



**Politechnika  
Śląska**

## **PROJEKT INŻYNIERSKI**

Tytuł pracy dyplomowej inżynierskiej

**Jakub KULA**

**Nr albumu: 296849**

**Kierunek:** Automatyka i Robotyka

**Specjalność:** Technologie Informatyczne

**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 słów (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

<b>1</b>	<b>Wstęp</b>	<b>1</b>
1.1	Cel i zakres pracy . . . . .	1
1.2	Aktualny stan wiedzy . . . . .	1
1.3	Charakterystyka rozdziałów . . . . .	1
1.4	Wkład autora . . . . .	1
<b>2</b>	<b>Zastosowane narzędzia w pracy</b>	<b>3</b>
2.1	Python . . . . .	3
2.2	Tensorflow . . . . .	3
2.3	Inne biblioteki . . . . .	4
2.3.1	Pandas . . . . .	4
2.3.2	Matplotlib . . . . .	4
2.3.3	Numpy . . . . .	4
2.3.4	Skit-learn . . . . .	4
2.4	CUDA toolkit . . . . .	4
<b>3</b>	<b>Modelowanie sieci neuronowej</b>	<b>5</b>
3.1	Metodologia projektowania modelu sieci neuronowej . . . . .	5
3.2	Dane wejściowe i proces ich przetwarzania . . . . .	6
3.3	Projektowanie i ocena modeli . . . . .	8
<b>4</b>	<b>Modelowanie zbiornika CWU</b>	<b>11</b>
4.1	Metodologia . . . . .	11
4.1.1	Opis matematyczny modelu . . . . .	11
4.2	Wyniki symulacji . . . . .	11
<b>5</b>	<b>Optymalizacja</b>	<b>13</b>
5.1	Funkcja kosztów . . . . .	13
5.2	Funkcja komfortu . . . . .	13
<b>6</b>	<b>Podsumowanie i wnioski</b>	<b>15</b>

Bibliografia	17
Spis skrótów i symboli	21
Źródła	23
Lista dodatkowych plików, uzupełniających tekst pracy	25
Spis rysunków	27
Spis tabel	29

# Rozdział 1

## Wstęp

### 1.1 Cel i zakres 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 głównego języka programowania zastosowanego w projekcie, wiązał się z postawieniem pewnych wymagań. Pierwszym z tych wymagań była dostępność dedykowanej biblioteki do uczenia maszynowego, która posiada narzędzia do efektywnej pracy nad modelami czy ich testowanie. Użycie biblioteki która jest dobrze utrzymana zapewni ogromne wsparcie społeczności, które może okazać się nieocenione w procesie nauki czy rozwiązywania problemów.

Kolejnym wymaganiem jest aby wybrana technologia była ciągle wspierana i aktualizowana. Machine learning jest aktualnie jedną z najszybciej rozwijających się dziedzin programowania, co wiąże się z szybkimi zmianami.(DOPISAC COŚ TUTAJ)

- R
- Python

R jest językiem skoncentrowanym na analizie danych i statystyce. Posiada on bardzo bogaty ekosystem jednak może stanowić to przyczynę wielu konfliktów pomiędzy pakietami. Największą wadą tego języka 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 bibliotekę standardową jak i bardzo liczne zewnętrzne biblioteki. 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 bibliotek do uczenia maszynowego i głębokiego w Pythonie. Głównym konkurentem tensorflow jest PyTorch który jest roz-

wijany przez Facebook.

## **2.3 Inne biblioteki**

### **2.3.1 Pandas**

### **2.3.2 Matplotlib**

### **2.3.3 Numpy**

### **2.3.4 Sckit-learn**

## **2.4 CUDA toolkit**

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

# Rozdział 3

## Modelowanie sieci neuronowej

### 3.1 Metodologia projektowania modelu sieci neuronowej

Wstęp teoretyczny o modelowaniu, opisanie rzeczy takich jak, warstwy, neurony, funkcje aktywacji, funkcje kosztu, optymalizator, liczba epok, batch size, walidacja, funkcja strat

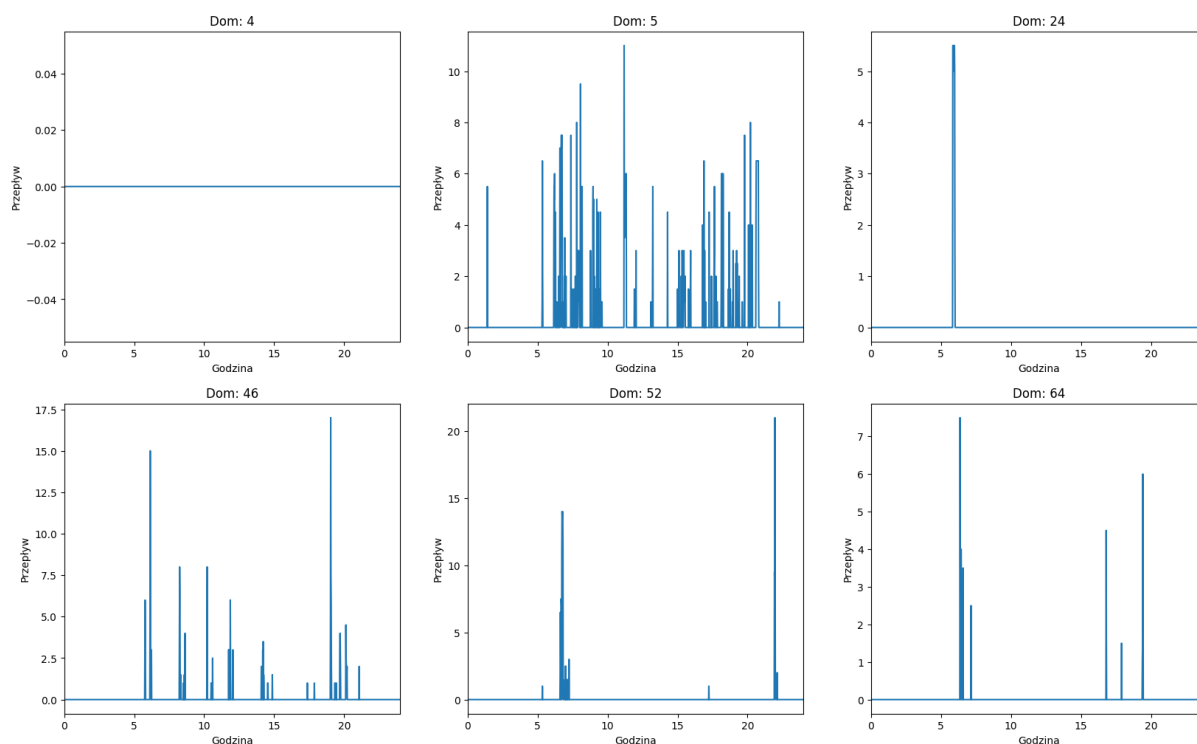
## 3.2 Dane wejściowe i proces ich przetwarzania

Projekt wykorzystuje dane zebrane przez instytut ASHRAE (American Society of Heating, Refrigerating and Air-Conditioning Engineer) w 2018 roku. Informacje o zużyciu wody zostały pozyskane z 77 domostwo znajdujących się na terenie Kanady, zamieszkanym przez dwoje osób dorosłych oraz dwójkę dzieci oraz posiadających podstawowe urządzenia AGD takie jak pralka czy zmywarka. Dane były zbierane przez 16 tygodni, w cyklach trwających 4 tygodnie, aby równomiernie obejmować każdą porę roku. Próbkowanie danych odbywało się co minute, nieprzerwanie przez całą dobę. Całość danych została podzielona na 73 oddzielne pliki w formacie .csv. W każdym pliku znajdował się timestamp w formacie RR-MM-DD HH:MM:SS, który określał dzień i godzinę pomiaru danych. Oprócz tego zawierał informacje o zużyciu wody, oraz temperaturę otoczenia. Dodatkowo dane były podzielone na poszczególne pory roku.



Rysunek 3.1: Podpis rysunku zawsze pod rysunkiem.

Pierwszy wykres przedstawia 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 przedstawia nieregularne piki o nierównomiernym rozkładzie. Okresami o zwiększonym przepływie są godziny 7-13 oraz 14-15. Okresy mniejszej aktywności możemy zaobserwować w godzinach późno popołudniowych oraz nocnych. Drugi i trzeci wykres przedstawiają zbliżenie na godzinie 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.



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.

---

```
1 for f in csv_files:
2     dataset = pd.read_csv(f)
3     for i in range(len(dataset)):
4         dt = dataset.loc[i, "Summer_Timestamps"]
5         data, time = dt.split('□')
6         year, month, day = (int(x) for x in data.split('-'))
7         ans = datetime.date(year, month, day)
8         dzien_tygodnia = dni_tygodnia_mapa[ans.strftime("%A")]
9
10        hours, minutes, null = time.split(":")
11        time = (int(hours)*60+int(minutes))/(60*24)
12
13        przeplyw = dataset.loc[i, "Summer_Water_Consumption"]
14
15        dane.loc[len(dane)] = {'Pora_roku': 2,
16                               'Dzien_tygodnia': dzien_tygodnia,
17                               'Czas_dnia': time,
18                               'Przeplyw': przeplyw}
```

---

Rysunek 3.3: Fragment skryptu przetwarzającego dane.

W ramach procesu dostawania formatu aby przystosować go do wymagań tensorflow, dzień tygodnia został zamieniona na etykiety liczbową, która przyjmuje wartość 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 zimą. 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.

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'

Tabela 3.1: Hiperparametry Sieci Neuronowej

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

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

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

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.





# Rozdział 4

## Modelowanie zbiornika CWU

### 4.1 Metodologia

#### 4.1.1 Opis matematyczny modelu

$$\frac{dT_{wo}^3}{dt} = b_1^3 F_z(T_{zi} - T_{wo}^3) - b_2^3 F_w(T_{wo}^3 - T_{wo}^2) - b_3^4(T_{wo}^3 - T_{ot}) \quad (4.1)$$

$$\frac{dT_{zi}}{dt} = p_1 Q_g - p_2 F_z(T_{zi} - T_{wo}^3) - p_3(T_{zi} - T_{ot}) \quad (4.2)$$

$$\frac{dT_{wo}^2}{dt} = b_1^2 F_z(T_{zi} - T_{wo}^2) - b_2^2 F_w(T_{wo}^2 - T_{wo}^1) - b_2^3(T_{wo}^2 - T_{ot}) - b_2^4(T_{wo}^2 - T_{wo}^1) + b_2^5(T_{wo}^3 - T_{wo}^2) \quad (4.3)$$

$$\frac{dT_{wo}^1}{dt} = -b_1^1 F_w(T_{wo}^1 - T_{wi}) - b_1^3(T_{wo}^1 - T_{ot}) + b_1^5(T_{wo}^2 - T_{wo}^1) \quad (4.4)$$

Przedstawienie modelu warstwowego, równań stanu, pokazanie wyników symulacji modelu

### 4.2 Wyniki symulacji

Krótką wstawkę kodu w linii tekstu jest możliwa, np. **int a**; (biblioteka **listings**). Dłuższe fragmenty lepiej jest umieszczać jako rysunek, np. kod na rys 4.1, a naprawdę długie fragmenty – w załączniku.

---

```
1 class test : public basic
2 {
3     public:
4         test (int a);
5         friend std::ostream operator<<(std::ostream & s,
6                                         const test & t);
7     protected:
8         int _a;
9
10 };
```

---

Rysunek 4.1: Pseudokod w `listings`.

# Rozdział 5

## Optymalizacja

### 5.1 Funkcja kosztów

$$G = \int p_1 Q_g dt \quad (5.1)$$

### 5.2 Funkcja komfortu

$$J = \int (T_{wo} - T_{wym})^2 \left| \frac{\text{sign}(T_{wo} - T_{wym} - \delta) + \text{sign}(T_{wo} - T_{wym} + \delta)}{2} \right| dt \quad (5.2)$$

- 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 5.1: Nagłówek tabeli jest nad tabelą.

$\zeta$	metoda						
	alg. 1	alg. 2	alg. 3			alg. 4, $\gamma = 2$	
			$\alpha = 1.5$	$\alpha = 2$	$\alpha = 3$	$\beta = 0.1$	$\beta = -0.1$
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 6

## 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 konferencji*. 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

$\mu$  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

3.1	Podpis rysunku zawsze pod rysunkiem. . . . .	6
3.2	Porównanie przepływów dla przykładowych domów w dniu 05/02/2018 . .	7
3.3	Fragment skryptu przetwarzającego dane. . . . .	8
4.1	Pseudokod w <code>listings</code> . . . . .	12



# Spis tabel

3.1	Hiperparametry Sieci Neuronowej . . . . .	9
5.1	Nagłówek tabeli jest nad tabelą. . . . .	14