Laboratory Report No. 4

Student of group IU1-41M Goriainov Igor

1 Цель работы

Целью данной лабораторной работы является исследование скрытых характеристик сигналов электроэнцефалографии на основе датасета записей ЭЭГ с приступами.

2 Ключевой навык

Ключевой получаемый навык - умение расчитывать спектрограмму и скейлограммусигнала и применять вейвлет преобразование.

3 Задание

В ходе выполнения лабораторной работы требуется:
1. Загрузите запись ЭЭГ из базы данных, выбрать любой файл EDF; 2. По аннотации из датасета определить, где в записи ЭЭГ указан приступ; 3. Построить график временной зависимости ЭЭГ в момент приступа (выбрать диапазон времени и масштаб так, чтобы было наглядно). Сделать усреднение всех каналов ЭЭГ в один. Удалить из сигнала все частоты выше 60 Гц. Для этого преобразованного сигнала:
1) построить спектрограмму сигнала;
2) построить вейвлет-преобразование (скейлограмму).

1 工作目的

本实验的目的是基于脑电图(EEG)数据集研究癫痫发作期间脑电信号的隐藏特征。

2 关键技能

获得的核心技能是计算信号的频谱图和尺度图，并应用小波变换。

3 实验任务

在实验过程中需要完成：
1. 从数据库中加载EEG记录，选择任意EDF格式文件；
2. 根据数据集中的标注确定EEG记录中癫痫发作的位置；
3. 绘制癫痫发作时EEG的时间依赖图（选择合适的时间范围和比例使其清晰可见）。将所有EEG通道平均为一个信号。去除信号中60Hz以上的频率。对于这个转换后的信号：
1) 绘制信号的频谱图；
2) 绘制小波变换(尺度图)。

4 Описание метода исследования

В ходе лабораторной работы после предварительной обработки ЭЭГ-сигнала (усреднение по каналам и фильтрация частот выше 60 Гц), производится спектральный анализ сигнала двумя методами[1]: спектрограммой и непрерывным вейвлет-преобразованием (CWT). Оба метода позволяют изучить, как энергетика сигнала распределена по частотам во времени, что особенно важно для детектирования кратковременных патологических состояний, таких как эпилептический приступ.

1. Построение спектрограммы сигнала

Спектрограмма строится на основе оконного преобразования Фурье (Short-Time Fourier Transform, STFT)[2]. В этом методе сигнал разбивается на перекрывающиеся окна, внутри которых выполняется обычное преобразование Фурье. Это позволяет получить спектральную информацию, локализованную во времени.

Математически STFT определяется как:

$$\mathrm{STFT}_x(t,\omega) = \int_{-\infty}^{\infty} x(\tau)\cdot w(\tau - t)\cdot e^{-j\omega\tau}\, d\tau \tag{1}$$

где:

- $x(\tau)$ — исходный сигнал,
- $w(\tau - t)$ — оконная функция, сдвинутая по времени (например, окно Хэмминга),
- w — угловая частота,
- τ — момент времени.

4 研究方法描述

在实验室工作中，经过EEG信号的预处理（通道平均和过滤60Hz以上频率）后，[1] 采用两种方法进行信号频谱分析：频谱图和连续小波变换(CWT)。这两种方法都可以研究信号能量如何随时间在频率上分布，这对于检测短暂的病理状态（如癫痫发作）特别重要。

1. 信号频谱图的构建

频谱图基于短时傅里叶变换(STFT)[2] 构建。在这种方法中，信号被分成重叠的窗口，在每个窗口内执行常规傅里叶变换。这样可以获得时间上局部化的频谱信息。

STFT的数学定义为：

$$\mathrm{STFT}_x(t,\omega) = \int_{-\infty}^{\infty} x(\tau)\cdot w(\tau - t)\cdot e^{-j\omega\tau}\, d\tau \tag{1}$$

其中：

- $x(\tau)$ —— 原始信号，
- $w(\tau - t)$ —— 时间平移的窗函数（例如汉明窗），
- ω —— 角频率，
- τ —— 时间点。

频谱图是STFT模的平方：

$$\mathrm{Spectrogram}_x(t,\omega) = |\mathrm{STFT}_x(t,\omega)|^2 \tag{2}$$

这种二维表示显示了信号在每个时间点存在的频率分量。[3]

2. 小波变换(尺度图)

与STFT不同，连续小波变换(CWT)可以自适应地改变时间和频率分辨率，为高频分量提供高时间精度，为低频分量提供高频率精度。

CWT定义如下：

$$\mathrm{CWT}_x(a,b) = \frac{1}{\sqrt{|a|}} \int_{-\infty}^{\infty} x(t)\cdot \psi^*\left(\frac{t-b}{a}\right) dt \tag{3}$$

其中：

- $x(t)$ —— 待分析信号，

Спектрограмма получается как квадрат модуля STFT:

$$\text{Spectrogram}_x(t,\omega) = |\text{STFT}_x(t,\omega)|^2 \tag{2}$$

Это двумерное представление показывает, какие частотные компоненты присутствуют в сигнале в каждый момент времени[3].

2. Вейвлет-преобразование (скейлограмма)

В отличие от STFT, непрерывное вейвлет-преобразование (CWT) позволяет адаптивно изменять временное и частотное разрешение, обеспечивая высокую временную точность для высокочастотных компонентов и высокую частотную точность для низкочастотных.

CWT определяется следующим образом:

$$\text{CWT}_x(a,b) = \frac{1}{\sqrt{|a|}} \int_{-\infty}^{\infty} x(t) \cdot \psi^* \left(\frac{t-b}{a} \right) dt \tag{3}$$

где:

- $x(t)$ — анализируемый сигнал,
- $\psi(t)$ — вейвлет-функция (например, комплексный Морле),
- $\psi^*(t)$ — комплексно-сопряжённая функция,
- a — масштаб (обратно пропорционален частоте),
- b — временной сдвиг,
- $\frac{1}{\sqrt{|a|}}$ — нормирующий коэффициент, обеспечивающий сохранение энергии при изменении масштаба.

Результат отображается на скейлограмме, где по оси X — время, по оси Y — масштаб или эквивалентная частота, а цвет отражает энергию (амплитуду преобразования) на соответствующем масштабе и моменте времени:

- $\psi(t)$ —— 小波函数（例如复Morlet小波），
- $\psi^*(t)$ —— 复共轭函数，
- a —— 尺度（与频率成反比），
- b —— 时间平移，
- $\frac{1}{\sqrt{|a|}}$ —— 归一化系数，确保尺度变化时能量守恒。

结果在尺度图上显示，其中X轴是时间，Y轴是尺度或等效频率，颜色反映了相应尺度和时间点的能量（变换幅度）：

$$\text{Scalogram}_x(a,b) = |\text{CWT}_x(a,b)|^2 \tag{4}$$

两种方法的结合使用可以更全面地分析癫痫发作的动态：频谱图显示高振幅频率，而尺度图可以揭示隐藏的结构和大脑活动模式之间的转换。

$$\text{Scalogram}_x(a,b) = |\text{CWT}_x(a,b)|^2 \quad (4)$$

Использование обеих методик в комплексе позволяет более полно проанализировать динамику приступа: спектрограмма показывает частоты с высокой амплитудой, а скейлограмма позволяет выявить скрытые структуры и переходы между режимами мозговой активности.

5 Ход работы

5.1 Установка библиотеки

Устанавливаем библиотеку pywavelets mne и zenodo-get.

```
!pip install pywavelets mne
!pip install zenodo_get
```

Листинг 1 Установка библиотеки

5.2 Загрузка набора данных

Загружаем набор данных.

```
!zenodo_get "10.5281/zenodo.4940267" -k -g eeg19.edf
data_from_raw_edf = mne.io.read_raw_edf('/content/eeg19.edf', preload=True)
eeg_data, eeg_times = data_from_raw_edf.get_data(return_times=True)
```

Листинг 2 Загрузка набора данных

Строим исходную ЭЭГ.

5 实验流程

5.1 安装库

安装pywavelets、mne和zenodo-get库。

```
!pip install pywavelets mne
!pip install zenodo_get
```

代码 1 安装库

5.2 加载数据集

加载数据集。

```
!zenodo_get "10.5281/zenodo.4940267" -k -g eeg19.edf
data_from_raw_edf = mne.io.read_raw_edf('/content/eeg19.edf', preload=True)
eeg_data, eeg_times = data_from_raw_edf.get_data(return_times=True)
```

代码 2 加载数据集

绘制原始EEG。

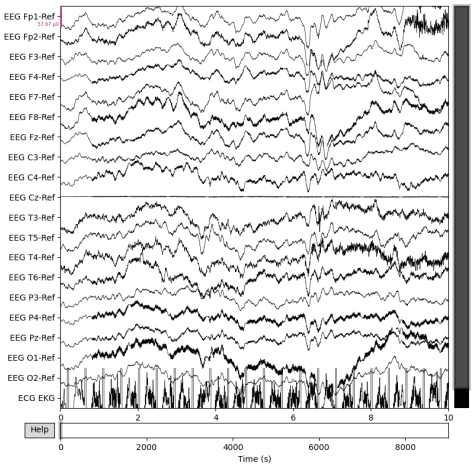

Рис. 1 Исходная ЭЭГ

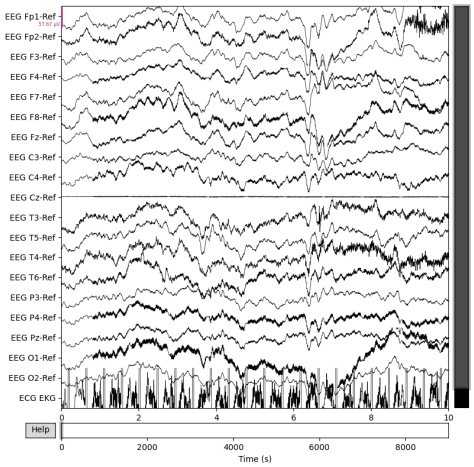

图 1 原始脑电图

Удаляем данные последнего канала ввиду неинформативности в контексте данной задачи.

```
eeg_data = np.delete(eeg_data, [
    len(eeg_data)-1], axis = 0)
```

Листинг 3 Удалаем ненужные сигналы из массива данных ЭЭГ

Построим график временной зависимости ЭЭГ в момент приступа (начало приступа 5800 отсчет).

```
t_index_begin = np.where(eeg_times >
    5500 - 480)[0][0]
t_index_end = np.where(eeg_times > 5800
    + 480)[0][0]
t = eeg_times[t_index_begin:t_index_end]
T = t[-1] - t[0]
N = len(t)
for i in range(
    len(data_from_raw_edf.ch_names)-2):
    plt.plot(t, eeg_data[i,
    t_index_begin:t_index_end],
    linewidth = 0.1)
plt.show()
```

Листинг 4 Построение графика временной зависимости ЭЭГ в момент приступа

删除最后一个通道的数据，因为在本任务背景下不具有信息性。

```
eeg_data = np.delete(eeg_data, [
    len(eeg_data)-1], axis = 0)
```

代码 3 删除不必要的脑电信号

绘制癫痫发作时EEG的时间依赖图（发作开始约5800个采样点）。

```
t_index_begin = np.where(eeg_times >
    5500 - 480)[0][0]
t_index_end = np.where(eeg_times > 5800
    + 480)[0][0]
t = eeg_times[t_index_begin:t_index_end]
T = t[-1] - t[0]
N = len(t)
for i in range(
    len(data_from_raw_edf.ch_names)-2):
    plt.plot(t, eeg_data[i,
    t_index_begin:t_index_end],
    linewidth = 0.1)
plt.show()
```

代码 4 绘制癫痫期间的脑电时间依赖图

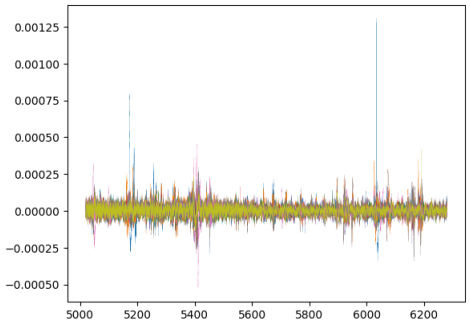

Рис. 2 Временная зависимость ЭЭГ в момент приступа

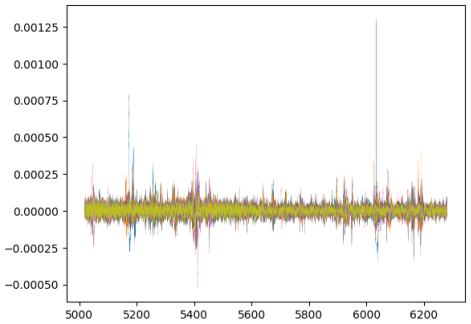

图 2 癫痫期间脑电时间依赖图

Сделаем простой устредненный сигнал - все каналы в один массив.

```
y=[]
for i in range(len(eeg_data)-2):
    y =+ eeg_data[i, t_index_begin:
     t_index_end]
y =y/(len(eeg_data)-1)
plt.plot(t,y)
```

Листинг 5 Устреднение сигнала

创建一个简单的平均信号 - 将所有通道合并为一个数组。

```
y=[]
for i in range(len(eeg_data)-2):
    y =+ eeg_data[i, t_index_begin:
     t_index_end]
y =y/(len(eeg_data)-1)
plt.plot(t,y)
```

代码 5 信号平均

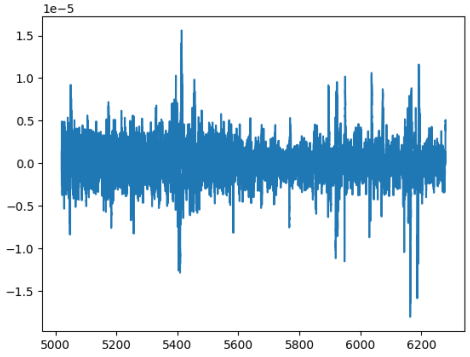

Рис. 3 Устредненный сигнал

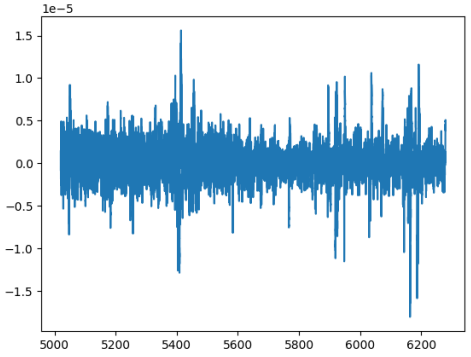

图 3 平均信号

5.3 Фильтрация сигнала

Фильтр Баттерфорда для вырезания полосы частот.

```
from scipy import signal
def butter_bandpass_filter(data, lowcut,
     highcut, fs, order=5):
```

5.3 信号滤波

巴特沃斯滤波器用于切除频带。

```
from scipy import signal
def butter_bandpass_filter(data, lowcut,
     highcut, fs, order=5):
    nyquist = 0.5 * fs
    low = lowcut / nyquist
    high = highcut / nyquist
```

```
    nyquist = 0.5 * fs
    low = lowcut / nyquist
    high = highcut / nyquist
    b, a = signal.butter(order, [low, high], btype='band')
    filtered_data = signal.lfilter(b, a, data)
    return filtered_data

fs = len(t)/(T)
lowcut = 1
highcut = 60
y_filt = np.apply_along_axis(butter_bandpass_filter, axis=0, arr=y, lowcut=lowcut, highcut=highcut, fs=fs)
plt.plot(t,y, t, y_filt)
```

Листинг 6 Фильтр Баттерфорда

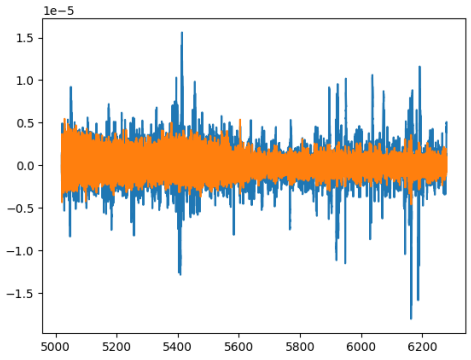

Рис. 4 Устредненный сигнал после фильтрации

5.4 Спектр Фурье

```
from scipy.fft import fft, fftfreq

y = y_filt
yfft = fft(y)
xf = fftfreq(N, T/N)[:N//2]
yf = np.abs(yfft[0:N//2])

f_viewmax = 60
if len(np.where(xf > f_viewmax)[0]) == 0:
  index_f_viewmax = len(xf)
else:
  index_f_viewmax = np.where(xf > f_viewmax)[0][0]
print('chislo tochek v diapazone preobrazovaniya Fur'e:',len(xf))
print('maksimal'naya chastota', xf[-1])
plt.figure(figsize=(10, 5))
```

```
    b, a = signal.butter(order, [low, high], btype='band')
    filtered_data = signal.lfilter(b, a, data)
    return filtered_data

fs = len(t)/(T)
lowcut = 1
highcut = 60
y_filt = np.apply_along_axis(butter_bandpass_filter, axis=0, arr=y, lowcut=lowcut, highcut=highcut, fs=fs)
plt.plot(t,y, t, y_filt)
```

代码 6 巴特沃斯滤波器

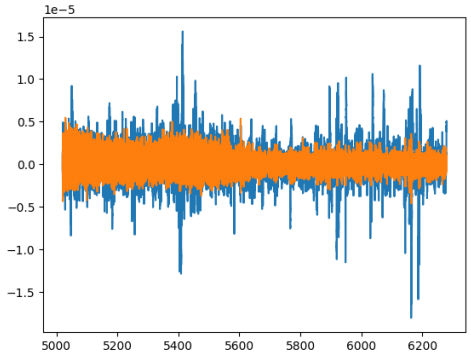

图 4 滤波后的平均信号

5.4 傅里叶频谱

```
from scipy.fft import fft, fftfreq

y = y_filt
yfft = fft(y)
xf = fftfreq(N, T/N)[:N//2]
yf = np.abs(yfft[0:N//2])

f_viewmax = 60
if len(np.where(xf > f_viewmax)[0]) == 0:
  index_f_viewmax = len(xf)
else:
  index_f_viewmax = np.where(xf > f_viewmax)[0][0]
print('Number of points in Fourier transform range:',len(xf))
print('Maximum frequency', xf[-1])
plt.figure(figsize=(10, 5))
plt.plot(xf[0:index_f_viewmax], yf[0:index_f_viewmax], label='FFT transform')
plt.xlabel('Frequency [Hz]', fontsize=12)
```

```
plt.plot(xf[0:index_f_viewmax], yf[0:
    index_f_viewmax], label='FFT
    preobrazovanie')
plt.xlabel('chastota[Hz]', fontsize=12)
plt.ylabel('spektr Fur'e', fontsize=12)
plt.grid()
plt.legend()
plt.show()
```

Листинг 7 Быстрое преобразование Фурье

Число точек в диапазоне преобразования Фурье: 161280 Максимальная частота 127.99960317337293

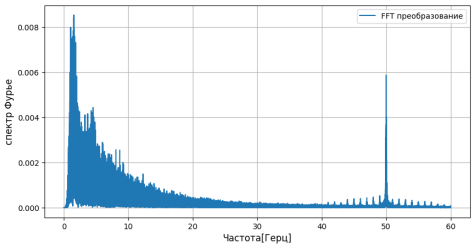

Рис. 5 FFT преобразование

5.5 Вейвлет-преобразование

```
wavlist_continuous = pywt.wavelist(kind=
    'continuous')
wavlist_discrete = pywt.wavelist(kind='
    discrete')
```

Листинг 8 Перечень вейвлетов

Изобразим график зависимости частоты и масштаба.

```
scale_max = 300
scale_min = 3
scales = np.linspace(scale_min,
    scale_max, num = 25, endpoint=True)

wavelet_core = 'morl'
fs = len(t)/(T)
dt = 1/ fs
coef, freqs = pywt.cwt(y, scales,
    wavelet_core, sampling_period = dt)

plt.figure(figsize=(7, 7))
plt.grid()
plt.yticks(np.arange(min(freqs),
    max(freqs), (max(freqs) -
    min(freqs))/10))
plt.xticks(np.arange(min(scales),
    max(scales), (max(scales) -
    min(scales))/10))
plt.ylabel('chastota[Hz]', fontsize=12)
plt.ylabel('Fourier spectrum
    ', fontsize=12)
plt.grid()
plt.legend()
plt.show()
```

代码 7 快速傅里叶变换

Number of points in Fourier transform range: 161280 Maximum frequency 127.99960317337293

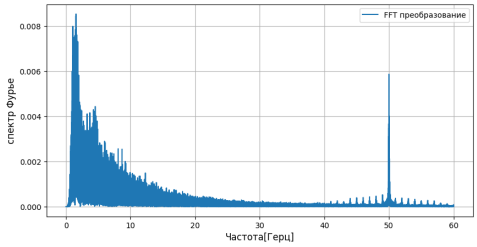

图 5 傅里叶变换

5.5 小波变换

```
wavlist_continuous = pywt.wavelist(kind=
    'continuous')
wavlist_discrete = pywt.wavelist(kind='
    discrete')
```

代码 8 小波函数列表

绘制频率与尺度的关系图。

```
scale_max = 300
scale_min = 3
scales = np.linspace(scale_min,
    scale_max, num = 25, endpoint=True)

wavelet_core = 'morl'
fs = len(t)/(T)
dt = 1/ fs
coef, freqs = pywt.cwt(y, scales,
    wavelet_core, sampling_period = dt)

plt.figure(figsize=(7, 7))
plt.grid()
plt.yticks(np.arange(min(freqs),
    max(freqs), (max(freqs) -
    min(freqs))/10))
plt.xticks(np.arange(min(scales),
    max(scales), (max(scales) -
    min(scales))/10))
plt.ylabel('Frequency [Hz]
    ', fontsize=12)
plt.xlabel('Scale in wavelet transform
    ', fontsize=12)
plt.plot(scales, freqs,'.-')
```

代码 9 频率与尺度的关系图

```
15 plt.xlabel('masshtab v veivlet
       preobrazovanii', fontsize=12)
16 plt.plot(scales, freqs,'.-')
```

Листинг 9 Вейвлет-преобразование

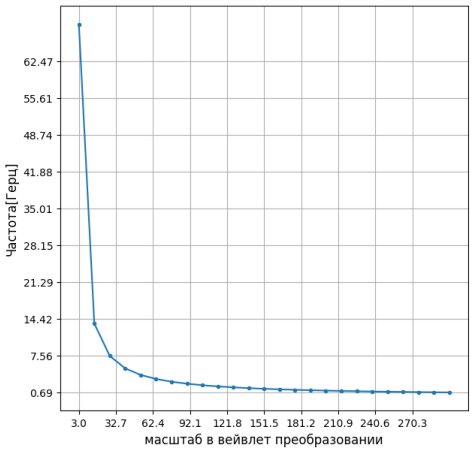

Рис. 6 График зависимости частоты и масштаба

Построим скейлограмму.

```
1 plt.figure(figsize=(10, 5))
2 plt.imshow(abs(coef), extent=[t[0], t
      [-1], max(scales),
      min(scales)], interpolation='
      bilinear', cmap='plasma', aspect='
      auto')
3 #plt.imshow(abs(coef), interpolation='
      bilinear', cmap='plasma', aspect='
      auto')
4 plt.gca().invert_yaxis ()
5 plt.yticks(np.arange(min(scales),
      max(scales), (max(scales) -
      min(scales))/10))
6 plt.ylabel('masshtab ', fontsize=12)
7 plt.xlabel('vremya ', fontsize=12)
8 plt.show()
```

Листинг 10 Скейлограмма

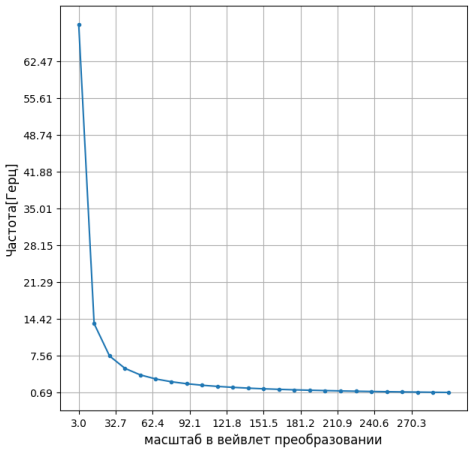

Рис. 6 频率与尺度关系图

绘制尺度图。

```
1 plt.figure(figsize=(10, 5))
2 plt.imshow(abs(coef), extent=[t[0], t
      [-1], max(scales),
      min(scales)], interpolation='
      bilinear', cmap='plasma', aspect='
      auto')
3 plt.gca().invert_yaxis ()
4 plt.yticks(np.arange(min(scales),
      max(scales), (max(scales) -
      min(scales))/10))
5 plt.ylabel('Scale', fontsize=12)
6 plt.xlabel('Time', fontsize=12)
7 plt.show()
```

代码 10 尺度图

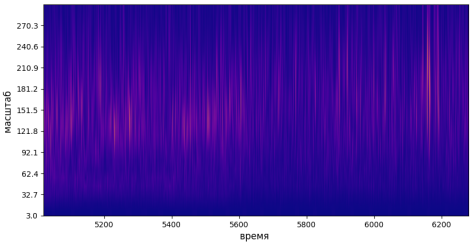

图 7 尺度图

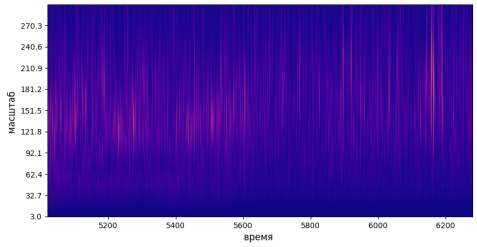

Рис. 7 Скейлограмма

6 Заключение

В ходе выполнения лабораторной работы были исследованы скрытые характеристики сигналов ЭЭГ на основе данных, содержащих приступы. Была загружена запись в формате EDF и определён интервал приступа по аннотациям из датасета. Для наглядного анализа построен график ЭЭГ во временной области, выполнено усреднение сигналов по каналам и применена фильтрация для удаления высокочастотных компонент (выше 60 Гц).

Основное внимание было уделено анализу преобразованного сигнала с помощью спектральных методов. Построены спектрограмма и скейлограмма, что позволило визуализировать изменение частотных характеристик сигнала во времени. В процессе работы был освоен навык применения спектрального и вейвлет-преобразования.

6 结论

在本次实验工作中，我们研究了包含癫痫发作的EEG数据的隐藏特征。加载了EDF格式的记录，并根据数据集中的标注确定了癫痫发作的时间间隔。为了直观分析，绘制了时域EEG图，执行了通道信号平均，并应用了滤波去除60Hz以上的高频分量。

主要关注点是通过频谱方法分析转换后的信号。构建了频谱图和尺度图，这使我们能够可视化信号频率特性随时间的变化。在工作过程中，掌握了应用频谱和小波变换的技能，这些技能也能在考虑的EEG信号数据上识别癫痫发作。

References

[1] Shoeb A. *Application of Machine Learning to Epileptic Seizure Onset Detection and Treatment*. PhD Thesis, Massachusetts Institute of Technology. (2009)

[2] Truong N. D., Nguyen A. D., Kuhlmann L., Bonyadi M. R., Yang J., Ippolito S., Kavehei O. *Convolutional Neural Networks for Seizure Prediction Using EEG Signals*. IEEE Transactions on Biomedical Engineering, 65(9): 2108–2118. (2018)

[3] Acharya U. R., Oh S. L., Hagiwara Y., Tan J. H., Adeli H. *Deep Convolutional Neural Network for the Automated Detection and Diagnosis of Seizure Using EEG Signals*. Computers in Biology and Medicine, 100: 270–278. (2018)