Politechnika
Śląska

PROJEKT INŻYNIERSKI

Tytuł pracy dyplomowej inżynierskiej

Jakub KULA
Nr albumu: 296849

Kierunek: Automatyka i Robotyka
Specjalność: Technologie Informacyjne

PROWADZĄCY PRACĘ
dr inż. Szymon Ogonowski, prof. PŚ
KATEDRA Katedry Pomiarów i Systemów Sterowania
Wydział Automatyki, Elektroniki i Informatyki

Gliwice 2023

Tytuł pracy

Tytuł pracy dyplomowej inżynierskiej

Streszczenie

(Streszczenie pracy – odpowiednie pole w systemie APD powinno zawierać kopię tego streszczenia.)

Słowa kluczowe

(2-5 slow (fraz) kluczowych, oddzielonych przecinkami)

Thesis title

Thesis title in English

Abstract

(Thesis abstract – to be copied into an appropriate field during an electronic submission – in English.)

Key words

(2-5 keywords, separated by commas)

Spis treści

Rozdział 1

Wstęp

1.1 Cel i zakre pracy

wprowadzenie w problem/zagadnienie

1.2 Aktualny stan wiedzy

osadzenie tematu w kontekście aktualnego stanu wiedzy (*state of the art*) o poruszanym problemie
studia literaturowe [3, 4, 2, 1] - opis znanych rozwiązań (także opisanych naukowo, jeżeli problem jest poruszany w publikacjach naukowych), algorytmów,

1.3 Charakterystyka rozdziałów

Krótkie wprowadzenie do zawartości Zarys głównych punktów i celów rozdziału

1.4 Wkład autora

jednoznaczne określenie wkładu autora, w przypadku prac wieloosobowych – tabela z autorstwem poszczególnych elementów pracy Wzory

$$y = \frac{\partial x}{\partial t} \tag{1.1}$$

jak i pojedyncze symbole x i y składa się w trybie matematycznym.

Rozdział 2

Zastosowane narzędzia w pracy

2.1 Python

Wybór głónego języka progarmowania zastowanego w projekcie, wiązał się z postawieniem pewnych wymagań. Pierwszym z tych wymagań była dostępność dedykowanej bibloteki do uczenia maszynowego, która posiada narzędzia do efektywnej pracy nad modelami czy ich tesowanie. Użycie bibloteki która jest dobrze utrzymana zapewnii ogromne wsparcie społeczności, które moze okazać sie nieocenione w procesie nauki czy rozwiązywnia problemów.
Kolejnym wymaganiem jest aby wybrana techonologia była ciągle wspierana i aktualizowana. Machine learning jest aktulanie jedną z najszyciej rozwijających sie dziedzin programowania, co wiąże sie z szybkimi zmianami.(DOPISAĆ COŚ TUTAJ)

- R
- Python

R jest językiem skoncetrowanym na analizie danych i statystyce. Posiada on bardzo bogaty ekosystem jednak może stanowić to przyczyne wielu konfiliktów pomiędzy pakietami. Największą wadą tego jest jezyka jest problem ze skalowalnością. Praca z dużą ilością danych skutkuje zużyciem ogromnej ilości pamięci RAM.
Skorzystanie z Pythona będzie lepiej spełniać wymogi projektu. Jest on językiem bardziej wszechstronny oraz posiada obszerną bibioteke standardową jak i bardzo liczne zewnętrzne bibloteki. Największą wadą Pythona jest jego wydajność. Gdyż jest językiem interpretowanym, więc nie jest on kompilowany do kodu maszynowego przed jego uruchomieniem.

2.2 Tensorflow

Tensorflow jest jedną z dwóch głównych otwartych biblotek do uczenia maszynowego i głębokiego w Pythonie. Głównym konkurentem tensorflow jest PyTorch który jest roz-

wijany przez Facebook.

2.3 Inne bibloteki

2.3.1 Pandas

2.3.2 Matlibplot

2.3.3 Numpy

2.3.4 Sckit-learn

2.4 CUDA toolkit

Opis narzędzi które zostały uzyte w celu optymalizacji pracy pythona, takie jak wirtalne środowisko Conda, czy nvdia CUDA

Rozdział 3

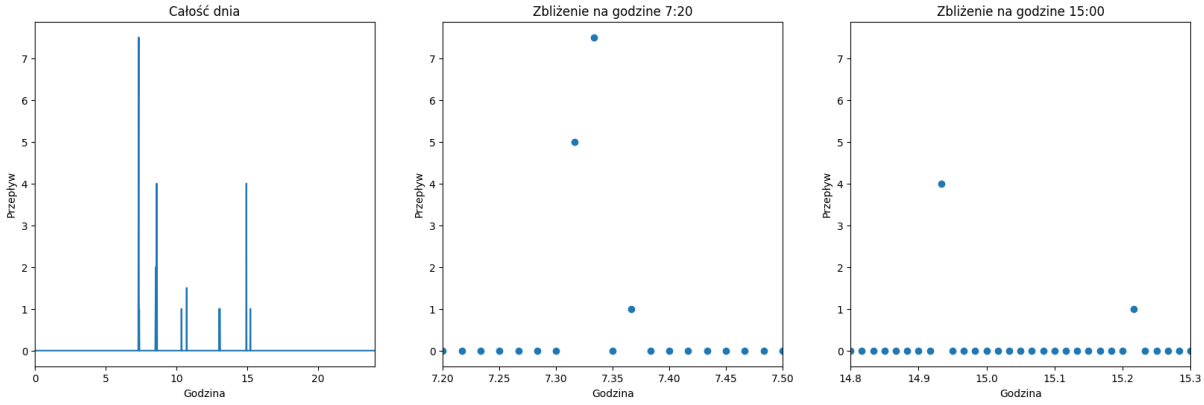
Modelowanie sieci neuronowej

3.1 Metodologia projektowania modelu sieci neuronowej

Wstep teoretyczny o modelowaniu, opisanie rzeczy takich jak, warstwy, neurony, funckje aktywacjie, funkcje kosztu, optymalizator, liczba epok, batch size, walidacha, funckaj strat

3.2 Dane wejściowe i proces ich przetwarzania

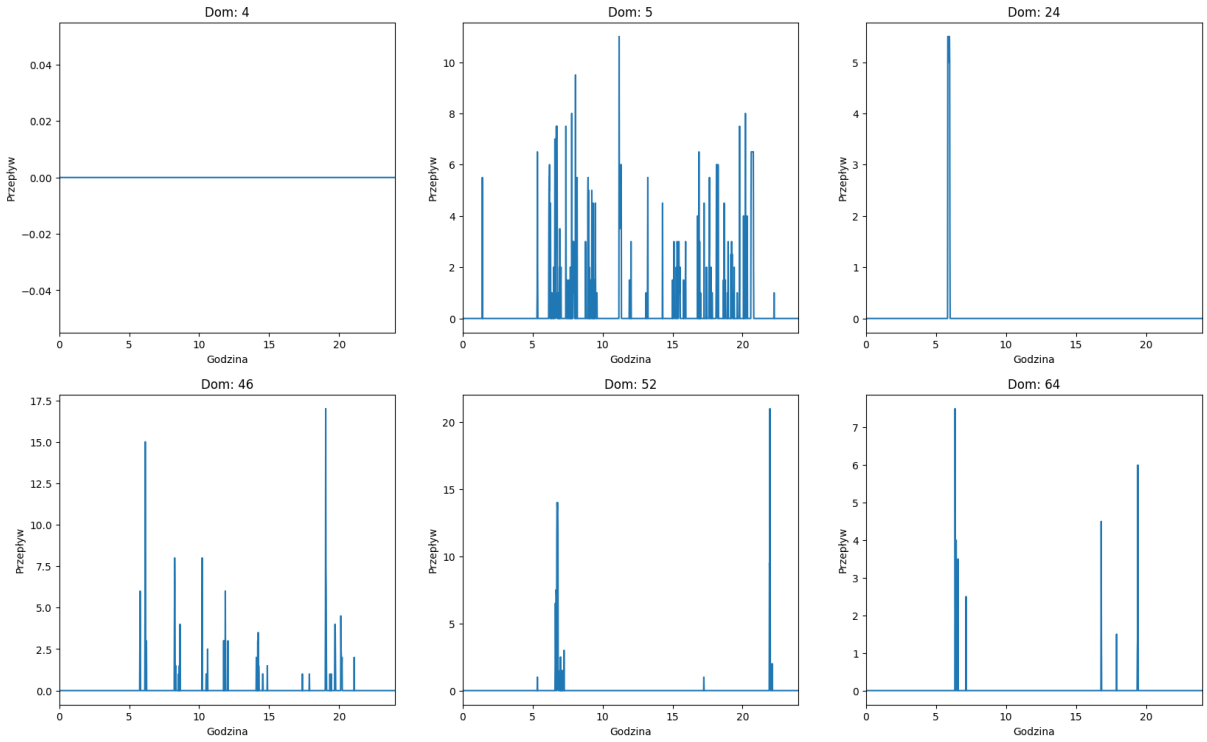
Projekt wykorzystuje dane zebrane przez instytut ASHRAE (American Society of Heating, Refrigerating and Air-Conditioning Enginee) w 2018 roku. Informacje o zużyciu wody zostały pozykane z 77 domostwo znajdujacych sie na terenie Kanady, zamieszkiwanym przez dwoje osób dorosłych oraz dwójke dzieci oraz posiadajacych podstawowe urządzenia AGD takie jak pralka czy zmywarka. Dane były zbierane przez 16 tygodni, w cyklach trwajacych 4 tygodnie, aby równomiernie obejmować każdą pore roku. Próbkowanie danych odbywało sie co minute, nieprzerwanie przez całą dobę. Całość danych zostało podzielone na 73 odzielne pliki w formacie .csv. W każdym plikuu znajdował sie timestamp w formacie RR-MM-DD HH:MM:SS, który określał dzien i godzine pomiaru danych. Oprócz tego zawierał informacje o zużyciu wody, oraz temperature otoczenia. Dodatkowo dane były podzielone na poszczególne poru roku. Pierwszy wykres przestawia



Rysunek 3.1: Podpis rysunku zawsze pod rysunkiem.

dane zebrane z domu nr. 13 zebrane w dniu 05/02/2018. Oś X oznaczająca godzinę, począwszy od północy. Oś Y reprezentuje przepływ wody w danym momencie dnia. Wykres Przestwia nieregularne piki o nierównomiernym rozkładzie. Okresami o zwiększonym przepływie są godziny 7-13 oraz 14-15. Okresy mniejszej aktywyności możemy zaobserwowac w godzinach późno popołudniowych oraz nocnych. Drugi i trzeci wykres przedstawiają zbiżenie na godzine 7 oraz 15. Dzięki zwężeniu analizowanego zakresu czasu, możliwe było dokładniejsze zbadanie struktury występujących pików. Ta metoda wizualizacji ujawniła, że poszczególne piki, które na ogólnym wykresie dobowym mogły sprawiać wrażenie pojedynczych punktów, w rzeczywistości są złożone z wielu pojedynczych zdarzeń. To odkrycie jest istotne, ponieważ wskazuje na bardziej złożoną dynamikę przepływu w określonych momentach doby, co na pierwszy rzut oka mogło umknąć uwadze.

W celu lepszego wstępnego zrozumienia charakterystyki analizowanego zestawu danych, niezbędne jest także szczegółowe przyjrzenie się kilku losowo wybranym domostwom.



Rysunek 3.2: Porównanie przepływów dla przykładowych domów w dniu 05/02/2018

Analiza przedstawionych histogramów przepływów dla poszczególnych losowo wybranych domostw, wykonana na podstawie danych z dnia 05/02/2018, ukazuje wyraźne różnice w charakterystyce przepływów dla każdego z nich. Zgodnie z Rysunkiem 3.2, można stwierdzić, że każdy dom prezentuje unikalny wzór aktywności, co może odzwierciedlać różnorodność nawyków, planów dnia lub specyficznych potrzeb mieszkańców. Na przykład, dla domu nr 5 zużycie wody jest rozłożone przez większą część dnia, z obserwowaną aktywnością w rozmaitych godzinach. Jednakże, zarejestrowano również pojedyncze zużycie w nocy. Z kolei Dom 52 charakteryzuje się wyraźnym szczytem wieczornym, co stanowi kontrast w stosunku do pozostałych domów.

W przypadku Domu nr 4, nie odnotowano żadnego przepływu w analizowanym dniu. Brak danych może wynikać z co najmniej dwóch potencjalnych przyczyn. Pierwszą z nich jest zastosowany czas próbkowania, który wynosił jedną minutę. Taki interwał może nie być wystarczająco krótki, aby zarejestrować sporadyczne lub krótkotrwałe zdarzenia przepływu. Drugą możliwością, która może wyjaśniać brak zarejestrowanej aktywności, jest potencjalna nieobecność mieszkańców w domu w danym dniu.

Pomimo iż zgromadzone dane charakteryzowały się wysoką jakością, ich specyficzny format wymagał przygotowania skryptu celem ich przetwarzania i ekstrakcję istotnych informacji.

```
for f in csv_files:
    dataset = pd.read_csv(f)
    for i in range(len(dataset)):
        dt = dataset.loc[i, "Summer_Timestamps"]
        data, time = dt.split(' ')
        year, month, day = (int(x) for x in data.split('-'))
        ans = datetime.date(year, month, day)
        dzied_tygodnia = dni_tygodnia_mapa[ans.strftime("%A")]

        hours, minutes, null = time.split(":")
        time = (int(hours)*60+int(minutes))/(60*24)

        przeplyw = dataset.loc[i, "Summer_Water_Consumption"]

        dane.loc[len(dane)] = {'Pora_roku': 2,
                               'Dzien_tygodnia': dzied_tygodnia,
                               'Czas_dnia': time,
                               'Przeplyw': przeplyw}
```

Rysunek 3.3: Fragment skryptu przetwarzającego dane.

W ramach procesu dostowania formatu aby przystosować go do wymagań tensorflow, dzień tygodnia zostal zamieniona na etykiete liczbową, która przyjmuje wartośc od 1 do 7, co odpowiada kolejnym dniom tygodnia. Podobny proces został zastosowany do etykietowania pór roku. Każda została zakodowana jako etykieta w zakresie od 1 do 4 co prezentuje kolejno, wiosne, lato, jesień i zimę. Dodatkowo czas dnia został zmieniony na procent dnia w skali od 0 do 1.

3.3 Projektowanie i ocena modeli

W ramach realizacji badań nad optymalizacją architektury sieci neuronowej oraz doborem hiperparametrów, zdecydowano się na podział danych uczących na trzy zbiory. Pierwszy z nich to zestaw który zawiera dane pochodzące z 12 losowo wybranych domostw, co ma na celu zapewnienie reprezentatywności i różnorodności w ramach próby badawczej. Drugi zestaw stanowi podzbiór zawierający dane z pojedynczego gospodarstwa domowego, co pozwala na szczegółową analizę wydajności modelu w warunkach bardziej jednorodnych danych. Dodatkowo, utworzony został trzeci zestaw danych, który obejmował informacje z wszystkich 77 domów biorących udział w badaniu.

Podział zbioru danych na trzy zestawy okazał się kluczowy dla efektywnego doboru hiperparametrów modelu, szczególnie biorąc pod uwagę, że cały zbiór danych zawierał aż 12,5 miliona wierszy. Zestaw wybranych domostw, zawierający blisko 2 miliony wierszy, oraz pojedyncze domostwo z 161 tysiącami wierszy, umożliwiły przeprowadzenie dokładniejszych i bardziej zróżnicowanych testów.

Czas uczenia sieci był znacząco różny dla poszczególnych zestawów danych. Przykładowo, dla całego zbioru danych proces uczenia trwający 10 epok przy rozmiarze partii równym 64 zajmował około 46 minut. Tymczasem dla wybranych domostw czas ten skracał się do 9 minut, a dla pojedynczego domostwa uczenie trwało zaledwie 40 sekund. W ramach badań podjęto próby wykorzystania Google Colab, będącego popularnym narzędziem służącym do programowania i przetwarzania danych w chmurze. Po odpowiednim skonfigurowaniu środowiska, napotkano na pierwszy znaczący problem - czas trwania uploadu pliku. Zaskakująco, przesyłanie pliku o rozmiarze 300 MB, zawierającego 12,5 miliona wierszy, zajęło znacznie więcej czasu, niż można było przewidywać. Kolejnym krokiem było przeprowadzenie procesu uczenia maszynowego na danych, zaplanowanego na 10 epok. Niestety, cały proces trwał ponad 100 minut, co wskazuje na ograniczenia wersji darmowej Google Colab. W związku z tym, stwierdzono, że bez inwestycji w wersję płatną, Google Colab nie zapewnia oczekiwanej redukcji czasu niezbędnego do nauki modelu

Początkowo hiperparametry były testowane na najmniejszym zbiorze, co pozwalało na szybką i efektywną ocenę różnych konfiguracji. Po uzyskaniu zadowalających wyników na zbiorze pojedynczego domostwa, testy były rozszerzane kolejno na zbiór wybranych domostw, a następnie na pełny zbiór danych. Taka strategia pozwoliła na stopniowe i metodyczne dostosowywanie hiperparametrów, minimalizując przy tym czas i zasoby potrzebne do przeprowadzenia eksperymentów, a jednocześnie maksymalizując ogólną skuteczność modelu.

Tabela 3.1: Hiperparametry Sieci Neuronowej

Optymalizator	Funkcja strat	Początkowy współczynnik uczenia	Rozmiar partii
Adam	mse	0.0001	64

Po przeprowadzeniu serii eksperymentów, w procesie selekcji optymalnej architektury sieci neuronowej, najbardziej efektywną konfiguracją okazała się struktura składająca się z sześciu warstw, z których cztery pełniły funkcję warstw ukrytych. W procesie iteracyjnego dostosowywania i ewaluacji różnych architektur sieci, model o takiej budowie wykazał najlepsze wyniki w zakresie dokładności i generalizacji na testowanych zbiorach danych. Architektura ta charakteryzowała się kolejno malejącą liczbą neuronów w poszczególnych warstwach: pierwsza warstwa zawierała 512 neuronów, druga 256, trzecia 128, czwarta 64, piąta 32, a szósta, będąca warstwą wyjściową, miała 1 neuron. Wszystkie warstwy, z wyjątkiem ostatniej, wykorzystywały funkcję aktywacji ReLU. Natomiast

ostatnia warstwa, pełniąca rolę warstwy wyjściowej, zastosowała funkcję aktywacji typu 'linear'

W ramach opracowanego modelu sieci neuronowej zastosowano dynamicznie zmieniający się współczynnik uczenia, oparty na metodzie wykładniczego spadku, opisanego wzorem:

$$\text{Współczynik uczenia}(epoka) = \begin{cases} \text{Początkowy współczynik uczenia} & \text{jeżeli } epoka < 5 \\ \text{Współczynik uczenia}(epoka - 1) \times e^{-0.1} & \text{jeżeli } epoka \geq 5 \end{cases} \tag{3.1}$$

Użycie tej motyody pozwoliło na zmniejszanie wartości współczynnika uczenia w trakcie procesu trenowania, co zwiększyło zdolności adaptacyjne sieci. Został on zastosowany gdyż częstym zjawiskiem było generowanie przez sieć stałej wartości wyjściowej, niezależnie od różnych danych wejściowych.

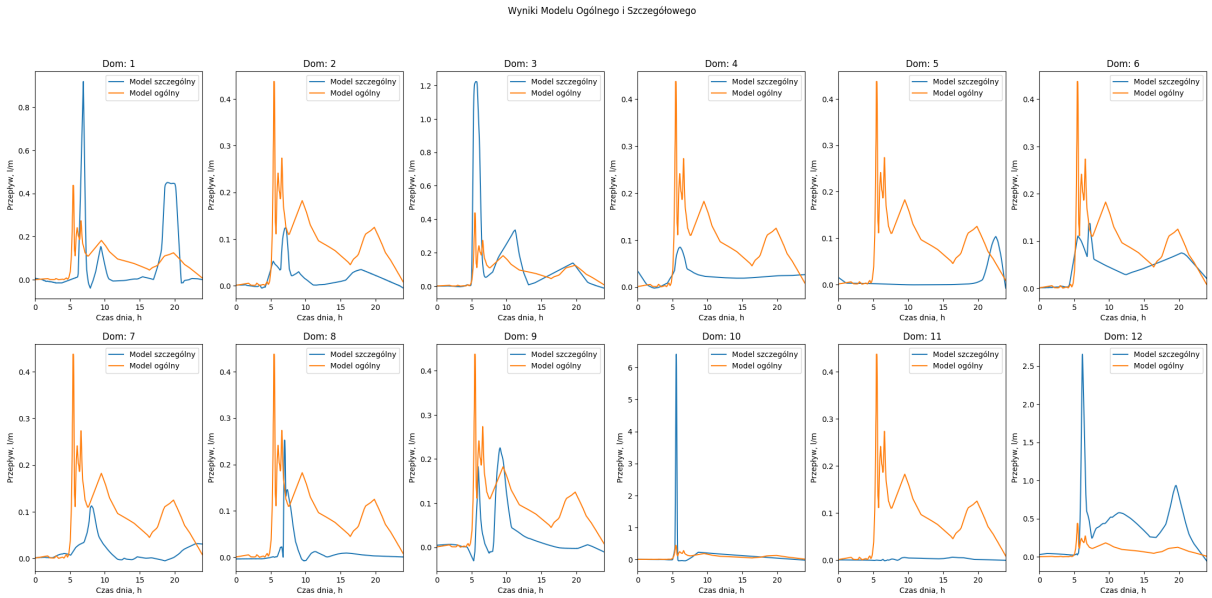
W ramach procesu testowania różnych konfiguracji sieci neuronowej zaproponowano eksplorację wydajności modeli przy różnorodnych kombinacjach wejść. Celem tego podejścia było zbadanie, jak zmiana danych wejściowych wpłynie na zdolność modelu do nauki i generalizacji przewidywania przepływu. Poniżej przedstawiono zestawienie modeli, które zostały uwzględnione w analizie:

1. Model A
   (a) Wejścia: Dzień tygodnia, pora dnia
   (b) Wyjście: Przepływ
2. Model B
   (a) Wejścia: pora dnia
   (b) Wyjście: Przepływ
3. Model C
   (a) Wejścia: pora roku, dzień tygodnia, pora dnia
   (b) Wyjście: Przepływ
4. Model D
   (a) Wejścia: pora roku, pora dnia
   (b) Wyjście: Przepływ

Wyniki te dostarczą wglądu w to, które wejścia są najbardziej wartościowe dla modelowania przepływu oraz czy dodanie dodatkowych informacji kontekstowych przyczynia się do znaczącej poprawy wyników predykcyjnych.

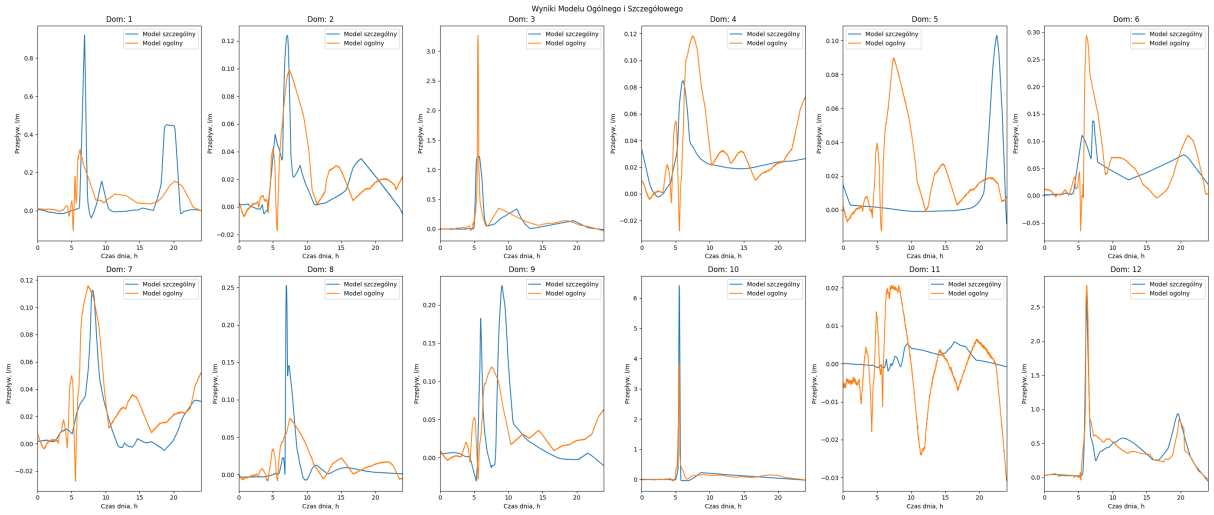
3.4 Walidacja i próby dostrajania (?)

W celu weryfikacji poprawności i efektywności opracowanego modelu sieci neuronowej, przeprowadzono porównanie modelu nauczonych na danych ze wszystkich 12 domostw z modelami utworzonymi dla każdego z tych domów osobno. Taki eksperyment miał na celu ocenę zdolności generalizacji modelu nauczonych na zbiorze 12 domostw w porównaniu z modelami specyficznymi dla poszczególnych domów.



Rysunek 3.4: Porównianie modelu ogólnego z modelami szczególnymi

W celu dalszego zwiększenia skuteczności modelu sieci neuronowej zaproponowano wprowadzenie dodatkowego wejścia do systemu – tygodniowego zużycia. Implementacja tego rozwiązania została przeprowadzona w specyficzny sposób, mający na celu uniknięcie przekształcenia tego parametru w niezamierzony label identyfikujący poszczególne domy. W fazie uczenia modelu, do każdego tygodnia przypisywano sumę zużycia zarejestrowanego w tym okresie. Natomiast w fazie testowania, model otrzymywał średnią wartość tygodniowego zużycia. Celem tej strategii było umożliwienie modelowi korzystania z danych historycznych zużycia w sposób, który poprawiałby jego zdolność do przewidywania, jednocześnie zachowując elastyczność i możliwość generalizacji wyników na różne domostwa.



Rysunek 3.5: Porównianie modelu ogólnego z modelami szczególnymi po dodaniu kolejnego wejścia do sieci

Analizując charakterystykę modelu ogólnego i szczególnego dla każdego z domostw, przedstawioną na Rysunku 3.5, można zauważyć, że dla domostwa nr 10 i 12 model ogólny wykazał się wysoką zgodnością z modelem szczegółowym, co świadczy o jego zdolności do precyzyjnego odwzorowania charakterystyki przewidywania przepływu w ciągu dnia. W przypadku tych dwóch domów, przebieg przewidywań dla obu modeli jest podobny, co wskazuje na to, że model ogólny efektywnie uchwycił dynamikę zużycia charakterystyczną dla tych konkretnych domostw. Natomiast w kontekście Domu nr 5, wyniki ukazują, że model ogólny miał znaczne trudności z dopasowaniem się do wzorców przepływu.

W celu dokładniejszej oceny i porównania efektywności modelu ogólnego, nauczonego na danych z 12 domostw, z modelami szczegółowymi, nauczonymi dla poszczególnych domów, zastosowano wskaźnik błędu średniokwadratowego. MSE, obliczany jako średnia kwadratów różnic między wartościami przewidywanymi przez model a rzeczywistymi danymi, posłużył jako miara odchylenia modelu ogólnego od wyników modeli szczegółowych. W tym kontekście, niższa wartość MSE wskazywała na lepszą zgodność modelu ogólnego z wynikami modeli szczegółowych, sugerując, że model ogólny skuteczniej generalizuje dane, zbliżając się do precyzji modeli trenowanych na danych z pojedynczych domostw.

Tabela 3.2: Porównianie wartości MSE dla każdego obu modelów

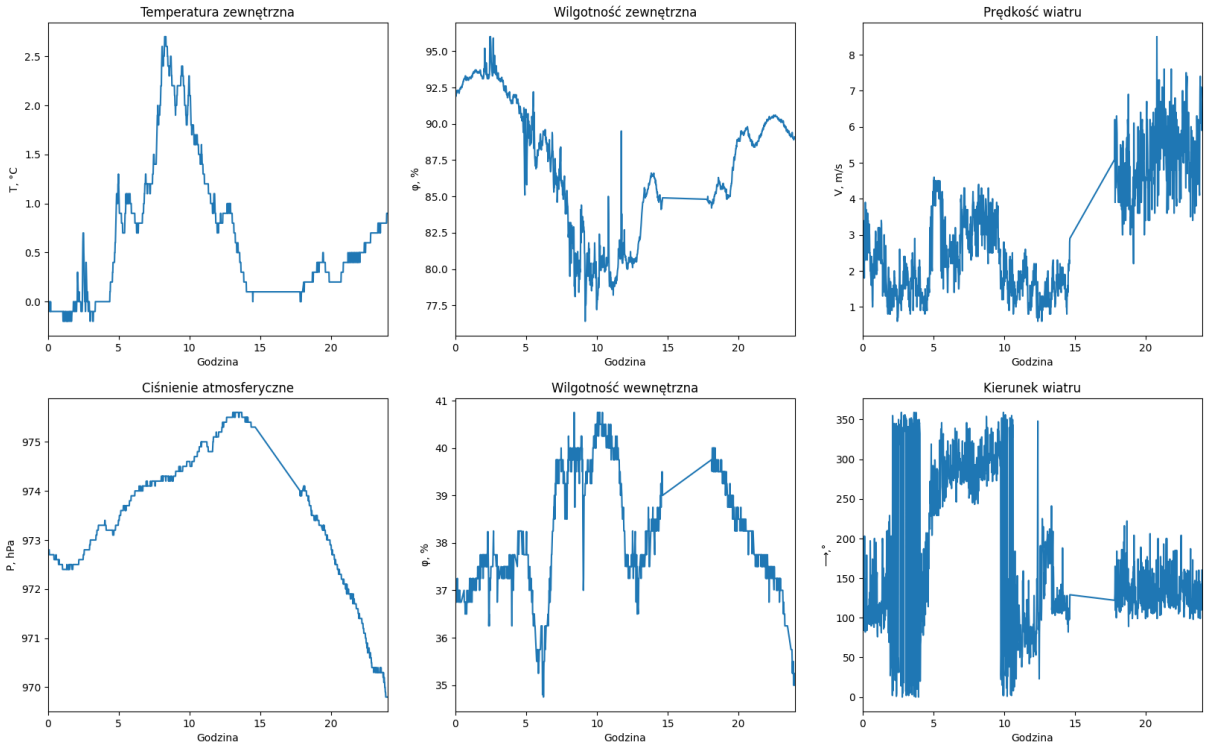
MSE	DOMOSTWO											
	1	2	3	4	5	6	7	8	9	10	11	12
Model 1	0.022	0.0074	0.037	0.0067	0.011	0.0039	0.0084	0.0095	0.0072	0.22	0.010	0.19
Model 2	0.019	0.00053	0.034	0.00085	0.0013	0.0025	0.00055	0.00069	0.0027	0.075	0.00013	0.019

Analizując przedstawione dane w tabeli, można zauważyć, że po dodaniu dodatkowego wejścia do systemu, czyli tygodniowego zużycia, model 2 (Model z dodatkowym wejściem) osiągnął znacznie lepsze wyniki w porównaniu z modelem bazowym. Wartości błędu średniokwadratowego dla modelu 2 są niższe w porównaniu do modelu 1 dla każ-

dego z domostw, co wskazuje na poprawę dokładności predykcji. Na przykład, dla Domu nr 2, MSE zmniejszyło się z 0.0074 w modelu 1 do 0.00053 w modelu 2, co jest znaczącą poprawą. Podobne znaczące redukcje można zaobserwować w przypadku Domu nr 12, gdzie MSE spadło z 0.19 do 0.019

Rozdział 4

Pogoda



Rysunek 4.1: Porównanie warunkuch atmosferycznych na przestrzeni dnia

W celu zbadania wpływu wielkości modelu na wyniki oraz sprawdzenia ważności różnych wejść, zaproponowano w badaniu dwa modele o różnej złożoności architektury. Oba modele korzystały z tej samej funkcji harmonogramowania tempa uczenia, która redukowała szybkość uczenia po czwartej epoce, oraz z tych samych parametrów kompilacji, w tym optymalizatora Adam z początkową szybkością uczenia 0.001, funkcji straty MSE Pierwszy model składał się z mniejszej liczby warstw i neuronów: warstwa normalizująca, trzy warstwy gęste z odpowiednio 64, 32 i 1 neuronami, używając funkcji aktywacji 'relu' dla pierwszych dwóch warstw i 'linear' dla warstwy wyjściowej. Drugi model był znacznie większy, zawierając więcej warstw i neuronów: warstwa normalizująca, sześć warstw

gęstych o zwiększającej się liczbie neuronów: 1024, 512, 256, 128, 64, 32, zakończonych warstwą wyjściową z 1 neuronem, używając funkcji aktywacji 'relu' dla warstw ukrytych i 'linear' dla warstwy wyjściowej. Oba modele były trenowane przez 50 epok z rozmiarem partii równym 32. Celem porównania tych dwóch modeli było ustalenie, czy zwiększenie liczby warstw i neuronów w modelu wpłynie na jego

W ramach procesu weryfikacji skuteczności zastosowanych modeli sieci neuronowych, dane zostały podzielone w proporcji 80% do 20%. Ta strategia podziału danych miała na celu zapewnienie solidnej bazy do nauki modeli oraz efektywnej oceny ich wydajności. Model numer 1 wykazał się niższą skutecznością w porównaniu do modelu numer 2. Świadczy o tym wartość błędu średniokwadratowego, która dla modelu pierwszego wyniosła 0.0668, natomiast dla modelu drugiego było to znacząco niższe, a mianowicie 0.0237. Ta różnica w wartościach MSE wskazuje na wyższą precyzję i efektywność modelu numer 2 w procesie uczenia i oceny na podstawie dostępnych danych.

4.1 Permutacyjna Ważność Cech

Permutacyjna Ważność Cech, to technika stosowana w uczeniu maszynowym do oceny znaczenia poszczególnych wejść dla modelu predykcyjnego. Metoda ta jest stosowana zarówno dla modeli klasyfikacyjnych, jak i regresyjnych. Proces ten rozpoczyna się od trenowania modelu na oryginalnym zestawie danych, co pozwala na ustalenie bazowej wydajności modelu. Następnie przeprowadza się permutację każdej cechy z osobna w zbiorze testowym, losowo mieszając jej wartości, podczas gdy wszystkie inne cechy pozostają niezmienione. Po dokonaniu permutacji, model jest ponownie oceniany na zmodyfikowanym zbiorze danych. Wyniki tej oceny są następnie porównywane z wynikami uzyskanymi na oryginalnym, niezmodyfikowanym zbiorze. Różnica w wydajności modelu, taka jak spadek dokładności w klasyfikacji lub wzrost błędu średniokwadratowego w regresji, jest wykorzystywana do oceny ważności danej cechy. Im większy spadek wydajności, tym większa uważana jest ważność tej cechy dla modelu.

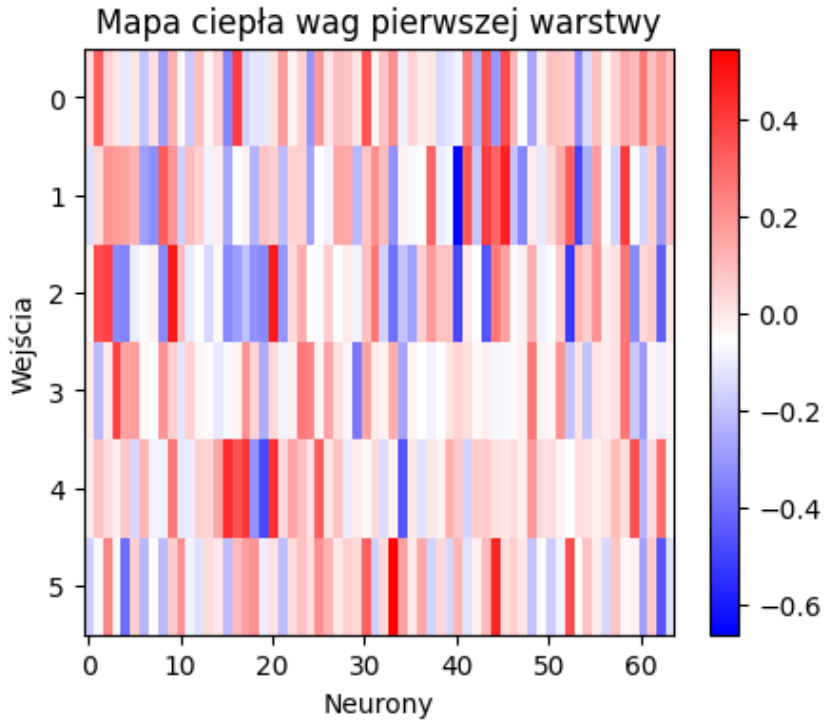
W kontekście wykorzystania PFI do oceny ważności cech w modelu, istotne jest zwrócenie uwagi na kwestię korelacji między danymi. PFI opiera się na permutacji pojedynczych cech, co oznacza zmianę wartości jednej zmiennej niezależnie od pozostałych. W przypadku, gdy dane są silnie skorelowane, taka metoda permutacji może prowadzić do błędnych wniosków

W celu implementacji techniki PFI w pracy, został wykorzystany obiekt PermutationImportance z biblioteki sklearn. Zastosowanie tego dedykowanego narzędzia umożliwiło nie tylko dokładną, ale i wydajną realizację tej metody, co pozwoliło na jej skuteczną integrację z procesem badawczym.

Tabela 4.1: Porównianie wartości MSE dla każdego obu modelów

	Temperatura zewnętrzna	Wilgotność zewnętrzna	Ciśnienie atmosferyczne	Prędkość wiatru	Kierunek wiatru	Wilgotność wewnętrzna
Model krótki						
Model długi						

4.2 Badanie wag wejściowych pierwszej warsty



Rysunek 4.2: Mapa ciepła wag pierwszej warstwy dla modelu o uproszczonej architekturze

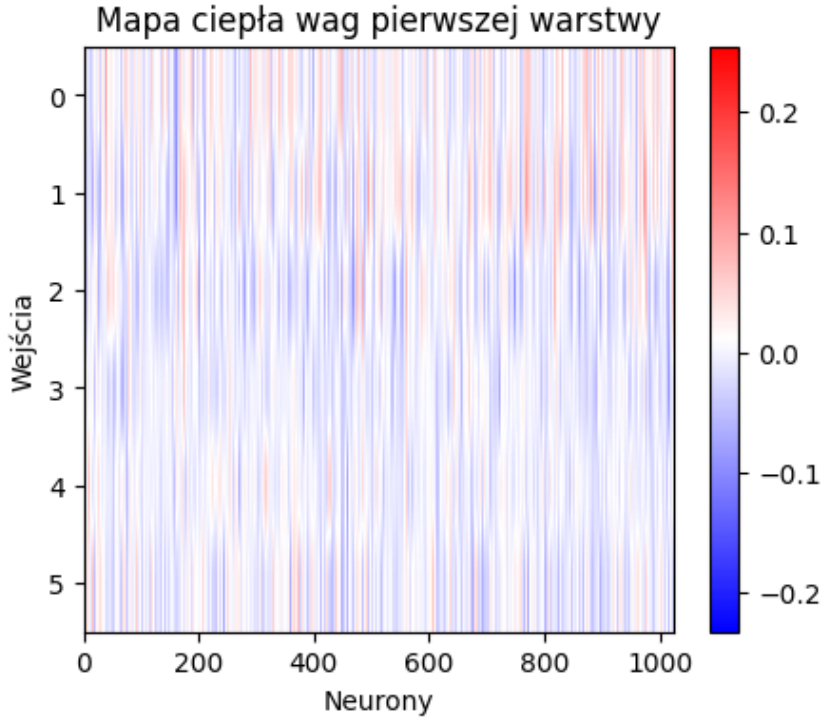
Mapa ciepła wag pierwszej warstwy sieci neuronowej stanowi cenne narzędzie analityczne, pozwalające na wizualną interpretację i zrozumienie wpływu dużej liczby cech na proces uczenia. Jest to instrument szczególnie użyteczny w kontekście wysokowymiarowych zbiorów danych, gdzie tradycyjne metody analizy mogą okazać się niewystarczające. W przedstawionym przypadku, analiza mapy ciepła nie ujawnia istotnych anomalii w rozkładzie wag. Obserwuje się jedynie sporadyczne wartości, które odstają od średnich wag, lecz nie osiągają one poziomu znacząco wpływającego na wyniki modelu.

W celu dokładnej analizy ważności poszczególnych wag wejściowych sieci neuronowej przeprowadzono obliczenie średniej wartości wagi dla każdego z wejść. Procedura ta umożliwiła identyfikację względnej ważności cech poprzez porównanie ich przeciętnego wpływu na aktywację neuronów w modelu. Następnie, aby umożliwić porównywalność wyników niezależnie od ich pierwotnej skali, dokonano normalizacji obliczonych średnich wag.

Tabela 4.2: Todo

Temperatura zewnętrzna	Wilgotność zewnętrzna	Ciśnienie atmosferyczne	Prędkość wiatru	Kierunek wiatru	Wilgotność wewnętrzna

Analizy bardziej złożonych modeli sieci neuronowych, metoda interpretacji wag pierwszej warstwy może okazać się nieefektywna. Ze względu na zwiększoną głębokość i złożoność architektury, wagi w pierwszej warstwie tracą bezpośrednią i jednoznaczną interpretowalność, która jest charakterystyczna dla prostszych modeli. W modelach rozbudowanych, cechy wejściowe przechodzą przez wiele warstw transformacji, co skutkuje utratą bezpośredniego powiązania między wagami pierwszej warstwy a wynikowymi decyzjami modelu. W efekcie, interpretacja tych wag może nie odzwierciedlać faktycznego wpływu poszczególnych cech na decyzje modelu, co jest spowodowane nakładaniem się, transformacją i połączeniem informacji w kolejnych warstwach sieci.



Rysunek 4.3: Mapa ciepła wag pierwszej warstwy dla modelu o rozbudowanej architekturze

Tabela 4.3: Todo

Temperatura zewnętrzna	Wilgotność zewnętrzna	Ciśnienie atmosferyczne	Prędkość wiatru	Kierunek wiatru	Wilgotność wewnętrzna

Rozdział 5

Modelowanie zbiornika CWU

5.1 Metodologia

5.1.1 Opis matematyczny modelu

$$\frac{dT^3_{wo}}{dt} = b^3_1 F_z(T_{zi} - T^3_{wo}) - b^3_2 F_w(T^3_{wo} - T^2_{wo}) - b^4_3(T^3_{wo} - T_{ot}) \tag{5.1}$$

$$\frac{dT_{zi}}{dt} = p_1 Q_g - p_2 F_z(T_{zi} - T^3_{wo}) - p_3(T_{zi} - T_{ot}) \tag{5.2}$$

$$\frac{dT^2_{wo}}{dt} = b^2_1 F_z(T_{zi} - T^2_{wo}) - b^2_2 F_w(T^2_{wo} - T^1_{wo}) - b^3_2(T^2_{wo} - T_{ot}) - b^4_2(T^2_{wo} - T^1_{wo}) + b^5_2(T^3_{wo} - T^2_{wo}) \tag{5.3}$$

$$\frac{dT^1_{wo}}{dt} = -b^1_1 F_w(T^1_{wo} - T_{wi}) - b^3_1(T^1_{wo} - T_{ot}) + b^5_1(T^2_{wo} - T^1_{wo}) \tag{5.4}$$

Przedstawienie modelu warstwowego, równań stanu, pokazanie wyników symulacji modelu

5.2 Wyniki symulacji

Krótka wstawka kodu w linii tekstu jest możliwa, np. `int a;` (biblioteka `listings`). Dłuższe fragmenty lepiej jest umieszczać jako rysunek, np. kod na rys 3.3, a naprawdę długie fragmenty – w załączniku.

Rozdział 6

Optymalizacja

6.1 Funkcja kosztów

$$G = \int p_1 Q_g \, dt \tag{6.1}$$

6.2 Funkcja komfortu

$$J = \int (T_{wo} - T_{wym})^2 \left| \frac{\text{sign}(T_{wo} - T_{wym} - \delta) + \text{sign}(T_{wo} - T_{wym} + \delta)}{2} \right| dt \tag{6.2}$$

- sposób testowania w ramach pracy (np. odniesienie do modelu V)
- organizacja eksperymentów
- przypadki testowe zakres testowania (pełny/niepełny)
- wykryte i usunięte błędy
- opcjonalnie wyniki badań eksperymentalnych

Tabela 6.1: Nagłówek tabeli jest nad tabelą.

	metoda						
			alg. 3			alg. 4, $\gamma = 2$	
ζ	alg. 1	alg. 2	$\alpha = 1.5$	$\alpha = 2$	$\alpha = 3$	$\beta = 0.1$	$\beta = -0.1$
0	8.3250	1.45305	7.5791	14.8517	20.0028	1.16396	1.1365
5	0.6111	2.27126	6.9952	13.8560	18.6064	1.18659	1.1630
10	11.6126	2.69218	6.2520	12.5202	16.8278	1.23180	1.2045
15	0.5665	2.95046	5.7753	11.4588	15.4837	1.25131	1.2614
20	15.8728	3.07225	5.3071	10.3935	13.8738	1.25307	1.2217
25	0.9791	3.19034	5.4575	9.9533	13.0721	1.27104	1.2640
30	2.0228	3.27474	5.7461	9.7164	12.2637	1.33404	1.3209
35	13.4210	3.36086	6.6735	10.0442	12.0270	1.35385	1.3059
40	13.2226	3.36420	7.7248	10.4495	12.0379	1.34919	1.2768
45	12.8445	3.47436	8.5539	10.8552	12.2773	1.42303	1.4362
50	12.9245	3.58228	9.2702	11.2183	12.3990	1.40922	1.3724

Rozdział 7

Podsumowanie i wnioski

- uzyskane wyniki w świetle postawionych celów i zdefiniowanych wyżej wymagań
- kierunki ewentualnych danych prac (rozbudowa funkcjonalna . . .)
- problemy napotkane w trakcie pracy

Bibliografia

[1] Imię Nazwisko i Imię Nazwisko. *Tytuł strony internetowej*. 2021. URL: `http://gdzies/w/internecie/internet.html` (term. wiz. 30. 09. 2021).

[2] Imię Nazwisko, Imię Nazwisko i Imię Nazwisko. „Tytuł artykułu konferencyjnego". W: *Nazwa konferecji*. 2006, s. 5346–5349.

[3] Imię Nazwisko, Imię Nazwisko i Imię Nazwisko. „Tytuł artykułu w czasopiśmie". W: *Tytuł czasopisma* 157.8 (2016), s. 1092–1113.

[4] Imię Nazwisko, Imię Nazwisko i Imię Nazwisko. *Tytuł książki*. Warszawa: Wydawnictwo, 2017. ISBN: 83-204-3229-9-434.

Dodatki

Spis skrótów i symboli

DNA kwas deoksyrybonukleinowy (ang. *deoxyribonucleic acid*)

MVC model – widok – kontroler (ang. *model–view–controller*)

N liczebność zbioru danych

μ stopnień przyleżności do zbioru

$\mathbb{E}$ zbiór krawędzi grafu

$\mathcal{L}$ transformata Laplace'a

Źródła

Jeżeli w pracy konieczne jest umieszczenie długich fragmentów kodu źródłowego, należy je przenieść w to miejsce.

```
1 if (_nClusters < 1)
2     throw std::string ("unknown number of clusters");
3 if (_nIterations < 1 and _epsilon < 0)
4     throw std::string ("You should set a maximal number of
          iteration or minimal difference — epsilon.");
5 if (_nIterations > 0 and _epsilon > 0)
6     throw std::string ("Both number of iterations and minimal
          epsilon set — you should set either number of iterations
          or minimal epsilon.");
```

Lista dodatkowych plików, uzupełniających tekst pracy

W systemie do pracy dołączono dodatkowe pliki zawierające:

- źródła programu,
- dane testowe,
- film pokazujący działanie opracowanego oprogramowania lub zaprojektowanego i wykonanego urządzenia,
- itp.

Spis rysunków

Spis tabel