

# Raport z końcowego zadania projektowego

Tomasz Dróżdż, Maciej Markiewicz, Dawid Motyka

## 1 Eksploracyjna analiza danych

Zadanie było wykonywane na zbiorze *Room Occupancy Estimation Data Set*<sup>1</sup>. Zawiera on dane z sensorów znajdujących się w pokoju oraz liczbę osób przebywających w tym pokoju. Pomiary były wykonywane co 30 sekund, na przestrzeni kilku dni.

Atrybuty dostępne w zbiorze to wartości odczytywane z:

- czterech czujników światła,
- czterech czujników temperatury,
- czterech czujników dźwięku,
- dwóch czujników dwutlenku węgla,
- dwóch czujnika ruchu.

Plan pokoju w których przeprowadzane były pomiary przedstawiono na Rysunku 1. Czujniki temperatury, dźwięku oraz światła były rozmieszczone w pozycjach S1-S4, czujnik dwutlenku węgla w pozycji S5, a czujniki ruchu w S6 oraz S7.

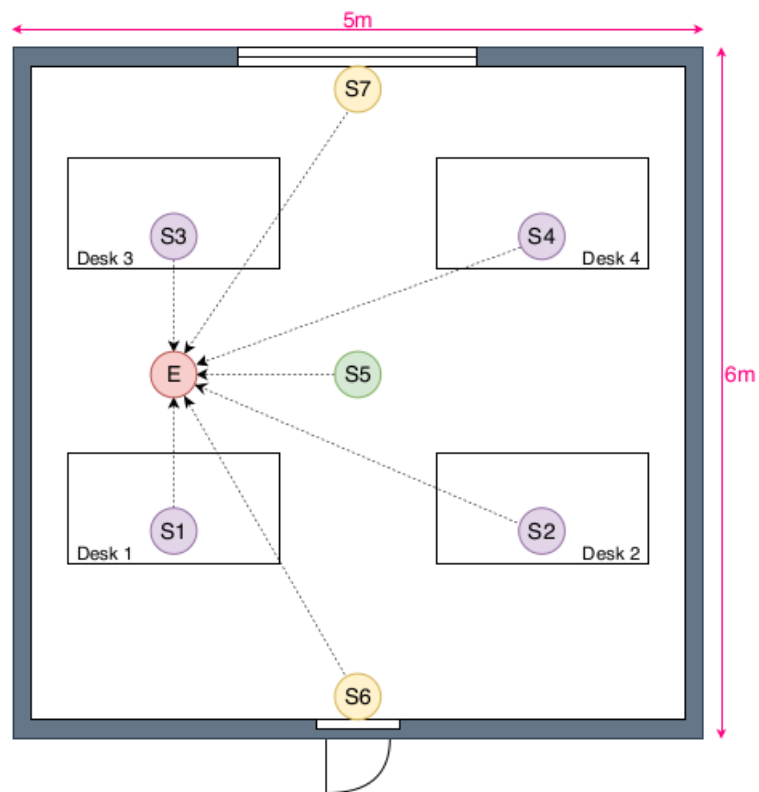
Liczba klas w zbiorze jest niezbalansowana, co przedstawia Tabela 1.

Klasa	Liczba instancji
0	8228
1	459
2	748
3	694

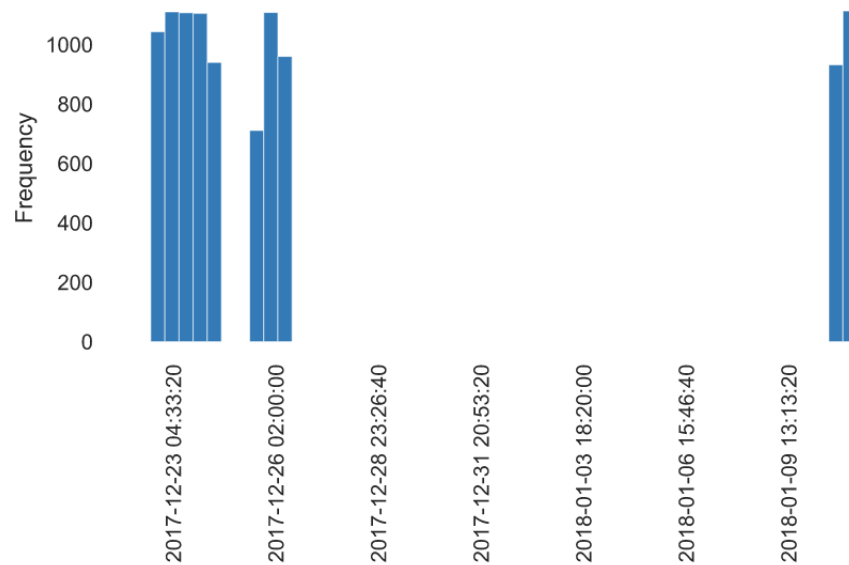
Tablica 1: Rozkład klas w zbiorze

Na zbiorze przeprowadzona została eksploracyjna analiza przy pomocy biblioteki *pandas-profiling*. Rysunek 2 przedstawia rozmieszczenia próbek w czasie. Można zauważyć długie przerwy w zbieraniu danych, co może utrudniać klasyfikację. Na Rysunku 3 pokazano współczynniki korelacji Spearmana dla całego zbioru. Można zauważyć duże korelacje większości cech z etykietą (Occupancy), jak również korelację cech między sobą. Szczególnie skorelowane są temperatury oraz światło. Warto mieć na uwadze, że korelacje temperatur ze względu na opóźnienia czasowe nie są liniowe.

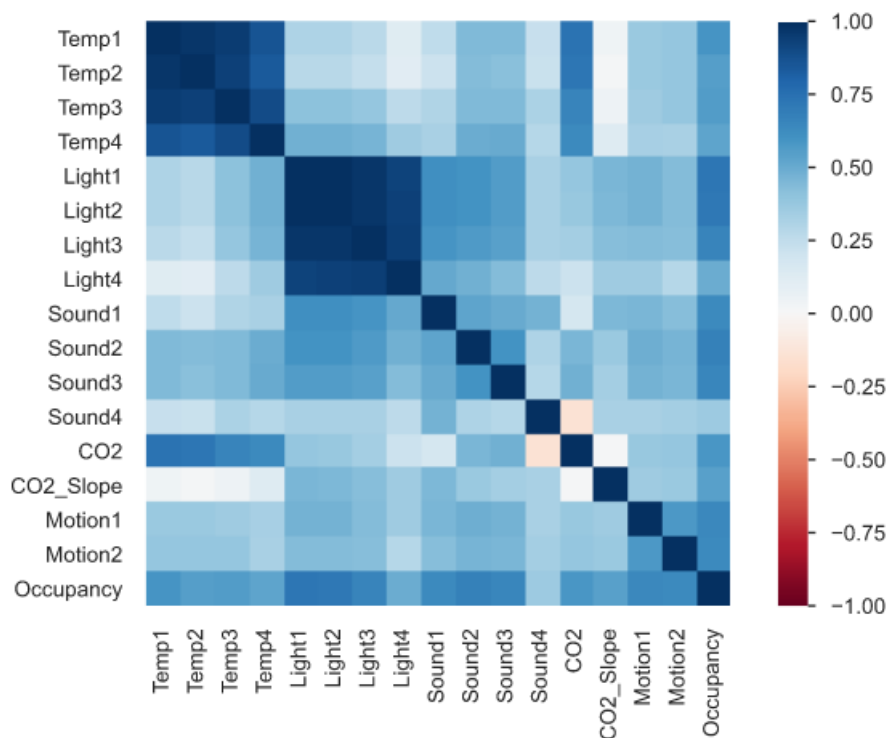
Na Rysunku 4 przedstawione zostały wartości poszczególnych atrybutów, wraz z liczbą osób przebywającą w pokoju w ciągu pierwszego dnia ze zbioru. Jak można zauważyć,



Rysunek 1: Plan pokoju z rozmieszczeniem czujników z których pobierane były dane.



Rysunek 2: Liczba próbek danych w czasie.



Rysunek 3: Współczynniki korelacji Spearmana dla cech oraz etykiety.

najbardziej skorelowane z ilością osób w pokoju są wartości odczytywane z czujników światła.

Potwierdzają to eksperymenty przeprowadzone na lasach losowych, gdzie wartości metryki F1 macro z wszystkimi atrybutami wynosi 0.98, a bez odczytów z czujników światła 0.36. Na podstawie tych przesłanek podjęta została decyzja o odrzuceniu czterech atrybutów z natężeniem światła. Poza tym, wszystkie atrybuty są skorelowane z liczbą osób przebywających w pokoju, jednak już nie tak jednoznacznie. Cechy takie jak temperatura, czy poziom dwutlenku węgla będą rosły i spadały, w zależności od liczby osób, z pewną inercją, co będzie utrudniało klasyfikację.

## 2 Modele

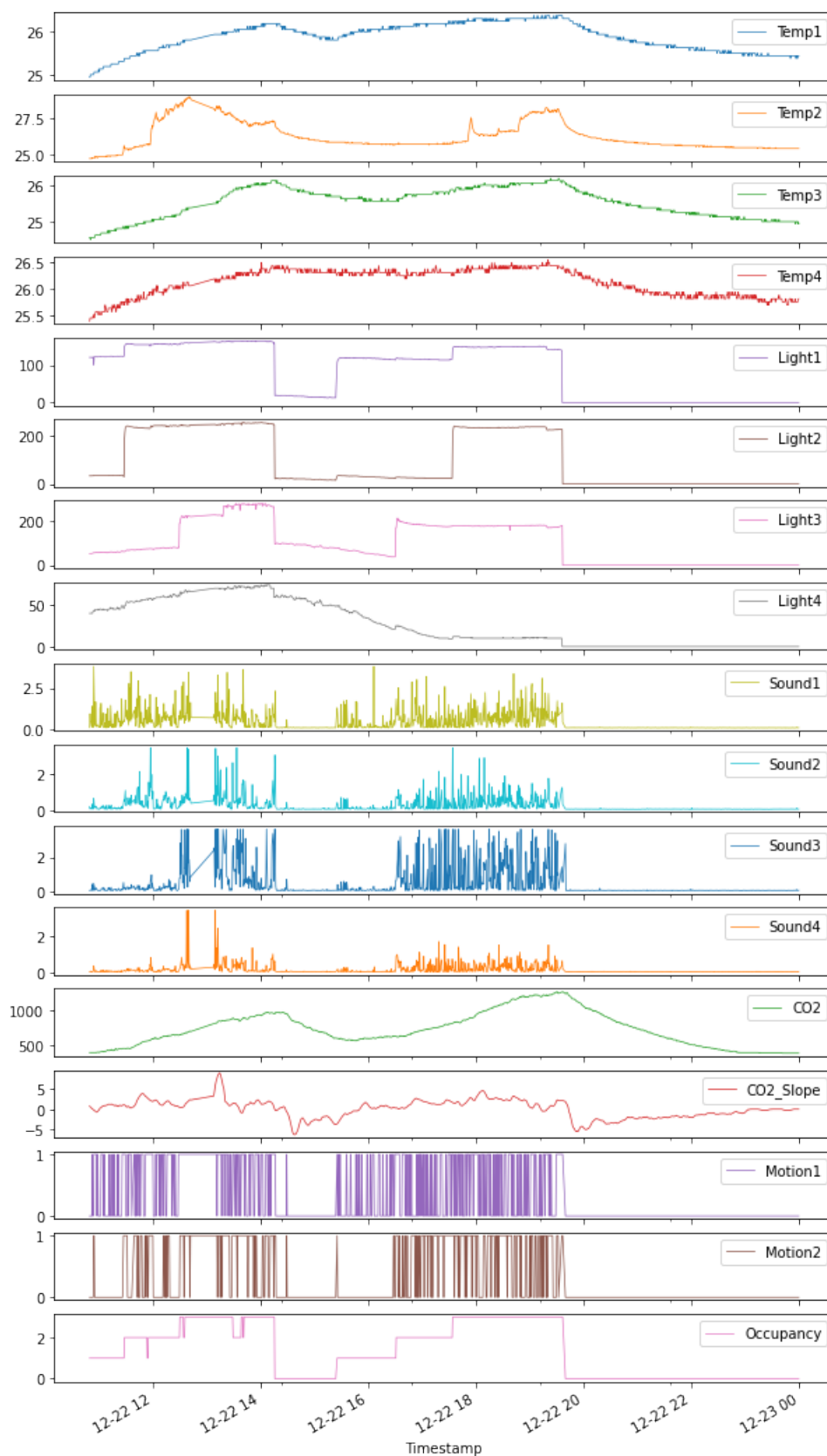
Do rozwiązania problemu zostały wybrane 3 modele:

1. Model wykorzystujący Conditional Random Fields (CRF) i MLP.
2. Bayesowska sieć neuronowa (BNN).
3. Sieć rekurencyjna LSTM wykorzystująca metodę Monte Carlo dropout.

Model wykorzystujący CRF został wybrany, jako że pozwala on na wykorzystanie zależności pomiędzy kolejnymi elementami w sekwencjach, co powinno się sprawdzić w przypadku pomiarów wykonywanych w równych odstępach czasu. Dodatkowo, modelowanie przez CRF przejść pomiędzy etykietami powinno pomóc dokonywać poprawnych predykcji, gdyż liczba osób w pokoju będzie się zazwyczaj zmieniać co najwyżej o 1.

Bayesowska sieć neuronowa pozwala na uzyskanie pewności klasyfikatora, co może posłużyć do interpretacji wyników, w przypadku gdy w pokoju jest kilka osób. Ze względu

<sup>1</sup><https://archive.ics.uci.edu/ml/datasets/Room+Occupancy+Estimation>



Rysunek 4: Wartości atrybutów w zestawieniu ze zmienną docelową z pierwszego dnia pomiarów.

na liczność klasy "0" i większą łatwość jej odróżnienia od pozostałych klas, może być to przydatne. W przypadku tego zadania, które jest z pogranicza klasyfikacji i regresji, może pozwolić na otrzymanie klas "pomiędzy" (jak w regresji), gdzie sieć nie jest pewna dokładnej ilości osób w pomieszczeniu i waha się pomiędzy dwiema klasami.

W związku z tym, że w zestaw jest serią czasową, a część parametrów takich jak temperatura będzie zależała w dużym stopniu od poprzednich wartości, zaproponowano wykorzystanie sieci LSTM, która może pozwolić wykorzystać takie zależności. Struktura sieci to jedna lub wiele warstw komórek LSTM, a wyjście ostatniej komórki trafia do warstwy w pełni połączonej. Wykorzystywano konfiguracje w z jednym wyjściem (regresja) oraz wieloma (klasyfikacja). Między komórkami LSTM oraz przed warstwą w pełni połączoną stosowano dropout.

Zastosowanie regularyzacji dropout wykorzystuje technikę Monte Carlo dropout, co czyni model probabilistycznym i pozwala określać niepewność predykcji. Metoda została zaprezentowana w roku 2015 (Gal, Ghahramani)[1]. Dla określenia niepewności należy znaleźć rozkład posterior  $p(W|X, Y)$ . W przypadku regresji można estymować rozkład prawdopodobieństwa klasy dla danej próbki poprzez wykonanie  $N$  klasyfikacji w modelu z aktywnym dropoutem (dla każdej klasyfikacji sieć ma inną strukturę). Dzięki temu można estymować rozkład normalny za pomocą średniej oraz odchylenia standardowego otrzymanych wyników. W przypadku klasyfikacji można określać prawdopodobieństwa klas dla próbki estymując rozkład kategoriowy, dla każdej klasy dzieląc ilość jej wystąpień przez ilość próbkowań.

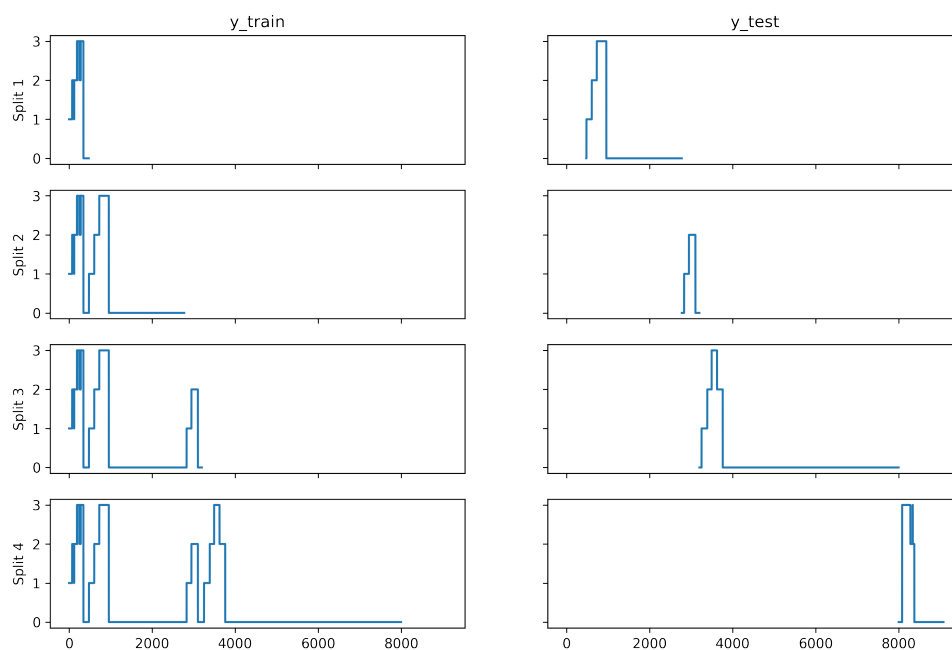
### 3 Eksperymenty

Dla każdego z wybranych modeli zostały przeprowadzone badania wpływu hiperparametrów na uzyskiwane wyniki. Jako metryki zostały wykorzystane Mean Squared Error i miara F1 w wariancie macro. Miara F1 macro została wybrana ze względu na niezbalansowanie klas. Problem jest jednoetykietowy, więc miara F1 micro zachowywałaby się jak dokładność (accuracy). Dla niektórych podziałów dla miary F1 występował problem nadawania etykiety, która nie występowała w zbiorze treningowym (nie dało się podzielić danych tak, żeby tego uniknąć) co sprawiało, że F1 dla niektórych klas było niezdefiniowane i otrzymywało wartość 0, co zaniżało wynik. Z tego powodu F1 nie było odpowiednie do porównywania modeli w treningu hiperparametrów, a podawane było dodatkowo. Jako kryterium jakości modelu wybrano miarę MSE, która jest odpowiednia ze względu na to, że przykładowo pomyłkę klasyfikacji 0 osób jako 3 uznaje za większy błąd, niż pomyłkę klasyfikacji 0 osób jako 1.

Do walidacji modeli został wykorzystany podział na 4 serie czasowe. Podział na poszczególne serie został wykonany ręcznie. Jest on widoczny na Rysunku 5. Dołożono starań, aby każda klasa była reprezentowana w danych testowych każdej serii. Podczas tworzenia serii czasowych dane treningowe były normalizowane do wartości znajdujących się w przedziale  $[0, 1]$ . Dane testowe również były normalizowane na podstawie minimalnych i maksymalnych wartości atrybutów ze zbioru treningowego (zatem mogą one wykraczać poza ustalony przedział). Końcowe wartości metryk zostały uśrednione dla wszystkich 4 serii czasowych.

#### 3.1 Conditional Random Fields

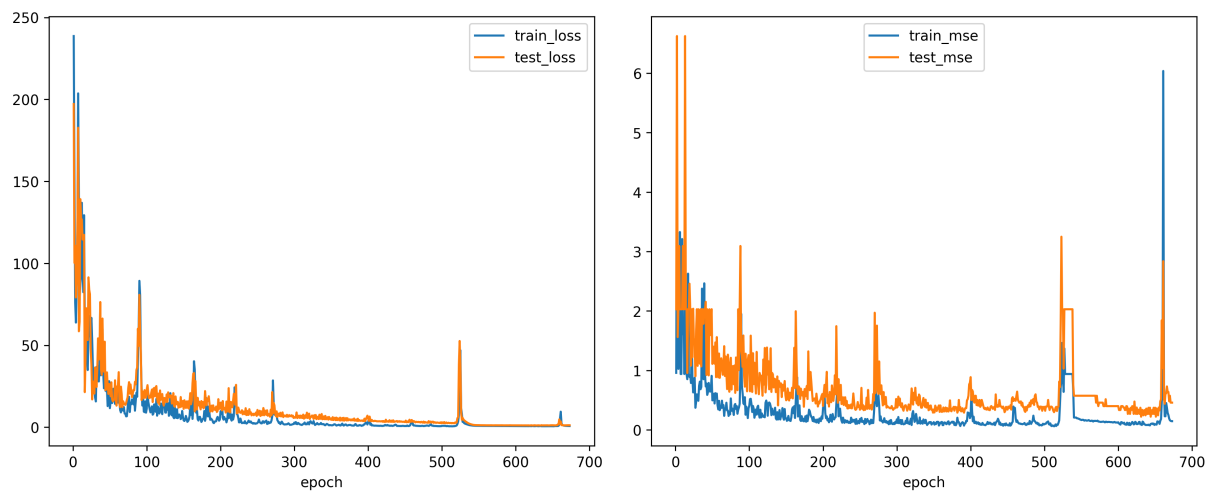
Dla perceptronu z warunkowymi polami losowymi zostały przeprowadzone badania wpływu następujących hiperparametrów:



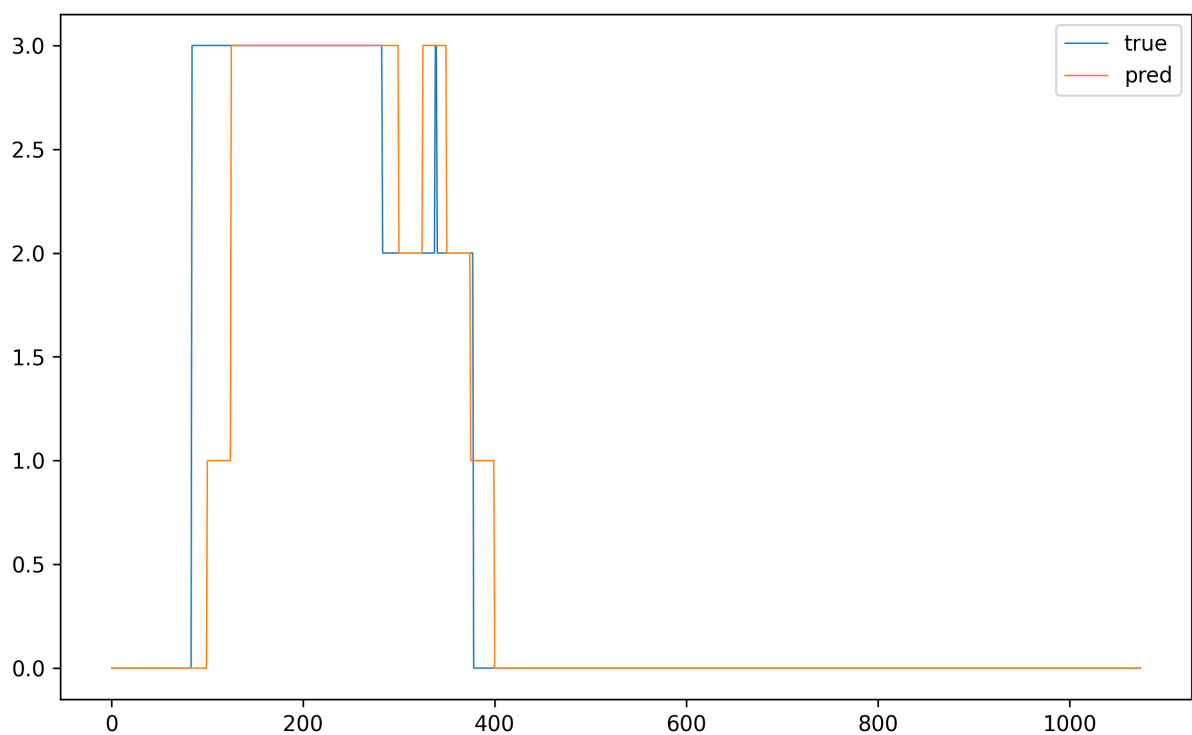
Rysunek 5: Serie czasowe wykorzystywane do walidacji modeli

Liczba neuronów	Wsp. uczenia	Dł. sekwencji	MSE	F1 macro	F1 micro
32	0.0007	25	0.2879	0.5296	0.8264
128	0.0007	20	0.3702	0.4979	0.8277
32	0.001	25	0.4023	0.5081	0.8391
128	0.001	20	0.405	0.56	0.8214
64	0.001	15	0.4439	0.4708	0.8095
32	0.001	15	0.5058	0.4654	0.7862
64	0.0003	10	0.5553	0.4288	0.7896
32	0.0001	20	0.5655	0.4582	0.8038
32	0.0001	25	0.5733	0.3782	0.7584
128	0.0007	15	0.5766	0.4698	0.8094
64-32	0.0001	20	0.5817	0.4028	0.7669
128	0.0003	10	0.5823	0.3882	0.7666
32	0.001	10	0.5918	0.5508	0.8057
64-32-16	0.0003	25	0.5951	0.4312	0.7664
32	0.0007	20	0.6109	0.4202	0.7972
32	0.001	20	0.6211	0.4184	0.7941
64	0.0007	10	0.6535	0.444	0.794
128	0.0007	25	0.6806	0.4136	0.7834
32	0.0003	10	0.6825	0.5038	0.6631

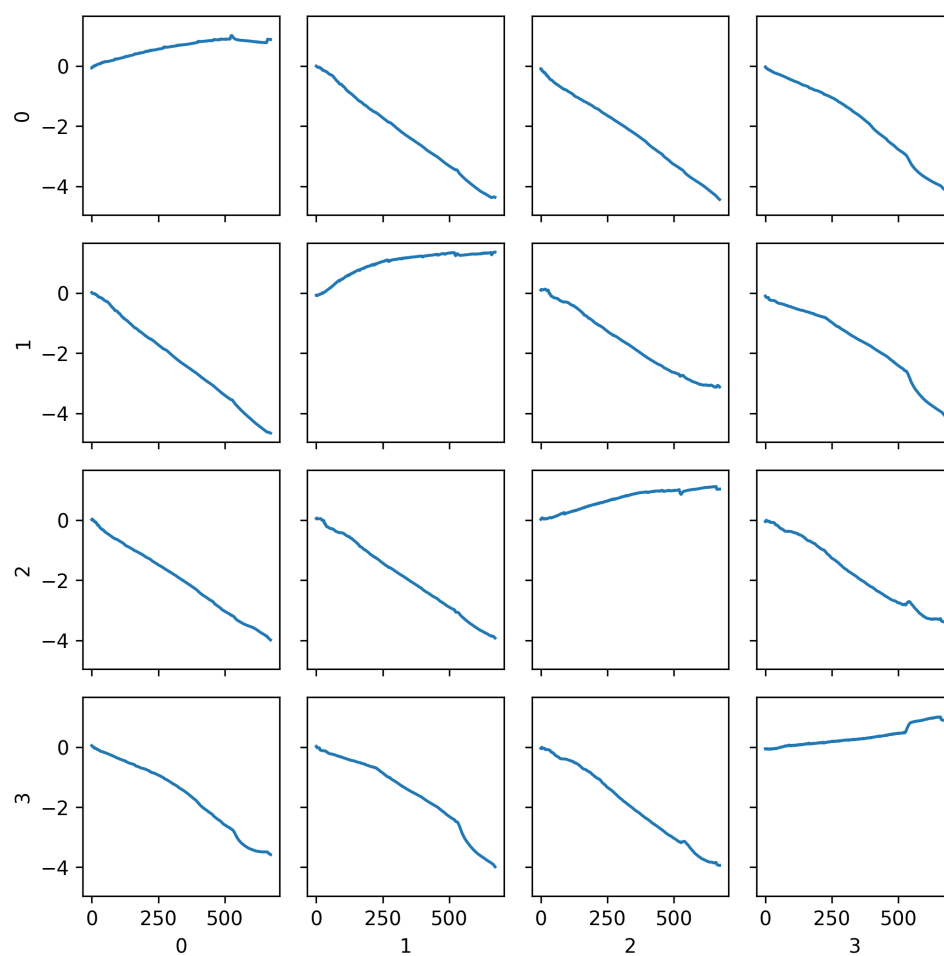
Tablica 2: Wyniki 20 najlepszych eksperymentów CRF



Rysunek 6: Przebieg treningu najlepszego modelu z CRF dla ostatniego podziału danych



Rysunek 7: Predykcje najlepszego modelu z CRF dla ostatniego podziału danych



Rysunek 8: Macierz przejść najlepszego modelu z CRF dla ostatniego podziału danych na przestrzeni epok uczenia



- liczba warstw ukrytych i liczba neuronów w warstwach ukrytych (wartości: 32, 64, 128, 64-32, 128-64, 64-32-16),
- współczynnik uczenia (wartości: 0.001, 0.0007, 0.0003, 0.0001),
- długość sekwencji (wartości: 10, 15, 20, 25)

Przy parametrze długości sekwencji przetwarzanych przez model należy wspomnieć, że nie został tutaj wykorzystany padding - ostatnie sekwencje danych krótsze niż wybrana długość były odrzucane. Po każdej warstwie ukrytej stosowana była funkcja aktywacji ReLU.

Z racji dość szybkiego czasu trenowania tego modelu, zbadano wszystkie kombinacje poszczególnych parametrów. Do treningu została wykorzystana technika wczesnego zatrzymywania, obserwująca wartość funkcji straty na zbiorze testowym i *cierpliwością* (*patience*) równą 50. Maksymalna liczba epok treningu wynosiła 1000 i została ona osiągnięta tylko w kilku przypadkach - większość treningów została zatrzymana wcześniej, więc nie występuje tutaj problem z niedouczeniem. Wielkość paczki danych w każdym przypadku wynosił 16.

Wyniki najlepszych eksperymentów zostały przedstawione w Tabeli 2. Najlepszą wartość metryki MSE uzyskano dla konfiguracji z najmniejszą liczbą warstw, najmniejszą liczbą neuronów ukrytych w tej warstwie i najdłuższymi sekwencjami. Różnica w MSE pomiędzy pierwszym a drugim modelem wynika głównie z predykcji dokonywanych na zbiorze testowym dla drugiego podziału danych - najlepszy model dla danych z etykietą o wartości 3 często predykuje wartość 2, a drugi najlepszy model predykuje w tych samych przypadkach wartość 0.

Największy problem pojawiający się w modelu z CRF-em można zaobserwować na Rys. 8. Po ostatnim wyuczeniu na trzech podziałach danych i testowaniu na czwartym podziale, szansa przejścia z dowolnej etykiety na etykietę o innej wartości jest bardzo niska. W znacznej większości przypadków etykiety pomiędzy poszczególnymi elementami w sekwencji nie będą się zmieniać, co bardzo dobrze widać na Rys. 7. Nigdzie tutaj nie występuje zmiana etykiety w środku sekwencji - etykieta całej sekwencji zależy więc tylko od pierwszego elementu. Problem ten wynika ze specyfiki zbioru danych - zmiany etykiet występują w nim zbyt rzadko, aby CRF mógł je predykować z większą dokładnością, przez co taki model może nie być najlepszym wyborem do tego zastosowania.

Pomimo powyższego problemu, w najlepszych 10 eksperymentach znajduje się tylko jeden model, który działał na sekwencjach o najmniejszej długości. W większości przypadków modele te miały podobnie prezentujące się macierze przejść, jednak znacznie częściej predykowały niewłaściwe wartości początkowe etykiet. Może to być powiązane nieodpowiednim doбором współczynników uczenia, jako że skrócenie sekwencji można porównać do zmniejszenia wielkości paczek danych przekazywanych do sieci, co ma istotny wpływ właśnie na współczynnik uczenia.

W najlepszych wynikach pojawiły się tylko 2 modele, które zawierały więcej niż 1 warstwę ukrytą. W takich modelach pojawiał się inny problem. Dla większych wartości współczynnika uczenia i przy ostatnim podziale danych, który zawierał najwięcej etykiet o wartości 0, modele predykowały dla danych testowych same etykiety o wartości 0. Problem ten nie występował przy mniejszych wartościach współczynnika uczenia, jednak uzyskiwane wyniki wciąż były gorsze niż dla prostszych modeli.

### 3.2 Bayesowska sieć neuronowa

Początkowo zastosowano prostą sieć z jedną warstwą liniową. Badanymi parametrami był współczynnik uczenia i rozmiar warstwy ukrytej. Domyślną wartością współczynnika było 0.01, a rozmiaru warstwy 64. Wyniki strojenia hiperparametrów przedstawia Tabela 3.

Parametr	MSE	F1
lr = 0.01	0.57	0.46
lr = 0.005	1.06	0.44
lr = 0.001	1.07	0.45
lr = 0.0001	1.36	0.28
hidden size = 16	0.49	0.45
hidden size = 64	0.54	0.46
hidden size = 128	1.25	0.38

Tablica 3: Badanie hiperparametrów dla BNN.

Podczas uczenia można było zaobserwować, że funkcja straty spada w przewidywalny sposób i stabilizuje się na pewnym poziomie pod koniec uczenia (lub zaczyna się lekko przeuczać). Wartości metryk na zbiorze treningowym rosną, choć można zaobserwować spore wahania; na zbiorze testowym natomiast utrzymują się na podobnym poziomie. Przykładowe krzywe uczenia przedstawia Rys. 9. Dla większości badanych kombinacji były one podobne - jedynie dla najmniejszych wartości lr widać było, że model nie douczył się do końca, a dla dużych rozmiarów warstwy ukrytej nieco mocniej widać przeuczanie się.

Wyniki po dodaniu kolejnej warstwy ukrytej nie poprawiają się znacząco, natomiast model bardzo szybko się przeucza i konieczne jest zatrzymanie uczenia wcześniej. Wyniki przedstawia je Tabela 4.

Parametr	MSE	F1
hidden size = 16	2.65	0.48
hidden size = 64	0.82	0.48
hidden size = 128	0.75	0.51

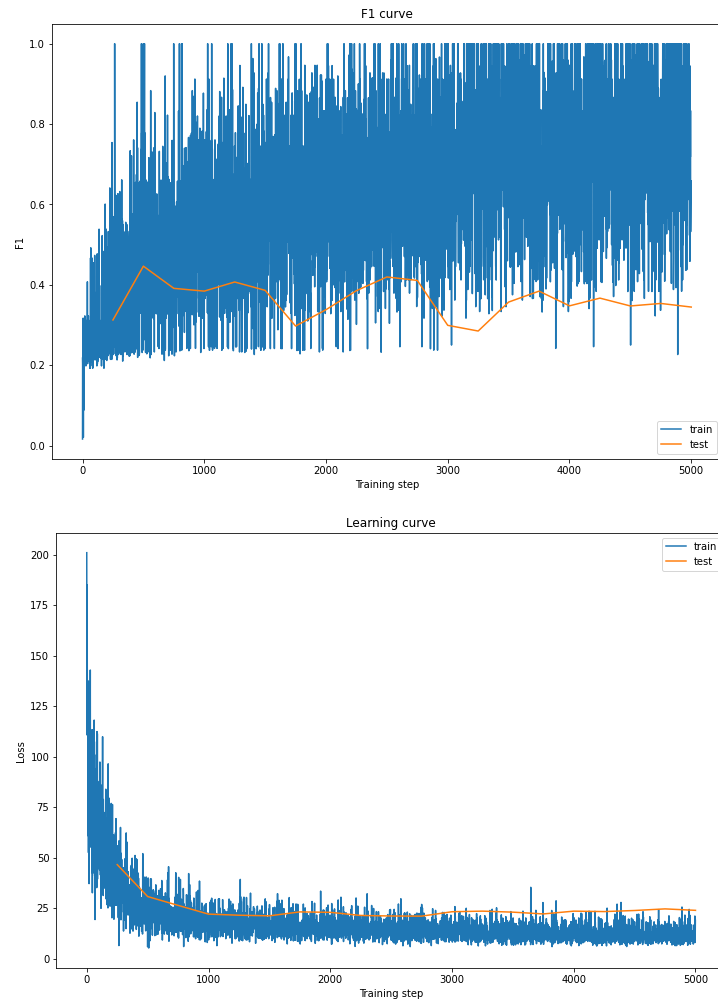
Tablica 4: Badanie hiperparametrów dla BNN z 2 warstwami ukrytymi.

Najlepszym modelem okazał się ten z domyślnymi parametrami, czyli lr = 0.01 i rozmiarem warstwy 64 (1 warstwa). Sprawdzono go dodatkowo na zbiorze testowym (ostatnia seria czasowa) i pokazano niepewność sieci. Wyniki widać na Rys. 10. Sieć potrafi napotkać na problem opisany przy poprzednim modelu - dla niektórych konfiguracji model uczy się przewidywać same zera, z prawdopodobieństwem bliskim 1. Dla ostatecznego modelu pewność predykcji nie zmienia się znacząco, oscylując wokół jednej wartości.

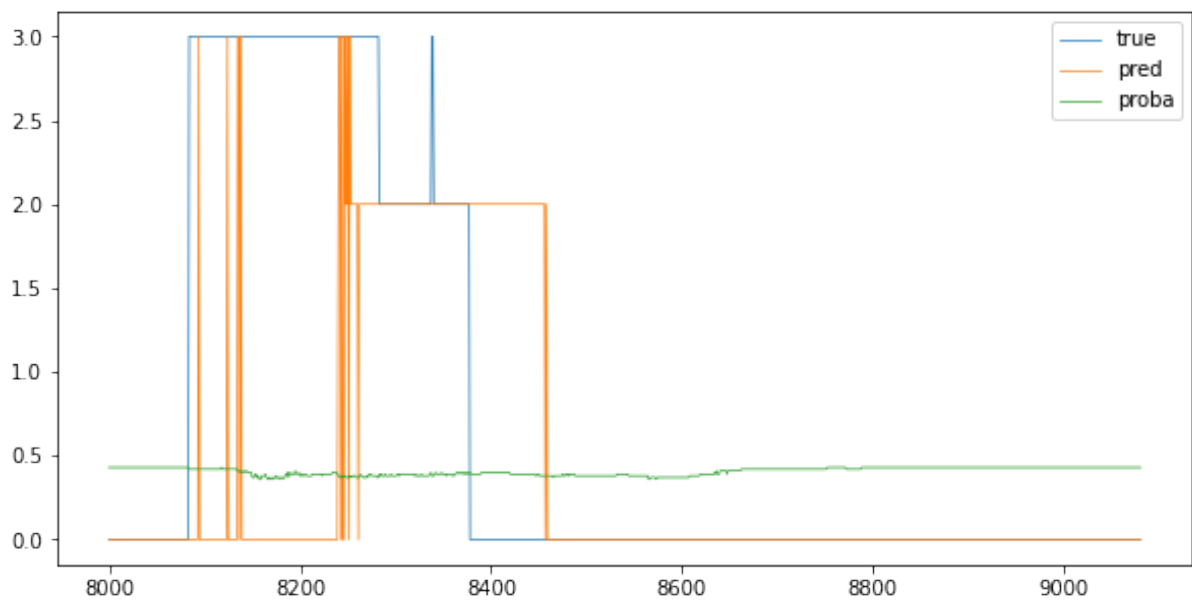
### 3.3 Sieć LSTM

Dla modelu LSTM wykonano badania wpływu następujących hiperparametrów:

- długość sekwencji,
- liczba warstw,



Rysunek 9: Krzywe uczenia BNN dla  $lr = 0.005$  i rozmiaru warstwy ukrytej 64.



Rysunek 10: Ostateczny model BNN wraz z niepewnościami sieci.

	Model regresji	Model klasyfikacji
Warstwa ukryta	64	32
Liczba warstw	3	4
Długość sekwencji	15	15
Dropout	0.5	0.5
Wsp. uczenia	0.003	0.0001
MSE	0.133 (0.058)	0.173 (0.093)
F1	0.659 (0.196)	0.627 (0.193)

Tablica 5: Najlepsze hiperparametry oraz wyniki (z odchyleniami standardowymi) dla modelu regresji oraz klasyfikacji.

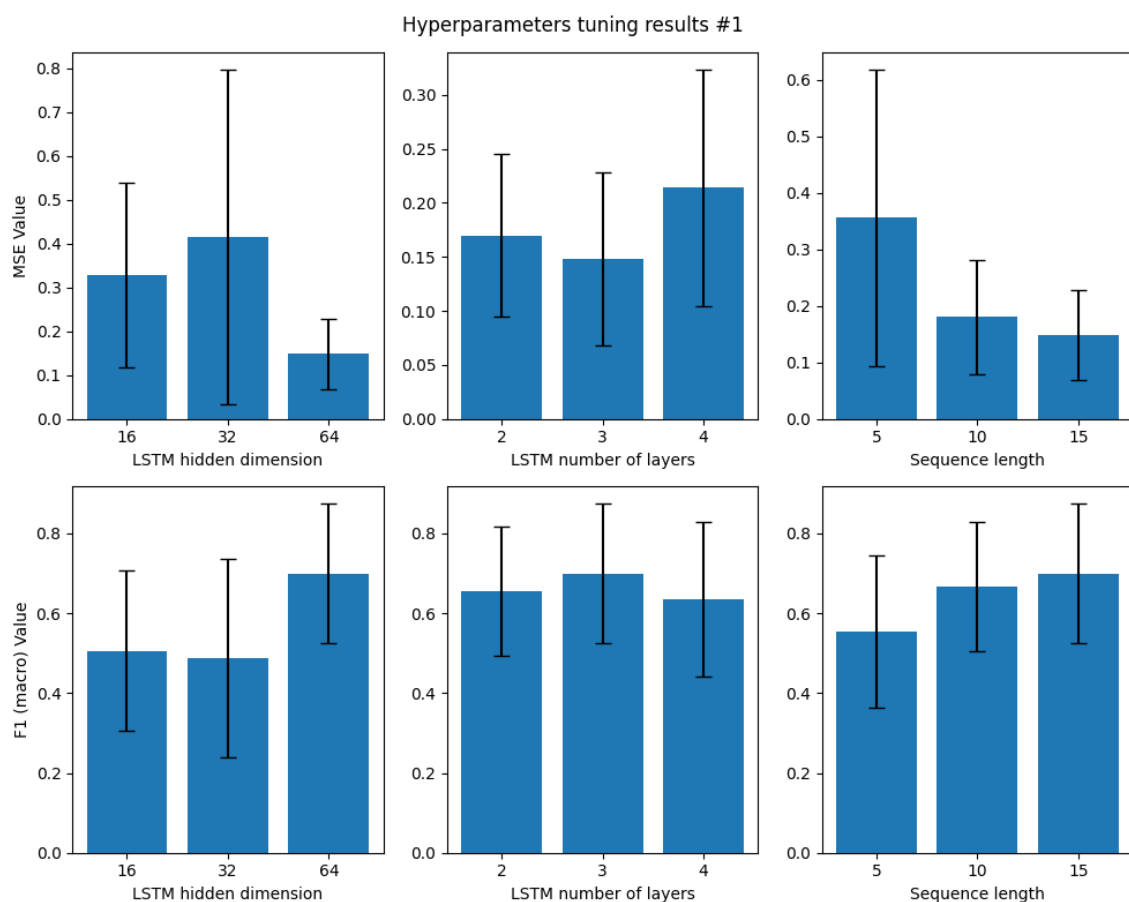
- rozmiar warstwy ukrytej,
- prawdopodobieństwo dropout,
- współczynnik uczenia.

W związku z tym, że uczenie modelu LSTM trwa długo, liczba parametrów do zbadania w rozsądnym czasie była ograniczona. Wykonano dwie serie doboru hiperparametrów: najpierw struktury sieci, a następnie prawdopodobieństwo dropoutu oraz współczynnik uczenia dla najlepszych znalezionych wcześniej hiperparametrów, co pozwoliło zmniejszyć liczbę kombinacji. Przed pierwszą serią badań ustalono empirycznie współczynnik uczenia obserwując krzywe uczenia - tak, aby obserwować stabilny spadek funkcji starty. Dla wszystkich badań stosowano wczesne zatrzymanie ze współczynnikiem patience równym 10, pozwalając przy każdym uczeniu na osiągnięcie wczesnego zatrzymania.

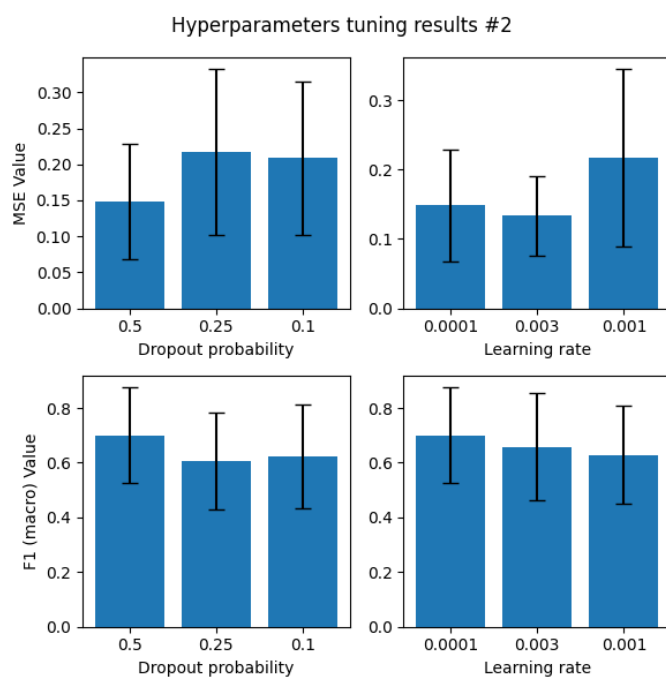
Osobno badano hiperparametry dla modelu regresji oraz modelu klasyfikacji. Dla możliwości porównania wartości MSE oraz F1 z modelu regresji oraz klasyfikacji, wynik dla regresji zaokrąglano do pełnych wartości, aby odzwierciedlały poszczególne klasy. Wyniki dla doboru hiperparametrów określano z wyłączoną opcją dropout (model nie jest probabilistyczny), jednak ostatecznie dokonania zostanie ewaluacja modeli z techniką Monte Carlo dropout.

Najlepszy wynik osiągnął model regresji i dla niego przestawiono wpływ hiperparametrów (Rysunek 11, Rysunek 12). Najlepsze hiperparametry przedstawiono w tabeli 5. Na wykresach można zaobserwować duże odchylenia standardowe, wynikające z tego, że niektóre podziały danych okazywały się trudniejsze lub nie zawierały wszystkich etykiet (w przypadku części testowych). Dobre wyniki osiągnano dla długich sekwencji oraz dla większych rozmiarów sieci - warstwa ukryta LSTM o wielkości 64 oraz liczba warstw LSTM równa 3 lub 4. Najlepsze okazały się niskie współczynniki uczenia, zwykle była to początkowo dobrana wartość - 0.0001.

Następnie dokonano ewaluacji najlepszych modeli stosując metodę Monte Carlo dropout. Dla modelu regresji dla każdej próbki testowej wykonywano 100 klasyfikacji z aktywnym dropoutem, a licząc średnią i odchylenie standardowe estymowano rozkład normalny odzwierciedlający niepewność predykcji. Dla modelu klasyfikacji również wykonywano 100 próbkowań, a dzieląc liczbę otrzymania wybranej etykiety przez liczbę próbkowań otrzymywano jej prawdopodobieństwa. W ten sposób można przykładowo oszacować pewność nadanej przez model etykiety. W tabeli 6 przedstawiono porównanie wyników. Okazało się, że model regresji w wersji probabilistycznej nie pozwolił na uzyskanie dobrych wyników jako klasyfikator (przy zaokrąglaniu wyników), jednak spo-



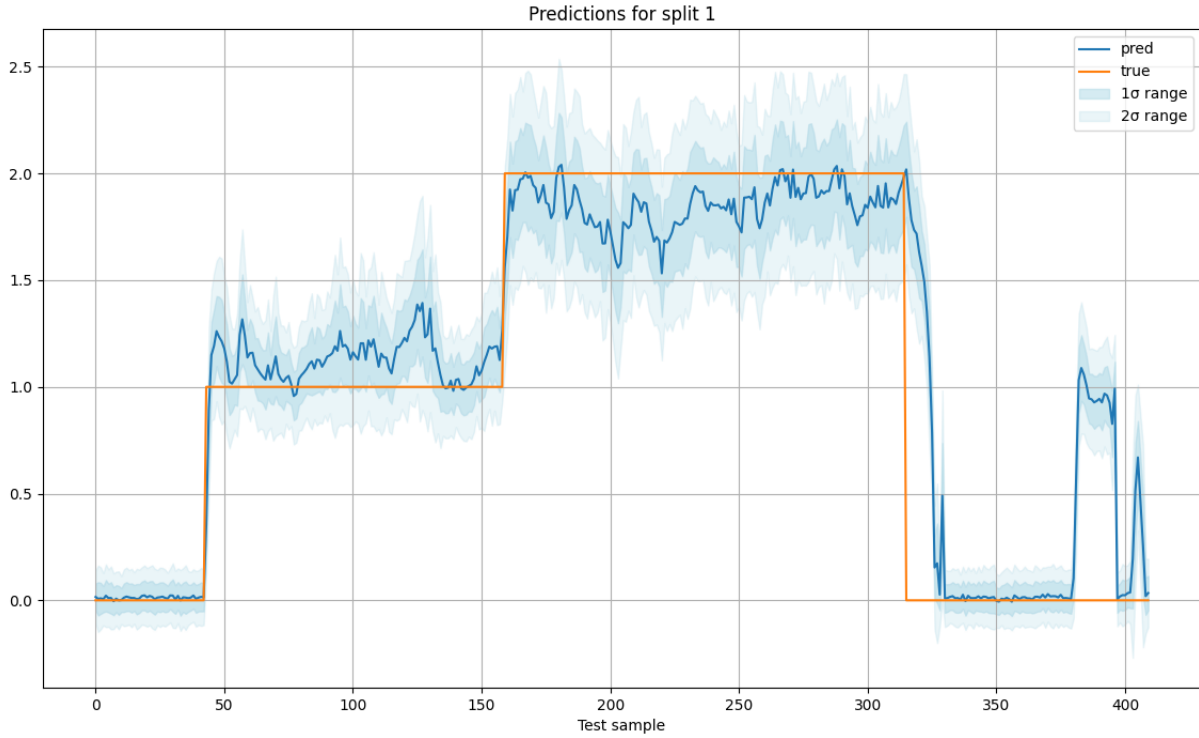
Rysunek 11: Wyniki badania hiperparametrów dla modelu regresji, 1 seria



Rysunek 12: Wyniki badania hiperparametrów dla modelu regresji, 2 seria

	MSE (std)	F1 macro (std)
Model regresji bez MCD	0.133 (0.058)	0.659 (0.196)
Model regresji z MCD	0.661 (0.194)	0.131 (0.058)
Model klasyfikacji bez MCD	0.173 (0.093)	0.634 (0.192)
Model klasyfikacji z MCD	0.173 (0.087)	0.627 (0.193)

Tablica 6: Wyniki modeli w wersjach z oraz bez Monte Carlo dropout



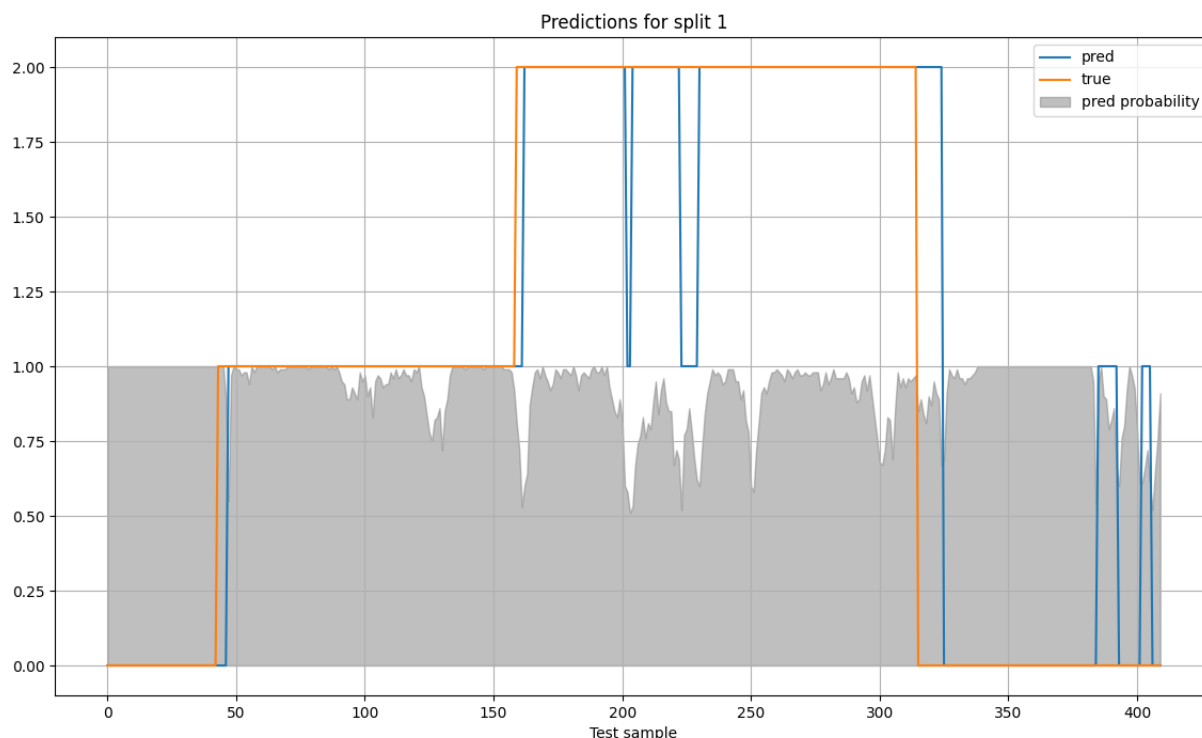
Rysunek 13: Przykładowe predykcje modelu regresji dla pierwszego podziału danych z niepewnościami

sób modelowania niepewności jest użyteczny, co zostanie przedstawione dalej. Wyniki modelu klasyfikacji w wersji probabilistycznej nie odbiegają od tych w wersji bez Monte Carlo dropout.

Na rysunkach 13 oraz 14 przedstawiono przykładowe działanie modelu regresji oraz klasyfikacji z Monte Carlo dropout dla 100 próbkowań.

Dla regresji (Rysunek 13) można zaobserwować, że mimo słabych wyników kiedy model był oceniany jako klasyfikator, predykcje dla wybranych danych zdają się być dobre. Zaobserwować można stosunkowo małą niepewność (odchylenie standardowe) w okolicach wartości 0, która była dobrze przewidywana przez wszystkie modele. Warto zauważyć, że anomalie w predykcji klasy 0 (okolice 400 próbki) wiążą się z wzrostem niepewności. Dla pozostałych wartości niepewność jest dość duża i porównywalna. Zdecydowana większość próbek mieści się w zakresie  $2\sigma$ .

Dla klasyfikacji (Rysunek 14) można zaobserwować wysokie, bliskie 1 prawdopodobieństwa predykcji dla klasy 0, które jest zwykle dobrze przydzielana. Błędy w okolicy 400 próbki wiążą się ze spadkiem prawdopodobieństwa. Dla innych klas, w momencie kiedy model przydziela dobrą klasę, można zwykle zaobserwować duże prawdopodobieństwa (okolice próbek 170 oraz 270) i ich spadek, kiedy model się myli (np. okolice próbek 200).



Rysunek 14: Przykładowe predykcje modelu klasyfikacji dla pierwszego podziału danych z prawdopodobieństwem nadanej klasy

Można więc wywnioskować, że niepewności obu modeli dają użyteczną informację. Szczególnie w przypadku modelu klasyfikacji miejsca gdzie model wykazuje tendencję do pomyłek, jego pewność znacznie spada. Wykorzystując prawdopodobieństwa można więc obniżyć częstość predykcji fałszywie pozytywnych (jednak nie zawsze zostanie przydzielona jakakolwiek klasa).

### 3.4 Porównanie wyników wybranych modeli

Model	MSE	F1 macro
Las losowy	0.323	0.350
CRF	0.288	0.530
BNN	0.540	0.480
LSTM (regresja bez MCD)	0.133	0.659
LSTM (klasyfikacja bez MCD)	0.173	0.634
LSTM (klasyfikacja z MCD)	0.173	0.627

Tablica 7: Porównanie najlepszych zbadanych modeli

Do porównania wybrane zostały modele wraz z hiperparametrami, które uzyskiwały najlepsze wartości wybranych metryk w eksperymentach opisanych powyżej. Wyniki zostały przedstawione w Tabeli 7.

Model nieuwzględniający zależności czasowych - bayesowska sieć neuronowa osiągał najniższe wyniki, co można wytłumaczyć tym, że zależności czasowe w danych, takie jak opóźnienia w zmianach cech, czy tendencja do utrzymywania stałej wartości etykiety przez dłuższe okresy czasu były użyteczne w predykcji. Model CRF uwzględniając zależność między obecną a poprzednią etykietą osiągnął lepsze rezultaty, a na wizualizacjach

predykcji można zauważyć mniejszą tendencję do nagłych zmian etykiety dla kolejnych próbek. Ma on jednak problem odwrotny - prawdopodobieństwa przejść pomiędzy różnymi etykietami są zbyt małe, aby można było uwzględnić bardziej dynamiczne zmiany klas. Model LSTM potrafił uwzględnić zależności czasowe w cechach, co pozwoliło na uzyskanie jeszcze wyższych wyników.

Bayesowska sieć neuronowa oraz sieć LSTM z Monte Carlo dropout pozwalają na predykcję niepewności modelu. Jak pokazano na wynikach, szczególnie dla modelu LSTM modelowane niepewności mogą być przydatne w praktycznym zastosowaniu modelu, na przykład zmniejszeniu liczby fałszywie pozytywnych predykcji lub określaniu na podstawie danego wyniku regresji powinno się podejmować decyzje.



## Bibliografia

- [1] Yarin Gal i Zoubin Ghahramani. *Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning*. 2015. DOI: 10.48550/ARXIV.1506.02142. URL: <https://arxiv.org/abs/1506.02142>.