the model is provided by:

$$O_i^5 = \sum_{i=1}^{2} \overline{w}_i f_i = \frac{\sum_{i=1}^{2} w_i f_i}{w_1 + w_2} \quad (56)$$

The architecture constructed for FANFIS is an adaptive network equivalent to type -3 fuzzy inference system.

Learning algorithm of FANFIS

The basic learning rule for any adaptive networks is based on gradient descent. Certain limitations faced by this method are the sluggish performance and the tendency of being trapped in local minima. In order to substantially speed up the learning process a hybrid learning algorithm is being adopted in the proposed work. The learning algorithm is intended to adjust all the modifiable parameters such as { a_i, b_i and c_i } and { p_i, q_i and r_i }, for the purpose of matching the output with the training data. If the parameters such as a_i, b_i and c_i of the membership function are unchanging, the outcome of the model can be given by:

$$f = \frac{w_1}{w_1 + w_2} f_1 + \frac{w_2}{w_1 + w_2} f_2 \quad (57)$$

Substituting Eq. (54) into Eq. (57) yields:

$$f = \overline{w}_1 f_1 + \overline{w}_2 f_2 \quad (58)$$

Substituting the fuzzy if-then rules into Eq. (14), it becomes:

$$f = \overline{w}_1 (p_1 x + q_1 y + r_1) + \overline{w}_2 (p_2 x + q_2 y + r_2) \quad (59)$$

After rearrangement, the output can be expressed as:

$$1+¿(\overline{w}_1)r_1+(\overline{w}_2x)\,p_2+(\overline{w}_2y)\,q_2+(\overline{w}_2)r_2$$
$$f=(\overline{w}_1x)\,p_1+(\overline{w}_1y)\,q_¿ \qquad (60)$$

which is a linear arrangement of the adjustable resulting parameters p_1, q_1 , r_1 , p_2, q_2 and r_2. If the basis parameters are not adjustable, it results in a larger search space search space becomes larger and needs more time for convergence. In order to solve the above mentioned problem a hybrid algorithm encompassing a forward pass and a backward pass was adopted in ANFIS, which merges the least squares and the gradient descent technique. The least squares technique which acts as a forward pass is utilized to determine the resulting parameters with the premise parameters. Once the optimal consequent parameters are determined, the backward pass begins by invoking the gradient descent technique. This pass is utilized to fine-tune the premise parameters equivalent to the fuzzy sets in the input domain. The outcome of the model is determined by using the resulting parameters identified in the forward pass. The output error is used to alter the premise parameters with the help of the standard back propagation method. It has been confirmed that this hybrid technique is very proficient in training the model. But still gradient descent is a sloth performer and hence in order to fasten up the performance of gradient descent, a Modified Levenberg Marquardt (MLM) algorithm is introduced in this research, which replaces the gradient descent and congregates FANFIS.

3.2.3.5 Levenberg Marquadt Algorithm (LM)

The Levenberg-Marquardt (LM) algorithm is one of the widely used optimization algorithm that outperforms simple gradient descent and other flavours of gradient methods. It was designed to approach a second order training speed without computing the Hessian matrix and highly suitable for small and medium sized networks. LM is considered to be a blend of Gradient Descent (GD) and Gauss-Newton (GN) iteration which addresses the limitations of each of these techniques. Both the methods are not optimal in search of global minimum, as with GN it is due to the divergence of successive iterations, and with GD, it is due to its slower

convergence. It combines the beneficiaries of GN and GD and arrives at the optimization.

Some of the highlights of GN method are its excellence in convergence properties. These properties are highly reliant on the selection of initial values. There is a tendency for the algorithm to diverge, if the initial values are improperly selected. Moreover, the accuracy of the GN method becomes trivial once it reaches the propinquity of a minimum. On the other hand, although GD algorithm reveals a slow speed of convergence, it exhibits excellent behaviour in the locality of a minimum point. The performance of the gradient descent algorithm is also not affected by the choice of the initial values. Now the optimization criteria arrived by LM successfully chooses the speed of the GN and localization at the minimum point of GD.

The mathematical derivation of Levenberg Marquardt as proposed by Hagan and Menhaj (1994) is given as follows. According to GN method the update rule for the weights of the neural network is denoted by:

$$\begin{matrix} \nabla^2 E(w) \\ ¿ \\ ¿ \\ w = -¿ \end{matrix} \qquad (61)$$

where $\nabla E(w)$ represents the Laplacian function of the energy and is also referred to as the Hessian matrix. The Hessian term can be represented as:

$$\nabla^2 E(w) = J^T(w).J(w) + S(w) \qquad (62)$$

$$S(w) = \sum_{i=1}^{N} e_i(w).\nabla^2 e_1(w) \qquad (63)$$

where the term $e_i(w)$ denotes the error vector of the neural network for pattern i**,** and J (w) represents the Jacobian matrix.

The GN method is normally assumed to be similar to the Taylor partial series expansion and denoted as:

$$S(w) \approx 0, \tag{64}$$

The term S (w) involves the second order derivatives of the network error and is very expensive to compute because the number of computations increases exponentially with the size of the network. By combining (61, 62) equations, the update rule for the GN method can be denoted as:

$$\Delta w ¿ -\left[J^T(w).J(w)\right]^{-1}.J^T(w)e(w) \tag{65}$$

The LM modification to the updated GN method is as follows:

$$\Delta w ¿ -\left[J^T(w).J(w)+\mu I\right]^{-1}.J^T(w)e(w) \tag{66}$$

LM is a damped square technique, where a positive constant (μ) called as the damping factor is added to the Jacobian matrix in order to control the behaviour of the system. When the damping factor μ is large, the above expression approximates gradient descent (with learning rate **1/** μ) and for a smaller damping factor, the algorithm approximates the GN method (with learning rate **μ**). By adaptively adjusting the parameter μ, the LM algorithm contrives between the two extremes – the gradient descent and the GN algorithm. By doing so, the LM method combines the advantage of gradient descent and the GN algorithms, while circumventing their limitations. The adaptive change in the damping factor is similar to the modification of the adaptive learning rate in the back - propagation algorithm. When moving through a gently sloping area of error surface it is more convenient to take large steps. This is performed by multiplying μ by a constant μ_{Inc} **(**i.e. an amplification factor) to drive the algorithm towards the gradient descent algorithm and, to obtain more stability. On the other hand, when descending through a steep valley in error surface it is best to use a small step size to avoid missing the minimum of the valley. Here μ is multiplied by another constant $\mu_{Dec} = \frac{1}{\mu_{Inc}}$ (i.e. a reduction factor) inorder to drive the algorithm towards the GN algorithm, to gain more celerity. Hence the

damping factor is dependent on μ with an initial value, amplification factor μ_{Inc} , reduction factor μ_{Dec} and a minimum value of μ The algorithm is assumed to converge when the sum of square has been reduced to some predefined error goal and is depicted in the Figure. 3.19.

Step 1: For the given inputs compute the corresponding outputs and the errors $e_q = t_q - a_q^m$ and compute the sum of squared errors (SSE) over all inputs.

Step 2: Compute the Jacobian matrix and their sensitivities with the recurrence relations $S_q^m = f^m\left(n_q^m\right)\left(w^{m+1}\right)^T . S_q^{m+1}$ ¿ , after initializing with the Eq. $S_q^m = -f^m\left(n_q^m\right)$ ¿ . Augment the individual matrices into the Marquardt sensitivities using Eq. $\left(S^m = \left[S_1^m, S_2^m, \ldots\ldots, S_n^m\right]\right)$ and then compute the elements of the Jacobian matrix with $\left(\left[J\right]_{h,l} = S_{l,h}^m \times \mu_{j,k}^{m-1}\right)$¿.

Step 3: Compute Eq. Δw_k¿$\left[J^T\left(w_k\right).J\left(w_k\right) + \mu.l\right]^T.J^T\left(w_k\right).e\left(w_k\right)$¿ and obtain *the value of* Δw_k .

Step 4: Recompute SSE based on $w_k + \Delta w_k$. If this new SSE is less than the SSE computed in step1, then multiply μ by μ_{dec} , and go back to step1, else multiply μ by μ_{Inc} and go back to step 3.

Figure. 3.19 Algorithm for LM

3.2.3.6 Modified Levenberg Marquardt (MLM) Algorithm

MLM is formulated by considering the simplicities of LM algorithm and attempts at examining and improving the efficiency of the method (Evelyn Arenda , 1999). MLM aspires to achieve the following:

i) It aims at minimizing the number of dependable parameters of the damping factor, thereby minimizing the modeling time as well as the error associated with a large number of variables.

ii) To find an optimal strategy so as to reduce the number of iterations with an effect to reduce the execution time.

It can be noted in LM that the performance of the algorithm wholly depended on the value of the damping factor, which in turn is dependent upon four factors μ_{Inc}, μ_{Dec} initial dampness and minimum dampness. There is a possibility of getting trapped in local minima due to two reasons.

i) If the value of minimum dampness is large, then it has an adverse effect on convergence.

ii) If the value of minimum dampness is very small, then a problem of singularity arises.

Another important factor of consideration about damping is to avoid singularity by increasing the values of the diagonal matrix and by evading zero or small Eigen values. This is attained by adding a positive constant δ to the diagonal matrix,

which circumvents the matrix from being singular. The high value of δ makes the Hessian matrix stable but leads to poor convergence. Hence the value of δ is central to the function and needs to be selected optimally. Now the modified LM is only dependent upon one factor i.e. δ instead of four factors.

3.2.3.7 EXTREME LEARNING MACHINES

Traditional learning methods aim at determining the hidden structure of the neural network, which in turn concentrates on determining the number of hidden layers and the number of neurons in each layer by training the network using conventional back propagation algorithm. The topology of the network also plays a high influential role in the performance of neural networks. The Feedforward neural network has a dominating role due to its ability in approximating complex nonlinear mappings directly from its input samples. It also provides models for complex large samples involved in natural and artificial phenomenon with ease. ELM has adopted the single layer feed forward neural network based on norm least square solution introduced by Huang. et al. (2006). It is known that traditionally all the parameters of the feed forward networks need to be tuned and thus there exists the dependency between different layers of parameters. Gradient descent-based methods have been used in various learning algorithms of feed forward neural networks. It is very clear that a gradient descent based learning methods are generally very slow due to improper learning steps or may easily converge to local minima. Many iterative learning steps are required by such learning algorithms in order to obtain better learning performance. In conventional learning theory and implementations, the training data has to be seen before generating the hidden node parameters. In ELM, learning is performed without iterative tuning i.e. the input weights of SLFNs does not require to be tuned and are randomly generated, whereas the output weights are analytically determined using the least-square method, thereby providing less time for training. In ELM learning theory and implementations, the hidden node parameters can be generated before seeing the training data. All the parameters of ELMs can be

analytically determined rather than being tuned. ELM randomly chooses and fixes the weights between the input neurons and hidden neurons based on some continuous probability density function, and then analytically determines the weights between hidden neurons and output neurons. This algorithm provides good generalization performance at extremely fast learning speed

Neural networks and Support Vector Machines have contributed greatly to computational intelligence techniques till recent years. The major bottlenecks faced by these two popular techniques are issues such as slow learning speed in case of NNs, poor computational and learning scalability and requirement of human intervention. Extreme Learning Machines (ELM) have a very high potential that can resolve challenges faced by these techniques and provides a better generalization with faster learning speed and lesser human intervention. Recent developments of ELM technique have shown that ELM inherits the advantages of both neural networks and support vector machines. It possesses faster learning speed, requires less human intervention and acquires robust property with respect to the parameter selection.

3.2.3.7.1 Salient features of ELM

ELM has several salient features when compared to the other traditional algorithms like Back Propagation (BP) Algorithm and Support Vector Machine (SVM).Some of the features are discussed below:

- ***Ease of use***: Once the architecture is predefined, there is no necessity to tune the parameters manually. This prevents the user straying and spending several hours in training the learning machines.
- ***Faster learning speed***: Conventional methods cannot provide a rapid learning rate as opposed to ELM, where the time taken for most of the training will be in milliseconds, seconds and minutes.
- ***Higher generalization performance***: The generalization performance of ELM is better than that of SVM and back propagation.

- ***Applicable for all nonlinear activation functions***: All piecewise continuous functions which include discontinuous, differential, non-differential functions can be used as activation functions in ELM.
- ***Applicable for fully complex activation functions***: Complex functions can also be used as activation functions in ELM.

The Structure of ELM is clearly shown in Figure. 3.20

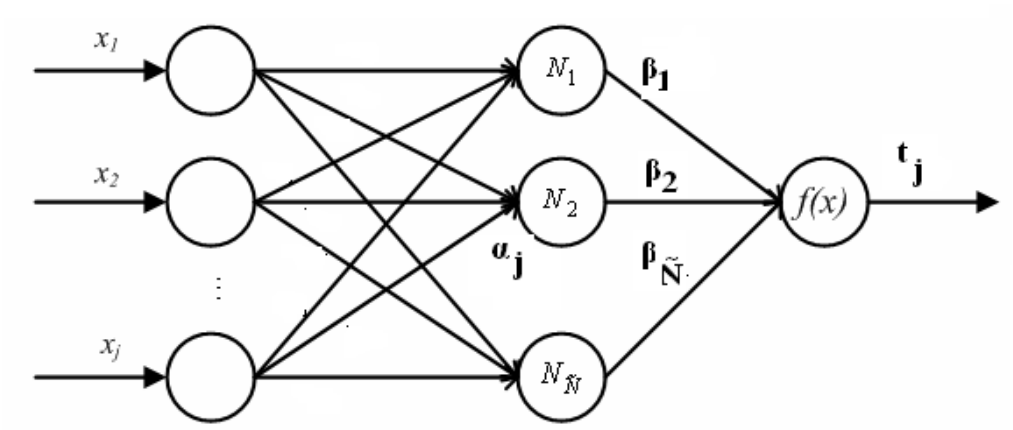


Figure. 3.20. Structure of ELM

3.2.3.7.2 ELM Algorithm

In supervised batch learning, the learning algorithms use a finite number of input-output samples for training. N arbitrary distinct samples $(x_i, t_i) \in R^n \times R^m$ are considered, where x_i is $n \times 1$ input vector and t_i is a $m \times 1$ target vector. If an SLFN with $\tilde{N}$ hidden nodes can approximate these N samples with zero error, it then implies that there exist β_i, a_i and b_i such that

$$t_j, j = 1, ..., N \quad (67)$$

$$f_{N^-}(x_j) = \beta \sum_{(i=1)}^{\tilde{N}} G(a_1, b_i x_i)_j =$$

Equation (67) can be written compactly as

$$H\beta = T \quad (68)$$

where

$$a_1, \ldots, a_{\tilde{N}}, b_1, \ldots, b_{\tilde{N}}, x_1, \ldots x_N = \begin{bmatrix} G(a_1, b_1, x_1) & \ldots & G(a_{\tilde{N}}, b_{\tilde{N}}, x_1) \\ \vdots & \ldots & \vdots \\ G(a_1, b_1, x_N) & \ldots & G(a_{\tilde{N}}, b_{\tilde{N}}, x_N) \end{bmatrix}_{N \times \tilde{N}} \quad (69)$$

H is called the hidden layer output matrix of the network, the i^{th} column of H is the i^{th} hidden node's output vector with respect to inputs $x_1, x_2, \ldots, x_N$ and j^{th} row of H is the output vector of the hidden layer with respect to Input x_j.

$$\beta = \begin{bmatrix} \beta_1^T \\ \vdots \\ \beta_{\tilde{N}}^T \end{bmatrix}_{\tilde{N} \times m} \quad \text{and} \quad T = \begin{bmatrix} t_1^T \\ \vdots \\ t_N^T \end{bmatrix}_{N \times m} \quad (70)$$

In real applications, the number of hidden nodes, $\tilde{N}$ will always be less than the number of training samples, N, and, hence, the training error cannot be made exactly zero but can approach a nonzero training error $\in$.

The hidden node parameters a_i and b_i (input weights and biases or centers and impact factors) of SLFNs need not be tuned during training and may simply be assigned with random values according to any continuous sampling distribution.

Thus the linear system is obtained and the output weights β are estimated with the generalized inverse of the hidden layer output matrix H.

$$\beta = H^{-1}T \quad (71)$$

The algorithm for ELM is depicted in the Figure. 3.21

Given a training set $x = \{(x_i, t_i) \mid x_i \in R^n, t_i \in R^m, i = 1,...,N\}$, Activation function g(x), and hidden node number $\tilde{N}$, the algorithm can be given as follows.

Step1: Assign random hidden nodes by randomly generating parameters (a_i, b_i) according to any continuous sampling distribution, i = 1, .., $\tilde{N}$.

Step2: Calculate the hidden layer output matrix H.

Step3: Calculate the output weight β.

Figure. 3.21. Algorithm for ELM

In this research work, Hybrid ELM was used as a classification model. This classification model uses the Analytical Hierarchy Process (AHP) method to select the input weights and hidden biases, the ELM algorithm to analytically determine the output weights and Modified Levenberg Marquardt (MLM) algorithm to learn the network.

3.2.3.8 Analytic Hierarchical Process (AHP)

Analytic Hierarchy Process (AHP), a popular and widely used method is eponymously referred to as the Saaty method (1980) after its inventor. Figure 3.22 and 3.23 represent the algorithm and flow structure of AHP algorithm respectively.

s

Algorithm for AHP

Step 1: Influential selection criteria are identified and the relative importance of each criterion is evaluated.
Step 2: Criterion specific EEG features are constructed based on comparison templates.
Step 3: Extract priority vector for each comparison template and convert them into decision matrix.
Step 4: Apply the scale of relative importance to create criteria judgement matrices and extract the criteria priority vector.
Step 5: Derive the ranking vector from the EEG decision matrix and a criterion priority vector

3.22 Algorithm for AHP

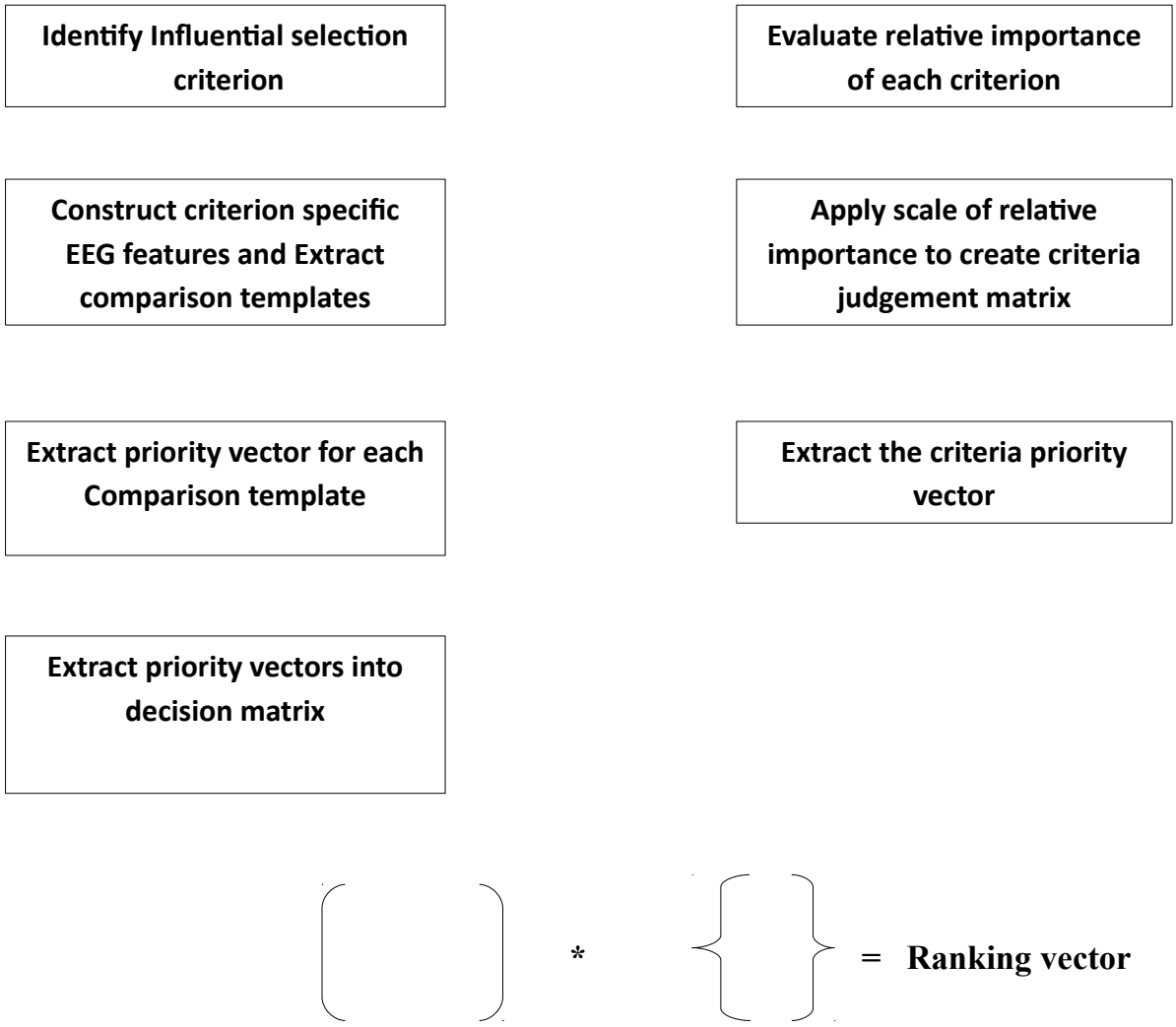


Figure. 3.23: Flow diagram for AHP

It is one of the extensively used Multi Criteria Decision Making (MCDM) methods. The main advantage of using this method is that it does not involve cumbersome mathematics and can effectively handle both qualitative and quantitative data. It is easier to understand AHP as it easily handles multiple criteria. It is mainly based on the principles of decomposition, pair-wise comparisons, and priority vector generation and synthesis. Pair wise comparisons are used to compute the performance

score of alternatives with respect to each attribute by using a pair wise comparison matrix and appropriate weights are evaluated based on them.

3.2.3.9 Hybrid ELM (HELM)

The Hybrid Extreme Learning algorithm is adopted as the proposed method for classification by merging ELM with the modified Levenberg Marquardt algorithm. The input weights and hidden biases are generated using the AHP method. Its corresponding output weights are analytically computed using ELM algorithm and the output hidden biases are randomly generated. Updation of the parameters is performed using MLM algorithm. HELM shows that the hidden node parameters are completely independent of the training data. The proposed algorithm tends to reach the smallest training error and also has the smaller norm of weights. Bartlett's theory (1998) on the generalization performance of feedforward neural networks is stated as follows: Any feedforward network that reaches a smaller training error tends to have smaller norm of weights. Hence the proposed learning algorithm tends to have a better generalization performance of feedforward neural networks. Figure. 3.24 represents the algorithm for HELM

Step 1: Generate input weight vector w_1 and hidden bias vector b_1 and b_2 using the AHP method .

Step 2: Calculate the first hidden layer output matrix a_1.

Step 3: Generate the corresponding output weight w_2 by using ELM algorithm .

$$w_2 = a_1^{-1} \cdot t \quad (72)$$

Step 4: Calculate the first hidden layer output matrix a_2 and error e_1

$$e_1 = t - a_2 \quad (73)$$

Step 5: Update the weight vectors w_1, w_2 and bias vectors b_1, b_2 using modified LM algorithm.

Step 6: Recompute the sum of squared errors (SSE) e_2 using $w_k + \Delta w_k$.

Step 7: If $e_2 < e_1$ then compute $\mu = \mu * \mu_{dec}$ else compute $\mu = \mu * \mu_{inc}$

Step 8: Compute $w_{k+1} = w_k + \Delta w_k$. Repeat the steps until the error goal is reached.

Figure. 3.24. Algorithm for HELM

The overall schematic diagram for the proposed algorithms used in this research is represented in Figure. 3.25

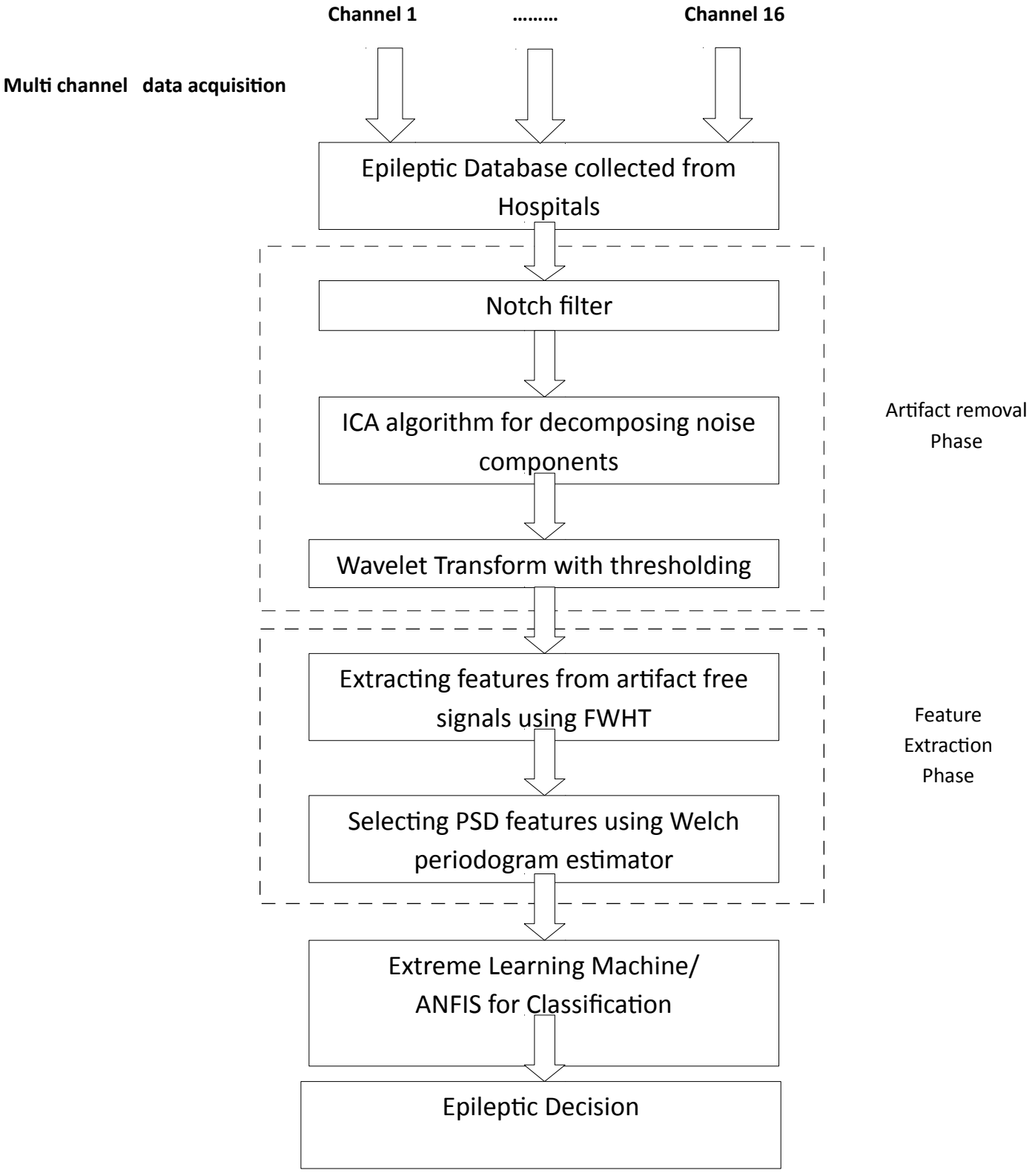


Figure. 3.25. Schematic diagram of the proposed algorithms used in this research work

3.3 Research Contributions

This work contributes to the field of Epilepsy Detection by analyzing several pre-processing, feature extraction and classification techniques in order to derive thriving software. Examination of each of the phases and the contributions with respect to each phase are discussed below.

A momentary look into **pre-processing** phase with its contributions is delineated subsequently. Dominance of placing spatial constraints over ICA algorithms is the main projection of this research work. Very meagre coalesces of BSS algorithms with their implementations in EEG signal processing have their roots intensified in adopting spatial constraints. This research work aims at extending the usage of spatial constraints by reinforcing three underlying algorithms the Infomax, the Extended Infomax and the FastICA ensuing into Spatially Constrained InformaxICA, Spatially Constrained Extended InformaxICA and Spatially Constrained FastICA respectively in order to attain dimensionality reduction.

This hybridization approach for pre-processing is the ultimate formula for removing artifacts. Line noise and a substantial number of artifacts like Electrical, Eyeball movement, Eye blink, Spit Swallowing and Jaw Clenching is considered in this research work. Much of the work in the field of EEG artifact removal revolves around only Ocular artifacts. Derivation of components using SCICAs provides the quintessence of scalability in removing any type of artifact and assimilating it with DWT, eliminates the trade-off imposed by each of the methods. A new combination of DWT with Otsu thresholding and Fuzzy shrinkage thresholding was arrived. Outcome of hybridization by blending appropriate algorithms was a tedious task and a major novelty.

Glimpses of contributions traced in the phase of **Feature extraction and Selection** are discussed subsequently. Extracting the most distinguishing features appropriate for EEG signals were the highlights of the second phase. EEG waves characterizing Delta, Theta, Alpha, and Beta were the relevant features extracted using

Fast Walsh Hadamard Transform (FWHT), which attenuates the complexity perceived in FFT. A noteworthy point is that FWHT is the first of its kind to be implemented for detection of epileptic seizures and is one of the major contributions for feature extraction. Confiscating redundant data is based on the statistical and spectral features of EEG signals. The solitude spectral feature used is the PSD, which surpasses the performance of classification by circumventing large number of features. The contributions in this phase are characterized by assimilating an appropriate choice of windowing, transformation and selecting the pertinent features which do not ascribe redundancy.

A quick glance on the contributions of the **Classification phase** is comprehended subsequently. A minor amendment proposed in the backward learning pass of ANFIS is performed by adopting MLM in order to augment the speed of error convergence. Such an alteration challenges to derive FANFIS, which fastens up ANFIS. A major contribution to the classification phase dwells in the implementation of a very powerful classifier ELM, which effortlessly handles the enormity of the data. Reviews emphasized that much work has not been attributed in Epilepsy detection using ELM. Major contributions were derived by implementing an enhancement of ELM by restructuring it as Hybrid ELM (HELM). The implementation of MLM in order to fasten the learning process is also an additional contributing factor in the accomplishment of the novice algorithm in classification.

3.4 Performance Evaluation

Performance evaluation for three phases has been executed and evaluated. Figure. 3.22 denotes how the evaluation of methods is performed in each of the phases. The results obtained are compared with their traditional counterparts in each of the phases. The first phase of experiment deals with artifact removal. The diverse artifacts under consideration are electrical disturbances, eye ball movement, and eye blink, spit swallowing and jaw clenching artifact. In this phase, performance is appraised for all the above mentioned artifacts in terms of Peak Signal to Noise Ratio (PSNR), Mean Square Error (MSE), and time taken to perform artifact removal.

The second phase of experiment extracts the spectral features as delta, theta, alpha, beta and gamma. The performance is evaluated in terms of Power Spectral Density (PSD) in each of the bands. The final phase of the experiment is used to evaluate the classification parameters. The parameters under consideration were Specificity, Sensitivity and Accuracy. The overall performance of the system is relented by the strength of the methods instituted in each phase is indicated in Figure. 3.26.

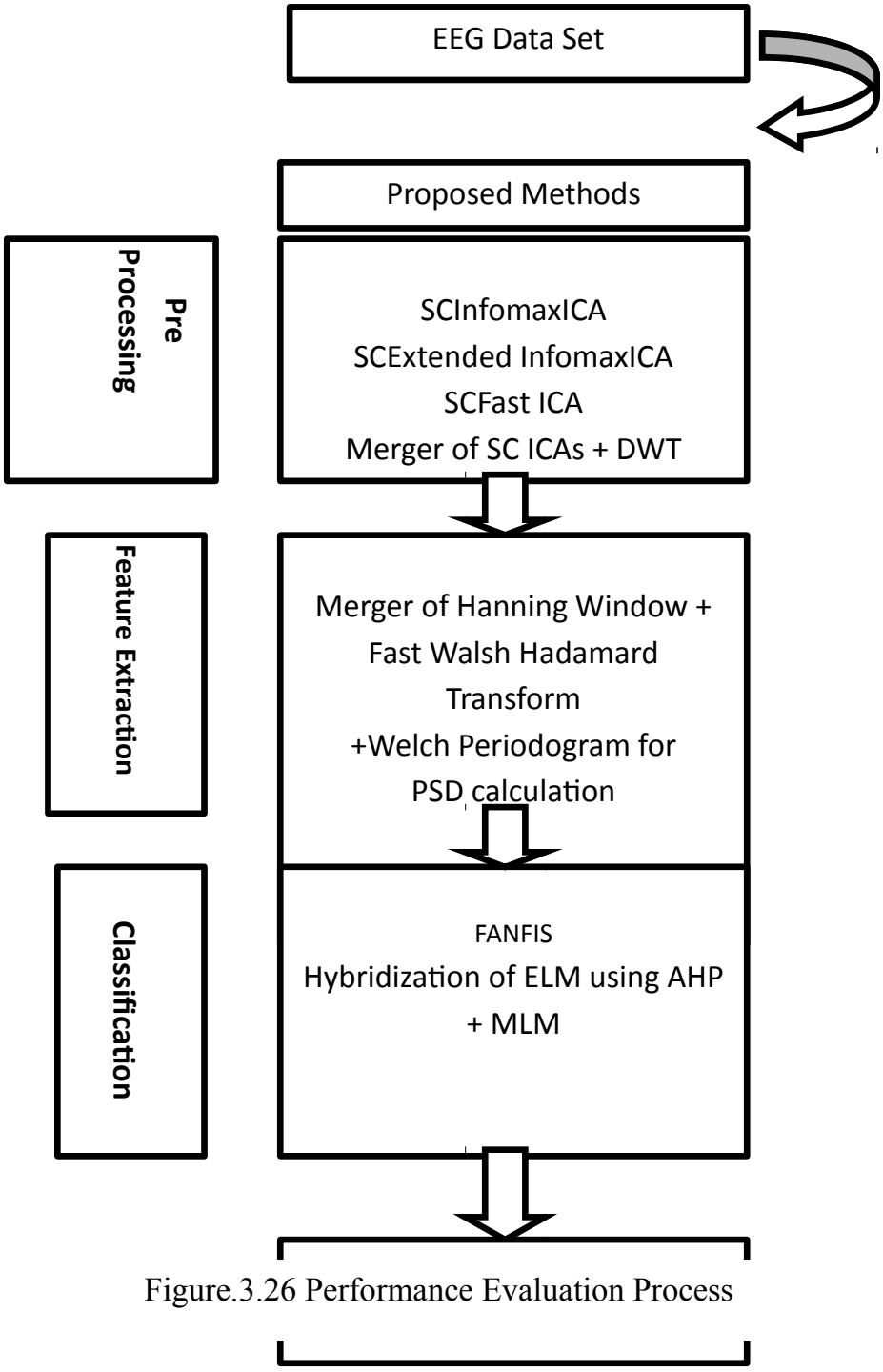


Figure.3.26 Performance Evaluation Process

3.5 CHAPTER SUMMARY

The current research work focuses on an automated system to detect seizures efficiently, for the signals captured using EEG sensors**.** Experiments conducted showed the performance of various proposed methods. The results of these methods are presented in the subsequent chapter, Results and Discussion.