Давид Христов – 221085
Огнен Трајковски – 221199

ФАКУЛТЕТ ЗА ИНФОРМАТИЧКИ НАУКИ

И КОМПЈУТЕРСКО ИНЖЕНЕРСТВО

Предмет: Вовед во наука на податоци

Наслов на проектот: Predicting Epileptic Seizures through EEG Signal Processing

Студенти: Давид Христов**,** Огнен Трајковски

Индекси: 221085, 221199

Професор: Димитар Трајанов

Датум: 20.07.2025

1.Домен на проектот и опис на проблемот

EEG, односно **електроенцефалографија** е неврофизиолошка дијагностичка метода со која се испитува мозочната активност преку добивање на сигнали од мозокот, лекарот утврдува дали мозочната активност е нормална или можеби има одредени нарушувања.

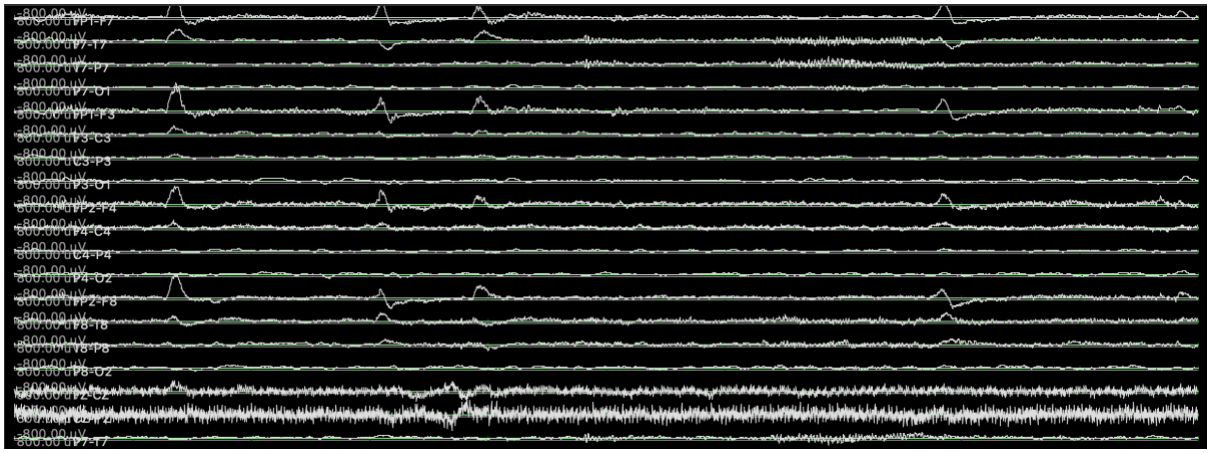
Овие сигнали се добиваат преку поставување на електроди на површината на главата. Сигналите добиени преку EEG ја рефлектираат невронската активност и се користат во широк спектар на медицински и когнитивни истражувања и дијагнози, како што се:

- Дијагноза и следење на **епилепсија**
- Анализа на **нарушувања на спиењето**
- Проценка по **мозочен удар**
- Истражувања на **ментално здравје и когнитивни состојби**

Давид Христов – 221085
Огнен Трајковски – 221199

EEG сигналите се електрични шеми генерирани од невроните на мозокот кои комуницираат едни со други. Невроните комуницираат преку електрични импулси, а кога многу неврони се активираат заедно, нивната комбинирана електрична активност може да се открие со **EEG**.

Електричната активност е претставена како брановидни линии со различни фреквенции и амплитуди што одговараат на различни состојби на мозокот.



EEG сигналите се **шумливи, нелинеарни, временски зависни и полни со артефакти**, што претставува голем предизвик при нивната обработка и анализа. Тие се со ниска амплитуда (обично во μV) кои се појавуваат како бранови со различни фреквенции и амплитуди

Видови бранови според фреквенција:

Име на бран	Фреквенција	Што претставува
Delta (δ)	0.5 – 4 Hz	Длабок сон, несвесна состојба.
Theta (θ)	4 – 8 Hz	Лесен сон, длабока релаксација, медитација.
Alpha (α)	8 – 13 Hz	Смирена будност, затворени очи, без ментален напор.
Beta (β)	13 – 30 Hz	Активно размислување, концентрација, алармна состојба.
Gamma (γ)	30 – 100+ Hz	Висока когнитивна активност, свесност, обработка на сетила.

Во случај на епилепсија, најважно е да се следат **spikes и sharp waves** (остри врвови), **spike-and-wave комплекси** кој се типичен шаблон кај генерализирани напади.

Давид Христов – 221085
Огнен Трајковски – 221199

2. Прибирање на податоци (Data Acqusition)

Користена е јавна база на податоци како што е

- CHB-MIT Scalp EEG Database

    - Собрана од **Children's Hospital Boston**, од страна на MIT, се состои од **EEG** снимки од испитаници со нерешливи напади.

    - Снимките се групирани во 23 случаи и се собрани од 22 испитаници (5 мажи, на возраст од 3 до 22 години; и 17 жени, на возраст од 1,5 до 19 години)

    - Секој случај (chb01, chb02, итн.) содржи помеѓу 9 и 42 континуирани .edf датотеки од еден испитаник, каде што има и **chb_summary** каде што кажува во која снимка, во кој период пациентот имал напад.

    - Секоја сесија содржи и анотации за почеток и крај на епилептичен напад.

    - Сигналите се снимани со 256hz во секунда со 16-битна резолуција

Значење на сигналот

Секој сигнал е разлика во напонот (во микроволти, μV) со текот на времето, обично семплирана на 256 Hz во овој збир на податоци

Всушност, сигналот е разликата во напонот помеѓу електродите на FP1 (фронтополарна лева страна) и F7 (фронтална лева страна). Сите абнормални скокови овде укажуваат на активност во тој регион на мозокот.

EEG канали

Секој .edf фајл се состои од **повеќе канали.**

Секој канал го покажува напонот помеѓу 2 електроди (Fp1-F7, F7-T7)

EEG канали: Fp1-F7, F7-T7, T7-P7, и слично – се електродни парови од 10–20 EEG системот.

Вредности: **μV (микроволти)** – електричен потенцијал меѓу двата електроди.

Поделба на EEG канали според регионите на мозокот

FP1-F7, F7-T7, T7-P7, P7-O1 (лева хемисфера) FP1-F3, F3-C3, C3-P3, P3-O1 (средно лева) FP2-F4, F4-C4, C4-P4, P4-O2 (средно десна) FP2-F8, F8-T8, T8-P8, P8-O2 (десна хемисфера)

Отприлика 23 канали, понекогаш 24-26

Снимање на податоците

Снимање на податоците значи процесот на регистрирање и зачувување на EEG сигналите од електродите на главата на пациентот.
Овие снимани податоци се користат за анализа, обработка и детекција на мозочни активности.

Се снима со 256 Hz → 256 примероци во секунда, поточно 256 мерења во секунда по канал).

23+ канали (електроди).

Времетраење: ~1 час по фајл (~3600 × 256 ≈ 921,600 редови)

Значење на карактеристиката

Во обработката на биомедицински сигнали (EEG, ECG, EMG), **карактеристика** е мерка што ја извлекуваме од сигналот за да добиеме корисна информација.

Тоа е како да ја „сумираме“ или „опишеме“ активноста во дел од сигналот со бројки што ја претставуваат неговата форма, енергија, сложеност, фреквенција и слично.

Извлекуваме карактеристики со цел да се намали димензијата на податоците, да се истакне некоја битна информација и да се олесни класификацијата и анализата.

Карактеристика	Значење
Базна линија (Baseline)	Рамна или конзистентна линија без големи осцилации – типична за **опуштена состојба на мозокот**, особено кога лицето мирува или спие.
Мали осцилации	Нормални мозочни ритмови, како што се **алфа (8–12 Hz)**, **бета (13–30 Hz)**, **тета (4–7 Hz)** и **делта (0.5–4 Hz)** бранови. Се јавуваат во различни состојби – будност, концентрација, длабок сон и слично.
Остри спајкови или изблици	Брзи и високи скокови во сигналот. Можат да бидат индикатор за **почеток на епилептичен напад** или друга **абнормална мозочна активност**.
Ритмички, повторливи бранови	Постои **регуларен, синхронизиран шаблон** во сигналот. Најчесто се гледа при **епилептични напади**, каде што одреден дел од мозокот испраќа координирани сигнали.
Ненадејни скокови или падови	Изненадни промени во сигналот. Може да се јават поради шум, артефакти (движење на телото, мускулна активност)

EDF Format

Самата база на податоци е во формат **.edf (European Data Format)** кој е стандардизиран формат за складирање на биомедицински сигнали, како што се:

- EEG (електроенцефалограм)
- EMG (електромиограм)
- ECG (електрокардиограм)

Структура на .edf фајл

1. *Заглавие (header)*
   ASCII текст со фиксна должина од 256 бајти: содржат основни информации:
   - Име на пациентот
   - Датум и време на снимање
   - Број на сигнали (електроди)
   - Времетраење на секој блок од податоци
   - **Канал по канал**: име, единица (μV), минимум/максимум вредности (23-26 канали)

2. *Податоци (signal data)*
    - бинарни податоци што складираат примероци од секој канал
    - Се снимаат податоци за **секој канал** (на пр. FP1-F7)
    - Се организирани по **временски блокови** (на пр: секоја секунда)
    - Податоците се **дигитални вредности** (обично цели броеви), кои можат да се претворат во микроволти

Below is the header record of a 24h recording of EEG and Rectal temperature sampled at 500Hz and 0.1Hz, respectively. The recording starts at September 16, 1987 at 20:35hr and ends 1440 minutes (2880 x 30s) later. Note that the offsets of EEG Fpz-Cz and Rectal temperature are 35uV and 37.3 degC (degrees centigrade), respectively while the gains are 4.31/uV and 706.2/degC, respectively. Each 30s data record contains 15000 samples of the EEG followed by 3 samples of the Rectal temperature signal. All fields are not only EDF but also EDF+ compatible, except the 'Reserved field of 44 characters' which is EDF but not EDF+ compatible.

0	MCH-0234567 F 16-SEP-1987 Haagse_Harry							Startdate 16-SEP-1987 PSG-1234/1987 NN Telemetry03								16.09.8720.35.00768	Reserved
field of 44 characters		2880	30	2	EEG Fpz-Cz		Temp rectal	AgAgCl cup electrodes						Rectal thermistor			
			uV		degC	-440	34.4	510	40.2	-2048	-2048	2047	2047	HP:0.1Hz LP:75Hz N:50Hz			LP:0.1Hz
(first order)					15000	3		Reserved for EEG signal						Reserved for Body temperature			

---

3. Препроцесирање на податоци за ML модели

Чистење на сигналите од шумови и непотребни компоненти.

- **Филтрирање**:
    - Примена на **Bandpass Filter (0.5 – 40 Hz)** за отстранување на DC offset и високофреквентни шумови.
- **Артефакт редукција**:
    - Отстранување на артефакти од трепкање, движење на очи, мускулна активност
- **Нормализација**
    - Сигналите се нормализираат на нула средна вредност и единечна стандардна девијација.
- **Сегментација**:
    - EEG сигналите се делат на **движечки прозорци (sliding windows) од фиксна должина**,
        - на пр. 10 секунди или 1 секунда со пречекорување (overlap).
    - Секој прозорец се обележува како:
        - **Preictal** – пред напад (часови или минути)
        - **Ictal** – време на напад
        - **Interictal** – помеѓу напади

Давид Христов – 221085
Огнен Трајковски – 221199

Signal Processor

Оваа класа ќе ја користиме за обработка на самите EEG сигнали

Дефинираме:

- **sampling_rate**: → е фреквенцијата на семплирање на самиот сигнал, односно колку пати во секунда се земаат вредности од EEG сигналот.
    - **Типична фреквенција на овие сигнали, како што споменавме погоре е 256hz , што значи дека за секоја секунда имаме 256 вредности**

- **листа на фреквентни рабови (frequency band edges)** → овие рабови ги дефинираат интервалите на EEG бранови

  Фреквентни појаси (frequency bands)

1-5 Hz (δ - delta)
5-10 Hz (θ - theta)
10-15 Hz (α - alpha)
15-20 Hz (β - low beta)
20-25 Hz (β - high beta)

  Во нашиот случај ги земаме како листа од самите **рабови (edges)** бидејќи оваа структура ни овозможува **лесно и автоматизирано** креирање на фреквентни рабови (**frequent band edges)**

  Пр. **[1,5,10,15,20,25]**

  Со овие рабови го делиме спектрумот на фреквенцијата на мали интервали

  Овие опсези одговараат на **различни типови на мозочни активности**, кои имаат клиничко значење, како што споменавме погоре

Давид Христов – 221085
Огнен Трајковски – 221199

3.1. Feature Extraction

Имаме **3 функции** за екстракција на карактеристики од EEG сигналите.

Овие функции автоматски извлекуваат

- **temporal features** → темпорални / временски статистички (во време),
- **spectral features** → спектрални (во фреквенција) и
- **non-linear features** → нелинеарни (комплексност)

карактеристики од EEG сигналите што е од клучно значење за ефикасна анализа и предикција на епилептични напади.

Temporal features (темпорални / временски) карактеристики

Карактеристика	Објаснување
mean	Просек на амплитудата на сигналот (дали сигналот е наклонет во одредена насока)
variance	Колку сигналот варира (нестабилност), колку вредностите се распространети од просекот
skewness	Асиметрија на сигналот (дали е повеќе над/под средината)
kurtosis	Острината на пиковите во сигналот
rms	Root Mean Square – мерка на енергијата на сигналот (signal power)
zero_crossings	Број на премини преку нулата (индикатор за фреквенција)
peak_amp	Највисока апсолутна вредност на сигналот
peak_count	Број на локални максимуми (пикови)

Овој вид на карактеристики ни е потребен бидејќи го опишуваат основното однесување на сигналот, всушност EEG сигналите се **временски серии** и тие варираат со текот на времето што значи дека **временските статистики** ни даваат основен увид во тоа како сигналот се однесува во еден сегмент.

Пр. Ако имаме сегмент од EEG сигнал и ја гледаме неговата RMS (Root Mean Square)

- **Нормален период**: RMS = 15
- **Pre-ictal период (пред напад)**: RMS = 28

Оваа разлика во вредности е доволна за **класификаторот да ја научи границата** и да го предвиди нападот.

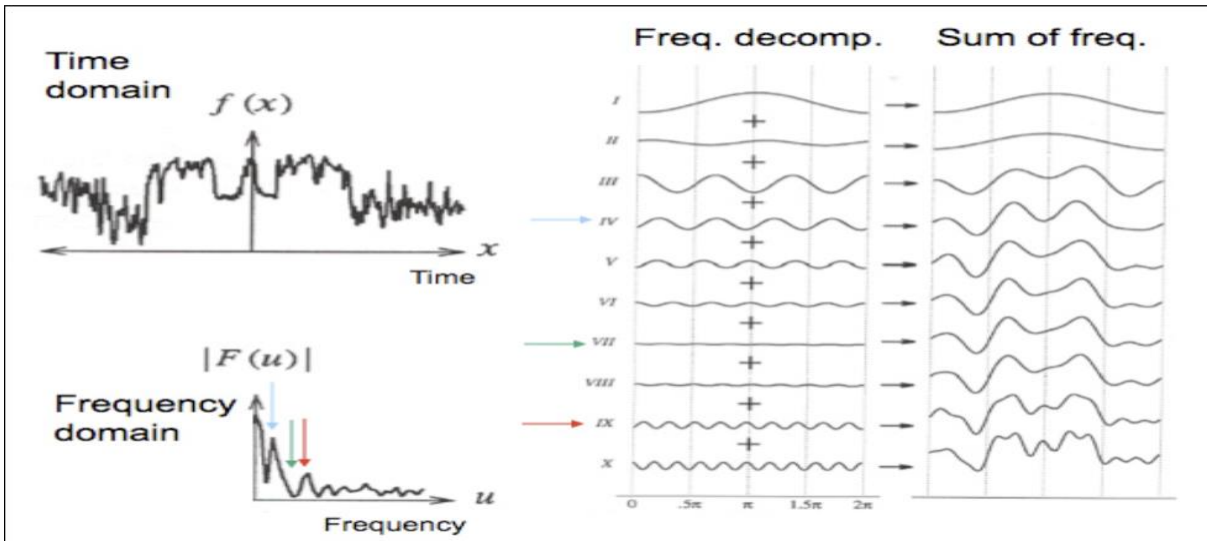
Spectral features (спектрални) карактеристики

Овој вид на карактеристики ни служи за вадење на **фреквенциски карактеристики** од ЕЕГ сигналот.

Фреквенциските карактеристики ни помагаат да го **разбереме како се распределува енергијата на сигналот** низ различни фреквенциски области (фреквентни појаси).

EEG сигналите се **нестационарни** – нивната фреквентна содржина се менува со текот на времето.

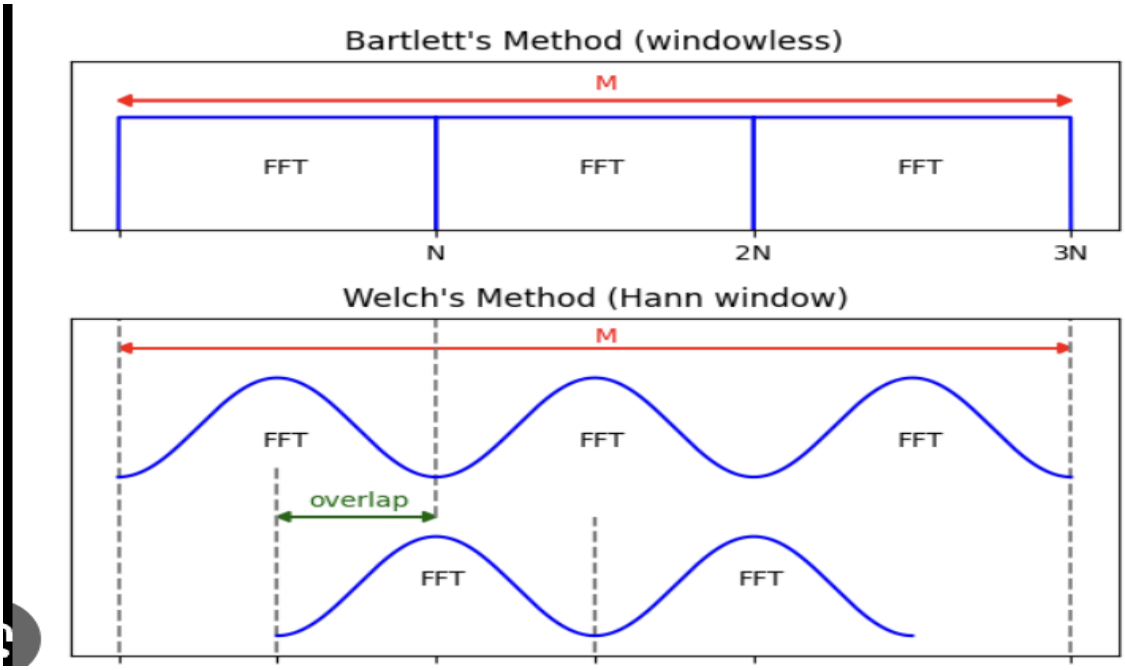
Во случај на епилептични напади, често има **нагли промени во одредени фреквентни појаси** (на пример, **ненадеен раст на енергија во делта или тета).**



Давид Христов – 221085
Огнен Трајковски – 221199

Welch's Method

Користиме **Welch-ов метод** за да добиеме **Power Spectral Density (PSD)** — т.е. колкава е енергијата на сигналот на секоја фреквенција.



*Оваа слика прикажува споредба помеѓу **Bartlett-овиот метод** и **Welch-овиот метод** за проценка на **спектралната густина на моќноста (Power Spectral Density – PSD)**.*
Овие методи се користат за да се анализира како енергијата (или моќта) на сигналот се распределува низ фреквенциите.

Bartlett's Method (без прозорец)

- ***Без преклопување**: Сигналот е поделен на **несекојдневни (непреклопувачки)** сегменти.*
- *Секој сегмент има должина **M**, а тие се поставени еден до друг (на растојание **N**, 2N...).*
- *Секој сегмент се обработува со **FFT (Fast Fourier Transform)**.*
- *Резултатите од сите сегменти се **средуваат (averaged)** за да се добие PSD.*
- *Нема применето прозорец (window function), што може да доведе до спектрален „leakage“.*

Давид Христов – 221085
Огнен Трајковски – 221199

Welch's Method (со Hann прозорец)

- ***Со преклопување**: Сигналот повторно е поделен на сегменти, но тука има **преклопување (overlap)** меѓу сегментите (обично 50%).*
- *На секој сегмент се применува **прозоречна функција (Hann window)** пред да се направи FFT.*
- *Hann прозорецот го **намалува влијанието на краевите** од секој сегмент и го **намалува спектралниот leakage**.*
- *Потоа се прави **FFT на секој сегмент** и се **средуваат резултатите**, како кај Bartlett.*

---

*Welch-ов метод е техника која се користи за процена на **Power Spectral Density (PSD)** – односно како е распределена моќта (енергијата) на сигналот низ различни фреквенции.*

*PSD покажува колку **енергија (моќ)** има сигналот во различни фреквентни компоненти.*

Наместо да гледаме сигнал во временски домен (time series), со **PSD гледаме колку фреквенции се присутни и колку се "силни" тие.**

EEG сигналите се **шумовити и нестабилни**, па класична ***Fourier трансформација*** може да даде непостојани резултати, и затоа ќе користиме **Welch-ов метод** кој ја подобрува прецизноста и стабилноста на PSD со следниве чекори:

1. **Сегментирање на сигналот:**

- Сигналот се дели на overlapping windows (50% overlap).
- Секој прозорец содржи дел од сигналот
  - 256 точки = 1 секунда на 256 Hz.

2. **Windowing:**

- Секој сегмент се множи со window function (на пример Hamming window) за да се намалат рабните ефекти.

3. **Fourier трансформација:**

- Се прави **FFT (Fast Fourier Transform)** на секој сегмент.
    - **FFT (Fast Fourier Transform)** е брз алгоритам што се користи за пресметување на **Дискретната Фуриева Трансформација (DFT)** на сигнал.
    - *Таа ја претвора временската серија (time series), EEG сигналот во фреквентен домен, т.е. ни покажува кои фреквенции постојат во сигналот и со каква јачина (енергија).*
    - Ни *овозможува да разбереме кои фреквенции се присутни во еден сигнал и со која енергија (амплитуда)*

- EEG сигналите се временски сигнали кои содржат многу информации скриени во различни фреквентни појаси и со примена на **Фуриева трансформација** можеме да ги издвоиме фреквентните компоненти на мозочната активност.

4. **Се пресметува Power Spectral Density за секој сегмент.**

- Го делиме сигналот во фреквентни појаси (delta, theta, alpha…).
- Мериме колкава енергија има во секој појас (како "band power").
- Го карактеризираме мозочниот статус (сон, внимание, напади…).

5. **Просек**

- PSD од сите сегменти прави просек (average) за да се добие стабилна проценка на целокупниот PSD.

EEG сигналот (на пример, 10 секунди) е предолг за да се анализира како една целина, и може да има различни карактеристики во различни делови од времето. Затоа, сигналот се сече на **помали сегменти (прозорци) и користиме техника на движечки прозорец (sliding window)**

Имплементиравме метода за екстракција **на спектралните карактеристики** каде што

- на влез се прима 1D EEG сигнал (signal), т.е. вредности од една EEG електрода.
- Се користи **welch()** функцијата за да се добие **Power Spectral Density (PSD)** → покажува колку енергија има во секоја фреквенција.

- Прави **вкупна енергија (total_power)** → сума на сите PSD вредности, **медијална фреквенција (median_freq)** →фреквенција под која е 50% од енергијата и **најсилна фреквенција (peak_freq)** → фреквенцијата со најголема енергија.
- **Bin power** → пресметува колку енергија има во секој фреквентен појас (на пр. алфа, бета...).

Спектрална ентропија → мерач на хаотичност во сигналот.

Non-linear features (нелинеарни) карактеристики

Овој вид на **нелинеарни (complexity-based) карактеристики** од EEG сигнал, се карактеристики кои не можат да се опишат со едноставни статистички или фреквентни мерки, туку со **математички индикатори за хаотичност, фрактална структура и самосличност.** Овие мерки даваат длабоко разбирање на динамичноста и сложеноста на мозочната активност.

Методот за екстракција на нелинеарни карактеристики ги прима **сигналите (time series податок)** како влезни податоци кои се **2D NumPy низа** каде првата димензија се каналите, а втората се временските точки

Извлекуваме повеќе **видови на нелинеарни карактеристики**

1. **Hjorth параметри** се 3 три статистички мерки развиени од Бо Hjorth за квантификување на карактеристиките на EEG сигнали, *поточно ја опишува активноста на сигналот и комплексноста*

    - **Активност** ја мери варијансата на сигналот (не го користиме овој параметар)
        - Претставува “моќност” на сигналот

    - **Комплексност** мери колку сигналот се разликува од чиста синусоида
        - 1.0 = чиста синусоида
        - > 1.0 = покомплексен од синусоида

- **Повисоки вредности** = повеќе фреквенциски компоненти

- **Мобилност (подвижност)** ја мери средната фреквенција (просекот) на сигналот , поточно мери колку “брзо” се менува сигналот
  - **Повисоки вредности** = побрзи промени, повисоки фреквенции
  - **Пониски вредности** = побавни промени, пониски фреквенции

2. **Fractal dimension (фрактална димензија)** ја мери "грубоста" или комплексноста на формата. Фракталите се геометриски објекти што покажуваат самосличност на различни скали.

   - ***hfd (Higuchi Fractal Dimension)****:* ја мери фракталната димензија преку пресметување на должини на криви на различни скали, *поточно ја мери фракталната сложеност на сигналот.*

     1. Сигналот се дели на сегменти со различни чекори k (1 до Kmax)
     2. За секој k се пресметува должината на кривата
     3. Фракталната димензија се добива од наклонот во log-log координати

     Поголеми вредности = посложен, понеуреден сигнал.

   - ***pfd (Petrosian Fractal Dimension)***: ја мери комплексноста преку броење на промени во правецот на сигналот

     1. Го бинаризира сигналот (1 ако се зголемува, 0 ако се намалува)
     2. Брои колку пати се менува правецот
     3. Нормализира спрема должината на сигналот

     Повисоки вредности = повеќе промени во правец

3. ***Hurst експонент: (Хурст експонентот):*** мерка за долгорочна корелација, *поточно мери долгорочна меморија и самосличност на сигналот*

    1. Ги пресметува кумулативните отстапувања од средната вредност
    2. Ја пресметува разликата помеѓу мин/макс вредности (Range)
    3. Ја нормализира со стандардна девијација (Rescaled)
    4. Наклонот во log-log е Hurst експонентот

    H = 0.5: случаен процес (бел шум)
    H > 0.5: постојан тренд (позитивна корелација)
    H < 0.5: анти-постојан тренд (негативна корелација)

Обработка по канали:
За секој канал посебно:
- Пресметуваме Hjorth параметри
- Пресметуваме фрактални димензии (со намален Kmax=3 за побрзина и ефикасност)

Обработка на грешки: Ако било која пресметка не успее, се користат стандардни вредности:
- Hjorth параметри: 0
- HFD/PFD: 1.0 (неутрална фрактална димензија)
- Hurst: 0.5 (случаен процес)

---

4. Вчитување на податоците и филтрирање

Ги вчитуваме податоците од EDF фајл и применуваме филтрирање на сигналот. Правиме 2 клучни операции

Давид Христов – 221085
Огнен Трајковски – 221199

1.Вчитување на податоци

- Ги читаме податоците преку **read_raw_edf** функцијата , каде ставаме параметар **preload=True** за податоците да се вчитуваат директно во меморија која го зголемува перфомансот и ефикасноста.

2.*Филтрирање на сигналот*

-Применуваме **bandpass filter** со прекин на пониска фреквенција наместена на 0.25hz и прекин на повисока фреквенција наместена на 25hz
-**Ова филтрирање ги отстранува многу бавните отстапувања (под 0,25 Hz) и високофреквентниот шум (над 25 Hz),** задржувајќи го само фреквентниот опсег што е најрелевантен за ЕЕГ анализата.

3.*Процесирање на снимката*

- Ја земаме целата снимка и ја делиме на помали временски сегменти (epochs) и извлекуваме карактеристики од секој сегмент

4.Пресметување на епохи

-Ја делиме целата снимка на **преклопувачки временски прозорци (sliding windows)**

- **epoch_length**: должина на секој сегмент (нпр. 4 секунди)
- **step_size**: чекор помеѓу сегменти (нпр. 2 секунди)

EEG снимка: [0————————————————————————————60] секунди

epoch_length = 4s,
step_size = 2s

Epochs:
[0——4s]
  [2——6s]
    [4——8s]
      [6——10s]
        ... до крај

Давид Христов – 221085
Огнен Трајковски – 221199

5.Batch обработка за меморија
- Ако имаме 10,000 epochs, не сакаме да ги држиме сите во меморијата
- Обработуваме 100 epochs одеднаш, потоа ги ослободуваме од меморија

6.Извлекување на карактеристики за секоја епоха
- За секоја епоха добиваме

data.shape = (19, 1000) → 19 канали × 1000 samples (4s × 250Hz)

7.Feature extraction (клучен дел)

```
# Extract features for all channels at once
features = {"start_time": start_time}
channel_features = self.processor.extract_all_features_batch(data, raw.ch_names)
features.update(channel_features)
```

Резултат:

features = {
"start_time": 0.0,
"Fp1_mean": 12.5,
"Fp1_std": 8.2,
"Fp1_total_power": 145.7,
"Fp1_peak_freq": 10.2,
"Fp1_hfd": 1.4,
"Fp2_mean": -5.1,
}. → **за сите 19 канали × ~15 карактеристики = ~285 features**

Давид Христов – 221085
Огнен Трајковски – 221199

8. Лабелирање на seizure колоната (target колона)

```
# Seizure label
if seizure_intervals:
    features["seizure"] = any(
        start_time < end and start_time + self.epoch_length > start
        for start, end in seizure_intervals
    )
else:
    features["seizure"] = 0
```

Напади од 120-180s и 300-350s
seizure_intervals = [(120.0, 180.0), (300.0, 350.0)]

За epoch од 115s до 119s:
seizure = False → не се преклопува

За epoch од 118s до 122s:
seizure = True → се преклопува со напад 120-180s

За epoch од 175s до 179s:
seizure = True → се преклопува со напад 120-180s

Користиме и хардкодиран речник каде што има информации за секој напад за секој запис во даден временски интервал

Ова се користи за означување на податоците (seizure = 1, нормално = 0)

```
seizure_info = {
    "chb01_03": [[2996, 3036]], "chb01_04": [[1467, 1494]], "chb01_15": [[1732, 1772]],
    "chb01_16": [[1015, 1066]], "chb01_18": [[1720, 1810]], "chb01_21": [[327, 420]],
    "chb01_26": [[1862, 1963]], "chb02_16": [[130, 212]], "chb02_16+": [[2972, 3053]],
    "chb02_19": [[3369, 3378]], "chb03_01": [[362, 414]], "chb03_02": [[731, 796]],
    "chb03_03": [[432, 501]], "chb03_04": [[2162, 2214]], "chb03_34": [[1982, 2029]],
    "chb03_35": [[2592, 2656]], "chb03_36": [[1725, 1778]]
}
```

5. Спојување на сите csv фајлови

Направивме скрипта за спојување на сите процесирани csv фајлови во еден **subjects.csv** фајл кој ќе ни е потребен за тренирање и евалуација на моделите

Овој фајл се состои од:

- сите екстрактнати карактеристики (темпорални, спектрални и нелинеарни)
- “seizure” колона (1 или 0) која кажува дали епохата има активност за напад
- “subject” колона која идентификува од која снимка , дадена редица дошла
- “start_time” колона која кажува кога дадена епоха започнала (временски интервал)

Скриптата има и **перфомациски оптимизации** како:

- Multithreading за I/O операции
- Memory usage следење
- Ефикасно читање / пишување параметри од/во CSV
- Progress monitoring

6. Тренирање на моделите

Скриптата **classifier.py** е организирана околу главната класа **SeizureClassifier** која имплементира комплетен pipeline за класификација со неколку различни алгоритми.

1. Дефиниција на модели

Во скриптата дефинираме 5 различни машински алгоритми:

- **MLP (Multi-Layer Perceptron)** - невронска мрежа
- **SVM (Support Vector Machine)** - со RBF kernel
- **Random Forest** – "ensemble" од дрва за одлучување
- **AdaBoost** - адаптивен boosting алгоритам
- **KNN (K-Nearest Neighbors)** - класификација врз база на најблиски соседи

2.Подготовка на податоци

- Првично ги читаме CSV датотека со EEG податоци
- Ги одделуваме **features од target променливата ('seizure')** и делиме на training и тест множество
- Отстрануваме колоните со **мала варијанса** (речиси константни) бидејќи вредностите се речиси исти, не можат да се разликуваат меѓу класите или дека постои артефакт во обработката
    - **Повеќе features = поголема сложеност**
    - **Бескорисни features** го влошуваат учењето
    - Не помагаат во класификацијата
        - feature_1 = [1.001, 1.002, 1.001, 1.000, 1.001] → Речиси константна
        - feature_2 = [10, 50, 80, 30, 90] → Добра варијанса
- Стандардизираме податоците со **StandardScaler**
    - Градиентот се пропагира подобро со стандардизирани податоци
    - Избегнува проблеми со **vanishing/exploding gradients**
    - Активациските функции (ReLU) работат подобро

3.Евалуација на модели

Скриптата користи напредна евалуација:

- 5-fold Cross Validation со **StratifiedKFold** за балансирани подели
- Пресметува комплетни метрики: accuracy, sensitivity (TPR), specificity (TNR), precision, F1-score
- Мери време на тренирање

4. Метрики за евалуација

За секој модел се пресметуваат:

- **Accuracy** - вкупна точност
- **TPR (True Positive Rate)** - чувствителност, колку добро детектира напади
- **TNR (True Negative Rate)** - специфичност, колку добро избегнува лажни аларми
- **Precision** - точност на позитивни предвидувања
- **F1-score** - хармониска средина на precision и recall

5. Зачувување резултати

Резултатите се зачувуваат во:

- Посебни CSV датотеки за секој модел
- Confusion matrices за детална анализа
- Комбинирана датотека со сите резултати

7. Оптимизација

Batch обработка

Batch значи **групно процесирање на повеќе канали одеднаш**, наместо еден по еден.

Во нашиот случај моравме да ги ставаме податоците во **batch** бидејќи **python overhead-от (преоптоварувањето**) нè “уби”

Првично ги процесиравме каналите еден по еден, наместо сите заедно

- **Python function calls**
  - Секоја епоха бара повеќекратни повици на функции, креирање на променливи, алокација на меморија
- **Memory fragmentation**
  - Постојано креирање / уништување на мали објекти
- **Cache**
  - Кешот на CPU се инвалидираше меѓу епохи

После користење **batch procesing**

For each batch (36 times per file instead of 3,590):
1. Load 100 epochs → 0.1s
2. Extract features (vectorized) → 8s
3. Store 100 results → 0.1s
4. Python overhead → 0.8s
Total per batch: ~9s
Per epoch: 9s ÷ 100 = 0.09s per epoch

Ги процесира епохите во batches од 100 наместо една по една епоха

Подобар memory management имаме и намалено преоптоварување (overhead)

Обработка на сите карактеристики во еден **batch**

Архитектура на batch обработката

Влез: signals [n_channels × n_samples]
↓
temporal = extract_temporal_features() → Dict co arrays
spectral = extract_spectral_features() → Dict co arrays
nonlinear = extract_nonlinear_features() → Dict co arrays
↓
Flattening (рамнење) во индивидуални карактеристики

Користиме **batch** обработка бидејќи драстично ја подобрува брзината (2.3x побрзо),организиран е кодот (има сепарација на одговорности), има скалабилност на повеќе канали и има можности за паралелизација

Влезни податоци

```
signals = np.array([
    [EEG_channel_Fp1_data],  # 1000 samples
    [EEG_channel_Fp2_data],  # 1000 samples
    [EEG_channel_C3_data],   # 1000 samples
    [EEG_channel_C4_data]    # 1000 samples
])
channel_names = ["Fp1", "Fp2", "C3", "C4"]
```

Излезни податоци

```
features = {
    "Fp1_mean": 12.5,
    "Fp1_std": 8.2,
    "Fp1_total_power": 145.7,
    "Fp1_peak_freq": 10.2,
    "Fp1_hfd": 1.4,
    "Fp1_hjorth_mobility": 0.8,
    "Fp2_mean": -5.1,
    "Fp2_std": 9.8,
    # ... за сите канали и сите карактеристики
}
```

Векторизирани операции

Векторизацијата е техника каде наместо да обработуваме елементи еден по еден, ги обработуваме сите одеднаш користејќи специјализирани процесорски инструкции.
Векторизацијата значи обработка на цели низи одеднаш, наместо циклус низ поединечни елементи. Таа користи:

- **SIMD инструкции** – Single Instruction Multiple Data (единечна инструкција, повеќе податоци) - процесорот обработува повеќе вредности истовремено
- **Оптимизирани библиотеки C/Fortran под NumPy**
- **Подобро користење на кешот на процесорот** - пристап до континуирана меморија

Процесорот ги процесира сите 23 канали истовремено

Давид Христов – 221085
Огнен Трајковски – 221199

Наместо да користиме циклуси , во голем дел заменивме со **NumPy операции** бидејќи ги прави математичките пресметки побрзо

Го намаливме и **преклопувањето** , поточно сменивме **step_size** да биде од 1 секунда на 5 секунди (помалку епохи за процесирање)

Временски карактеристики

Стар начин (циклично)

```
for ch in channels:
    mean = np.mean(ch)
    std = np.std(ch)
```

Нов начин (векторизирано):

```
means = np.mean(signals, axis=1)  # За сите канали одеднаш
stds = np.std(signals, axis=1)    # Користи SIMD инструкции
```

Спектрални карактеристики

```
# Иако сè уште има циклус, намалена е overhead-от од повикување

for ch_idx in range(n_channels):
    # Welch се повикува еднаш по канал, но другите операции се
векторизирани
```

Паралелна обработка

```python
# Process files in parallel
max_workers = min(cpu_count() - 1, len(edf_files), 4)  # Limit to 4 concurrent files
logger.info(f"Processing {len(edf_files)} files using {max_workers} workers")

with ProcessPoolExecutor(max_workers=max_workers) as executor:
    results = list(tqdm(executor.map(process_single_file, *iterables: file_args),
                        total=len(file_args), desc="Processing files", unit="file"))

for result in results:
    logger.info(result)

elapsed_time = time.time() - start_time
logger.info(f"Total processing time: {elapsed_time:.1f} seconds")
logger.info("✅ All files processed!")
```

- Одредуваме **број на workers**
  - максимум 4, или колку CPU cores имаме
- Подготвуваме аргументи за секој фајл
- Користи **multiprocessing** за паралелна обработка

8. Резултати

All Algorithms

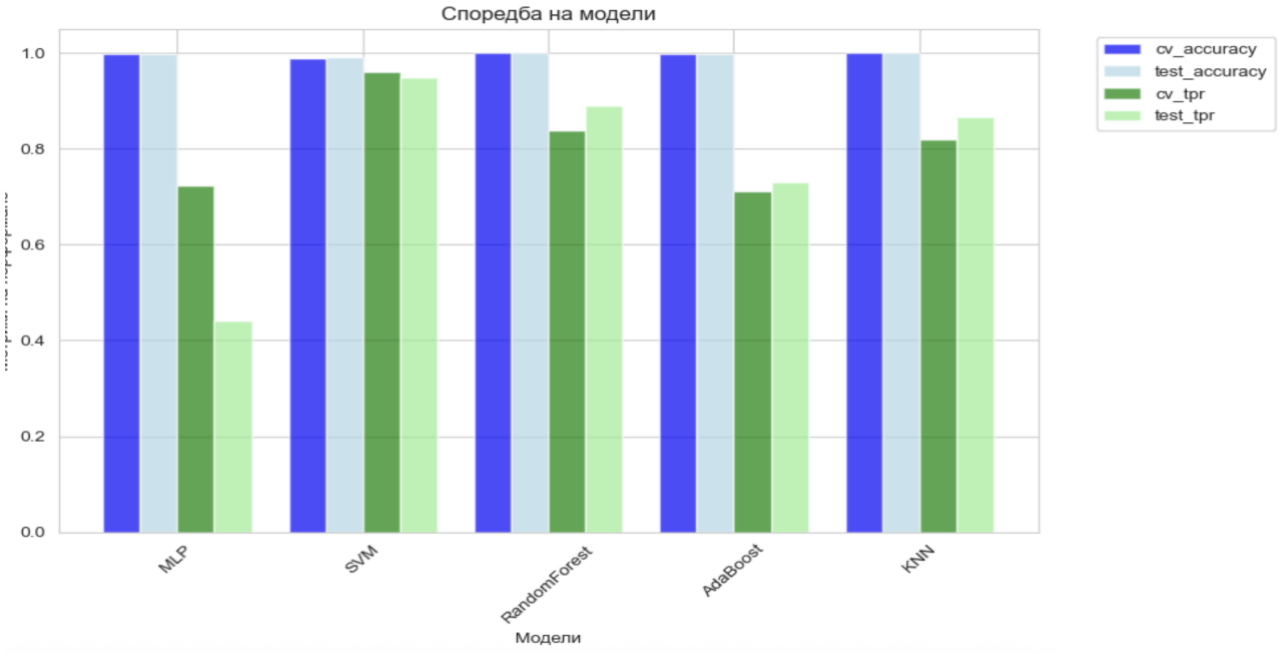
Model	CV Acc	CV TPR	CV FPR	Test Acc	Test TPR	Test FPR	Time (s)
MLP	0.998	0.721	0.0002	0.996	0.441	0.0004	350.5
SVM	0.988	0.959	0.011	0.990	0.948	0.010	37928.3
Random Forest	0.999	0.839	0.00005	0.999	0.889	0.00005	589.3
AdaBoost	0.998	0.712	0.0005	0.998	0.730	0.0005	1397.3
KNN	0.999	0.820	0.00002	0.999	0.867	0.000	175.4

Користиме повеќе видови на пристапи за машинско учење

- невронски мрежи (MLP),
- кернел методи (SVM),
- “ensemble” методи (DecisionTrees),
- Boosting (AdaBoost), и
- instance-based learning (KNN) .

Давид Христов – 221085
Огнен Трајковски – 221199

Перфоманс на моделите



Време на тренирање на моделите