Hilbert marginal spectrum analysis for automatic seizure detection in EEG signals

Aravind Ravi and Manpreet Singh Minhas

ECE 603 - Statistical Signal Processing - Course Project

Introduction

Epileptic seizure is a group of brain disorders characterized pre-dominantly by recurrent and unpredictable interruptions of normal brain function. Epilepsy is one of the most common neurological disorders, 0.6–0.8% of the world's population suffers. Electroencephalogram (EEG) is a valuable measure of the brain's electrical function and generated by the cerebral cortex's nerve cells, it has been a valuable clinical tool for epilepsy evaluation and treatment. EEG signal is highly nonlinear and nonstationary. Fourier analysis which expands the signals in terms of sinusoids cannot appropriately represent the amplitude contribution from each frequency value. Hilbert-Huang transform (HHT) is a new signal analyzing method. This does not handle the signal processing problems from the view that the basic component of signal is sinusoid, instead it is of signals called intrinsic mode functions (IMFs). HHT has well solved the contradiction of imperfectness of non-stationary signals processed by Fourier transform and established the foundation in analyzing signals of time-varying frequency and amplitude.

The Hilbert Marginal Spectrum (HMS) analysis is based on HHT and is used to derive features for epileptic seizure classification. HMS is derived from the empirical mode decomposition (EMD) which decomposes the signal into a collection of intrinsic mode functions (IMFs). Since this decomposition is based on the local characteristic time scale of the signal, it can be well applied to nonlinear and nonstationary processes. Band energy features and spectral entropies are extracted as features from the HMS. A support vector machine (SVM) with a radial basis function kernel is used for seizure detection. This method is compared with the same features extracted based on the Fourier Spectrum of the EEG signals.

Problem Formulation

EEG signal is highly nonlinear and nonstationary and Fourier analysis cannot appropriately represent the amplitude contribution from each frequency value. The HMS analysis can be well applied to nonlinear and nonstationary processes. This work presents a new technique for automatic seizure detection in electroencephalogram (EEG) signals by using Hilbert marginal spectrum (HMS) analysis.

Method Proposed in the Paper

Fig. 1. Block diagram of the proposed method

The proposed method makes use the Hilbert–Huang transform (HHT). HHT consists of two components: Empirical Mode Decomposition (EMD) and Hilbert Spectral Analysis. This decomposition is based on local characteristic of local time domain of data. Based on this characteristic, any nonlinear and nonstationary signal can be decomposed into a set of intrinsic mode functions (IMFs) which are amplitude and frequency modulated signals. Each IMF should satisfy two basic conditions: (1) The number of extreme points and the number of zero crossings must be either equal or differ at most by one; (2) At any time point, the local mean value of the envelope which is defined by the average of the maximum and minimum envelopes is zero. The first condition is similar to the narrowband requirement for a stationary Gaussian process. The second condition modifies a global requirement to a local one, and is necessary to ensure that the instantaneous frequency will not have unwanted fluctuations as induced by asymmetric waveforms.The IMFs are obtained by using the EMD decomposition algorithm and are denoted as $c_i(t)$ $[1]$.

At the end of the algorithm, the original signal $x(t)$ can be represented as:

$$x(t) = \sum_{i=1}^{n} c_i(t) + r_n(t)$$

'n' is the number of intrinsic modes. $c_i(t)$ is the ith IMF, and $r_n(t)$ is the final residual which can be interpreted as the DC component of the signal.For any real IMF $c(t)$,its Hilbert transform $c_H(t)$ is defined as:

$$c_H(t) = \frac{1}{\pi} P \int_{-\infty}^{\infty} \frac{c(\tau)}{t-\tau} d\tau$$

where P is the Cauchy principal value of the singular integral. Then the analytic signal of the $c(t)$ can be defined as below:

$$z(t) = c(t) + jc_H(t) = a(t)e^{j\phi(t)}$$

The amplitude of pre envelope $a(t)$ and instantaneous phase $\phi(t)$ are defined as:

$$a(t) = \sqrt{c(t)^2 + c_H(t)^2}$$

$$\phi(t) = arctan\frac{c_H(t)}{c(t)}$$

The instantaneous frequency can then be written as the time derivative of the phase, as shown below:

$$w(t) = \frac{d\phi(t)}{dt}$$

The marginal hilbert spectrum is defined as:

$$h(w) = \int_0^T H(w,t)dt$$

In order to calculate the entropy, the spectrum should be converted into a probability mass function by normalizing the spectrum firstly.

Normalized Spectrum - Probability Mass Function

$$p_i = \frac{P_i}{\sum_{i=1}^{n} P_i}$$

where P_i is the energy of the ith frequency component of the spectrum and p_i is the probability mass function of the spectrum.

Spectral Entropies are defined as follows:

Shannon Entropy

$$SEN = -\sum_{i=1}^{n} p_i \log p_i$$

where p_i is the probability of occurrence of an event, here it refers the probability density of the spectrum.

Renyi Entropy

$$REN_\alpha = \frac{1}{1-\alpha} \log \sum_{i=1}^{n} p_i^\alpha$$

where the additional parameter α is used to make it more or lesssensitive to the shape of probability distributions.

Tsallis Entropy

$$TEN_\alpha = \frac{1}{\alpha-1}(1 - \sum_{i=1}^{n} p_i^\alpha)$$

Renyi and Tsallis entropies reduce to the Shannon entropy incase of α = 1. In this work, the parameter α of Renyi and Tsallis entropy are both set to 2.

Band Energy Features

The energy features are effective features for epileptic seizure detection. The energy features based on the frequency bands of the rhythms in EEG signals (namely delta: 0–4 Hz; theta:4–8 Hz; alpha: 8–12 Hz; beta: 12–30 Hz; gamma: 30–50 Hz) are extracted. The sub-band energy isdefined as the summation of the magnitude of squared spectrum components.

$$e_i = \sum_{l=0}^{k-1} h_l^2$$

where h_i is the i sub band of the spectrum and k is the total number of frequency in each band. Thus, the energy feature vector can be formed with the five features as E = [e1, e2, e3, e4, e5].

Implementation

1. A-E Classification (HMS Features)

```
%System Requirements
%MATLAB 2018a - Signal Processing Toolbox
%Download the dataset from the link [3]

clear all;
close all;

%Number of IMFs to use
for comp = 1:5
    %Iterate through the datasets
    for dataset = 1:100

        %Load the dataset for Non-Seizure Data
        data = load(strcat('Z/Z',sprintf('%03d', dataset),'.txt'));
        sampleRate = 173.61; %Hz

        %Resolution
        len=512;
        resolution=sampleRate/512;

        %Empirical Mode Decomposition
        imf = emd(data);

        %Hilbert-Huang Transform
        [HSpect,F(:,dataset),T] = hht(imf(:,1:comp),sampleRate,'FrequencyResolution',resolution);

        %Computing the Marginal Spectrum
        margSpect(:,dataset) = sum(HSpect,2);

        %Computing the Probability distribution
        margSpectEner(:,dataset) = sum(HSpect.^2,2);
        marSpectProb(:,dataset) = margSpectEner(:,dataset)./sum(margSpectEner(:,dataset));

        %Shannon-Entropy
        SEN_non(dataset) = -1*sum((marSpectProb(:,dataset)).*log(0.0001+marSpectProb(:,dataset)));

        alpha=2;
        %Renyi Entropy
        REN_non(dataset) = (1/(1-alpha))*log(sum(marSpectProb(:,dataset).^alpha));

        %Tsallis Entropy
        TEN_non(dataset) = (1/(alpha-1))*(1-(sum(marSpectProb(:,dataset).^alpha)));

        %Compute Energy Features
        e1_non(dataset) = computeEnergy(margSpect(:,dataset),0,4,resolution);
        e2_non(dataset) = computeEnergy(margSpect(:,dataset),4,8,resolution);
        e3_non(dataset) = computeEnergy(margSpect(:,dataset),8,12,resolution);
        e4_non(dataset) = computeEnergy(margSpect(:,dataset),12,30,resolution);
        e5_non(dataset) = computeEnergy(margSpect(:,dataset),30,50,resolution);

    end


    for dataset = 1:100
        %Load the dataset for Seizure Data
        data = load(strcat('S/S',sprintf('%03d', dataset),'.txt'));
        sampleRate = 173.61;

        %Empirical Mode Decomposition
        imf = emd(data);

        %Hilbert-Huang Transform
        [HSpect,F(:,dataset),T] = hht(imf(:,1:comp),sampleRate,'FrequencyResolution',resolution);

        %Computing the Marginal Spectrum
        margSpectSez(:,dataset) = sum(HSpect,2);

        %Computing the Probability distribution
        margSpectEnerSez(:,dataset) = sum(HSpect.^2,2);
        marSpectProbSez(:,dataset) = margSpectEnerSez(:,dataset)./sum(margSpectEnerSez(:,dataset));

        %Shannon-Entropy
        SEN_sez(dataset) = -1*sum((marSpectProbSez(:,dataset)).*log(0.0001+marSpectProbSez(:,dataset)));

        alpha=2;
        %Renyi Entropy
        REN_sez(dataset) = (1/(1-alpha))*log(sum(marSpectProbSez(:,dataset).^alpha));

        %Tsallis Entropy
        TEN_sez(dataset) = (1/(alpha-1))*(1-(sum(marSpectProbSez(:,dataset).^alpha)));

        e1_sez(dataset) = computeEnergy(margSpectSez(:,dataset),0,4,resolution);
        e2_sez(dataset) = computeEnergy(margSpectSez(:,dataset),4,8,resolution);
        e3_sez(dataset) = computeEnergy(margSpectSez(:,dataset),8,12,resolution);
        e4_sez(dataset) = computeEnergy(margSpectSez(:,dataset),12,30,resolution);
        e5_sez(dataset) = computeEnergy(margSpectSez(:,dataset),30,50,resolution);

    end

    %Feature Aggregation
    features_non = full([SEN_non;REN_non;TEN_non;e1_non;e2_non;e3_non;e4_non])';
    features_sez = full([SEN_sez;REN_sez;TEN_sez;e1_sez;e2_sez;e3_sez;e4_sez])';
```

```
    features = [features_non;features_sez];
    labels = [ones(1,length(features_non)),-1*ones(1,length(features_sez))]';

    %SVM Classification
    cl_svm = fitcsvm(features,labels,'KernelFunction','rbf','ClassNames',[1,-1],'BoxConstraint',5.7,'KernelScale',85.36);

    %10-fold Cross-validation
    svm_models = crossval(cl_svm);
    accuracy(comp) = (1-kfoldLoss(svm_models))*100;
end
accuracyHMS = mean(accuracy)
```

2. ABCD-E Classification (HMS Features)

```
clear all;
close all;

driveletter = {'Z/Z','O/O','F/F','N/N'};
%Number of IMFs to use

for comp = 1:5
    for subset = 1:4
    %Iterate through the datasets
    for dataset = 1:100


        %Load the dataset for Non-Seizure Data
        data = load(strcat(driveletter{subset},sprintf('%03d', dataset),'.txt'));
        sampleRate = 173.61; %Hz

        %Resolution
        len=512;
        resolution=sampleRate/512;

        %Empirical Mode Decomposition
        imf = emd(data);

        %Hilbert-Huang Transform
        [HSpect,F(:,dataset),T] = hht(imf(:,1:comp),sampleRate,'FrequencyResolution',resolution);

        %Computing the Marginal Spectrum
        margSpect(:,dataset) = sum(HSpect,2);


        %Computing the Probability distribution
        margSpectEner(:,dataset) = sum(HSpect.^2,2);
        marSpectProb(:,dataset) = margSpectEner(:,dataset)./sum(margSpectEner(:,dataset));

        %Shannon-Entropy
        SEN_non(subset,dataset) = -1*sum((marSpectProb(:,dataset)).*log(1e-4+marSpectProb(:,dataset)));

        alpha=2;
        %Renyi Entropy
        REN_non(subset,dataset) = (1/(1-alpha))*log(sum(marSpectProb(:,dataset).^alpha));

        %Tsallis Entropy
        TEN_non(subset,dataset) = (1/(alpha-1))*(1-(sum(marSpectProb(:,dataset).^alpha)));


        %Compute Energy Features
        e1_non(subset,dataset) = computeEnergy(margSpect(:,dataset),0,4,resolution);
        e2_non(subset,dataset) = computeEnergy(margSpect(:,dataset),4,8,resolution);
        e3_non(subset,dataset) = computeEnergy(margSpect(:,dataset),8,12,resolution);
        e4_non(subset,dataset) = computeEnergy(margSpect(:,dataset),12,30,resolution);
        e5_non(subset,dataset) = computeEnergy(margSpect(:,dataset),30,50,resolution);

       end
    end

    for dataset = 1:100
        %Load the dataset for Seizure Data
        data = load(strcat('S/S',sprintf('%03d', dataset),'.txt'));
        sampleRate = 173.61;

        %Empirical Mode Decomposition
        imf = emd(data);

        %Hilbert-Huang Transform
        [HSpect,F(:,dataset),T] = hht(imf(:,1:comp),sampleRate,'FrequencyResolution',resolution);

        %Computing the Marginal Spectrum
        margSpectSez(:,dataset) = sum(HSpect,2);

        %Computing the Probability distribution
        margSpectEnerSez(:,dataset) = sum(HSpect.^2,2);
        marSpectProbSez(:,dataset) = margSpectEnerSez(:,dataset)./sum(margSpectEnerSez(:,dataset));

        %Shannon-Entropy
        SEN_sez(dataset) = -1*sum((marSpectProbSez(:,dataset)).*log(1e-4+marSpectProbSez(:,dataset)));

        alpha=2;
        %Renyi Entropy
        REN_sez(dataset) = (1/(1-alpha))*log(sum(marSpectProbSez(:,dataset).^alpha));

        %Tsallis Entropy
        TEN_sez(dataset) = (1/(alpha-1))*(1-(sum(marSpectProbSez(:,dataset).^alpha)));

        e1_sez(dataset) = computeEnergy(margSpectSez(:,dataset),0,4,resolution);
        e2_sez(dataset) = computeEnergy(margSpectSez(:,dataset),4,8,resolution);
        e3_sez(dataset) = computeEnergy(margSpectSez(:,dataset),8,12,resolution);
        e4_sez(dataset) = computeEnergy(margSpectSez(:,dataset),12,30,resolution);
        e5_sez(dataset) = computeEnergy(margSpectSez(:,dataset),30,50,resolution);

    end
```

```
    %Feature Aggregation
    features_non = full([SEN_non(:),REN_non(:),TEN_non(:),e1_non(:),e2_non(:),e3_non(:),e4_non(:)]);
    features_sez = full([SEN_sez;REN_sez;TEN_sez;e1_sez;e2_sez;e3_sez;e4_sez])';

    features = [features_non;features_sez];
    labels = [ones(1,length(features_non)),-1*ones(1,length(features_sez))]';

    %SVM Classification
    cl_svm = fitcsvm(features,labels,'KernelFunction','rbf','ClassNames',[1,-1],'BoxConstraint',89.75,'KernelScale',16.6);

    %10-fold Cross-validation
    svm_models = crossval(cl_svm);
    accuracy(comp) = (1-kfoldLoss(svm_models))*100;
end
accuracyHMS = mean(accuracy)
```

3. A-E Classification (Fourier Spectrum Features)

```
clear all;
close all;

%Iterate through the datasets
for dataset =1:100

    %Load the dataset for Non-Seizure Data
    data = load(strcat('Z/Z',sprintf('%03d', dataset),'.txt'));

    sampleRate = 173.61;
    len = 512;

    NFFT1 =(2^(nextpow2(len)));

    %Computing Fourier Spectrum
    signalFft = fft(data,NFFT1)/len;
    signalFftMagClass2(1:NFFT1/2,dataset)= 2*abs(signalFft(1:NFFT1/2));

    %Computing the Probability distribution
    signalFftprob(1:NFFT1/2)=signalFftMagClass2(1:NFFT1/2,dataset)./sum(signalFftMagClass2(1:NFFT1/2,dataset));

    fft_axis =(0:NFFT1/2-1)*(sampleRate/NFFT1);

    %Shannon-Entropy
    SEN_non(dataset) = -1*sum((signalFftprob).*log(signalFftprob));

    %Renyi Entropy
    alpha=2;
    REN_non(dataset) = (1/(1-alpha))*log(sum(signalFftprob.^alpha));

    %Tsallis Entropy
    TEN_non(dataset) = (1/(alpha-1))*(1-(sum(signalFftprob.^alpha)));

    resolution = sampleRate/len;

    %Compute Energy Features
    e1_non(dataset) = computeEnergy(signalFftMagClass2(:,dataset),0,4,resolution);
    e2_non(dataset) = computeEnergy(signalFftMagClass2(:,dataset),4,8,resolution);
    e3_non(dataset) = computeEnergy(signalFftMagClass2(:,dataset),8,12,resolution);
    e4_non(dataset) = computeEnergy(signalFftMagClass2(:,dataset),12,30,resolution);
    e5_non(dataset) = computeEnergy(signalFftMagClass2(:,dataset),30,50,resolution);
end

%Iterate through the datasets
for dataset = 1:100
        %Load the dataset for Seizure Data
        data = load(strcat('S/S',sprintf('%03d', dataset),'.txt'));
        sampleRate = 173.61;

        len = 512;
        NFFT1 =(2^(nextpow2(len)));

        %Computing Fourier Spectrum
        signalFft = fft(data,NFFT1)/len;
        signalFftMagClass2(1:NFFT1/2,dataset)= 2*abs(signalFft(1:NFFT1/2));

        %Computing the Probability distribution
        signalFftprob(1:NFFT1/2)=signalFftMagClass2(1:NFFT1/2,dataset)./sum(signalFftMagClass2(1:NFFT1/2,dataset));
        fft_axis =(0:NFFT1/2-1)*(sampleRate/NFFT1);

        %Shannon-Entropy
        SEN_sez(dataset) = -1*sum((signalFftprob).*log(signalFftprob));

        %Renyi Entropy
        alpha = 2;
        REN_sez(dataset) = (1/(1-alpha))*log(sum(signalFftprob.^alpha));

        %Tsallis Entropy
        TEN_sez(dataset) = (1/(alpha-1))*(1-(sum(signalFftprob.^alpha)));

        resolution = sampleRate/len;

        %Compute Energy Features
        e1_sez(dataset) = computeEnergy(signalFftMagClass2(:,dataset),0,4,resolution);
        e2_sez(dataset) = computeEnergy(signalFftMagClass2(:,dataset),4,8,resolution);
        e3_sez(dataset) = computeEnergy(signalFftMagClass2(:,dataset),8,12,resolution);
        e4_sez(dataset) = computeEnergy(signalFftMagClass2(:,dataset),12,30,resolution);
        e5_sez(dataset) = computeEnergy(signalFftMagClass2(:,dataset),30,50,resolution);

end

%Feature Aggregation
features_non = [SEN_non;REN_non;TEN_non;e1_non;e2_non;e3_non;e4_non]';
features_sez = [SEN_sez;REN_sez;TEN_sez;e1_sez;e2_sez;e3_sez;e4_sez]';
```

```
features = [features_non;features_sez];
labels = [ones(1,length(features_non)),-1*ones(1,length(features_sez))]';

%SVM Classification
cl_svm = fitcsvm(features,labels,'KernelFunction','rbf','ClassNames',[1,-1],'BoxConstraint',42.34,'KernelScale',27.30);

%10-fold Cross-validation
svm_models = crossval(cl_svm);
accuracyFFT = (1-kfoldLoss(svm_models))*100
```

4. ABCD-E Classification (Fourier Spectrum Features)

```
clear all;
close all;

driveletter = {'Z/Z','O/O','F/F','N/N'};

for subset = 1:4
    %Iterate through the datasets
    for dataset = 1:100

        %Load the dataset for Non-Seizure Data
        data = load(strcat(driveletter{subset},sprintf('%03d', dataset),'.txt'));

        sampleRate = 173.61;
        len = 512;
        NFFT1 =(2^(nextpow2(len)));

        %Computing Fourier Spectrum
        signalFft = fft(data,NFFT1)/len;
        signalFftMagClass2(1:NFFT1/2,dataset)= 2*abs(signalFft(1:NFFT1/2));

        %Computing the Probability distribution
        signalFftprob(1:NFFT1/2)=(signalFftMagClass2(1:NFFT1/2,dataset).^2)./sum((signalFftMagClass2(1:NFFT1/2,dataset).^2));

        fft_axis =(0:NFFT1/2-1)*(sampleRate/NFFT1);

        %Shannon-Entropy
        SEN_non(subset,dataset) = -1*sum((signalFftprob).*log(signalFftprob));

        %Renyi-Entropy
        alpha=2;
        REN_non(subset,dataset) = (1/(1-alpha))*log(sum(signalFftprob.^alpha));

        %Tsallis-Entropy
        TEN_non(subset,dataset) = (1/(alpha-1))*(1-(sum(signalFftprob.^alpha)));

        %Compute Energy Features
        resolution = sampleRate/len;

        e1_non(subset,dataset) = computeEnergy(signalFftMagClass2(:,dataset),0,4,resolution);
        e2_non(subset,dataset) = computeEnergy(signalFftMagClass2(:,dataset),4,8,resolution);
        e3_non(subset,dataset) = computeEnergy(signalFftMagClass2(:,dataset),8,12,resolution);
        e4_non(subset,dataset) = computeEnergy(signalFftMagClass2(:,dataset),12,30,resolution);
        e5_non(subset,dataset) = computeEnergy(signalFftMagClass2(:,dataset),30,50,resolution);
    end
end


for dataset = 1:100
    %Load the dataset for Seizure Data
    data = load(strcat('S/S',sprintf('%03d', dataset),'.txt'));

    sampleRate = 173.61;
    len = 512;
    NFFT1 =(2^(nextpow2(len)));

    %Computing Fourier Spectrum
    signalFft = fft(data,NFFT1)/len;
    signalFftMagClass2(1:NFFT1/2,dataset)= 2*abs(signalFft(1:NFFT1/2));

    %Computing the Probability distribution
    signalFftprob(1:NFFT1/2)=(signalFftMagClass2(1:NFFT1/2,dataset).^2)./sum((signalFftMagClass2(1:NFFT1/2,dataset).^2));

    fft_axis =(0:NFFT1/2-1)*(sampleRate/NFFT1);

    %Shannon-Entropy
    SEN_sez(dataset) = -1*sum((signalFftprob).*log(signalFftprob));

    %Renyi-Entropy
    alpha = 2;
    REN_sez(dataset) = (1/(1-alpha))*log(sum(signalFftprob.^alpha));

    %Tsallis-Entropy
    TEN_sez(dataset) = (1/(alpha-1))*(1-(sum(signalFftprob.^alpha)));

    %Compute Energy Features
    resolution = sampleRate/len;
    e1_sez(dataset) = computeEnergy(signalFftMagClass2(:,dataset),0,4,resolution);
    e2_sez(dataset) = computeEnergy(signalFftMagClass2(:,dataset),4,8,resolution);
    e3_sez(dataset) = computeEnergy(signalFftMagClass2(:,dataset),8,12,resolution);
    e4_sez(dataset) = computeEnergy(signalFftMagClass2(:,dataset),12,30,resolution);
    e5_sez(dataset) = computeEnergy(signalFftMagClass2(:,dataset),30,50,resolution);
end

%Feature Aggregation
features_non = full([SEN_non(:),REN_non(:),TEN_non(:),e1_non(:),e2_non(:),e3_non(:),e4_non(:)]);
features_sez = [SEN_sez;REN_sez;TEN_sez;e1_sez;e2_sez;e3_sez;e4_sez]';

features = [features_non;features_sez];
labels = [ones(1,length(features_non)),-1*ones(1,length(features_sez))]';

%SVM Classification
```

```
cl_svm = fitcsvm(features,labels,'KernelFunction','rbf','ClassNames',[1,-1],'BoxConstraint',71.31,'KernelScale',13.72);

%10-fold Cross-validation
svm_models = crossval(cl_svm);
accuracyFFT = (1-kfoldLoss(svm_models))*100
```

Band Energy Calculation

```
function bandEnergy = computeEnergy(margSpect,startFreq,endFreq,resolution)
        %Calculate frequency start and end index
        hftIndexStart = round(startFreq/resolution)+1;
        hftIndexEnd = round(endFreq/resolution)+1;
        %Calculate Band Energy
        bandEnergy = log(sum(margSpect(hftIndexStart:hftIndexEnd,:).^2));
end
```

Entropy Plots and Box Plot

```
close all

figure(1)
scatter(1:100,SEN_non(1,:),'ro')
hold on
scatter(1:100,SEN_sez(1,:),'b*'); title('Shannon');
legend('NS','S','Location','southeast'); box on;

figure(2)
scatter(1:100,TEN_non(1,:),'ro')
hold on
scatter(1:100,TEN_sez(1,:),'b*'); title('Tsallis');
legend('NS','S','Location','southeast'); box on;

figure(3)
scatter(1:100,REN_non(1,:),'ro')
hold on
scatter(1:100,REN_sez(1,:),'b*'); title('Renyi');
legend('NS','S','Location','southeast'); box on;


figure;
boxplot([e1_non',e1_sez',e2_non',e2_sez',e3_non',e3_sez',e4_non',e4_sez',e5_non',e5_sez']);
box on
set(gca, 'XTick',[1:10],'xticklabel',{'NS';'S';'NS';'S';'NS';'S';'NS';'S';'NS';'S';});xlabel('  Delta              Theta              Alpha              Beta              Gamma');ylabel('log(EN)');
```

Results and Discussions

Fig. 2 illustrates exemplary fourier spectra from the non-siezure and siezure datasets. Fig. 3. shows exemplary Hilbert Marginal Spetra for siezure and non-siezure Signals using the first three IMF components.

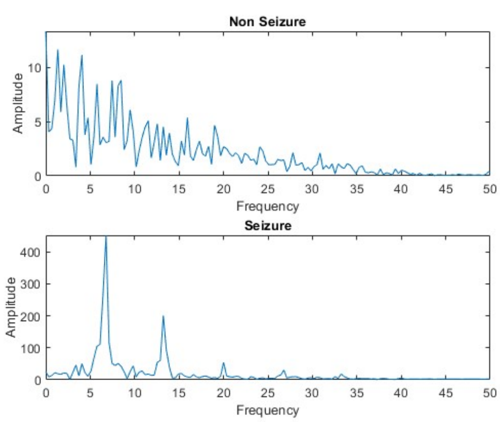


Fig. 2. Fourier Spectra for siezure and non-siezure signals

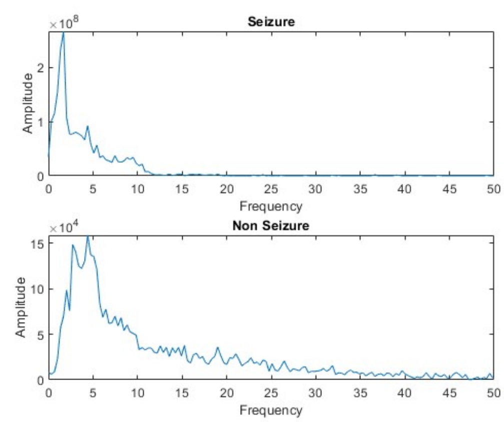
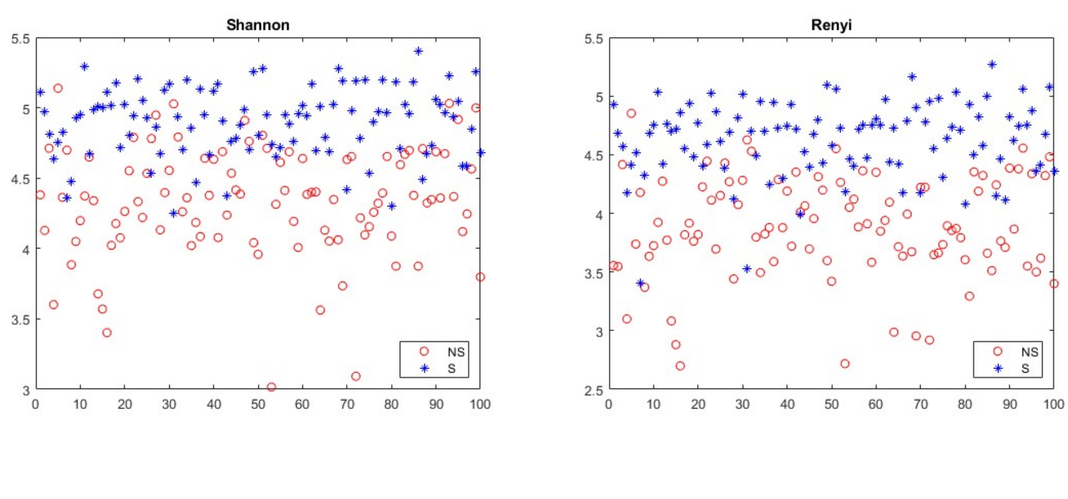


Fig. 3. Hilbert Marginal Spectrum for siezure and non-siezure signals

Fig. 4 shows the relative entropy distribution of Shannon, Renyi and Tsallis of seizure and nonseizure EEG signals extracted using the HMS method. We observe that the spectral entropy values of the seizure EEG signal are greater than the non-seizure EEG signal for first IMF component. However, the entropies for seizure signals are lower than the non-seizure signals when more than one IMFs were considered. These spectral entropies features along with the band energy features are used for automatic seizure detection using a SVM classifier. Fig. 5 illustrates the box plots of all the five sub-band energy features of seizure and non-seizure EEG signals. It can be observed that sub band energies for seizure signals are greater than non-seizure signals.



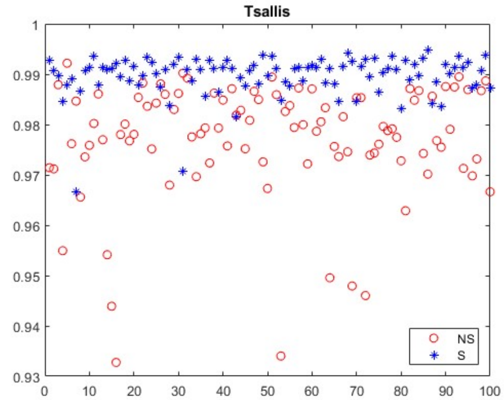


Fig. 4. Relative entropy distribution features based on Shannon, Renyi and Tsallis entropies of seizure (S) and non-seizure (NS) EEG signals for the first IMF component

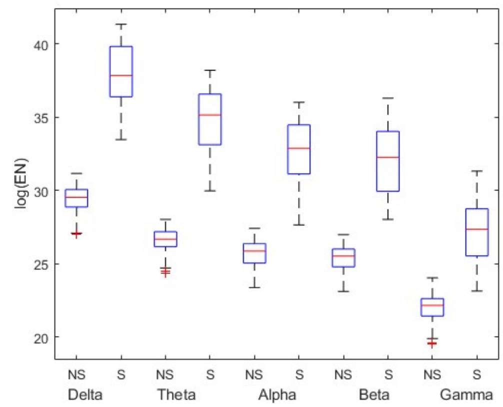


Fig. 5. Box plot of each sub-band energy features of seizure (S) and non-seizure (NS) EEG signals

The classification between Seizure and Non-Seizure signals has been performed based on two groups of datasets. The first group is classification between A-E and the second group is the classification between ABCD-E. The HMS and Fourier Spectrum based features are extracted in these two cases. The groups are as follows:

1. A-E : Dataset A consisting of segments acquired from surface EEG recordings of five healthy volunteers with eyes open and closed. Subset E contains the seizure activity
2. ABCD-E : The features from datasets ABCD are combined and used for non-seizure features training and test data

A Support Vector Machine (SVM) with a radial basis function (RBF) kernel is used as the classifier. The RBF kernel performs a non-linear transformation which projects the features into a higher dimensional feature space. The parameters (c,g) used in the current implementation are provided by the authors. This classifier is validated based on a 10-fold cross-validation technique. The classification accuracies are as shown in Table 1. The accuracies for HMS based feature extraction for the A-E group is 99.70% and ABCD-E group is 98.28% ; providing an improvement of 0.7% and 0.2% respectively, compared to the Fourier Spectrum based features.

Table 1. Classification accuracies for HMS and FFT based feature extraction methods

	Hilbert Marginal Spectrum		Fourier Spectrum	
	A-E (c = 5.70, g = 85.36)	ABCD–E (c = 89.75, g = 16.60)	A–E (c = 42.34, g = 27.30)	ABCD–E (c = 71.31, g = 13.72)
Project Implementation Results	**99.70%**	**98.28%**	**99.00%**	**98.08%**
Results Reported in the Paper	99.85%	98.80%	99.30%	98.16%

Conclusions

It is evident from the classification accuracies that the proposed feature extraction method based on HMS provides a good representation of the data for seizure detection. The results obtained from our implementation are consistent with the findings reported in the paper. However, when comparing the individual accuracies, the paper presents higher accuracies. These can be attributed to the following factors: length of the signal considered for the analysis and number of IMFs used in the analysis. On performing further analysis, we find that the classification accuracy of SVM changes with varying number of IMF components. There is a notable difference in the statistics of the entropy features derived in our analysis and of that presented in the paper. When more than one IMF is used, the entropy values of seizure data is lower than non-seizure data. When using only the first IMF, the results are consistent with the results reported in the paper. Regardeless of the number of IMFs used, the SVM trained on the HMS based features always performs better compared to FFT based features.

References

[1] Fu, K., Qu, J., Chai, Y., & Zou, T. (2015). Hilbert marginal spectrum analysis for automatic seizure detection in EEG signals. *Biomedical Signal Processing and Control*, *18*, 179-185.

[2] Andrzejak, R. G., Lehnertz, K., Mormann, F., Rieke, C., David, P., & Elger, C. E. (2001). Indications of nonlinear deterministic and finite-dimensional structures in time series of brain electrical activity: Dependence on recording region and brain state. *Physical Review E*, *64*(6), 061907.

[3] Dataset - http://epileptologie-bonn.de/cms/front_content.php?idcat=193&lang=3&changelang=3 (Accessed on 16/08/2018)