Python i analiza EEG

Poradnik: instalacja, import danych, przetwarzanie wstępne, ICA, ERP, PSD, analiza czasowo-częstotliwościowa, łączność, statystyka i klasyfikacja.

Wprowadzenie do problematyki: Python jako środowisko badań EEG

Niniejszy materiał jest przeglądem możliwości i ograniczeń analizy EEG w języku Python. Przykłady poleceń pełnią funkcję ilustracyjną: pokazują związki między pojęciami, danymi i narzędziami, ale nie stanowią kompletnego protokołu laboratoryjnego. Dobór parametrów zależy od układu elektrod, celu badania, charakterystyki sygnału oraz planu analizy. Dotychczasowe zestawienia narzędzi i fragmenty kodu pozostawiono jako materiał referencyjny.

Python pozwala zestawić w jednym, audytowalnym środowisku import danych z wielu urządzeń, kontrolę jakości, analizę sygnałów, statystykę i modele predykcyjne. Jego przewaga nie polega na istnieniu jednego „najlepszego" algorytmu, lecz na przejrzystych strukturach danych, integracji bibliotek i możliwości odtworzenia analizy. Podstawą ekosystemu EEG jest MNE-Python, ale dla pytań dotyczących snu, sieci funkcjonalnych lub uczenia głębokiego potrzebne mogą być osobne biblioteki [1, 5, 9, 10].

1. Cel i wymagania

Celem jest nauczenie studentów wykonywania powtarzalnej analizy EEG i krytycznej oceny jej wyników. Wiodącą biblioteką jest MNE-Python. Dodatkowe pakiety należy instalować tylko w razie potrzeby.

- **Python**: dla współczesnego MNE zalecany Python 3.11 lub nowszy; przed instalacją należy sprawdzić wymagania konkretnej wersji pakietu [1].
- **System operacyjny**: aktualny Windows, Linux lub macOS. Zalecane co najmniej 16 GB RAM, szybki dysk SSD i zapas przestrzeni na dane oraz ich kopie. To zalecenia dydaktyczne, nie gwarantowane minima producenta.
- **Środowisko**: Miniforge/conda-forge albo venv i pip. Dla uczelni należy rozważyć warunki licencyjne dystrybucji Anaconda [1].
- **Edytor**: JupyterLab lub Visual Studio Code. Każde ćwiczenie powinno zawierać dane, kod, wersje pakietów oraz zapis wyników.

2. Instalacja i kontrola środowiska

Poniższe polecenia wykonuj w terminalu systemowym, a nie w interpreterze Python.

```
conda create -n eeg-lab -c conda-forge python=3.11 mne numpy scipy pandas matplotlib
scikit-learn jupyterlab
conda activate eeg-lab
python -m pip install mne-bids mne-connectivity autoreject pyedflib neurokit2
python -m pip check
```

w środowisku opartym tylko na venv można użyć poleceń `python -m venv .venv`, aktywować środowisko i zainstalować analogiczne pakiety przez pip. Po instalacji zapisz listę zależności:

```
python -m pip freeze > requirements-lock.txt
```

Test uruchomienia (kod Python):

```
import mne, numpy as np
print("MNE:", mne.__version__)
print("NumPy:", np.__version__)
```

3. Najważniejsze biblioteki

Pakiet	Zastosowanie	Uwagi
MNE-Python	Surowe EEG, epoki, ERP, PSD, TFR, ICA, statystyka	Podstawowe narzędzie [1,2]
MNE-BIDS	Import i eksport danych w strukturze BIDS	Wymaga prawidłowych metadanych [3]
MNE-BIDS-Pipeline	Automatyzacja przetwarzania BIDS	Osobna instalacja i konfiguracja .py [4]
MNE-Connectivity	Koherencja, PLV, wPLi i inne miary	Odrębny pakiet, nie mne.connectivity [5]
autoreject	Adaptacyjne odrzucanie/naprawa epok	Wymaga kontroli jakości [6]
pyEDFlib	Niskopoziomowe czytanie/zapis EDF/BDF	Nie zastępuje analiz MNE [7]
NeuroKit2	Sygnały fizjologiczne, dodatkowe analizy EEG	APi zależne od wersji [8]
YASA	Analiza EEG snu, wrzeciona, stadia snu	Nie jest podstawowym pakietem ERP [9]
Braindecode	Uczenie głębokie na EEG	Wymaga właściwego podziału danych [10]
NumPy, SciPy, pandas	Obliczenia i zestawienia tabelaryczne	Zależności naukowe
Matplotlib, seaborn, Plotly	Wykresy statyczne i interaktywne	Nie zastępują kontroli jednostek
scikit-learn, joblib	Uczenie maszynowe i równoległość	Ważna walidacja grupowa
Eelbrain, nitime, napari	Specjalistyczne analizy lub wizualizacje	Opcjonalne, do uzasadnionych zastosowań

FieldTrip jest przede wszystkim środowiskiem MATLAB. Integracji fieldtrip-python nie należy przedstawiać jako pełnego, stabilnego odpowiednika pakietu FieldTrip.

4. Import danych i metadane

Format	Funkcja / pakiet	Kontrola po imporcie
EDF/EDF+	mne.io.read_raw_edf	Jednostki, typy kanałów, adnotacje
BDF	mne.io.read_raw_bdf	Referencja, status/zdarzenia
BrainVision (.vhdr/.vmrk/.eeg)	mne.io.read_raw_brainvision	Wymagany komplet plików
EEGLAB (.set/.fdt)	mne.io.read_raw_eeglab	Dane i położenia elektrod
GDF	mne.io.read_raw_gdf	Mapowanie zdarzeń
BIDS	mne_bids.read_raw_bids	Spójność JSON/TSV i struktury
XDF	pyxdf, potem konwersja do MNE	Synchronizacja strumieni i znaczniki czasu

Przykład importu BrainVision:

```
import mne
raw = mne.io.read_raw_brainvision("sub-01_task-oddball_eeg.vhdr", preload=True)
print(raw.info["sfreq"], raw.ch_names)
print(raw.get_channel_types())
print(raw.annotations)
raw.plot(n_channels=min(20, len(raw.ch_names)))
```

Plik EDF można wczytać przez `mne.io.read_raw_edf("badanie.edf", preload=True)`. Dla BDF stosuje się `read_raw_bdf`, a dla EEGLAB `read_raw_eeglab`. Nazwy wywołań nie są zamienne.

5. Przetwarzanie wstępne

5.1. Inspekcja, filtracja i próbkowanie

Najpierw sprawdź typy kanałów, częstotliwość próbkowania, jakość sygnału, przerwy w zapisie, elektrodę odniesienia i poprawność znaczników zdarzeń. Parametry filtracji dobiera się do pytania badawczego; nie istnieje jeden uniwersalny filtr 0,5–40 Hz.

Dla analiz ERP często potrzebne jest zachowanie wolnych komponentów, dlatego filtr górnoprzepustowy 1 Hz nie powinien być bezrefleksyjnie stosowany do danych końcowych. Do estymacji ICA często przygotowuje się oddzielną kopię danych filtrowaną od 1 Hz, po czym rozwiązanie ICA stosuje się do kopii przeznaczonej do ERP [2].

```
raw_erp = raw.copy().filter(l_freq=0.1, h_freq=30.0)
raw_ica = raw.copy().filter(l_freq=1.0, h_freq=30.0)
```

Filtry FIR mogą być liniowofazowe; `fir_design="firwin"` określa metodę projektowania filtru FIR, a nie filtr nieliniowofazowy. Usuwanie zakłóceń 50 Hz filtrem zaporowym jest uzasadnione po ocenie widma i powinno uwzględniać pasmo zainteresowania. Przed zmniejszeniem częstotliwości próbkowania konieczna jest filtracja antyaliasingowa; `Raw.resample()` wykonuje odpowiednie przetwarzanie, ale należy kontrolować dokładność znaczników zdarzeń.

```
raw_erp.resample(250)  # jeśli pozwala na to pasmo analizowane
```

5.2. Kanały uszkodzone i odniesienie

Kanały o płaskim sygnale, nadmiernym szumie lub niestabilnym kontakcie elektrody oznacz jako złe. Interpolację przeprowadzaj po kontroli rozmieszczenia elektrod i dostępności poprawnych współrzędnych. Wybór odniesienia powinien być uzasadniony przez system rejestracyjny i metodę analizy.

```
raw_erp.info["bads"] = ["Fp1"]  # tylko przykład; kanał musi istnieć
# Interpolacja wymaga poprawnego montage/położeń elektrod
# raw_erp.interpolate_bads(reset_bads=False)
raw_erp.set_eeg_reference("average")
```

5.3. ICA i kontrola artefaktów

ICA może pomóc oddzielić komponenty związane z mruganiem, EOG lub ECG, ale wynik zależy od rzędu danych, liczby kanałów, jakości i długości zapisu. Nie wybieraj liczby komponentów arbitralnie i nie odrzucaj składowych wyłącznie na podstawie ich indeksów. Automatycznie wskazane składowe trzeba zweryfikować topograficznie i czasowo.

```
from mne.preprocessing import ICA

ica = ICA(n_components=0.99, method="fastica", random_state=42, max_iter="auto")
ica.fit(raw_ica, picks="eeg")
```

```
ica.plot_components()
# Przy dostępnym kanale EOG można dodatkowo użyć find_bads_eog().
# Po inspekcji wpisz tylko potwierdzone numery do ica.exclude:
ica.exclude = []
clean_erp = ica.apply(raw_erp.copy())
```

Powyższy przykład celowo pozostawia listę składowych do usunięcia pustą. Jeżeli dane nie zawierają odpowiednich kanałów EOG, automatyczna klasyfikacja komponentów może wymagać odrębnej metody lub ręcznej oceny. AutoReject służy przede wszystkim do oceny i naprawy epok; nie jest prostym zamiennikiem ICA [6].

6. Epokowanie i ERP

Zdarzenia mogą być zapisane w adnotacjach lub kanale stymulacyjnym. Nie należy zakładać istnienia kanału STi 014 dla wszystkich systemów EEG. Przykład dla danych z adnotacjami:

```
events, event_id = mne.events_from_annotations(clean_erp)
print(event_id)  # dopasuj zdarzenia do swojego eksperymentu
epochs = mne.Epochs(clean_erp, events, event_id=event_id,
                    tmin=-0.2, tmax=0.8, baseline=(-0.2, 0),
                    preload=True, reject_by_annotation=True)
print(epochs.drop_log)
evoked = epochs.average()
evoked.plot()
```

Dla eksperymentu Go/No-Go wybierz najpierw poprawne etykiety zdarzeń. P300 nie musi pojawić się w każdym warunku ani osiągnąć maksimum dokładnie po 300 ms; latencja i amplituda zależą od paradygmatu, liczby prób, lokalizacji elektrod i badanej populacji. Nie obiecuj istotnej statystycznie różnicy jako oczekiwanego wyniku każdego ćwiczenia.

7. Gęstość widmowa mocy (PSD)

Aktualne APi MNE wykorzystuje `raw.compute_psd()` lub `epochs.compute_psd()`. Dawne funkcje `psd_welch` i `psd_multitaper` z modułu mne.time_frequency zostały zastąpione interfejsem metod obiektów. PSD MNE podawane jest zasadniczo w V^2/Hz dla kanałów EEG; przy przeliczeniu na $\mu V^2/Hz$ należy pomnożyć przez 10^{12} [1].

```
import matplotlib.pyplot as plt
spectrum = clean_erp.compute_psd(method="welch", fmin=1, fmax=40, picks="eeg", n_fft=512)
psd, freqs = spectrum.get_data(return_freqs=True)
mean_psd_uV2 = psd.mean(axis=0) * 1e12
plt.plot(freqs, mean_psd_uV2)
plt.xlabel("Częstotliwość [Hz]")
plt.ylabel("PSD [μV²/Hz]")
plt.tight_layout()
plt.show()
```

Rozdzielczość fizyczną estymacji ogranicza przede wszystkim długość analizowanego okna; samo zwiększenie n_fft przez dopełnienie zerami nie zwiększa zdolności rozróżniania bliskich składowych. Widoczne maksimum alfa około 10 Hz zależy od danych i nie jest gwarantowane.

8. Analiza czasowo-częstotliwościowa (TFR)

w analizie TFR dobiera się częstotliwości i liczbę cykli falki do kompromisu między rozdzielczością czasową a częstotliwościową. Baseline dla mocy (np. zmiana względna lub log-ratio) ma inną interpretację niż odejmowanie średniego napięcia ERP.

```
import numpy as np
freqs = np.arange(4., 31., 2.)
power = epochs.compute_tfr(method="morlet", freqs=freqs,
                           n_cycles=freqs / 2, average=True)
power.plot(picks="eeg", baseline=(-0.2, 0), mode="logratio")
```

Krótka epoka może nie wystarczyć do rzetelnej estymacji niskich częstotliwości. Przed obliczeniami dobierz długość segmentu oraz okres odniesienia do analizowanych falek.

9. Analiza łączności funkcjonalnej

Funkcje analizy łączności przeniesiono do osobnego pakietu `mne-connectivity`. Koherencja i PLV mierzą statystyczną zależność, a nie bezpośrednią komunikację anatomiczną ani przyczynowość. Na poziomie elektrod mogą być zniekształcone przez przewodnictwo objętościowe i wspólne odniesienie [5].

```
from mne_connectivity import spectral_connectivity_epochs
conn = spectral_connectivity_epochs(epochs, method="coh",
                                    mode="fourier", fmin=8, fmax=12,
                                    faverage=True)
connectivity_matrix = conn.get_data(output="dense")
print(connectivity_matrix.shape)
```

Miary PLI/wPLi są w określonych sytuacjach mniej wrażliwe na sprzężenia o zerowym przesunięciu fazy, ale również mają ograniczenia. w porównaniach należy kontrolować liczbę epok oraz jakość sygnału.

10. Statystyka i wnioskowanie

Dobór testu musi odpowiadać planowi eksperymentu: dane zależne versus niezależne, poziom analizy (epoka/osoba), liczba porównań i hipoteza. `mne.stats.permutation_t_test` jest permutacyjną procedurą opartą na statystyce t, nie „testem t nieparametrycznym” w prostym sensie. Testy klastrowe zwykle kontrolują błąd rodzinny dla klastrów, nie pozwalają automatycznie lokalizować dokładnego początku lub granic efektu.

- Testy t, ANOVA i modele mieszane stosuj wtedy, gdy odpowiadają projektowi i założeniom. Testy permutacyjne wymagają poprawnej wymienialności obserwacji.
- FDR kontroluje oczekiwany odsetek fałszywych odkryć wśród odrzuconych hipotez; Bonferroni i odpowiednie testy maksymalnej statystyki kontrolują FWER.
- w analizie grupowej jednostką niezależną jest zwykle uczestnik, a nie pojedyncza epoka. Epok tego samego uczestnika nie należy traktować jako niezależnych osób.

11. Uczenie maszynowe

w klasyfikacji EEG najważniejsze są uniknięcie przecieku informacji, prawidłowy podział uczestników/sesji, walidacja krzyżowa i metryki odporne na nierównowagę klas. Wszystkie transformacje uczone na danych (skalowanie, selekcja cech) muszą być dopasowywane wyłącznie do części treningowej.

Poniżej jest minimalny, wykonywalny wzorzec klasyfikatora SVM na cechach widmowych, zakładający istnienie co najmniej dwóch klas epok:

```
import numpy as np
from sklearn.pipeline import make_pipeline
from sklearn.preprocessing import StandardScaler
from sklearn.svm import SVC
from sklearn.model_selection import StratifiedKFold, cross_val_score

X_psd = epochs.compute_psd(method="welch", fmin=8, fmax=30).get_data()
X = np.log10(X_psd.mean(axis=-1) + 1e-30)  # epoki × kanały
y = epochs.events[:, 2]
cv = StratifiedKFold(n_splits=5, shuffle=True, random_state=42)
model = make_pipeline(StandardScaler(), SVC(kernel="linear"))
scores = cross_val_score(model, X, y, cv=cv, scoring="balanced_accuracy")
print("Balanced accuracy:", scores.mean(), scores.std())
```

UWAGA: Ten kod prezentuje walidację na poziomie epok wyłącznie dydaktycznie. Nie stanowi właściwej walidacji uogólnienia między uczestnikami. Dla wielu osób lub sesji użyj GroupKFold, StratifiedGroupKFold lub LeaveOneGroupOut, ze świadomie zdefiniowanymi grupami. Próg „60% skuteczności” nie jest uniwersalnym kryterium poprawności modelu.

12. Format BIDS i automatyzacja

BIDS standaryzuje strukturę katalogów oraz metadane; dane źródłowe i przetworzone powinny być wyraźnie rozdzielone. `write_raw_bids` wymaga poprawnych danych i metadanych oraz należy sprawdzić ograniczenia funkcji, w tym związane z formatem i stanem danych [3].

```
from mne_bids import BIDSPath, read_raw_bids
bids_path = BIDSPath(subject="01", task="rest", datatype="eeg",
                     root="./bids_dataset")
raw_bids = read_raw_bids(bids_path)
```

MNE-BIDS-Pipeline instaluje się osobno. Konfigurację zapisuje się w pliku Python `.py`, a nie domyślnie w `myconfig.json` [4].

```
python -m pip install mne-bids-pipeline
mne_bids_pipeline --create-config=custom_config.py
mne_bids_pipeline --config=custom_config.py
```

14. Jak Python reprezentuje sygnał EEG

14.1. Napięcie, próbki, kanały i zdarzenia

EEG opisuje zmiany różnicy potencjałów w czasie. Pojedynczy zapis jest zwykle macierzą kanały × próbki, a częstotliwość próbkowania pozwala przeliczyć indeks próbki na czas. Typowy sygnał zawiera także metadane: rodzaj kanałów EEG/EOG/ECG, położenia elektrod, odniesienie, filtrację sprzętową i adnotacje zdarzeń. w MNE wartości napięcia EEG przechowuje się w woltach; wykresy mogą prezentować mikrovolty. Pomylenie V i µV powoduje błąd skali o czynnik milion [1].

Struktura	Co przedstawia	Główne zastosowanie
Raw	Wielo-kanałowy zapis ciągły wraz z metadanymi	Przegląd rejestracji i wstępne przetwarzanie
Epochs	Próby wycięte względem zdarzeń	Porównywanie warunków, ERP, TFR, klasyfikacja
Evoked	Uśrednienie epok danego warunku	Analiza odpowiedzi związanych ze zdarzeniem
Spectrum / EpochsSpectrum	Oszacowania widma mocy	Analiza pasm częstotliwościowych
EpochsTFR / AverageTFR	Moc w funkcji czasu i częstotliwości	Dynamika oscylacji
Info / Annotations	Parametry rejestracji i opisy fragmentów	Interpretacja, selekcja i kontrola jakości

Rozróżnienie tych obiektów jest ważniejsze niż zapamiętanie ich metod: nie należy traktować średniej ERP, surowej epoki i widma jako tego samego rodzaju obserwacji. Uśrednianie zmniejsza część aktywności nieskoordynowanej czasowo z bodźcem, ale może też zatrzeć zmienność między próbami.

14.2. Metadane i montaż elektrod

Montaż (montage) definiuje przestrzenne położenie elektrod. Jest niezbędny do wiarygodnych map topograficznych i interpolacji kanałów. Standard nazewniczy 10–20 nie gwarantuje, że współrzędne i rozmieszczenie elektrod w konkretnym badaniu są identyczne ze standardowym szablonem. Lokalizację elektrod i odniesienie należy dokumentować także wtedy, gdy badanie nie obejmuje rekonstrukcji źródeł [1, 14].

```
print(raw.info["sfreq"], raw.get_channel_types())
print(raw.info["bads"], raw.get_montage())
```

15. Co można naukowo wywnioskować z EEG

Pytanie badawcze	Rodzina analiz	Główne zastrzeżenie
Czy bodziec wywołuje powtarzalną odpowiedź?	ERP oraz modele statystyczne	Latencja i amplituda zależą od paradygmatu
Czy zmienia się aktywność w pasmach?	PSD, moc względna, parametry aperiodiczne	Moc nie jest tożsama z aktywacją regionu
Kiedy pojawia się zmiana oscylacji?	TFR, ERD/ERS	Zależność rozdzielczości od liczby cykli
Czy kanały współzmieniają się w czasie?	Koherencja, PLV, wPLI	Przewodnictwo objętościowe i odniesienie
Gdzie mogły powstać sygnały?	Modelowanie źródeł	Niejednoznaczny problem odwrotny
Czy można przewidzieć kategorię/stan?	scikit-learn, Braindecode	Przeciek danych i uogólnianie międzyosobnicze

w badaniu neuropsychologicznym analiza EEG staje się użyteczna dopiero po powiązaniu miar sygnału z konstruktem poznawczym, zachowaniem oraz projektem eksperymentu. Zwiększona moc theta nie stanowi samodzielnie dowodu na określony proces psychiczny ani rozpoznania klinicznego; interpretacja wymaga warunków kontrolnych, porównywalnych procedur i ostrożności wobec zakłóceń.

16. Sygnał użyteczny i źródła zakłóceń

16.1. Artefakty fizjologiczne i techniczne

w zapisie EEG nakładają się aktywność neuronalna oraz potencjały oczno-ruchowe, mięśniowe, sercowe i sygnały pochodzące z ruchu przewodów czy niestabilnego kontaktu elektrody. Artefakty nie zawsze można wiarygodnie odseparować pojedynczym filtrem. Analiza jakości powinna poprzedzać decyzję o odrzuceniu epoki, interpolacji elektrody lub zastosowaniu ICA [2].

Zjawisko	Częsty obraz	Konsekwencja analityczna
Mruganie, ruchy oczu	Wolne, duże odchylenia, często czołowo	Wpływ na ERP i moc niskich częstotliwości
EMG twarzy/szyi	Szerokopasmowa aktywność, często wyższe pasma	Ryzyko pozornej aktywności beta/gamma
Zakłócenia sieciowe	Wąski „pik” ok. 50 Hz w Polsce	Zafałszowanie PSD i niektórych analiz łączności
Słaby kontakt elektrody	Szum, dryf lub przerwy	Zaburzenie odniesienia, interpolacji i ICA
Ruch i przesunięcie czepka	Zmiana topografii i niestacjonarność	Spadek porównywalności warunków

16.2. Filtracja nie jest neutralna

Filtr górnoprzepustowy osłabia powolne składowe i dryf, natomiast dolnoprzepustowy ogranicza szybkie zmiany. Filtracja może powodować rozmycie czasowe, oscylacje przy ostrych przejściach i zmiany amplitudy lub kształtu ERP. Przed zastosowaniem filtru warto rozważyć jego charakterystykę częstotliwościową, odpowiedź impulsową, długość sygnału oraz sposób obsługi granic danych. Parametr 0,1 Hz odpowiedni dla jednej analizy ERP może być niewystarczający albo nieuzasadniony w innej [1, 2].

16.3. Odniesienie, ICA i interpolacja

Każdy kanał EEG opisuje różnicę potencjałów względem odniesienia. Zmiana odniesienia zmienia interpretację amplitud i rozkład przestrzenny sygnału. Średnie odniesienie ma inne własności przy gęstym, równomiernym rozmieszczeniu elektrod niż przy niewielkiej liczbie kanałów. Interpolacja nie odzyskuje rzeczywistego pomiaru, tylko przybliża go na podstawie sąsiednich elektrod. ICA jest dekompozycją statystyczną, a nie biologicznym rozdzieleniem źródeł; usuwanie składowych wymaga kontroli topografii, widma i przebiegu czasowego [2, 14].

17. ERP, widmo i analiza czasowo-częstotliwościowa: interpretacja

17.1. ERP i pytania o przetwarzanie poznawcze

Potencjały związane ze zdarzeniem są uśrednionymi przebiegami zsynchronizowanymi z bodźcem lub odpowiedzią. w analizach N1, N2, P3/P300 czy późnych potencjałów dodatnich należy z góry określić okno czasowe, regiony zainteresowania i sposób pomiaru amplitudy. Wybieranie największej różnicy

z tych samych danych, na których następnie testuje się jej istotność, może prowadzić do błędnych wniosków. Należy opisać liczbę zachowanych epok, sposób korekcji linii bazowej i warunki kontrastowania [13].

17.2. Widmo mocy i pasma EEG

PSD opisuje rozkład mocy sygnału w funkcji częstotliwości. Nazwy delta, theta, alfa, beta i gamma są konwencjonalnymi przedziałami, których granice różnią się między publikacjami. w badaniach wieku, snu lub neuroatypowości warto unikać automatycznego przypisywania zakresom jednoznacznych funkcji. Moc absolutna, moc względna i charakterystyka 1/f nie są zamiennymi miarami; podanie samego „wzrostu alfa" bez definicji miary jest niepełne.

w metodzie Welcha segmenty są okienkowane i ich periodogramy są uśredniane. Dłuższy segment poprawia zdolność rozdzielenia bliskich częstotliwości, ale daje mniej fragmentów do uśredniania. Dopełnianie zerami zwiększa gęstość próbkowania osi częstotliwości, nie tworząc dodatkowej informacji. w analizie EEG należy odróżnić parametr częstotliwości próbkowania od rozdzielczości widma.

17.3. Dynamika oscylacji

Analiza TFR umożliwia rozpoznanie okresowych zmian mocy po zdarzeniu. Krótsze falki lepiej lokalizują zmiany w czasie, dłuższe pozwalają dokładniej oceniać częstotliwości. ERD oznacza zmniejszenie mocy względem odniesienia, ERS jej zwiększenie, a nie bezpośrednią dezaktywację lub aktywację określonego obszaru mózgu. Sposób normalizacji do okresu bazowego należy podawać w metodach.

18. Łączność i rekonstrukcja źródeł

Analiza łączności funkcjonalnej mierzy statystyczne zależności między szeregami czasowymi. Koherencja mierzy zależność widmową, a miary fazowe pozwalają analizować synchronizację faz. Ich podwyższone wartości nie dowodzą ani połączenia anatomicznego, ani kierunku przepływu informacji. Wynik może zależeć od wspólnego odniesienia, przewodnictwa objętościowego, długości epok i składowych artefaktowych [5].

Modelowanie źródeł EEG próbuje przypisać potencjały na skórze głowy do aktywności wewnątrzczaszkowej. Wymaga modelu przewodzenia, współrzędnych elektrod i założeń regularizacyjnych; korzysta często z obrazu MRi i modelu geometrycznego głowy. Problem odwrotny jest niejednoznaczny: kilka różnych rozkładów źródeł może wyjaśniać te same pomiary. Modele te należy traktować jako oszacowania zależne od modelu, nie bezpośrednie obrazy miejsca generowania potencjałów [1].

19. Analiza statystyczna: jednostka obserwacji i wnioskowanie

Najczęstszym błędem jest mylenie dużej liczby próbek lub epok z dużą liczbą niezależnych badanych. Sto epok od jednej osoby nadal nie oznacza próby liczącej sto osób. Przy porównaniach międzyosobniczych jednostką replikacji jest zazwyczaj uczestnik. w projektach z powtarzanymi pomiarami przydatne są modele uwzględniające strukturę zależności oraz zmienność między uczestnikami.

Testy permutacyjne wymagają wymienialności elementów podlegających permutacji w ramach hipotezy zerowej. w testach klastrowych istotność klastra nie upoważnia do precyzyjnego lokalizowania granic czasowych efektu. Wielokrotne porównania kanałów, punktów czasu i częstotliwości wymagają jawnego planu kontroli błędów. Warto raportować wielkość efektu, przedziały ufności tam, gdzie mają uzasadnienie, oraz pełną liczbę analizowanych porównań [13].

20. Uczenie maszynowe i dekodowanie stanów

W EEG można trenować klasyfikatory rozpoznające warunek eksperymentalny, rodzaj wyobrażonego ruchu albo inne etykiety, ale dokładność predykcji nie jest tożsama z trafnością interpretacji neuropsychologicznej. Model może opierać się na artefaktach aparatury, różnicach między sesjami lub cechach ubocznych protokołu. Interpretacja modeli wymaga odróżnienia przewidywania od wyjaśniania mechanizmu.

Walidacja wewnątrz osoby mierzy inne zdolności niż ocena na nowych uczestnikach. Dla generalizacji międzyosobniczej dane jednej osoby nie mogą jednocześnie zasilać treningu i testu. Selekcja cech, skalowanie i strojenie hiperparametrów muszą być realizowane wyłącznie na danych treningowych w każdym podziale; wynik ostateczny należy wyznaczyć na niezależnym zbiorze testowym lub za pomocą właściwie skonstruowanej walidacji zagnieżdżonej. Nierównowagę klas warto opisywać m.in. balanced accuracy, F1, czułością i swoistością, adekwatnie do celu.

Biblioteka scikit-learn dobrze odpowiada modelom bazującym na cechach takich jak moc pasmowa i parametry ERP. Braindecode integruje głębokie sieci neuronowe z reprezentacjami EEG, ale bardziej rozbudowany model nie musi zapewniać lepszej generalizacji od poprawnie zwalidowanego modelu liniowego [10].

21. Powtarzalność, interoperacyjność i otwarta nauka

Przejrzysty projekt EEG oddziela dane źródłowe od pochodnych, opisuje metadane uczestników oraz zdarzeń i przechowuje wersje bibliotek. BIDS dostarcza uzgodnioną strukturę danych i nazewnictwo, natomiast MNE-BIDS umożliwia import i eksport zgodny z tym standardem. Nie należy utożsamiać samego formatu pliku EDF lub BrainVision ze zgodnością całego zbioru z BIDS [3].

Element dokumentacji	Dlaczego jest ważny
Parametry akwizycji	Wyjaśniają ograniczenia fizyczne sygnału
Mapowanie zdarzeń	Pozwala odtworzyć warunki eksperymentu
Kryteria wykluczania kanałów/epok	Umożliwiają audyt decyzji badacza
Wersje bibliotek i środowiska	Ograniczają ryzyko zmian APi i wyników
Kod i parametry transformacji	Wspierają replikację obliczeń
Dane i pochodne w BIDS	Ułatwiają interoperacyjność i współdzielenie
Raport jakości oraz ograniczenia	Zapobiegają nieuzasadnionym interpretacjom

Dane EEG mogą być danymi osobowymi lub danymi dotyczącymi zdrowia; publikowanie sygnałów wymaga analizy prawnej, etycznej i oceny ryzyka ponownej identyfikacji. Informacje o zgodzie, polityce dostępu oraz zakresie anonimizacji nie są zastępowane samym usunięciem nazwiska z pliku.

22. Jak dobierać narzędzia do konkretnego celu

Cel	Rozsądny punkt wyjścia	Rozszerzenia
Klasyczne ERP i TFR	MNE-Python	autoreject, statystyka SciPy
Porządkowanie większego zbioru	MNE-BIDS	MNE-BIDS-Pipeline
Badania snu	MNE-Python	YASA
Łączność fazowa i widmowa	MNE-Connectivity	Kontrola artefaktów, analizy źródłowe
Klasyfikacja tradycyjna	MNE + scikit-learn	Walidacja grupowa, raportowanie metryk
Głębokie uczenie	Braindecode/PyTorch	Zewnętrzny zbiór testowy
Integracja ECG/EMG/EDA	MNE + NeuroKit2	Synchronizacja multimodalna
Prezentacja wyników	MNE + Matplotlib	Plotly, seaborn

W neuropsychologii najpierw warto nauczyć się rozróżniać pytanie badawcze, wielkość fizyczną i metodę wnioskowania. Dopiero wówczas dobór biblioteki ma sens. Python jest narzędziem analizy, a nie źródłem dowodu naukowego: to projekt eksperymentu, jakość danych i metodologia decydują o trafności wnioskowania.

23. Uzupełniająca literatura metodologiczna

[16] Gramfort, A. i in. (2013). MEG and EEG data analysis with MNE-Python. Frontiers in Neuroscience, 7, 267. https://doi.org/10.3389/fnins.2013.00267

[17] Pernet, C. R. i in. (2019). EEG-BIDS, an extension to the brain imaging data structure for electroencephalography. Scientific Data, 6, 103. https://doi.org/10.1038/s41597-019-0104-8

[18] Appelhoff, S. i in. (2019). MNE-BIDS: Organizing electrophysiological data into the BIDS format and facilitating their analysis. Journal of Open Source Software, 4(44), 1896. https://doi.org/10.21105/joss.01896

[19] Jas, M. i in. (2017). Autoreject: Automated artifact rejection for MEG and EEG data. NeuroImage, 159, 417–429. https://doi.org/10.1016/j.neuroimage.2017.06.016

[20] MNE-Python: przegląd struktur danych i podstawowych analiz. https://mne.tools/stable/auto_tutorials/intro/10_overview.html

24. Bibliografia i dokumentacja podstawowa

[1] MNE-Python, instalacja i dokumentacja API. https://mne.tools/stable/install/manual_install_python.html ; https://mne.tools/stable/

[2] MNE-Python, Repairing artifacts with ICA. https://mne.tools/stable/auto_tutorials/preprocessing/40_artifact_correction_ica.html

[3] MNE-BIDS, dokumentacja. https://mne.tools/mne-bids/stable/

[4] MNE-BIDS-Pipeline, Basic usage. https://mne.tools/mne-bids-pipeline/stable/getting_started/basic_usage.html

[5] MNE-Connectivity, APi spectral_connectivity_epochs. https://mne.tools/mne-connectivity/stable/generated/mne_connectivity.spectral_connectivity_epochs.html

[6] autoreject, dokumentacja. https://autoreject.github.io/

[7] pyEDFlib, dokumentacja. https://pyedflib.readthedocs.io/

[8] NeuroKit2, dokumentacja. https://neuropsychology.github.io/NeuroKit/

[9] YASA, dokumentacja. https://yasa-sleep.org/

[10] Braindecode, dokumentacja. https://braindecode.org/

[11] Eelbrain, dokumentacja. https://eelbrain.readthedocs.io/

[12] MNE-Python, import plików EEG. https://mne.tools/stable/auto_tutorials/io/20_reading_eeg_data.html

[13] MNE-Python, analiza statystyczna. https://mne.tools/stable/auto_tutorials/stats-sensor-space/index.html

[14] MNE-Python, EEG reference. https://mne.tools/stable/auto_tutorials/preprocessing/55_setting_eeg_reference.html

[15] PhysioNet Sleep-EDF. https://physionet.org/content/sleep-edfx/1.0.0/