

**WYDZIAŁ
ELEKTROTECHNIKI
I INFORMATYKI
POLITECHNIKI RZESZOWSKIEJ**

Daniel Kleczyński 3AA/EF-DI

Wykorzystanie rekurencyjnych sieci neuronowych w dekodowaniu
intencji ruchowych na podstawie sygnałów EEG

Projekt inżynierski

Opiekun pracy:
prof. dr hab. inż. Jacek Kluska

Rzeszów, 2025

Spis treści

1. Korelacja między ruchami a sygnałami EEG	5
1.1. Potencjały związane z ruchem	5
1.2. Wzorce czasowo-częstotliwościowe	6
1.3. Dekodowanie kierunku ruchu	6
2. Przegląd wybranych metod klasyfikacji sygnałów EEG dla zadania <i>Motor Imagery</i>	7
2.1. Przetwarzanie danych i segmentacja	7
2.2. Architektury modeli głębokiego uczenia	7
2.3. Porównanie wyników	9
3. Analiza zbioru danych	9
3.1. Opis klas i ich ilość	10
3.1.1. T0 (Stan spoczynku)	10
3.1.2. T1 (Ruch lub wyobrażenie ruchu lewej pięści lub obu pięści)	10
3.1.3. T2 (Ruch lub wyobrażenie ruchu prawej pięści lub obu stóp)	10
3.2. Charakterystyka sygnałów EEG	11
3.3. Metryki statystyczne danych	12
4. Podziału danych	14
4.1. Metody podziału danych na próbki	14
4.2. Podział zbioru danych na zbiory treningowy i walidacyjny	15
5. Przetwarzanie danych EEG do eksperymentu	17
5.1. Normalizacja sygnałów EEG	17
5.2. Normalizacja sygnałów EEG	17
5.3. Transformacja falkowa	17
5.3.1. Wzór na ciągłą transformację falkową	18
5.3.2. Typ falki	18
5.3.3. Wzór na falkę Daubechies 4 (db4)	19
5.3.4. Zastosowanie falki Daubechies w analizie sygnałów	19
5.3.5. Zastosowanie transformacji falkowej	19
5.4. Podsumowanie	19
6. Wybór modeli oraz ich struktur do eksperymentu	20
6.1. Long Short-Term Memory (LSTM)	20
6.1.1. Problemy klasycznych RNN	20
6.1.2. Architektura komórki LSTM	20
6.1.3. Przewagi LSTM nad klasycznymi RNN	21
6.1.4. Podsumowanie	22
6.2. CNN-LSTM: Hybrydowe podejście do analizy EEG	22
6.2.1. Architektura modelu	22
6.2.2. Przewagi CNN-LSTM	22
6.2.3. Wady CNN-LSTM	22

6.2.4. Zastosowanie modelu w analizie EEG	22
6.2.5. Podsumowanie	23
7. Eksperyment dla nieodseparowanych osób w zbiorze testowym i treningowym	23
7.1. Wybór osób oraz zadania do eksperymentu	23
7.2. Normalizacja danych	24
7.3. Przestrzeń wyszukiwania hiperparametrów	24
7.4. Konfiguracja modelu oraz procesu uczenia	24
7.5. Struktura modelu	26
7.6. Wyniki eksperymentu	26
7.6.1. Dokładność (<i>Accuracy</i>)	26
7.6.2. Pole pod krzywą ROC (<i>AUC</i>)	27
7.6.3. Miara F1 (<i>F1 Score</i>)	27
7.6.4. Strata (<i>Loss</i>)	27
7.6.5. Precyza (Precision)	27
7.6.6. Czułość (<i>Recall</i>)	28
7.6.7. Podsumowanie najlepszej konfiguracji hiperparametrów	28
7.6.8. Analiza korelacji hiperparametrów z dokładnością	29
7.7. Podsumowanie	29
8. Eksperyment dla odseparowanych osób w zbiorze testowym i treningowym	30
8.1. Wyniki eksperymentu	30
8.2. Wnioski	30
9. Podsumowanie	31

Abstrakt

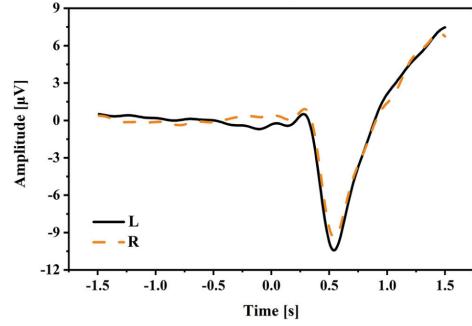
Celem niniejszej pracy jest opracowanie modelu dekodującego intencje ruchowe na podstawie sygnałów EEG, wykorzystującego różne konfiguracje rekurencyjnych sieci neuronowych (RNN), w tym w szczególności sieci LSTM (Long Short-Term Memory). Projekt bada możliwość precyzyjnego dekodowania intencji ruchowych w warunkach offline oraz analizuje wymagania, które muszą zostać spełnione, aby osiągnąć wysoką dokładność tego procesu. Istotnym aspektem jest ocena możliwości uogólnienia modelu na dane pochodzące od osób spoza zbioru uczącego, co stanowi kluczowy element w kontekście praktycznych zastosowań. W pracy wykorzystano dane z *EEG Motor Movement/Imagery Dataset* [5], dostępne w zbiorze PhysioNet. Analiza opierała się na dwóch wybranych klasach ruchowych oraz jednej spoczynkowej. Przeprowadzone eksperymenty wykazały, że przy odpowiednim podziale próbek pomiędzy zbiór treningowy i walidacyjny, z zachowaniem tej samej liczebności klas, model osiągnął dokładność na poziomie 0.9833, a pole pod krzywą ROC uzyskało wartość 0.99915 dla zbioru testowego zawierającego próbki od tych samych osób, które znajdowały się w zbiorze treningowym. Model, który osiągnął najwyższą skuteczność, charakteryzował się następującymi parametrami: rozmiar warstwy ukrytej (hidden size) = 334, współczynnik uczenia (learning rate) = 3.587×10^{-5} , liczba warstw = 2, współczynnik dropout = 0.393, długość sekwencji = 86. Natomiast próba wyszkolenia modelu LSTM na danych, w których osoby z testowego zbioru danych były wyseparowane ze zbioru treningowego, zakończyła się niepowodzeniem – dokładność nigdy nie przekroczyła 0.5. Sugeruje to bardzo odmienny charakter danych dla różnych osób lub niewystarczającą zdolność modeli LSTM do uogólniania. Praca ma na celu przyczynienie się do rozwoju interfejsów mózg-komputer (BCI – Brain-Computer Interface), które mogą znaleźć zastosowanie w poprawie jakości życia osób z niepełnosprawnościami. Szczególny nacisk kładzie się na potencjalne wykorzystanie tego typu modeli do sterowania urządzeniami, takimi jak protezy, wyłącznie za pomocą myśli użytkownika. Wyniki badań mogą również dostarczyć cennych informacji na temat warunków niezbędnych do skutecznego wykorzystania sygnałów EEG w praktyce.

1. Korelacja między ruchami a sygnałami EEG

Dekodowanie intencji ruchowych z sygnałów EEG jest jednym z kluczowych obszarów badań w dziedzinie interfejsów mózg-komputer Brain-Computer Interface(BCI). W szczególności potencjały korowe związane z ruchem Movement-Related Cortical Potentials (MRCPs) oraz wzorce czasowo-częstotliwościowe odgrywają kluczową rolę w przewidywaniu kierunku ruchu.

1.1. Potencjały związane z ruchem

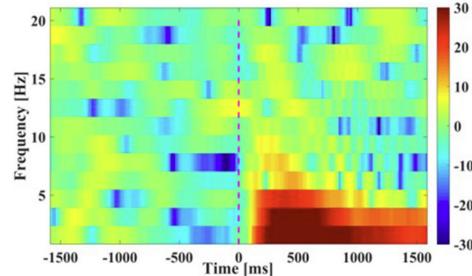
Jak pokazano w pracy [14], MRCPs mogą być używane do przewidywania ruchów ręki na podstawie niskoczęstotliwościowych sygnałów EEG (0.01–4 Hz). Rysunek 1.1 przedstawia średnie MRCPs w elektrodzie Cz w czasie przygotowania i wykonania ruchu. Amplitudy potencjałów pozostały stabilne w fazie przygotowania ruchu (od -1,5 do 0 s), natomiast podczas inicjacji ruchu zaobserwowano znaczną negatywną zmianę potencjału.



Rysunek 1.1: Średnie potencjały korowe (MRCPs) w elektrodzie Cz dla kierunków ruchu prawej ręki (lewo i prawo). Czas 0 s odnosi się do momentu inicjacji ruchu. Źródło: [14]

1.2. Wzorce czasowo-częstotliwościowe

W pracy [14] zbadano również wzorce czasowo-częstotliwościowe. Rysunek 1.2 przedstawia widmo czasowo-częstotliwościowe, które ukazuje wzrost energii w niskich częstotliwościach ($< 7 \text{ Hz}$) w czasie wykonywania ruchu. Wynik ten potwierdza, że główne modulacje mocy związane z ruchem są skoncentrowane w niskim zakresie częstotliwości.



Rysunek 1.2: Widmo czasowo-częstotliwościowe sygnałów EEG w kanale Cz dla ruchu prawej ręki. Czas 0 s odnosi się do momentu inicjacji ruchu. Źródło: [14]

1.3. Dekodowanie kierunku ruchu

Dzięki zaawansowanym metodom, takim jak wykorzystanie nieliniowych dynamicznych parametrów MRCPs, udało się osiągnąć wysoką dokładność w dekodowaniu kierunków ruchu. Zastosowana metoda oparta na analizie sieci ESN (Echo State Networks) - sieci echo-stanu osiągnęła średnią dokładność dekodowania na poziomie 89.48% [14].

Przeprowadzone badania potwierdzają, że sygnały EEG zawierają istotne informacje o ruchach, co uzasadnia możliwość tworzenia zaawansowanych modeli do dekodowania intencji ruchowych. Takie modele mogą być wykorzystane w praktycznych zastosowaniach, takich jak interfejsy BCI, systemy wspomagające rehabilitację, czy sterowanie protezami i egzoszkieletami.

2. Przegląd wybranych metod klasyfikacji sygnałów EEG dla zadania *Motor Imagery*

Motor Imagery (MI) to proces mentalnego wyobrażania sobie ruchu bez jego faktycznego wykonania, który może aktywować lub hamować aktywność w korze mózgowej. W kontekście Echo State Networks MI znajduje zastosowanie w modelowaniu i analizie procesów związanych z kontrolą ruchu.

W literaturze dotyczącej klasyfikacji sygnałów EEG w kontekście wyobrażeniowych ruchów (MI) kluczową rolę odgrywa analiza zbiorów danych, umożliwiających ocenę skuteczności różnych modeli uczenia maszynowego. Jednym z takich zbiorów jest EEG Motor Movement/Imagery Dataset, udostępniany przez repozytorium ogólnodostępnych danych badawczych zarządzane przez MIT Laboratory for Computational Physiology.

Zbiór ten obejmuje ponad 1500 nagrani z udziałem 109 ochotników, zarejestrowanych za pomocą systemu BCI2000 – wszechstronnej platformy oprogramowania zaprojektowanej do badań i aplikacji związanych z interfejsami mózg-komputer (BCI – Brain-Computer Interfaces). Dane te zawierają sygnały związane zarówno z rzeczywistymi ruchami kończyn, jak i ich wyobrażeniem, co czyni je wartościowym źródłem do testowania nowoczesnych metod głębokiego uczenia.

2.1. Przetwarzanie danych i segmentacja

Przed wykorzystaniem do uczenia modeli, sygnały EEG są zazwyczaj poddawane filtracji w celu redukcji szumów oraz artefaktów. Standardowo stosuje się filtr Notcha (usuwający składową 50/60 Hz związaną z zasilaniem) oraz filtr pasmowoprzepustowy ograniczający zakres częstotliwości do przedziału od 2–8 Hz do 30–60 Hz, w zależności od założeń eksperymentu [2, 11]. Sygnały mogą być również poddane ponownemu próbkowaniu (np. do 125 Hz lub 160 Hz) w celu standaryzacji oraz zmniejszenia wymiarowości danych.

W pracach [2, 11] podzielono nagrania na krótkie segmenty czasowe (ang. *segmentation* lub *windowing*), np. 4-sekundowe próbki. Każdą próbę można dodatkowo rozdzielić na mniejsze, niekolidujące okna (np. 8 okien po 0,5 s), co zwiększa liczbę dostępnych przykładów treningowych i ułatwia sieciom neuronowym wykrywanie lokalnych cech czasowo-częstotliwościowych.

2.2. Architektury modeli głębokiego uczenia

W kontekście klasyfikacji sygnałów EEG związanych z wyobrażeniami ruchowymi szczególnie skuteczne są splotowe sieci neuronowe (CNN). Autorzy [2] przeanalizowali kilka wariantów, w tym samą sieć CNN oraz jej hybrydy z LSTM (ang. *Long Short-Term Memory*):

- **CNN** – sieć koncentruje się na wyodrębnianiu cech przestrzennych (tj. między elektrodami) oraz krótkich zależności czasowych.
- **CNN-LSTM** – łączy warstwy splotowe z warstwami LSTM, co pozwala na uchwycenie

zarówno cech przestrzennych, jak i dłuższych zależności czasowych.

- **CNN-BiLSTM** – wykorzystuje dwukierunkowe 2.4 LSTM (ang. *Bidirectional LSTM*), co umożliwia analizę sygnału zarówno w przód, jak i wstecz wzdłuż osi czasowej.

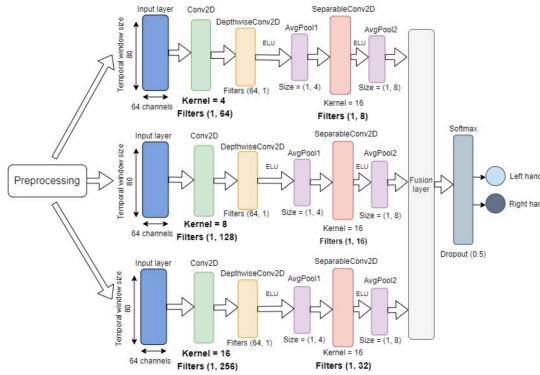
W pracy [2] modele trenowane na zbiorach danych podzielonych na:

- 70% danych do zbioru trenigowego w celu uczenia modelu,
- 10% danych do zbioru walidacyjnego w celu optymalizacji hiperparametrów,
- 20% danych do zbioru testowego w celu oceny modelu przy użyciu danych które nie były używane ani do treningu ani do dostarajania hiperparametów.

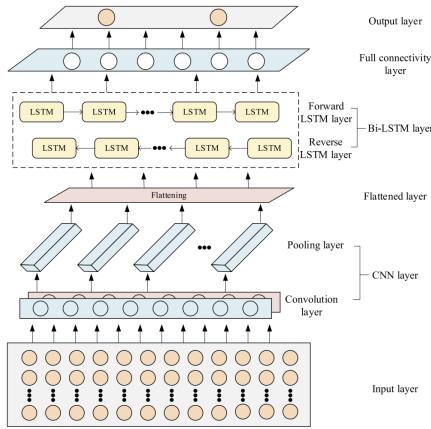
Wybrane eksperymenty i uczestnicy: W badaniu wykorzystano podzbiór danych EEG pochodzących od siedmiu losowo wybranych uczestników. Skupiono się na sześciu zadaniach związanych z wyobrażeniami ruchowymi, oznaczonych numerami 4, 6, 8, 10, 12 i 14. Każdy z tych zadań dotyczył wyobrażonych ruchów, takich jak zamknięcie i otwieranie lewej lub prawej dłoni, a także obu pięści i obu stóp.

Liczba próbek: W przypadku analizowanych danych, liczba pojedynczych próbek wynosiła 826 560 przy zadaniu zdefiniowanych powyżej oraz losowaniu 7 uczestników. Przyjmując, że jeden pomiar składał się średnio z 19 721 próbek, całkowita liczba próbek dla siedmiu uczestników i sześciu zadań wynosiła $19721 \times 7 \times 6 = 828324$, co jest zbliżone do podanej wartości 830 880. Różnica może wynikać z nierównych czasów trwania zadań.

Trening modeli: Trening trwał przez 100 epok z użyciem optymalizatora Adam, wskaźnika uczenia 0,0001 i *batch size* wynoszącego 32. Wyniki dla modelu CNN osiągnęły dokładność 99,86%, natomiast dla CNN-LSTM 98,39% [2]. W pracy [11] zaproponowano podejście *Fusion Convolutional Neural Network* (tzw. *EEGNet Fusion*), które obejmuje kilka równoległych gałęzi splotowych o różnych hiperparametrach (liczba filtrów, rozmiar jąder splotowych) 2.3. Uzyskane cechy są następnie łączone w specjalnej warstwie fuzji (*fusion layer*).



Rysunek 2.3: Schemat architektury EEGNet Fusion [11].



Rysunek 2.4: Schemat architektury CNN-BiLSTM [2].

Wyniki dla modelu EEGNet Fusion oceniono w dwóch scenariuszach: rzeczywistych ruchów i wyobrażeniowych. Dokładność wyniosła odpowiednio 84,1% oraz 83,8%, a modele oceniono na odpowiednio podzielonych zbiorach danych [11].

2.3. Porównanie wyników

Poniżej przedstawiono porównanie wyników uzyskanych w obu pracach:

Tablica 2.1: Porównanie wyników dla różnych modeli [2, 11].

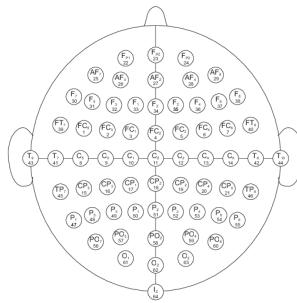
Model	Dokładność	Precyzja	Recall	F1-Score
CNN	99.86%	1.00	1.00	1.00
CNN-LSTM	98.39%	0.98	0.98	0.98
CNN-BiLSTM	99.27%	1.00	0.99	0.99
EEGNet Fusion	84.1% (ruch rzeczywisty)	0.84	0.84	0.84
EEGNet Fusion	83.8% (ruch wyobrażeniowy)	0.84	0.84	0.84

Podsumowując, modele hybrydowe, takie jak CNN-BiLSTM, skuteczniej analizują zależności czasowe w sygnałach EEG, podczas gdy standardowe CNN osiąga najwyższą dokładność przy prostszej strukturze. Podejście EEGNet Fusion zapewnia solidne wyniki, ale z nieco niższą dokładnością w porównaniu do klasycznych modeli CNN [2, 11].

3. Analiza zbioru danych

Prezentowany zbiór danych EEG składa się z ponad 1500 jedno- i dwuminutowych nagrani, zebranych od 109 ochotników. Dane zostały zarejestrowane przy użyciu 64-kanalowego EEG zgodnie z systemem 10-10, wykorzystując system BCI2000. Uczestnicy wykonywali różne zadania motoryczne

oraz ich wyobrażenia, takie jak otwieranie i zamykanie lewej lub prawej pięści, obu pięści czy obu stóp.



Rysunek 3.5: Rozkład na głowie oraz nazwy sensorów EEG.

3.1. Opis klas i ich ilość

Zbiór danych zawiera trzy główne klasy oznaczone jako T0, T1 i T2:

3.1.1. T0 (Stan spoczynku)

Odpowiada momentom, w których uczestnik jest w stanie relaksu, nie wykonuje żadnego ruchu ani nie wyobraża sobie jego wykonania.

3.1.2. T1 (Ruch lub wyobrażenie ruchu lewej pięści lub obu pięści)

- Pomiar 3, 7, 11: Fizyczne otwarcie i zaciśnięcie lewej pięści.
- Pomiar 4, 8, 12: Wyobrażenie otwarcia i zaciśnięcia lewej pięści.
- Pomiar 5, 9, 13: Fizyczne otwarcie i skurczanie obydwóch stóp.
- Pomiar 6, 10, 14: Wyobrażenie otwarcia i skurczania obydwóch stóp.

3.1.3. T2 (Ruch lub wyobrażenie ruchu prawej pięści lub obu stóp)

- Pomiar 3, 7, 11: Fizyczne otwarcie i zaciśnięcie prawej pięści.
- Pomiar 4, 8, 12: Wyobrażenie otwarcia i zaciśnięcia prawej pięści.
- Pomiar 5, 9, 13: Fizyczne otwarcie i zaciśnięcie obydwóch dloni.
- Pomiar 6, 10, 14: Wyobrażenie otwarcia i zaciśnięcia obydwóch dloni.

Klasy T0, T1, T2 są definiowane różnie w zależności od numeru zadania. Poniżej przedstawiono warunki zadań oraz odpowiadające im numery pomiarów:

- 1) **Pomiar referencyjny, otwarte oczy**
- 2) **Pomiar referencyjny, zamknięte oczy**

- 3) **Zadanie 1** (otwieranie i zamykanie lewej lub prawej pięści)
 - 4) **Zadanie 2** (wyobrażenie otwierania i zamykania lewej lub prawej pięści)
 - 5) **Zadanie 3** (otwieranie i zamykanie obu pięści lub obu stóp)
 - 6) **Zadanie 4** (wyobrażenie otwierania i zamykania obu pięści lub obu stóp)
- 7) **Zadanie 1**
- 8) **Zadanie 2**
- 9) **Zadanie 3**
- 10) **Zadanie 4**
- 11) **Zadanie 1**
- 12) **Zadanie 2**
- 13) **Zadanie 3**
- 14) **Zadanie 4**

Aby uprościć problem klasyfikacji oraz zgodnie z dobrymi praktykami rozpocząć od prostszego przypadku, zdecydowano się początkowo skupić na klasyfikacji trzech klas z zadania nr 1. Analizę przeprowadzono na danych pochodzących od 10 losowo wybranych osób o indeksach: [1, 2, 8, 9, 10, 13]. Takie podejście pozwala na opracowanie możliwie najlepszego modelu w ramach ograniczonego zakresu danych, który można następnie udoskonalać i rozszerzać na bardziej złożone przypadki oraz większe zbiory danych.

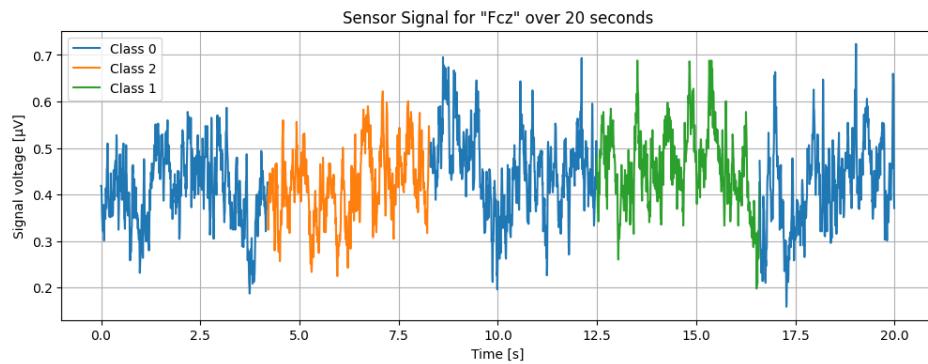
3.2. Charakterystyka sygnałów EEG

Sygnały EEG (elektroencefalograficzne) rejestrują aktywność elektryczną mózgu, charakteryzującą się niskimi amplitudami (rzędu mikrowoltów) oraz wysoką złożonością wynikającą z różnorodności stanów funkcjonalnych mózgu, takich jak czuwanie, sen czy aktywność poznawcza. Aktywność ta nie ogranicza się do jednego obszaru mózgu, lecz obejmuje różne regiony, które są aktywowane w różnych czasie i na różne sposoby. Utrudnia to jednoznaczne określenie, czy dany stan aktywności mózgu odpowiada konkretnej czynności.

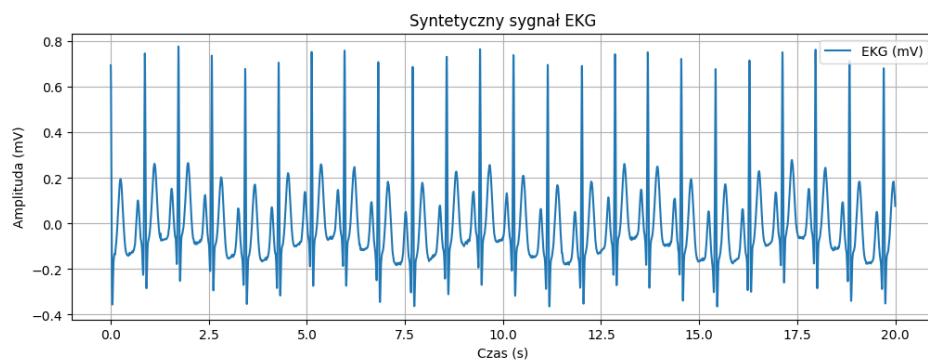
W celu przedstawienia charakterystyki sygnałów EEG wybrano kanał FCz , ponieważ jego lokalizacja obejmuje obszar pierwotnej kory ruchowej (M1), która odpowiada m.in. za kontrolę ruchów kończyn [13]. Należy jednak pamiętać, że jeden sensor nie jest w stanie w pełni odwzorować aktywności danego obszaru mózgu, co implikuje konieczność analizy danych z wielu sensorów jednocześnie, przez określony czas. Przykładowy zapis sygnału EEG o długości 20 sekund z sensora FCz przedstawiono na Rysunku 3.6. Widoczne są w nim fluktuacje o nieregularnym charakterze.

Dla porównania, sygnał EKG (elektrokardiograficzny) zaprezentowany na Rysunku 3.7 charakteryzuje się wyższymi amplitudami (rzędu miliwoltów) oraz wyraźną okresowością, która odzwierciedla

cykl pracy serca. Podczas gdy sygnały EEG zawierają informacje o procesach neurofizjologicznych, sygnały EKG są bezpośrednio związane z mechaniką pracy mięśnia sercowego. Z tego względu sygnały EEG należą do najbardziej złożonych sygnałów biologicznych, co czyni ich analizę niezwykle trudnym zadaniem.



Rysunek 3.6: Przykładowy sygnał EEG z kanału FCz rejestrowany przez 20 sekund. z wydzielonymi klasami 0, 1, 2.



Rysunek 3.7: Przykładowy sygnał ECG rejestrowany przez 20 sekund z częstotliwością próbkowania 160 hz. przy użyciu biblioteki python neurokit2

3.3. Metryki statystyczne danych

Zestawienie metryk statystycznych 3.2 pozwala wstępnie zinterpretować różnice w aktywności EEG pomiędzy stanem spoczynku (klasa 0) a dwoma klasami ruchu (1 i 2). W kontekście dekodowania ruchów z sygnału EEG warto zwrócić uwagę na kilka aspektów:

- **Średnia i mediana**
 - Dla stanu spoczynku (klasa 0) średnia wartość sygnału jest ujemna (ok. $-5.25\mu V$) i mediana również ($-5\mu V$), co wskazuje, że potencjał w tej klasie zwykle utrzymuje się poniżej zera.

Tablica 3.2: Metryki statystyczne dla sygnału EEG z kanału FCz (wartości w μV) oraz dla poszczególnych klas.

Metryka	Cały sygnał	Klasa 0	Klasa 1	Klasa 2
Liczba próbek	118320	59280	30832	28208
Maksimum	390.0	390.0	376.0	359.0
Minimum	-338.0	-338.0	-304.0	-278.0
Amplituda	728.0	728.0	680.0	637.0
Mediana	-4.0	-5.0	-1.0	-5.0
Maksimum wartości bezwzględnej	390.0	390.0	376.0	359.0
Średnia	-3.5714	-5.2485	2.7959	-7.0066
Średnia wartości bezwzględnych	45.1493	45.5618	45.1973	44.2300
Wariancja	4083.02	4134.47	4177.00	3810.15
Odchylenie standardowe	63.8985	64.2999	64.6297	61.7264
Energia	484.6e6	246.7e6	129.0e6	108.9e6
Energia sygnału wycentrowanego	483.1e6	245.1e6	128.8e6	107.5e6
Kurtoza	2.6358	2.6572	2.7884	2.2596
Skośność	0.3120	0.2355	0.5454	0.1892
Moment 5. rzędu	6452.7e6	5414.8e6	10084.6e6	4081.3e6
Moment 6. rzędu	4290.7e9	4545.4e9	4760.0e9	2930.3e9
Moment 7. rzędu	530.5e12	479.4e12	762.7e12	323.0e12
Moment 8. rzędu	2.79e17	3.06e17	3.11e17	1.62e17
Moment 9. rzędu	4.89e19	4.77e19	6.50e19	2.76e19
Moment 10. rzędu	2.26e22	2.57e22	2.46e22	1.14e22
Entropia Shannona	10.5300	9.8431	9.1462	9.1351

- W klasie 1 (ruch) średnia jest dodatnia ($2.80\mu V$) i mediana bliska zeru ($-1\mu V$), co sugeruje częstsze wychylenia w kierunku wartości dodatnich.

- W klasie 2 (drugi typ ruchu) średnia jest najbardziej ujemna ($-7.01\mu V$), a mediana wynosi $-5\mu V$, czyli wyraźnie utrzymuje się w dolnym zakresie rozkładu.

- **Amplituda, wariancja i odchylenie standardowe**

- Największą amplitudę (różnicę między maksymum a minimum) obserwujemy w klasie 0 ($728\mu V$) – czyli w stanie spoczynku. Klasy ruchowe mają nieco mniejszy rozrzut (klasa 1: $680\mu V$, klasa 2: $637\mu V$).

- Wariancja i odchylenie standardowe są najwyższe dla klasy 1 (odpowiednio $4177\mu V^2$ i $64.63\mu V$), a najniższe dla klasy 2 ($3810\mu V^2$ i $61.73\mu V$). Klasa 0 plasuje się między nimi ($4134.47\mu V^2$ oraz $64.30\mu V$).

- **Energia sygnału i energia wycentrowana**

- Największa energia całkowita i wycentrowana przypada na stan spoczynku (klasa 0). Może to wskazywać na silną składową rytmów tła (np. alfa, beta).
- Klasa ruchowa 1 i 2 mają wyraźnie mniejszą energię całkowitą, co sugeruje krótsze epizody sygnału o większej amplitudzie lub inną charakterystykę częstotliwościową.

- **Kurtoza i skośność**

klasy mają kurtozę poniżej 3, co oznacza, że rozkłady są raczej „spłaszczone” w porównaniu do rozkładu normalnego. Skośność jest dodatnia dla wszystkich klas, ale najwyższa dla klasy 1 (0.5454), co sugeruje, że rozkład próbek sygnału w ruchu 1 ma wydłużony „ogon” w kierunku wartości dodatnich.

- **Entropia Shannona**

- Najwyższą entropię wykazuje stan spoczynku (9.8431), co może wskazywać na większą różnorodność rytmów i potencjalów.
- Klasa 1 i 2 mają niższe wartości entropii (9.15), co może świadczyć o bardziej specyficznej aktywności związanej z planowaniem/wykonywaniem ruchu.

- **Implikacje dla dekodowania EEG**

- Różnice w średniej, rozkładach (skośność i kurtoza), a także w zmienności sygnału (wariancja, odchylenie standardowe) sugerują, że poszczególne stany (spoczynek vs. różne typy ruchu) mogą mieć rozróżnialne cechy w dziedzinie czasowej.
- Entropia i momenty wyższych rzędów mogą pomóc w klasyfikacji, umożliwiając rozróżnienie klas przez algorytmy uczenia maszynowego.

Podsumowując, metryki statystyczne wskazują, że sygnał w stanie spoczynku może być bardziej „rozproszony” i złożony (wyższa entropia, największa amplituda), natomiast w klasach ruchu (1 i 2) zaobserwować można różnice w średniej, odchyleniu standardowym czy skośności. Analiza tych parametrów jest przydatna w projektowaniu algorytmów dekodujących intencję ruchu z sygnałów EEG.

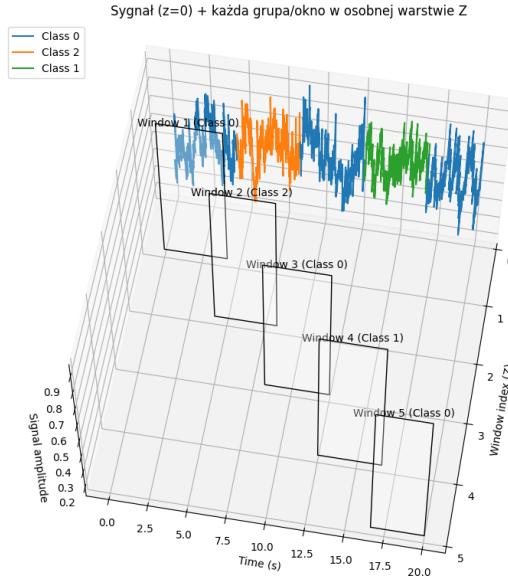
4. Podziału danych

Podział danych obejmuje dwa etapy: podział na próbki oraz na zbiory treningowe i testowe. Najpierw dane dzielone są na mniejsze sekwencje czasowe (próbki) przy użyciu różnych metod segmentacji. Następnie próbki te są rozdzielane na zbiory treningowy i testowy, z zachowaniem proporcji klas, co zapewnia reprezentatywność i poprawność oceny modelu.

4.1. Metody podziału danych na próbki

Tak jak w literaturze [2, 11] zdecydowano się na podział danych na krótkie sekwencje czasowe, lecz można rozważyć 3 różne podejścia poniżej przedstawiono opis oraz wizualizację każdej z tych metod.

Pełne okna na całą klasę W tej metodzie pełne 4-sekundowe sekwencje są traktowane jako pojedyncze próbki, bez fragmentacji. Liczba próbek w zbiorze treningowym i testowym odpowiada liczbie pełnych okien czasowych w danych. Ta metoda pozwala na utrzymanie stałej struktury danych, co ułatwia interpretację wyników. Przykład wizualizacji tej metody przedstawiono na rysunku 4.8.



Rysunek 4.8: Podział na pełne okna (bez nak adania). (Rysunek w asny)

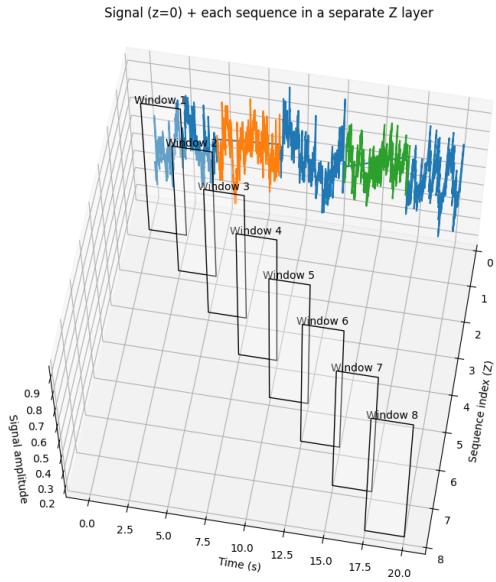
Okna o dowolnej d ugo ci bez nak adania si  Dane dzielone s a na okna o określonej d ugo ci sekwencji (np. `sequence_length = 2 s`) . Ka de okno jest niezale ne i nast puje bezpo rednio po poprzednim. Zwi kszenie liczby kr tszych okien pozwala na uzyskanie wi kszej liczby próbek w zbiorze treningowym, co mo e poprawi  zdolno ci generalizacji modelu. Przyk ad wizualizacji tej metody przedstawiono na rysunku 4.9.

Okna o dowolnej d ugo ci z nak adaniem si  W tej metodzie okna o d ugo ci sekwencji `sequence_length` nachodz  na siebie. Dla zobrazowania, przesuni cie `overlap = 50%` powoduje,  e ka de kolejne okno rozpoczyna si  w po owie poprzedniego. Dzi ki temu uzyskuje si  wi ksza liczb  sekwencji, kosztem mniejszej r  norodno ci danych. Przyk ad wizualizacji tej metody przedstawiono na rysunku 4.10.

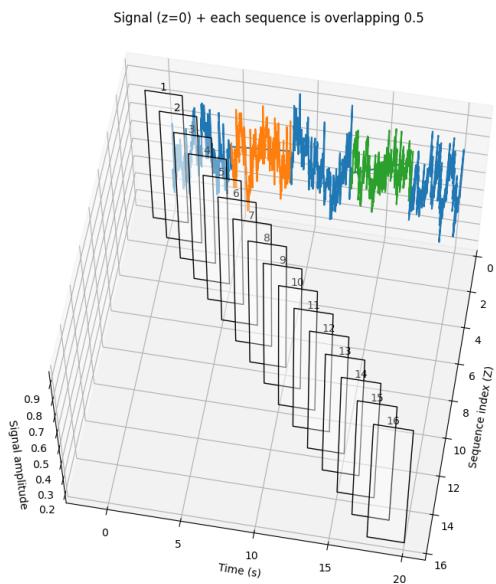
4.2. Podzia  zbioru danych na zbiory treningowy i walidacyjny

W ramach eksperymentu zdecydowano si  na podzia  danych na dwa zbiory: treningowy i testowy. Pomimo  e w wielu przypadkach stosuje si  dodatkowy zbi r walidacyjny, w tym badaniu ograniczono si  do dw ch zbiorów w celu oszcz dno ci czasu. Zbi r treningowy stanowi  80% danych, a zbi r walidacyjny 20% , przy zachowaniu proporcji klas, co zapewni o danych w obu zestawach.

Do bada  w tym eksperymencie wybrano metod  dzielenia na próbki z u yciem okien o dowolnej d ugo ci bez nak adania. Wyb r ten podyktowany by  mo liwo ci  elastycznego dopasowania d ugo ci okien.



Rysunek 4.9: Okna o dowolnej długości (bez nakładania). (Rysunek własny)



Rysunek 4.10: Okna o dowolnej długości (z nakładaniem 50%). (Rysunek własny)

gości sekwencji, co pozwala na dokładniejsze zbadanie wpływu tego parametru na wyniki modelu. Metoda z pełnymi oknami dla całych klas została odrzucona, gdyż nie umożliwia dostosowania długości sekwencji. Z kolei metoda z oknami o dowolnej długości z nakładaniem została wykluczona ze względu na generowanie dużej liczby próbek, co przy dostępnych zasobach obliczeniowych znaczaco wydłużałoby czas szkolenia modeli.

5. Przetwarzanie danych EEG do eksperymentu

Przetwarzanie sygnałów EEG przed szkoleniem modeli wymaga zastosowania odpowiednich technik wstępnej obróbki. W tej pracy skupiamy się na dwóch kluczowych etapach: normalizacji oraz transformacji falkowej.

5.1. Normalizacja sygnałów EEG

5.2. Normalizacja sygnałów EEG

W celu zapewnienia spójności danych i poprawy efektywności modeli uczenia maszynowego, sygnały EEG zostały poddane normalizacji min-max. Normalizacja ta jest wykonywana osobno dla każdego kanału w każdym eksperymencie, co pozwala na zachowanie względnych różnic między wartościami sygnału w obrębie jednego kanału, jednocześnie eliminując różnice w amplitudzie między kanałami i eksperymentami.

Wzór na normalizację min-max dla każdego kanału w danym eksperymencie przedstawiono poniżej:

$$X_{\text{norm}} = \frac{X - X_{\min}}{X_{\max} - X_{\min}}$$

Gdzie:

X – oryginalny sygnał EEG,

X_{\min} – minimalna wartość sygnału dla danego kanału w eksperymencie,

X_{\max} – maksymalna wartość sygnału dla danego kanału w eksperymencie,

X_{norm} – znormalizowany sygnał EEG, przyjmujący wartości z zakresu [0, 1].

Normalizacja min-max jest szczególnie przydatna w analizie sygnałów EEG, ponieważ zachowuje względne różnice między wartościami sygnału, jednocześnie ograniczając je do ustalonego zakresu. Przykłady zastosowania tego podejścia można znaleźć w literaturze, np. w [11], gdzie wykazano, że normalizacja sygnałów EEG istotnie poprawia jakość klasyfikacji.

Przykłady zastosowania tego podejścia można znaleźć w literaturze, np. w [11], gdzie wykazano, że normalizacja sygnałów EEG istotnie poprawia jakość klasyfikacji.

5.3. Transformacja falkowa

Transformacja falkowa (ang. *Wavelet Transform*, WT) jest techniką analizy sygnałów, która umożliwia dekompozycję danych na składowe o różnej skali. Jest to szczególnie użyteczne w analizie sygnałów EEG, które charakteryzują się zmiennością zarówno w czasie, jak i w częstotliwości.

5.3.1. Wzór na ciągłą transformatę falkową

Ciągła transformata falkowa sygnału $x(t)$ jest definiowana jako:

$$W_x(a, b) = \frac{1}{\sqrt{|a|}} \int_{-\infty}^{\infty} x(t) \psi^* \left(\frac{t-b}{a} \right) dt$$

Gdzie:

$W_x(a, b)$ – współczynnik transformaty falkowej dla skali a i przesunięcia b ,

$x(t)$ – analizowany sygnał,

$\psi(t)$ – funkcja falki (tzw. falka matka),

$\psi^*(t)$ – sprzężenie zespolone falki $\psi(t)$,

a – parametr skali (rozciągnięcie lub skurczenie falki),

b – parametr przesunięcia (lokalizacja czasowa),

t – czas.

Wzór ten opisuje, jak sygnał $x(t)$ jest korelowany z falką $\psi(t)$ w różnych skalach i przesunięciach, co umożliwia analizę lokalnych cech sygnału w dziedzinie czas-częstotliwość. Szczegółowe omówienie transformaty falkowej oraz jej zastosowań można znaleźć w literaturze, np. w [1], gdzie przedstawiono podstawy teoretyczne transformaty falkowej oraz przykłady jej zastosowań w analizie sygnałów.

5.3.2. Typ falki

W analizie sygnałów EEG często stosuje się różne rodzaje falek, w zależności od charakterystyki sygnału i celu analizy. Dwa popularne typy falek to falka Morleta oraz falki Daubechies.

Falka Morleta Falka Morleta $\psi(t)$ jest sinusoidą modulowaną oknem Gaussa i jest zdefiniowana jako:

$$\psi(t) = \exp \left(-\frac{t^2}{2} \right) \cdot \exp(j\omega t),$$

gdzie:

j – jednostka urojona ($j^2 = -1$),

ω – częstotliwość centralna falki.

Falka Morleta jest szczególnie przydatna w analizie sygnałów EEG, ponieważ dobrze odwzorowuje składowe o charakterze oscylacyjnym, co jest typowe dla sygnałów mózgowych.

Falki Daubechies Falki Daubechies to rodzina falek ortogonalnych, które są szeroko stosowane w analizie sygnałów ze względu na ich dobre właściwości czasowo-częstotliwościowe. Falki te charakteryzują się zwartym nośnikiem (są niezerowe tylko w skończonym przedziale czasu) oraz różną liczbą momentów znikających, co pozwala na efektywną dekompozycję sygnałów. W tej pracy zastosowano falę Daubechies 4 ($db4$), która jest jedną z najczęściej wykorzystywanych falek w analizie EEG. Falka $db4$ dobrze odwzorowuje szybkie zmiany sygnałów, co czyni ją odpowiednią dla danych EEG.

5.3.3. Wzór na falkę Daubechies 4 (db4)

Falka Daubechies 4 jest definiowana przez swoje współczynniki skalujące i falkowe. Współczynniki skalujące dla falki db4 są następujące:

$$\begin{aligned} h_0 &= \frac{1 + \sqrt{3}}{4\sqrt{2}}, \\ h_1 &= \frac{3 + \sqrt{3}}{4\sqrt{2}}, \\ h_2 &= \frac{3 - \sqrt{3}}{4\sqrt{2}}, \\ h_3 &= \frac{1 - \sqrt{3}}{4\sqrt{2}}, \end{aligned}$$

gdzie h_0, h_1, h_2, h_3 to współczynniki skalujące, a współczynniki falkowe g_k są obliczane jako:

$$g_k = (-1)^k h_{3-k} \quad \text{dla } k = 0, 1, 2, 3.$$

Funkcja skalująca $\phi(t)$ i falkowa $\psi(t)$ są definiowane rekurencyjnie poprzez te współczynniki:

$$\phi(t) = \sqrt{2} \sum_{k=0}^3 h_k \phi(2t-k),$$

$$\psi(t) = \sqrt{2} \sum_{k=0}^3 g_k \phi(2t-k).$$

Te równania opisują, jak funkcja skalująca i falkowa są generowane na podstawie współczynników h_k i g_k [4].

5.3.4. Zastosowanie falki Daubechies w analizie sygnałów

Falki Daubechies są szeroko stosowane w analizie sygnałów, szczególnie w kontekście dyskretnej transformacji falkowej (DWT). Są one wykorzystywane do dekompozycji sygnałów na składowe o różnej skali, co jest szczególnie przydatne w analizie sygnałów EEG, kompresji danych oraz usuwaniu szumów [7].

5.3.5. Zastosowanie transformacji falkowej

Transformacja falkowa jest wykorzystywana wyłącznie w modelach, które integrują splotowe sieci neuronowe (CNN). W artykule [10] wykazano, że transformacja falkowa w połączeniu z CNN znacząco poprawia dokładność klasyfikacji sygnałów EEG. CNN potrafią efektywnie uczyć się wzorców na podstawie cech czasowo-częstotliwościowych uzyskanych dzięki transformacji falkowej.

5.4. Podsumowanie

Zaproponowana metodologia przetwarzania danych EEG obejmuje normalizację danych dla każdego eksperymentu z osobna oraz transformację falkową z wykorzystaniem falki Daubechies 4. Transformacja falkowa umożliwia efektywne wydobycie cech czasowo-częstotliwościowych, co jest kluczowe w przypadku modeli opartych na sieciach spłotowych. Dzięki takiemu podejściu możliwe jest uzyskanie wysokiej dokładności klasyfikacji sygnałów EEG w ramach eksperymentów naukowych.

6. Wybór modeli oraz ich struktur do eksperymentu

W celu przeprowadzenia eksperymentów wybrano modele odpowiednie do analizy danych sekwencyjnych. Jednym z kluczowych zastosowanych modeli jest Long Short-Term Memory (LSTM), opisany szczegółowo w niniejszej sekcji.

6.1. Long Short-Term Memory (LSTM)

Model LSTM jest rozwinięciem klasycznej sieci rekurencyjnej (RNN), zaprojektowanym w celu rozwiązania problemów związanych z gradientem znikającym i eksplodującym. Problemy te pojawiają się w klasycznych RNN podczas trenowania sieci na długich sekwencjach danych, prowadząc do trudności w propagowaniu informacji przez wiele kroków czasowych. Sieci RNN mają trudności z uchwyceniem długoterminowych zależności, co ogranicza ich efektywność w zadaniach takich jak analiza szeregów czasowych czy przetwarzanie języka naturalnego [6].

6.1.1. Problemy klasycznych RNN

Gradienty znikające pojawiają się, gdy wartości gradientów stają się coraz mniejsze podczas propagacji wstecznej, przez co wagi nie są efektywnie aktualizowane. Gradienty eksplodujące to odwrotny problem, w którym wartości gradientów rosną wykładniczo, co prowadzi do niestabilności procesu uczenia [12].

LSTM został zaprojektowany, aby zachować długoterminowe informacje w stanie ukrytym, umożliwiając efektywne modelowanie zależności w długich sekwencjach danych.

6.1.2. Architektura komórki LSTM

Podstawowym elementem sieci LSTM jest pojedyncza komórka LSTM (rysunek 6.11), która składa się z trzech bramek kontrolujących przepływ informacji:

- **Bramka zapominania (forget gate):** Decyduje, które informacje z poprzedniego stanu komórki należy usunąć. Wzór:

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f)$$

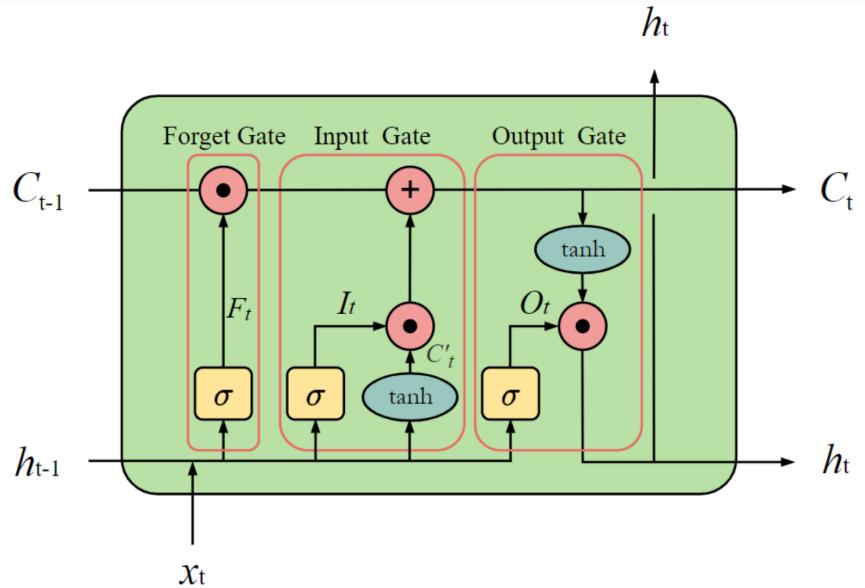
gdzie f_t to wartość bramki zapominania, σ to funkcja sigmoidalna, W_f to macierz wag, h_{t-1} to poprzedni stan ukryty, x_t to dane wejściowe, a b_f to wektor przesunięcia.

- **Bramka wejścia (input gate):** Odpowiada za aktualizację stanu komórki poprzez dodanie nowych informacji. Wzory:

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i),$$

$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C),$$

gdzie i_t to wartość bramki wejścia, a \tilde{C}_t to nowe kandydatury informacji, które mogą zostać zapisane.



Rysunek 6.11: Schemat komórki LSTM przedstawiający przepływ informacji oraz podział na bramki zapomnienia, wejścia i wyjścia [3].

- **Bramka wyjścia (output gate):** Określa, które informacje zaktualizowanego stanu komórki są wykorzystywane do generowania stanu ukrytego. Wzory:

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o),$$

$$h_t = o_t \cdot \tanh(C_t),$$

gdzie h_t to stan ukryty w bieżącym kroku czasowym.

Aktualizacja stanu komórki C_t przebiega zgodnie ze wzorem:

$$C_t = f_t \cdot C_{t-1} + i_t \cdot \tilde{C}_t.$$

6.1.3. Przewagi LSTM nad klasycznymi RNN

Dzięki zastosowaniu mechanizmów bramek, LSTM może efektywnie przechowywać informacje przez długi czas, eliminując problemy gradientów znikających i eksplodujących. LSTM znalazł zastosowanie w wielu dziedzinach, takich jak:

- przetwarzanie języka naturalnego (tłumaczenia maszynowe, analiza sentymentu),
- rozpoznawanie mowy,
- analiza szeregow czasowych (prognozowanie finansowe, analiza EEG).

Jak wskazuje Sherstinsky [12], użycie LSTM w zadaniach sekwencyjnych jest kluczowe dla skutecznego modelowania zależności czasowych.

6.1.4. Podsumowanie

LSTM jest wydajnym narzędziem do analizy danych sekwencyjnych, eliminując ograniczenia klasycznych RNN. Dzięki mechanizmom bramek i możliwości długoterminowego przechowywania informacji, LSTM jest obecnie standardem w wielu zastosowaniach związanych z przetwarzaniem danych czasowych.

6.2. CNN-LSTM: Hybrydowe podejście do analizy EEG

W ramach eksperymentu zastosowano model hybrydowy CNN-LSTM, łączący możliwości sieci splotowych (CNN) i długiej pamięci krótkoterminowej (LSTM). Model ten został zaprojektowany w celu efektywnej klasyfikacji sygnałów EEG, umożliwiając zarówno ekstrakcję cech przestrzennych, jak i modelowanie zależności czasowych. W literaturze wykazano, że takie podejście doskonale nadaje się do zadań związanych z analizą wyobrażonych ruchów (ang. motor imagery, MI) [2].

6.2.1. Architektura modelu

Struktura CNN-LSTM obejmuje:

- Warstwy CNN do ekstrakcji cech przestrzennych, takich jak rozkład elektrod i lokalne wzorce aktywności mózgowej.
- Warstwy LSTM, które modelują zależności czasowe w danych EEG, uwzględniając sekwencyjny charakter sygnałów.

Prezentowana architektura łączy kluczowe komponenty obu typów sieci, umożliwiając modelowanie zarówno statycznych, jak i dynamicznych cech danych czasoprzestrzennych.

6.2.2. Przewagi CNN-LSTM

- **Modelowanie czasoprzestrzenne:** Hybrydowe podejście pozwala na uchwycenie zarówno cech przestrzennych, jak i czasowych, co jest kluczowe w analizie sygnałów EEG związanych z ruchem wyobrażonym.
- **Efektywność w klasyfikacji:** Model CNN-LSTM osiąga wysoką dokładność klasyfikacji (98.39%), co potwierdza skuteczność w integracji obu podejść [2].

6.2.3. Wady CNN-LSTM

- **Złożoność obliczeniowa:** Dodanie warstw LSTM znaczaco zwiększa liczbę parametrów i wydłuża czas treningu w porównaniu do czystych modeli CNN.
- **Niższa dokładność niż w CNN:** W porównaniu do modelu CNN, który osiągnął dokładność 99.86%, model CNN-LSTM charakteryzuje się nieco niższą precyzją, co wskazuje na potencjalny kompromis między złożonością modelu a jego wydajnością.

6.2.4. Zastosowanie modelu w analizie EEG

Model CNN-LSTM został wykorzystany w badaniach nad dekodowaniem intencji ruchowych w kontekście interfejsów mózg-komputer (BCI). Dzięki zdolności do integracji cech przestrzennych i

czasowych, model ten stanowi odpowiednie narzędzie do analizy dynamicznych wzorców aktywności mózgowej, umożliwiając precyzyjną klasyfikację wyobrażonych ruchów. W kontekście zadań MI, podejście CNN-LSTM oferuje potencjalne korzyści w zastosowaniach takich jak robotyka wspomagająca czy zaawansowane systemy sterowania.

6.2.5. Podsumowanie

Model CNN-LSTM, choć bardziej złożony niż tradycyjne CNN, oferuje unikalne korzyści w kontekście analizy danych czasoprzestrzennych. Jego zastosowanie w klasyfikacji sygnałów EEG podkreśla potencjał w precyzyjnym dekodowaniu intencji ruchowych, jednak należy uwzględnić kompromis między dokładnością a złożonością obliczeniową.

7. Eksperyment dla nieodseparowanych osób w zbiorze testowym i treningowym

Celem przyświecającym temu eksperymentowi było jak największe uproszczenie analizy dla wybranego zbioru danych, co umożliwiło szybkie i efektywne badanie działania modelu przy różnych konfiguracjach hiperparametrów. Eksperyment miał również na celu zrozumienie, jak model zachowuje się w warunkach ograniczonej ilości danych i klas, aby w przyszłości móc stopniowo zwiększać rozmiar zbioru treningowego, liczbę zadań oraz uczestników, co pozwoli na bardziej kompleksową ocenę działania modelu.

7.1. Wybór osób oraz zadania do eksperymentu

Do eksperymentu wybrano osoby o numerach 1, 2, 8 oraz 9 z sposobem próbkowania okien o dowolnej długości (bez nakładania) 4.9. Ograniczenie liczby uczestników było świadomą decyzją mającą na celu redukcję danych, co z kolei pozwoliło na przyspieszenie procesu testowania różnych kombinacji hiperparametrów. Eksperyment został zaprojektowany tak, aby skupić się na szybkim iteracyjnym badaniu przestrzeni hiperparametrów oraz wstępnej ocenie skuteczności modelu w tym kontekście.

Wybrano zadanie numer 3, które polega na zaciskaniu lewej lub prawej pięści. Decyzja o wyborze tego zadania była motywowana potrzebą dodatkowego uproszczenia eksperymentu, co pozwoliło na szybszą analizę działania modelu. Rzeczywiście zaciskanie pięści generuje wyraźniejsze sygnały w danych EEG niż ruchy wyobrażone, dzięki czemu jest bardziej czytelne i spójne dla modeli dekodujące dane encefalograficzne. Ponadto, wybór tego zadania opierał się na jego lepszym zbadaniu w literaturze oraz potencjalnie wyższej jakości danych w porównaniu do bardziej złożonych zadań.

Wybranie czterech osób oraz jednego zadania przełożyło się na 78 960 pojedynczych odczytów z 64 czujników EEG. Ta liczba stanowi wystarczający zbiór do przeprowadzenia wstępnych eksperymentów z hiperparametrami, jednocześnie pozostając na poziomie umożliwiającym szybkie iteracje.

Podsumowując, ograniczenie liczby uczestników i wybranie prostego zadania pozwoliło na szybkie iteracje w badaniu hiperparametrów, jednocześnie tworząc solidną podstawę dla przyszłych eksperymentów z większą liczbą danych, klas oraz zadań.

7.2. Normalizacja danych

Normalizacja danych została przeprowadzona w celu ujednolicenia skali odczytów EEG i poprawy zbieżności podczas procesu uczenia modelu. Zastosowano metodę min-max, która przekształca wartości do zakresu $[0, 1]$. Proces normalizacji został przeprowadzony oddzielnie dla każdego zadania i dla każdej osoby osobno. Oznacza to, że dane każdej osoby były normalizowane niezależnie, co pozwoliło zachować specyfikę sygnałów pochodzących od różnych uczestników eksperymentu.

Taka strategia normalizacji umożliwiła modelowi skuteczniejsze przetwarzanie danych o różnej skali i charakterystyce, zwiększaając spójność wyników w trakcie eksperymentu.

7.3. Przestrzeń wyszukiwania hiperparametrów

W celu zbadania optymalnych wartości hiperparametrów zdefiniowano przestrzeń wyszukiwania 7.3 obejmującą różne zakresy i typy danych. Poniżej przedstawiono szczegóły w formie tabeli:

Hiperparametr	Typ danych	Zakres
Współczynnik uczenia	Logarytmiczny	$[1 \times 10^{-5}, 1 \times 10^{-1}]$
Wielkość wsadu	Pojedyncze wybory	$\{8, 16, 32, 64, 128\}$
Rozmiar warstwy ukrytych (hidden size)	Logarytmiczny	$[100, 10\,000]$
Liczba warstw	Liniowy	$[1, 4]$
Współczynnik wyłączenia	Liniowy	$[0, 0.4]$
Długość sekwencji	Liniowy	$[0, 800]$

Tablica 7.3: Przestrzeń wyszukiwania hiperparametrów

Do przeszukiwania przestrzeni hiperparametrów wykorzystano metodę *grid search*, co pozwoliło na systematyczne badanie wpływu poszczególnych parametrów na działanie modelu. Zakresy zostały dobrane w celu zbalansowania szczegółowości i szerokości eksploracji przestrzeni hiperparametrów, uwzględniając zarówno małe, jak i duże zmiany wpływające na wyniki.

7.4. Konfiguracja modelu oraz procesu uczenia

Proces uczenia modelu został skonfigurowany z wykorzystaniem hiperparametrów przedstawionych w tabeli 7.3. Model składał się z warstwy *LSTM*, do której wyjścia z ostatniej komórki (*cell*) były podłączone do w pełni połączonej warstwy wyjściowej (*fully connected layer*). Architektura ta umożliwia przetwarzanie sekwencji danych i generowanie wyników klasyfikacji.

Szczegóły architektury:

- Warstwa LSTM: rozmiar warstwy ukrytej (*hidden size*) był jednym z hiperparametrów definiujących model.
- W pełni połączona warstwa: przyjmuje jako wejście wyjście z ostatniej komórki warstwy

LSTM, a jej wymiar jest określony przez *hidden size* oraz liczbę klas, czyli 3 (zaciskanie lewej pięści, zaciskanie prawej pięści, stan spoczynku).

Do optymalizacji modelu zastosowano algorytm Adam (*Adaptive Moment Estimation*) [9], opisany poniższymi wzorami:

$$\begin{aligned} m_t &= \beta_1 m_{t-1} + (1 - \beta_1) g_t, \\ v_t &= \beta_2 v_{t-1} + (1 - \beta_2) g_t^2, \\ \hat{m}_t &= \frac{m_t}{1 - \beta_1^t}, \\ \hat{v}_t &= \frac{v_t}{1 - \beta_2^t}, \\ \theta_t &= \theta_{t-1} - \eta \frac{\hat{m}_t}{\sqrt{\hat{v}_t} + \epsilon} \end{aligned}$$

gdzie:

g_t – gradient funkcji straty względem parametrów w kroku t,

m_t – estymacja średniej ruchomej gradientów w kroku t,

v_t – estymacja średniej ruchomej kwadratów gradientów w kroku t,

β_1 – współczynnik kontrolujący wpływ poprzednich wartości momentu pierwszego rzędu (domyślnie $\beta_1 = 0,9$),

β_2 – współczynnik kontrolujący wpływ poprzednich wartości momentu drugiego rzędu (domyślnie $\beta_2 = 0,999$),

\hat{m}_t – skorygowana estymacja średniej ruchomej gradientów w kroku t,

\hat{v}_t – skorygowana estymacja średniej ruchomej kwadratów gradientów w kroku t,

η – współczynnik uczenia (*learning rate*),

ϵ – mała stała zapobiegająca dzieleniu przez zero (domyślnie $\epsilon = 10^{-8}$),

θ_t – aktualne wartości parametrów modelu w kroku t,

θ_{t-1} – wartości parametrów modelu w poprzednim kroku ($t-1$).

Jako funkcję straty zastosowano funkcję krzyżowej entropii (*Cross-Entropy Loss*) [8], definiowaną jako:

$$\mathcal{L} = -\frac{1}{N} \sum_{i=1}^N \sum_{j=1}^C y_{ij} \cdot \log(\hat{y}_{ij})$$

gdzie:

N – liczba próbek w zbiorze danych,

C – liczba klas,

y_{ij} – rzeczywista etykieta dla próbki i i klasy j (wartość binarna: 0 lub 1),

\hat{y}_{ij} – przewidywane prawdopodobieństwo dla próbki i i klasy j.

Powyższa konfiguracja modelu i procesu uczenia zapewniała stabilność trenowania oraz możliwość efektywnego testowania różnych kombinacji hiperparametrów.

Taka konfiguracja modelu, oparta na warstwie *LSTM* i w pełni połączonej warstwie wyjściowej, zapewnia zdolność do analizy sekwencji danych EEG oraz dokładnej klasyfikacji wyników.

7.5. Struktura modelu

Model składał się z jednej warstwy *LSTM* o 128 ukrytych jednostkach (*hidden size*). Wyjścia z warstwy *LSTM* były przekazywane do w pełni połączonej warstwy (*fully connected layer*) z liczbą neuronów równą liczbie klas (2 klasy: otwieranie i zamykanie pięści). Do modelu zaimplementowano mechanizm regularizacji w postaci warstwy *dropout* z prawdopodobieństwem $p = 0.5$.

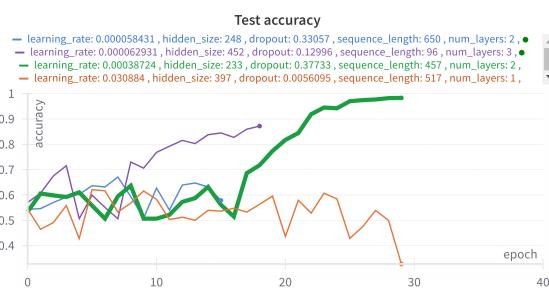
7.6. Wyniki eksperymentu

W ramach eksperymentu przeprowadzono 100 prób, z których każda testowała różne kombinacje hiperparametrów, zdefiniowane w tabeli 7.3 . Eksperymenty zostały wykonane przy użyciu optymalizacji hiperparametrów za pomocą narzędzia *Optuna* , co umożliwiło efektywne przeszukiwanie przestrzeni parametrów i identyfikację najlepszych konfiguracji dla modelu.

Dalsze wyniki eksperymentów, w celu poprawy czytelności, są przedstawiane na wykresach obejmujących najlepszą próbę, drugą najlepszą próbę, średni wynik oraz najgorszy wynik. Taka selekcja pozwala na dokładniejsze zrozumienie wpływu hiperparametrów na poszczególne metryki.

7.6.1. Dokładność (*Accuracy*)

Najlepsza próba osiągnęła wartość dokładności na poziomie 0.98331 , co wskazuje na bardzo wysoką skuteczność modelu w klasyfikacji danych (rys. 7.12).

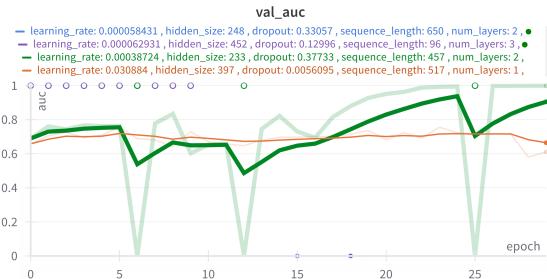


Rysunek 7.12: Wyniki dokładności (*Accuracy*) dla czterech wybranych prób

7.6.2. Pole pod krzywą ROC (*AUC*)

Najlepsza próba osiągnęła wartość *AUC* na poziomie 0.99915, co potwierdza zdolność modelu do doskonałego rozróżniania klas pozytywnych i negatywnych (rys. 7.13).

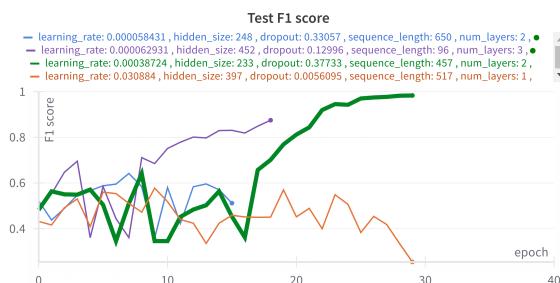
Należy jednak zauważać, że implementacja *AUC* prawdopodobnie zawierała błędy, ponieważ w niektórych przypadkach metryka ta nie zwracała wartości. Na wykresie można to zaobserwować jako brakujące dane, co wymaga dalszej analizy oraz potencjalnej poprawy obliczeń tej metryki.



Rysunek 7.13: Pole pod krzywą ROC (*AUC*) dla czterech wybranych prób

7.6.3. Miara F1 (*F1 Score*)

Miara F1 w najlepszej próbie osiągnęła wartość 0.98326%, co świadczy o znakomitym zbalansowaniu precyzji i czułości (rys. 7.14).



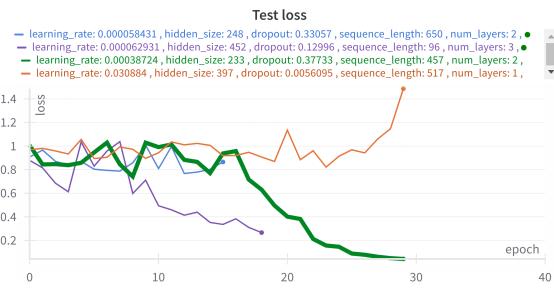
Rysunek 7.14: Miara F1 (*F1 Score*) dla czterech wybranych prób

7.6.4. Strata (*Loss*)

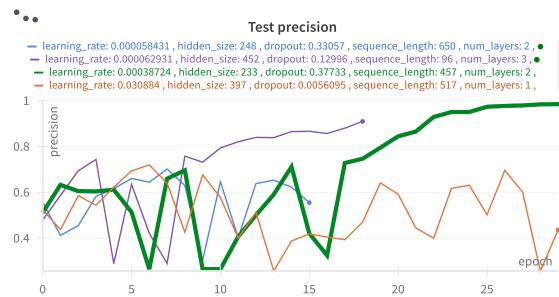
Najniższa strata w najlepszej próbie wyniosła 0,04264%, co oznacza bardzo dobре dopasowanie modelu do danych (rys. 7.15).

7.6.5. Precyzyja (*Precision*)

Precyzyja w najlepszej próbie wyniosła 0.98488%, co wskazuje na skuteczność modelu w unikaniu fałszywych pozytywów (rys. 7.16).



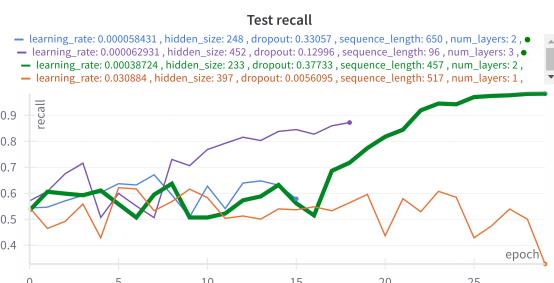
Rysunek 7.15: Strata (*Loss*) dla czterech wybranych prób



Rysunek 7.16: Precyzja (*Precision*) dla czterech wybranych prób

7.6.6. Czułość (*Recall*)

Czułość modelu w najlepszej próbie wyniosła 0.98328% , co potwierdza jego zdolność do wykrywania większości pozytywnych przypadków (rys. 7.17).



Rysunek 7.17: Czułość (*Recall*) dla czterech wybranych prób

7.6.7. Podsumowanie najlepszej konfiguracji hiperparametrów

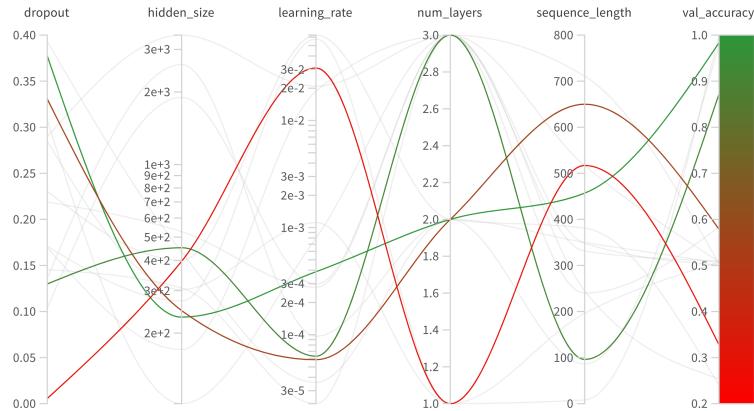
Najlepsza konfiguracja hiperparametrów, wybrana na podstawie najwyższej wartości pola pod krzywą ROC (*AUC*), została przedstawiona w tabeli 7.4.

Hiperparametr	Wartość
Learning rate	3.872e-4
Batch size	32
Rozmiar warstw ukrytych (hidden size)	233
Liczba warstw	3
Dropout	0.3773
Długość sekwencji	457

Tablica 7.4: Najlepsza konfiguracja hiperparametrów

7.6.8. Analiza korelacji hiperparametrów z dokładnością

Dla lepszego zrozumienia wpływu hiperparametrów na dokładność (*test_accuracy*), na rysunku 7.18 przedstawiono wykres typu parallel plot, który pokazuje zależności między wartościami hiperparametrów a wynikami modelu.



Rysunek 7.18: Korelacje hiperparametrów z dokładnością na wykresie typu (*parallel plot*)

7.7. Podsumowanie

W ramach pierwszego eksperymentu osiągnięto dokładność na poziomie 98% oraz pole pod krzywą ROC (*AUC*) na poziomie 99%, co można uznać za bardzo zadowalający wynik. Wyniki te pokazują potencjał zastosowanego podejścia oraz użytego zbioru danych do dalszych badań.

Niestety, należy zauważyc, że wynik może być niemiarodajny ze względu na ograniczoną liczbę próbek. Ponadto, zbiór testowy i walidacyjny zawierają dane pochodzące od tych samych osób, co, mimo rozdzielenia próbek, oznacza, że model nie został w pełni przetestowany pod kątem uniwersalności dla każdej osoby. W obecnej formie eksperyment nie tworzy modelu potencjalnie uniwersalnego, zdolnego do generalizacji dla nowych uczestników.

Wyniki te sugerują konieczność dalszych badań, w szczególności zwiększenia liczby próbek

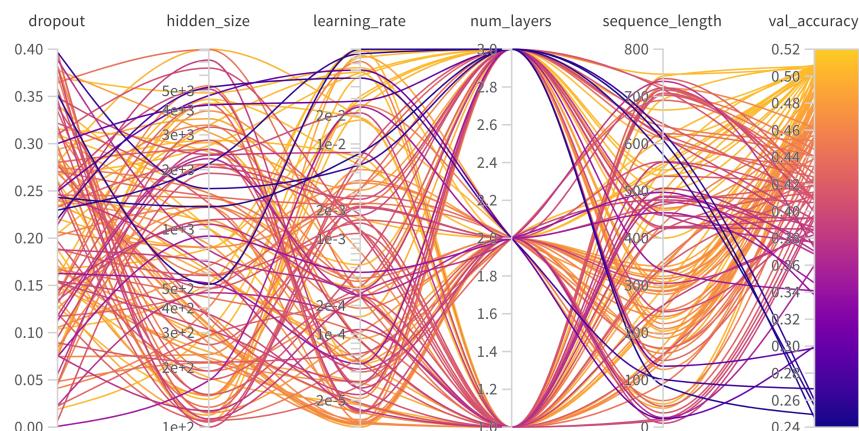
w zbiorze treningowym oraz testowym, a także uwzględnienia osób, które nie występują w zbiorze treningowym. Takie działania pozwolą na lepszą ocenę możliwości uogólniania modelu oraz dokładniejsze zrozumienie jego potencjału w kontekście bardziej zróżnicowanych danych.

8. Eksperyment dla odseparowanych osób w zbiorze testowym i treningowym

Celem tego eksperymentu było zbadanie zdolności modelu do uogólniania w warunkach, w których osoby w zbiorze treningowym i testowym są odseparowane. W zbiorze treningowym uwzględniono dane od osób o numerach 1, 2, 8, 9, 14, 17, 24 oraz 29, natomiast w zbiorze testowym znalazły się osoby o numerach 32 oraz 54. Pozostałe założenia eksperymentalne, takie jak wybór zadania, normalizacja danych, przestrzeń wyszukiwania hiperparametrów oraz struktura modelu i proces uczenia, były identyczne z opisanymi w sekcji 7.

8.1. Wyniki eksperymentu

Wyniki tego eksperymentu wskazują na trudności modelu w uogólnianiu danych od nowych osób. Żaden z testowanych konfiguracji nie osiągnął dokładności większej niż 51% , co świadczy o problemach z generalizacją w kontekście odseparowanych osób co obrzuca wykres równoległy hiperparametów i dokładności 8.19.



Rysunek 8.19: Koreacje hiperparametrów z dokładnością na wykresie typu *parallel plot* dla odseparowanych osób

8.2. Wnioski

Dane od nowych osób w zbiorze testowym znacznie różniły się od danych w zbiorze treningowym. Modele nie były w stanie dostatecznie dobrze uogólnić, co skutkowało niską dokładnością klasyfikacji nie przekraczającą 51% . Wyniki te sugerują potrzebę dalszej analizy, szczególnie w kontekście różnic

między indywidualnymi sygnałami EEG, oraz potencjalne zmiany w architekturze modelu lub strategii przetwarzania danych w przyszłych eksperymentach.

9. Podsumowanie

W niniejszej pracy przedstawiono wyniki eksperymentów dotyczących dekodowania intencji ruchowych na podstawie sygnałów EEG z wykorzystaniem modelu LSTM + **fully connected** (FC). W części przeglądowej omówiono różnorodne podejścia do klasyfikacji sygnałów EEG, w tym architektury typu CNN czy CNN-LSTM [2, 11], jednakże w ramach własnych badań skupiono się wyłącznie na modelu opartym na warstwie LSTM, do której wyjście zostało podłączone do w pełni połączonej warstwy końcowej (FC).

Wyniki dla osób nieodseparowanych w zbiorze treningowym i testowym

Pierwszy eksperyment zakładał, że te same osoby (uczestnicy badania EEG) występują zarówno w zbiorze treningowym, jak i testowym. W tej konfiguracji – po dopasowaniu odpowiednich hiperparametrów (m.in. *learning rate*, *hidden size*, *dropout* i długości sekwencji) – uzyskano bardzo wysokie wyniki klasyfikacji. Najlepsze modele osiągnęły dokładność na poziomie ok. 0,9833, a pole pod krzywą ROC (AUC) przyjmowało wartości sięgające nawet 0,99915 na zbiorze testowym. Tak wysoka skuteczność wskazuje, że zastosowany model LSTM potrafi skutecznie wyodrębniać wzorce związane z realnym lub wyobrażonym ruchem kończyn w sytuacji, gdy dane od tych samych osób pojawiają się już w fazie treningu.

Wyniki dla osób odseparowanych w zbiorze treningowym i testowym

Drugi eksperyment miał na celu zweryfikowanie, w jakim stopniu model jest w stanie uogólniać (generalizować) się na dane pochodzące od osób nieobecnych w zbiorze treningowym. W tym celu zbiór treningowy i testowy został podzielony tak, aby uczestnicy badania w zbiorze testowym byli całkowicie odseparowani od tych w zbiorze treningowym. Wyniki pokazały, iż model LSTM, w testowanej konfiguracji i bez dodatkowych technik adaptacyjnych, nie był w stanie przekroczyć dokładności ok. 0,51, co oznacza skuteczność zblżoną do losowej.

Wnioski główne

- 1) **Skuteczność modelu w przypadku nieodseparowanych danych** Uzyskane wysokie wyniki (dokładność powyżej 0.98 oraz AUC bliskie 1.0) wskazują, że model LSTM znakomicie identyfikuje specyficzne wzorce EEG w sytuacji, gdy w zbiorze treningowym znajdują się dane od tych samych osób, które później występują w zbiorze testowym.
- 2) **Brak uogólnienia na osoby spoza zbioru treningowego** W warunkach, gdy osoby ze zbioru testowego nie pojawiły się w zbiorze treningowym, dokładność modelu spada do

poziomu ok. 0.5. Fakt ten wskazuje na duże zróżnicowanie sygnałów EEG między różnymi uczestnikami oraz konieczność opracowania lub zastosowania metod pozwalających na adaptację modelu do nowych użytkowników.

- 3) **Perspektywy dla dalszych badań nad BCI** Zaprezentowane wyniki potwierdzają zaśadność wykorzystania rekurencyjnych sieci LSTM w zadaniach typu *Motor Imagery*. Jednocześnie podkreślają potrzebę wdrażania metod pozwalających na personalizację modelu bądź wykorzystanie transfer learningu, aby przełamać problem niskiej generalizacji na osoby nieobecne w fazie trenowania.

Dalsze kierunki rozwoju

- **Rozbudowa i zróżnicowanie zbioru danych:** Planuje się rozszerzyć liczbę uczestników badania EEG oraz włączyć dodatkowe zadania motoryczne lub wyobrażeniowe, co pozwoli na bardziej wiarygodną ocenę uogólniania modelu.
- **Personalizacja i transfer learning:** Jednym z kluczowych wyzwań w zastosowaniach *Brain-Computer Interface* (BCI) jest dostosowanie modeli do indywidualnych cech użytkownika. Wprowadzenie metod *transfer learningu* czy *few-shot learningu* może znaczco poprawić wyniki uzyskiwane dla nowych osób.
- **Porównanie z innymi modelami:** Chociaż w niniejszej pracy skupiono się wyłącznie na modelu LSTM + FC, przyszłe prace mogą uwzględniać również architektury hybrydowe (np. CNN-LSTM), które w literaturze nierzadko wykazują wysoką skuteczność w zadaniach klasyfikacji sygnałów EEG [2, 11].
- **Badanie stabilności i odporności na zakłócenia:** W praktycznych zastosowaniach BCI istotne jest, aby model pozostawał stabilny w obecności artefaktów (np. z ruchów gałek ocznych czy zakłóceń elektromagnetycznych). W związku z tym warto rozważyć mechanizmy dodatkowej filtracji sygnału i zaawansowane metody detekcji artefaktów.

Ostatecznie, uzyskane wyniki potwierdzają, iż sieci LSTM stanowią wartościowe narzędzie w dekodowaniu intencji ruchowych z EEG, zwłaszcza w warunkach, gdy dane uczące i testowe pochodzą od tych samych osób. Jednak w kontekście opracowywania uniwersalnych interfejsów mózg–komputer, zdolnych do natychmiastowego działania na nieznanych użytkownikach, konieczny jest dalszy rozwój metod adaptacyjnych i personalizowanych, które umożliwiają znaczące zwiększenie ogólnej skuteczności klasyfikacji.

Bibliografia

- [1] bibliotekanauki.pl. „Transformata falkowa wybranych sygnałów symulacyjnych jako narzędzie analizy”. W: *Biblioteka Nauki* (2023). URL: <https://bibliotekanauki.pl/articles/256150.pdf>.
- [2] N. Boutarfaia i in. „Deep Learning for EEG-Based Motor Imagery Classification: Towards Enhanced Human-Machine Interaction and Assistive Robotics”. W: *CEUR Workshop Proceedings*. 2023. URL: <https://api.semanticscholar.org/CorpusID:270258728>.
- [3] N. Chen. „Exploring the development and application of LSTM variants”. W: *Proceedings of the 4th International Conference on Signal Processing and Machine Learning*. 2024. DOI: 10.54254/2755-2721/53/20241288.
- [4] Ingrid Daubechies. *Ten Lectures on Wavelets*. Philadelphia, PA: SIAM, 1992.
- [5] A. L. Goldberger i in. „PhysioBank, PhysioToolkit, and PhysioNet: Components of a new research resource for complex physiologic signals”. W: *Circulation* 101.23 (2000), e215–e220. DOI: 10.1161/01.CIR.101.23.e215. URL: <https://physionet.org/content/eegmmidb/1.0.0/>.
- [6] S. Hochreiter i J. Schmidhuber. „Long Short-Term Memory”. W: *Neural Computation* 9.8 (1997), s. 1735–1780. DOI: 10.1162/neco.1997.9.8.1735. URL: <https://doi.org/10.1162/neco.1997.9.8.1735>.
- [7] Stéphane Mallat. *A Wavelet Tour of Signal Processing*. San Diego, CA: Academic Press, 1999.
- [8] PyTorch Developers. *torch.nn.CrossEntropyLoss — PyTorch 2.5 documentation*. Dostęp: 13 stycznia 2025. 2025. URL: <https://pytorch.org/docs/stable/generated/torch.nn.CrossEntropyLoss.html>.
- [9] PyTorch Developers. *torch.optim.Adam — PyTorch 2.5 documentation*. Dostęp: 13 stycznia 2025. 2025. URL: <https://pytorch.org/docs/stable/generated/torch.optim.Adam.html>.
- [10] S. Rajwal i S. Aggarwal. „Convolutional Neural Network-Based EEG Signal Analysis: A Systematic Review”. W: *Archives of Computational Methods in Engineering* 30 (2023), s. 3585–3615. DOI: 10.1007/s11831-023-09920-1. URL: <https://link.springer.com/article/10.1007/s11831-023-09920-1>.
- [11] K. Roots, Y. Muhammad i N. Muhammad. „Fusion Convolutional Neural Network for Cross-Subject EEG Motor Imagery Classification”. W: *Computers* 9.3 (2020), s. 72. DOI: 10.3390/computers9030072. URL: <https://www.mdpi.com/2073-431X/9/3/72>.
- [12] A. Sherstinsky. „Fundamentals of Recurrent Neural Network (RNN) and Long Short-Term Memory (LSTM) Network”. W: *Physica D: Nonlinear Phenomena* (2020). DOI: 10.1016/j.physd.2019.132306. arXiv: 1808.03314 [cs.LG]. URL: <https://doi.org/10.48550/arXiv.1808.03314>.

- [13] Leandro Moura Silva i in. „Localizing the Primary Motor Cortex of the Hand by the 10-5 and 10-20 Systems for Neurostimulation: An MRI Study”. W: *Clinical EEG and Neuroscience* 1 (2020). © EEG and Clinical Neuroscience Society (ECNS) 2020, s. 1–9. DOI: 10.1177/1550059420934590. URL: <https://journals.sagepub.com/doi/10.1177/1550059420934590>.
- [14] J. Wang, L. Bi i W. Fei. „Using Non-linear Dynamics of EEG Signals to Classify Primary Hand Movement Intent Under Opposite Hand Movement”. W: *Frontiers in Neurorobotics* 16 (2022), s. 845127. DOI: 10.3389/fnbot.2022.845127.