Politechnika Łódzka
Instytut Informatyki

RAPORT Z PROJEKTU KOŃCOWEGO

STUDIÓW PODYPLOMOWYCH

DATA SCIENCE

Zastosowanie metod uczenia maszynowego do predykcji opadów na podstawie ciśnienia, temperatury i wilgotności.

Wydział Fizyki Technicznej, Informatyki i Matematyki Stosowanej
Opiekun: dr inż. Jakub Walczak
Uczestnik: mgr inż. Wiktoria Świech

Łódź, 2024

Instytut Informatyki
90-924 Łódź, ul. Wólczańska 215, budynek B9
tel. 042 631 27 97, 042 632 97 57, fax 042 630 34 14 email: office@ics.p.lodz.pl

Spis treści

1 Wprowadzenie

Celem projektu jest przeprowadzenie analizy własnego zbioru danych, który zawiera temperaturę, wilgotność i ciśnienie zarejestrowaną przez czujnik Internetu Rzeczy (czujnik IoT) dla jednej lokalizacji w Łodzi na wysokości 236 m n.p.m. Realizowany projekt ma za zadanie zbadać, czy stosując tradycyjne techniki uczenia maszynowego oraz rekurencyjne sieci neuronowe możliwa jest predykcja opadów. W tym celu, oprócz własnego zbioru, zostaną również wykorzystane dane dotyczące sumy opadów pozyskane ze strony OpenWeather[1] dla tej samej lokalizacji, w której znajdował się czujnik IoT.

Zastosowany czujnik IoT został wyprodukowany przez firmę Aqara, która wykorzystała czujnik klasy przemysłowej dostarczony przez firmę Sensirion. Czujnik posiada dokładność wykrywania wilgotności ± 3% oraz wykrywania temperatury ± 0,3 °C [1]. Producent nie zamieścił na stronie internetowej informacji o pomiarze ciśnienia, ale z danych wysyłanych przez czujnik wynika, że dokładność tej wielkości to ± 1 hPa. Opisany czujnik IoT przesyła dane przez technologie ZigBee do oprogramowania Home Assistance działającego na lokalnym serwerze.

Do predykcji opadów deszczu zostały zastosowane tradycyjne techniki uczenia maszynowego (ML) tj.: regresor wektorów nośnych (SVR), regresor drzewa decyzyjnego (DTR), regresor lasu losowego (RFR) oraz regresja liniowa (LR). Ze względu na to, że predykcja opadów to zagadnienie związane z szeregami czasowymi, zastosowano także rekurencyjne sieci neuronowe LSTM *(ang. Long Short-Term Memory)*. Na wybór tradycyjnych technik uczenia maszynowego oraz typu sieci neuronowej miał wpływ przegląd literatury z pozycji [2], [3], [4], [5], [6].

W etapie końcowym projektu zostanie udzielona odpowiedź na przedstawiony problem badawczy, założone cele projektu opisane w Rozdziale 4 oraz przedstawione podsumowanie i wnioski.

2 Stos technologiczny

Projekt został zrealizowany w języku Python przy użyciu notatnika JupyterLab. W Tabeli 1 przedstawiono wykorzystane biblioteki tj. scikit-learn [7], pandas [8], plotly [9], keras [10] i numpy [11] oraz ich zastosowania. Bibliotekę scikit-learn wykorzystano do budowy wspomnianych modeli tradycyjnych technik uczenia maszynowego, strojenia ich hiperparametrów oraz do ich ewaluacji. Bibioteka pandas posłużyła do przeprowadzenia analizy i do manipulacji danymi. Plotly wykorzystano do generowania interaktywnych wizualizacji danych. Z pomocą biblioteki keras został zbudowany model rekurencyjnych sieci neuronowych LSTM. Natomiast biblioteka numpy została użyta do obsługi wielowymiarowych tablic i macierzy oraz do wykonania operacji logarytmu naturalnego.

[1] Dane pogodowe ze strony OpenWeather dostępne pod adresem: https://openweathermap.org/

Tabela 1: Biblioteki języka Python wykorzystane do realizacji projektu.

Nazwa biblioteki	**Zastosowanie**
scikit-learn	Predykcyjna analiza danych, w tym budowa modelów tradycyjnych technik uczenia maszynowego (SVR, DT, RFR, LR), strojenie modelu poprzez dobór hiperparametrów, dokonywanie predykcji oraz ewaluacji modelu.
pandas	Analiza i manipulacja danych.
plotly	Generowanie interaktywnych wizualizacji danych.
keras	Budowa modelu rekurencyjnych sieci neuronowych LSTM.
numpy	Obsługa wielowymiarowych tablic i macierzy oraz wykonanie operacji logarytmu naturalnego.

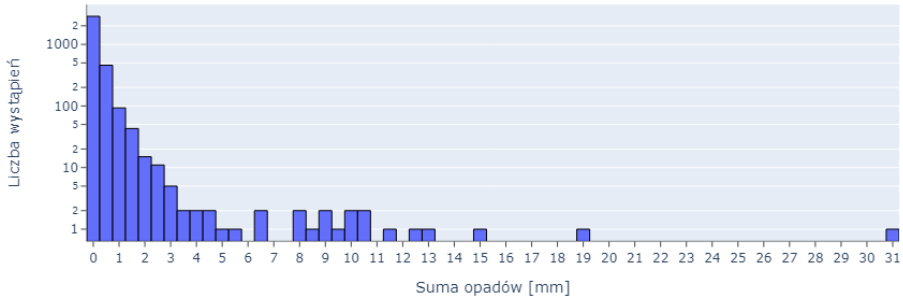
3 Charakterystyka danych

Dane zarejestrowane przez czujnik IoT to temperatura [°C], ciśnienie [hPa] oraz wilgotność [%] w okresie od 20.09.2023 godziny 00:00 do 12.02.2024 godziny 23:00. Informacja o każdej zmianie jednej z trzech wielkości jest wysyłana przez czujnik IoT na serwer, gdzie pomiary są przechowywane. W celu przeprowadzenia analizy dane zostały pobrane z serwera i przetworzone w języku programowania Python. Do przygotowanych danych została dodana informacja o sumie opadów pozyskana ze strony OpenWeather dla tej samej lokalizacji, w której znajdował się czujnik IoT. Poszczególne kroki podjęte w procesie przetwarzania danych zostały opisane w Rozdziale 5. Charakterystyka sumy opadów, temperatury, wilgotności i ciśnienia w zdefiniowanym zakresie dat została przedstawiona odpowiednio na Rysunku 1-4 oraz zbiorczo w Tabeli 2. Korelacje pomiędzy poszczególnymi wielkościami zostały przedstawione na Rysunku 5.

Tabela 2: Charakterystyka danych.

Charakterystyka	**Ciśnienie [hPa]**	**Temperatura [°C]**	**Wilgotność [%]**	**Suma opadów [mm]**
Minimalna wartość	950	-11,70	37,75	0,00
Maksymalna wartość	1 008	25,93	98,48	30,99
Średnia	982	5,67	82,54	0,19
Odchylenie standardowe	11	6,62	9,22	0,98

Rysunek 1 przedstawia histogram sumy opadów. Oś pionowa tego wykresu została przedstawiona w **skali logarytmicznej**, ponieważ istnieją duże dysproporcje w liczbie wystąpień opadów. Z danych wynika, że zarejestrowano 2 666 pomiarów, kiedy nie wystąpiły opady (0 mm), co stanowi 76% wszystkich obserwacji.

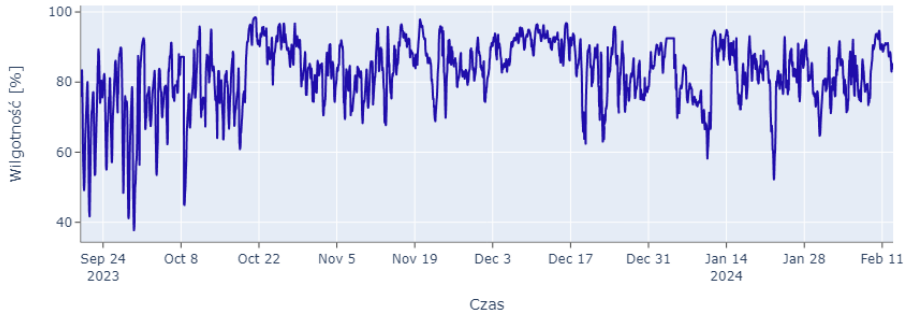


Rysunek 1: Histogram sumy opadów.

Rysunek 2 przedstawia pomiary temperatury w zdefiniowanym zakresie dat. Temperatury w tym okresie czasowym wahały się od -11,70 °C do 25,93 °C.

Rysunek 2: Pomiary temperatury w czasie.

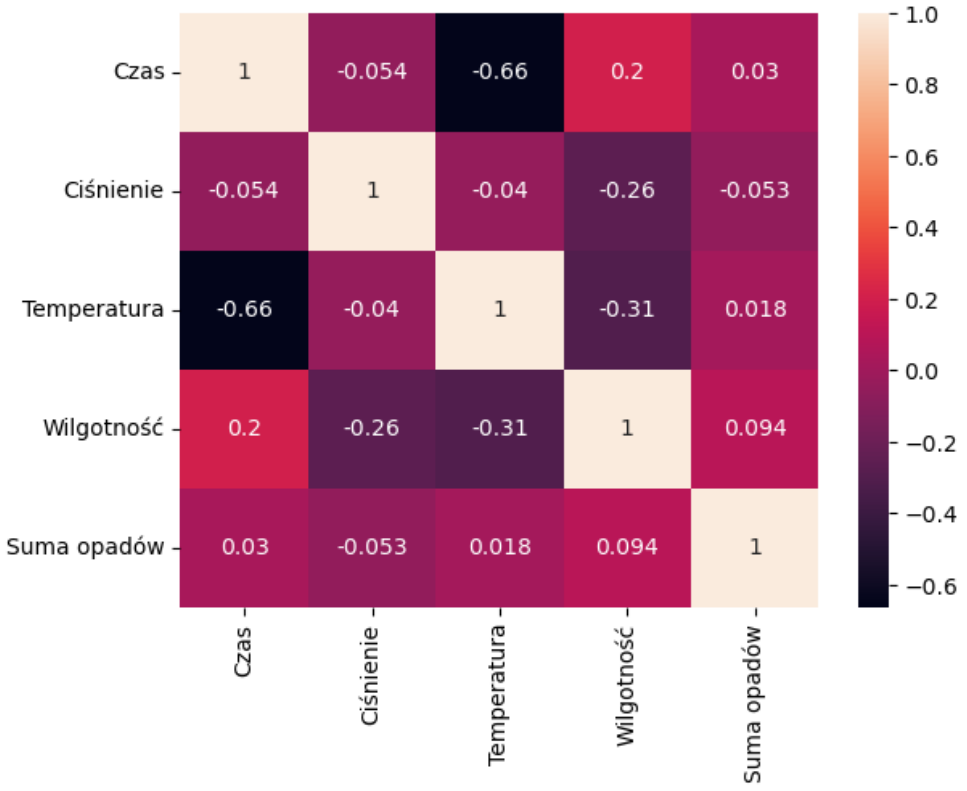
Rysunek 3 przedstawia pomiary wilgotności w zdefiniowanym zakresie dat. Zarejestrowane wartości wilgotności znalazły się w przedziale od 37,75% do 98,48%.



Rysunek 3: Pomiary wilgotności w czasie.

Rysunek 4 przedstawia pomiary ciśnienia w zdefiniowanym zakresie dat. Zarejestrowane wartości ciśnienia wahały się od 950 hPa do 1 008 hPa.

Rysunek 4: Pomiary ciśnienia w czasie.



Rysunek 5: Macierz korelacji poszczególnych cech w zbiorze.

Analiza macierzy korelacji z Rysunku 5 wykazała słabe zależności między ciśnieniem, temperaturą, wilgotnością i sumą opadów. Wartości współczynników korelacji dla wskazanych wielkości fizycznych były bliskie zeru.

4 Cele projektu

Celem projektu jest porównanie technik uczenia maszynowego w zagadnieniu predykcji opadów na podstawie informacji historycznych o ciśnieniu, temperaturze, wilgotności oraz sumie opadów. Celem pośrednim jest także zbadanie, czy dane wielkości fizyczne są wystarczające do predykcji opadów. Do ewaluacji modelów wykorzystano błąd średniokwadratowy (MSE) przedstawiony na Równaniu 1, średni błąd bezwzględny (MAE) przedstawiony na Równaniu 2, oraz współczynnik determinacji (R^2) przedstawiony na Równaniu 3. Zmienne występujące w równaniach oznaczają:

n – liczba obserwacji,

y_i – rzeczywista wartość,

$\hat{y}_i$ – przewidywana wartość,

$\bar{y}$ – średnia wartość obserwacji rzeczywistych.

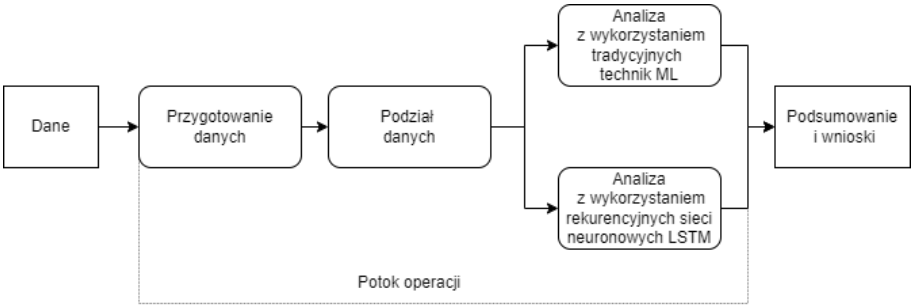
$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \tag{1}$$

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \tag{2}$$

$$R^2 = \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2} \tag{3}$$

5 Proces analizy

Potok operacji związanych z przetwarzaniem i analizą danych został przedstawiony na Rysunku 6. Poszczególne etapy zostały opisane w Rozdziałach 5.1-5.4.



Rysunek 6: Potok operacji przetwarzania i analizy danych.

5.1 Przygotowanie danych

Proces przygotowania danych wymagał usunięcia zbędnych kolumn oraz brakujących rekordów. Chcąc otrzymać pomiary temperatury, ciśnienia i wilgotności w tych samych odstępach czasu, została zastosowana technika interpolacji liniowej. Do przygotowanych danych dodano informacje o sumie opadów pozyskane ze strony OpenWeather dla tej samej lokalizacji, w której znajdował się czujnik IoT. Po zakończeniu etapu przygotowania danych otrzymano Tabelę 3 z czasem, pomiarem temperatury, ciśnienia, wilgotności i sumy opadów z godzinną rozdzielczością dla wskazanego zakresu dat. W przygotowanej tabeli znalazły się 3 504 wiersze z pomiarami.

Tabela 3: Przygotowane dane.

LP	**Czas**	**Ciśnienie [hPa]**	**Temperatura [°C]**	**Wilgotność [%]**	**Suma opadów [mm]**
0	2023-09-20 00:00:00+00:00	986	15,30	76,00	0,00
1	2023-09-20 01:00:00+00:00	985	14,53	78,90	0,00
2	2023-09-20 02:00:00+00:00	986	14,21	79,50	0,00
...	...	...	...	...	....
3 503	2024-02-12 23:00:00+00:00	977	6,19	84,13	0,00

Następnie Tabela 3 została przekształcona odpowiednio do zagadnień związanych z predykcją szeregów czasowych. Chcąc trenować modele, aby mogły dokonać predykcji opadów, w nowej tabeli w wierszach zostały zawarte informacje o pomiarach historycznych. Liczbę takich pomiarów definiuje argument funkcji, który może być dostosowywany do późniejszych potrzeb. Przykładowa tabela, o której mowa, zawierająca dwa rekordy historyczne została przedstawiona w Tabeli 4. Liczba rekordów w niej zawarta jest równa 3 501. Wynika to z faktu, że trzy wiersze zostały usunięte, ponieważ dla dwóch pierwszych rekordów nie istnieją pełne dane historyczne, a dla ostatniego rekordu nie można podać sumy opadów, która dopiero wystąpi.

Tabela 4: Przygotowane dane do prognozowania szeregów czasowych.

LP	*Czas* t_o	$TP\ t_{o+1}$	$P\ t_o$	$T\ t_o$	$H\ t_o$	$TP\ t_o$	$P\ t_{o-1}$	$T\ t_{o-1}$	$H\ t_{o-1}$	$TP\ t_{o-1}$	$P\ t_{o-2}$	$T\ t_{o-2}$	$H\ t_{o-2}$	$TP\ t_{o-2}$
0	2023-09-20 02:00:00+00:00	0,00	986	14,21	79,50	0,00	985	14,53	78,90	0,00	986	15,30	76,00	0,00
1	2023-09-20 03:00:00+00:00	0,25	985	14,04	81,30	0,00	986	14,21	79,50	0,00	985	14,53	78,90	0,00
2	2023-09-20 04:00:00+00:00	0,00	986	13,83	83,33	0,25	985	14,04	81,30	0,00	986	14,21	79,50	0,00
...	...	...	...	...	...	...	...	...	....	...	...	...	...	...
3 500	2024-02-12 22:00:00+00:00	0,00	977	6,37	83,87	0,00	977	6,57	84,03	0,00	976	7,02	84,93	0,00

Oznaczenia:

TP – suma opadów [mm]

P – ciśnienie [hPa]

H – wilgotność [%]

T – temperatura [°C]

t_o – czas t_o

t_{o+1} – pomiar o jedną godzinę później od t_o

t_{o-1} – pomiar o jedną godzinę wcześniej od t_o

t_{o-2} – pomiar o dwie godziny wcześniej od t_o

5.2 Podział danych

Do trenowania modeli tradycyjnych metod uczenia maszynowego oraz rekurencyjnych sieci neuronowych dane podzielono na zmienne zależne i niezależne. W zmiennych niezależnych znalazły się wartości ciśnienia, temperatury, wilgotności i sumy opadów dla pięciu pomiarów historycznych, czyli dla czasu od t_{0-5} do t_0. Natomiast zmienną zależną oznaczono przyszłe opady, czyli dla czasu t_{0+1}. Liczba wierszy dla pięciu pomiarów historycznych wyniosła 3 498. Wynika to z faktu, że sześć wierszy zostało usuniętych, ponieważ dla pięciu pierwszych rekordów nie istnieją pełne dane historyczne, a dla ostatniego rekordu nie można podać sumy opadów, która dopiero wystąpi. Otrzymane rekordy zostały podzielone do zbioru treningowego, walidacyjnego i testowego poprzez wybranie z przygotowanej ramki danych wierszy dla odpowiedniego zakresu dat. Stąd w zbiorze treningowym znalazły się rekordy z przedziału czasowego od 2023-09-20 05:00:00+00:00 do 2023-12-13 21:00:00+00:00, w zbiorze walidacyjny od 2023-12-13 22:00:00+00:00 do 2024-01-13 21:00:00+00:00, a w zbiorze testowym od 2024-01-13 22:00:00+00:00 do 2024-02-12 22:00:00+00:00. W Tabeli 5 dla poszczególnych zbiorów przedstawiono: liczbę wszystkich pomiarów, liczbę pomiarów z opadami, liczbę pomiarów bez opadów oraz zakres dat.

Tabela 5: Podział danych na zbiór treningowy, walidacyjny i testowy.

Nazwa zbioru	**Liczba wszystkich pomiarów**	**Liczba pomiarów z opadami (>0.0 mm)**	**Liczba pomiarów bez opadów (=0.0 mm)**	**Zakres dat**
Zbiór treningowy	2 033	442	1 591	od 2023-09-20 05:00:00+00:00 do 2023-12-13 21:00:00+00:00
Zbiór walidacyjny	744	231	513	od 2023-12-13 22:00:00+00:00 do 2024-01-13 21:00:00+00:00
Zbiór testowy	721	164	557	od 2024-01-13 22:00:00+00:00 do 2024-02-12 22:00:00+00:00

5.3 Analiza danych z użyciem tradycyjnych technik ML

Hiperparametry poszczególnych modelów tradycyjnych technik ML wypracowano na podstawie empirycznej analizy z wykorzystaniem pięciokrotnej walidacji krzyżowej. Wyniki zostały ocenione za pomocą MSE, R^2 oraz MAE obliczonych dla zbioru walidacyjnego i testowego. W kolejnych rozdziałach opisano wybrane techniki ML oraz osiągnięte rezultaty.

5.3.1 Regresor wektorów nośnych (SVR)

Technika maszyny wektorów nośnych (SVM) może być stosowana do rozwiązywania zarówno problemów klasyfikacji, jak i regresji [2]. Wykorzystanie SVM w zagadnieniu regresji, znane jako regresor wektorów nośnych (SVR), służy do przewidywania wartości ciągłych.

W [4], [5] i [12] technika SVR została wykorzystywana do predykcji opadów deszczu. Zadaniem SVR jest znalezienie najlepiej dopasowanej hiperpłaszczyzny w przestrzeni cech, która zmniejszy błąd funkcji straty. W tym celu metoda SVR znajduje wektory nośne, czyli punkty, które znajdują się najbliżej hiperpłaszczyzny. Wokół linii tworzony jest margines, czyli obszar, w którym przewidywania są akceptowane jako poprawne, pomimo wystąpienia błędu. Następnie dane są rzutowane na przestrzeń o wyższym wymiarze, aby lepiej dostosować aproksymacje. Zastosowanie funkcji jądra pozwala na obsługę wysokiej wymiarowości przestrzeni cech, a jego właściwy wybór może zwiększyć skuteczność modelu [3]. Wyróżnia się m.in. jądro:

- liniowe, gdzie dane są separowane za pomocą prostej linii lub hiperpłaszczyzny,
- wielomianowe (Poly), które obsługuje modele nieliniowe, poprzez przeniesienie wektorów do przestrzeni cech nad wielomianami zmiennych oryginalnych,
- RBF, które wykorzystuje funkcje radialną Gaussa.

Strojenie hiperparametrów modelu SVR polegało na testowaniu jądra, współczynnika regularyzacji (C) oraz wartości epsilon, a dla jądra wielomianowego dodatkowo stopnia funkcji. Testowane wartości poszczególnych parametrów zostały przedstawione w Tabeli 6.

Tabela 6: Przetestowane wartości hiperparametrów dla modelu SVR.

Testowany parametr	**Przetestowane wartości**
jądro	RBF, wielomianowe
C	10^{-1}, 1, 10, 20, 50, 10^2, 10^3, 10^4, 10^5, 10^6, 10^7, 10^8
epsilon	0,01, 0,05, 0,1, 0,5
stopień funkcji dla jądra wielomianowego	1, 2, 3, 4, 5, 6, 7, 8, 9

Hiperparametry, które dały najlepsze wyniki zostały przedstawione w Tabeli 7. Z przetestowanych parametrów okazało się, że najlepsze rezultaty osiągnięto dla jądra RBF, współczynnika regularyzacji o wartości 10^4 oraz wartości epsilon wynoszącej 0,1.

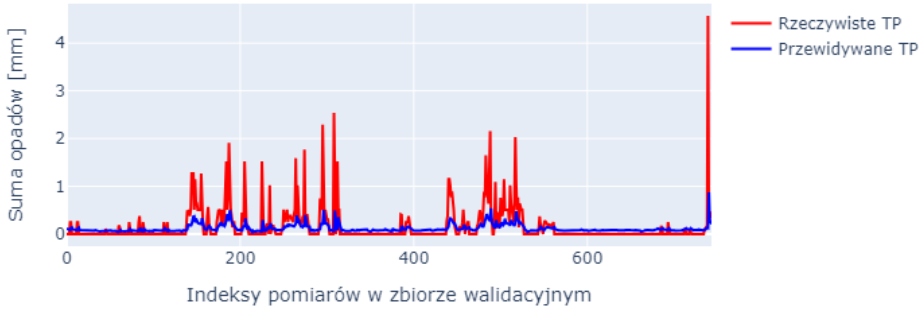
Tabela 7: Hiperparametry, które dały najlepsze wyniki dla modelu SVR.

Jądro	**Stopień funkcji**	**C**	**Epsilon**	**MSE**	$\boldsymbol{R^2}$	**MAE**
RBF	-	10^4	0,1	0,924	0,082	0,219

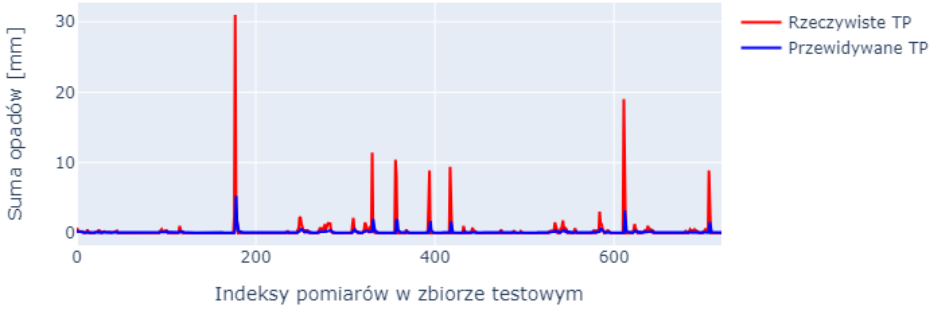
Wyniki MSE, R^2, i MAE na zbiorze walidacyjnym i testowym dla modelu SVR z hiperparametrami z Tabeli 7 zostały przedstawione zbiorczo w Tabeli 8 oraz odpowiednio na Rysunku 7 i Rysunku 8. Na zbiorze walidacyjnym osiągnięto MSE o wartości 0,109, MAE wynoszące 0,159 oraz wartość R^2 równą 0,169. Natomiast na zbiorze testowym MSE wyniósł 2,640, MAE osiągnęło wartość 0,323 a R^2 był równy 0,015.

Tabela 8: Wyniki MSE, MAE i R^2 na zbiorze walidacyjnym i testowym dla modelu SVR hiperparametrami z Tabeli 7.

Nazwa zbioru	**MSE**	**MAE**	$\boldsymbol{R^2}$
Zbiór walidacyjny	0,109	0,159	0,169
Zbiór testowy	2,640	0,323	0,015



Rysunek 7: Predykcja modelu SVR na zbiorze walidacyjnym.



Rysunek 8: Predykcja modelu SVR na zbiorze testowym.

5.3.2 Regresor Drzewa Decyzyjnego (DTR)

Drzewo decyzyjne (DT) to graf acykliczny, który składa się z wierzchołków (węzłów), gałęzi (krawędzi) oraz liści, czyli wierzchołków, które nie mają elementów potomnych. DT są wykorzystywane zarówno w zagadnieniach klasyfikacji jak i regresji (DTR). Model uczy się na podstawie danych, aby przybliżyć krzywą za pomocą reguł decyzyjnych. Każdy liść drzewa decyzyjnego w przypadku zadania regresji reprezentuje wartość numeryczną. Im głębsze drzewo, tym bardziej złożone reguły decyzyjne i lepiej dopasowany model. Aby ustalić najlepszy podział obserwacji, algorytm testuje wybrane wartości progowe oraz liczy sumę reszt podniesionych do kwadratu (RSS) na każdym etapie. Wartość progowa dla najmniejszej RSS zostaje wybrana na korzeń drzewa. Jeżeli istnieje kilka predyktorów, DTR znajduje najlepszą wartość progową dla każdego z nich, a następnie wybiera tego z najmniejszą RSS jako korzeń drzewa. Następnie operacje są dzielone oraz proces jest powtarzany do momentu, w którym węzły staną się liśćmi, czyli będą zawierać określoną liczbę obserwacji, która nie ulegnie dalszemu podziałowi [13]. W literaturze algorytmy DTR są wykorzystane m.in. do prognozowania opadów deszczu [4]. Strojenie hiperparametrów modelu DTR polegało na testowaniu minimalnej liczby próbek wymaganej do podziału węzła wewnętrznego oraz minimalnej liczby próbek wymaganej do utworzenia liścia. Testowane wartości tych parametrów zostały przedstawione w Tabeli 9.

Tabela 9: Przetestowane wartości hiperparametrów dla modelu DT.

Testowany parametr	**Przetestowane wartości**
Minimalna liczba próbek wymagana do podziału węzła wewnętrznego	10, 20, 50, 100, 200, 300, 400, 500, 600, 700, 800, 900, 1 000, 10 000
Minimalna liczba próbek wymagana do utworzenia liścia	2, 10, 50, 100, 400, 420, 430, 440, 450, 460, 470, 480, 490, 500, 510, 520, 530, 540, 550, 560, 570, 580, 590, 600, 700, 800, 900, 1 000

Hiperparametry, które dały najlepsze wyniki dla modelu DT w pięciokrotnej walidacji krzyżowej zostały przedstawione w Tabeli 10. Z przetestowanych parametrów najlepsze rezultaty osiągnięto dla minimalnej liczby próbek wymaganych do podziału węzła wewnętrznego równej 100 oraz dla minimalnej liczby próbek wymaganej do utowrzenia liścia wynoszącej 540. Przy takich parametrach MSE wyniósł 0,899, MAE osiągnął wartość 0,233, a R^2 był równy 0,096.

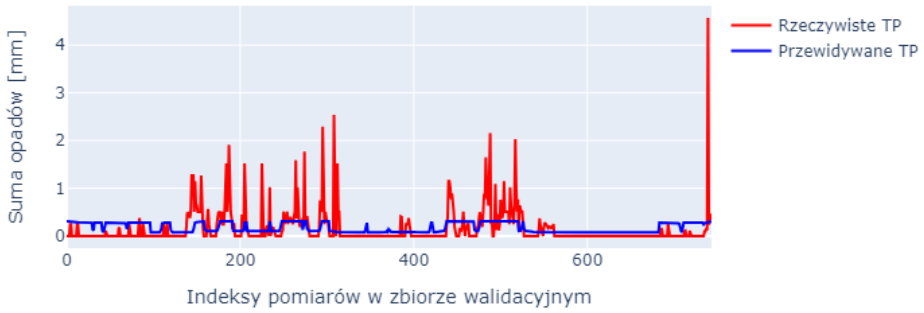
Tabela 10: Hiperparametry, które dały najlepsze wyniki dla modelu DT.

Minimalna liczba próbek wymagana do podziału węzła wewnętrznego	**Minimalna liczba próbek wymagana do utworzenia liścia**	**MSE**	**MAE**	$\boldsymbol{R^2}$
100	540	0,899	0,233	0,096

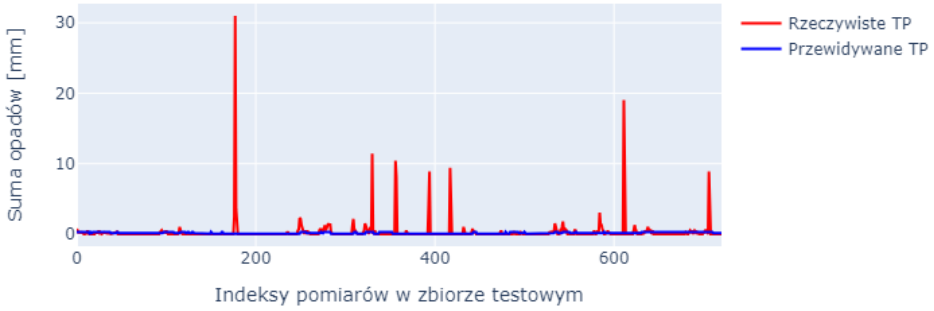
Wyniki MSE, R^2, i MAE na zbiorze walidacyjnym i testowym dla modelu DT z parametrami z Tabeli 10 zostały przedstawione odpowiednio na Rysunku 9, Rysunku 10 oraz zbiorczo w Tabeli 11. Na zbiorze walidacyjnym osiągnięto MSE o wartości 0,119, MAE wynoszące 0,194 oraz wartość R^2 równą 0,093. Natomiast na zbiorze testowym MSE wyniósł 2,712, MAE osiągnęło wartość 0,337 a R^2 był równy -0,012.

Tabela 11: Wyniki MSE, MAE i R^2 na zbiorze walidacyjnym i testowym dla modelu DT z hiperparametrami z Tabeli 10.

Nazwa zbioru	**MSE**	**MAE**	$\boldsymbol{R^2}$
Zbiór walidacyjny	0,119	0,194	0,093
Zbiór testowy	2,712	0,337	-0,012



Rysunek 9: Predykcja modelu DT na zbiorze walidacyjnym.



Rysunek 10: Predykcja modelu DT na zbiorze testowym.

5.3.3 Regresor lasu losowego (RFR)

Regresor lasu losowego (RFR) to model, który generuje dużą liczbę drzew decyzyjnych, a następnie wykorzystuje uśrednianie w celu poprawy dokładności predykcyjnej i kontroli nadmiernego dopasowania. Technika RFR do predykcji opadów deszczu jest wykorzystywana przez [4] oraz [5]. Strojenie modelu polegało na testowaniu liczby drzew, minimalnej liczby próbek wymaganej do podziału węzła wewnętrznego oraz minimalnej liczby próbek wymaganej do utworzenia liścia [14]. Testowane wartości wymienionych parametrów zostały przedstawione w Tabeli 12.

Tabela 12: Przetestowane wartości hiperparametrów dla modelu RFR.

Testowany parametr	**Przetestowane wartości**
Liczba drzew	400, 500, 550, 600, 650, 700, 800, 1 000, 1 500
Minimalna liczba próbek wymagana do podziału węzła wewnętrznego	440, 500, 510, 520, 530, 540, 600, 640
Minimalna liczba próbek wymagana do utworzenia liścia	50, 60, 70 , 80 , 90, 100, 150

Hiperparametry, które dały najlepsze rezultatu dla modelu RFR zostały przedstawione w Tabeli 13. Z przetestowanych parametrów najlepsze rezultaty osiągnięto dla liczby drzew równej 600, minimalnej liczby próbek wymaganych do podziału węzła wewnętrznego o wartości 80 oraz dla minimalnej liczby próbek wymaganej do utowrzenia liścia wynoszącej 510. Dla tych parametrów MSE wyniósł 0,902, MAE osiągnął wartość 0,229, a R^2 był równy 0,112.

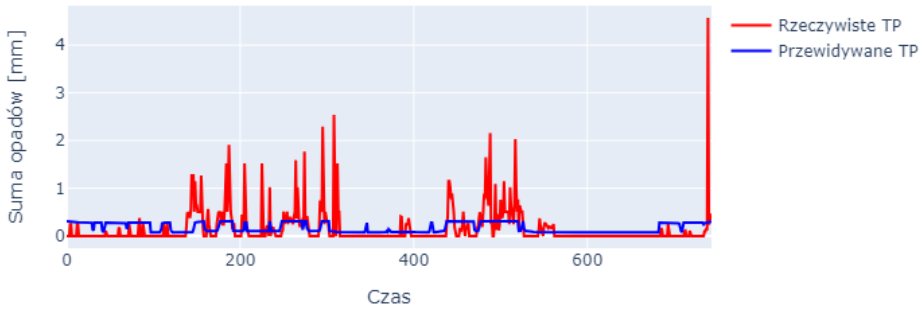
Tabela 13: Hiperparametry, które dały najlepsze wynki dla modelu RFR

Liczba drzew w lesie	**Minimalna liczba próbek wymagana do podziału węzła wewnętrznego**	**Minimalna liczba próbek wymagana do utworzenia liścia**	**MSE**	**MAE**	$\boldsymbol{R^2}$
600	80	510	0,902	0,229	0,112

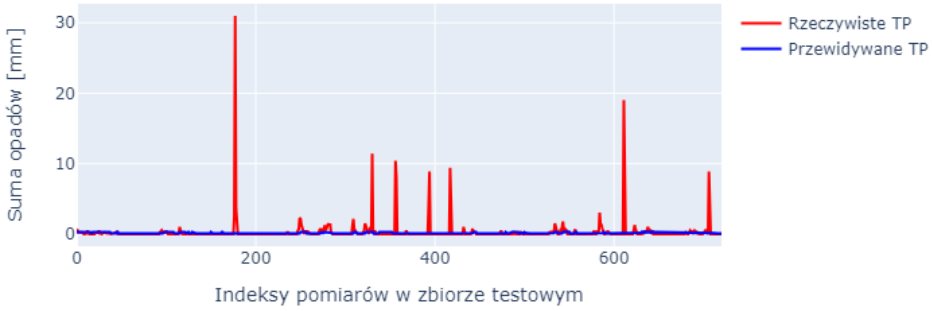
Wyniki MSE, R^2, i MAE na zbiorze walidacyjnym i testowym dla modelu RFR z hiperparamterami z Tabeli 13 zostały przedstawione odpowiednio na Rysunku 11, Rysunku 12 oraz zbiorczo w Tabeli 14. Na zbiorze walidacyjnym osiągnięto MSE o wartości 0,117, MAE wyoszące 0,198 oraz wartość R^2 równą 0,105. Natomiast na zbiorze testowym MSE wyniósł 2,701, MAE osiągnęło wartość 0,346 a R^2 był równy -0,007.

Tabela 14: Wyniki MSE, MAE i R^2 na zbiorze walidacyjnym i testowym dla modelu RFR z hiperparametrami z Tabeli 13.

Nazwa zbioru	**MSE**	**MAE**	$\boldsymbol{R^2}$
Zbiór walidacyjny	0,117	0,198	0,105
Zbiór testowy	2,701	0,346	-0,007



Rysunek 11: Predykcja modelu RFR na zbiorze walidacyjnym.



Rysunek 12: Predykcja modelu RFR na zbiorze testowym.

5.3.4 Regresja liniowa (LR)

Regresja liniowa (LR) to prosty model statystyczny, który może być używany do przewidywania opadów deszczu na podstawie danych historycznych [5] [15] [16]. Celem LR jest pokazanie relacji liniowej między dwiema lub wieloma zmiennymi. W tym celu funkcja liniowa jest dopasowywana do obserwacji, aby pokazać zależności między obserwacjami i móc na ich podstawie przewidzieć przyszłą wartość nowych danych [17]. Prostą regresję liniową opisuje równanie 4.

$$y = b + b_1 * x \tag{4}$$

W równaniu 4 y jest zmienną zależną, której wartość będzie przewidywana, a x jest zmienną niezależną, której wartość jest znana. Współczynnik b oznacza stałą wartość, a b_1 to współczynnik kierunkowy prostej. Rozszerzeniem LR jest regresja wieloraka, którą opisuje równanie 5.

$$y = b + b_1 * x_1 + b_2 * x_2 + \ldots + b_n * x_n \tag{5}$$

W równaniu 5 y jest zmienną zależną, a x_1,$x_2, \ldots, x_n$ są kolejnymi zmiennymi niezależnymi. W celu znalezienia najlepiej dopasowanej prostej minimalizowana jest suma reszt, czyli różnic między wartościami rzeczywistymi a przewidywanymi, podniesionych do kwadratu.

W przedstawionym eksperymencie zostanie wykorzystana regresja wieloraka ze względu na to, że w danych znajduje się 20 zmiennych niezależnych tj. ciśnienie, temperatura, wilgotnośc oraz suma opadów dla pięciu pomiarów historycznych. Wyniki LR dla pięciokrotnej walidacji krzyżowej zostały przedstawione w Tabeli 15. Z informacji zawartych w tabeli wynika, że model osiągnął MSE wynoszący 0,932, MAE równy 0,271 oraz R^2 o wartości 0,041.

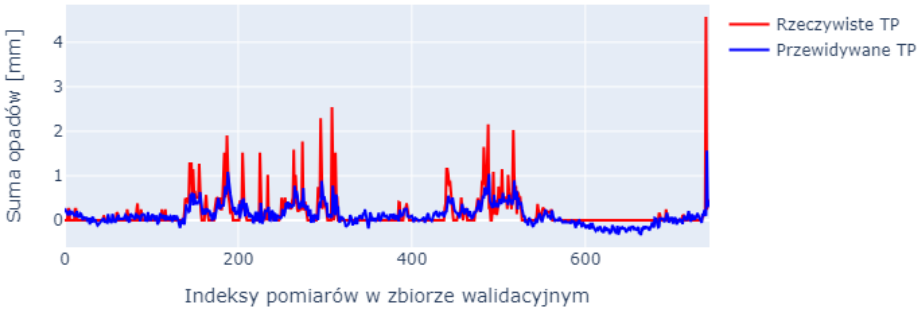
Tabela 15: Wyniki modelu LR dla pięciokrotnej walidacji krzyżowej.

MSE	**MAE**	$\boldsymbol{R^2}$
0,932	0,271	0,041

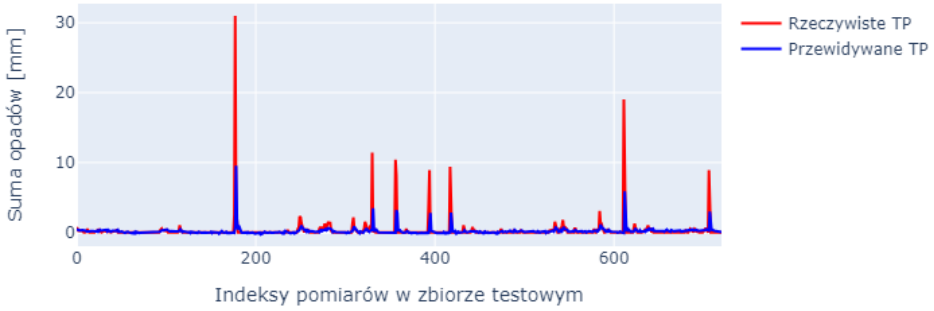
Wyniki MSE, MAE i R^2 na zbiorze walidacyjnym i testowym zostały przedstawione na Rysunku 13, Rysunku 14 oraz zbiorczo w Tabeli 15. Na zbiorze walidacyjnym osiągnięto MSE o wartości 0,103, MAE wynoszące 0,163 oraz wartość R^2 równą 0,214. Natomiast na zbiorze testowym MSE wyniósł 2,713, MAE osiągnęło wartość 0,343 a R^2 był równy -0,012.

Tabela 16: Wyniki MSE, MAE i R^2 na zbiorze walidacyjnym i testowym dla modelu LR.

Nazwa zbioru	**MSE**	**MAE**	$\boldsymbol{R^2}$
Zbiór walidacyjny	0,103	0,163	0,214
Zbiór testowy	2,713	0,343	-0,012



Rysunek 13: Predykcja modelu LR na zbiorze walidacyjnym.



Rysunek 14: Predykcja modelu LR na zbiorze testowym.

5.4 Analiza danych z użyciem rekurencyjnych sieci neuronowych LSTM

Rekurencyjne sieci neuronowe (RNN) są wykorzystywane w zadaniach analizy szeregów czasowych. Klasyczne RNN przetwarzają sekwencje danych poprzez iteracyjne podawanie na wejściu kolejnych elementów ze zbioru danych. Ich wyróżnikiem jest to, że zachowują informacje o elementach, które zostały już przetworzone w poprzednich krokach. Za realizację tego zadania odpowiada tzw. „wektor stanu", którego wartość zależy nie tylko od danych wejściowych analizowanych w danym kroku, ale także od postaci wektora stanu z kroku poprzedniego [18]. Chociaż RNN biorą pod uwagę dane dotyczące przeszłości, niestety nie zapamiętują dobrze odległej historii [19]. W odpowiedzi na problem zanikania gradientu w klasycznych sieciach RNN, czyli zapamiętywania rozległych w czasie zależności, zostały stworzone rekurencyjne sieci neuronowe LSTM [20]. Ich zmodyfikowana struktura posiada cztery warstwy sieci (zamiast jednej), które współdziałają ze sobą w ściśle określony sposób. Dodatkowo pomiędzy kolejnymi krokami są w stanie swobodnie przekazywać informacje o wektorze stanu komórki, którego wartości mogą być modyfikowane w każdym kroku [19]. Te zabiegi pomogły rozwiązać problem zanikającego gradientu i sprawiły, że rekurencyjne sieci neuronowe LSTM mogą zapamiętywać zależności w rozległym czasie tym samym stając się odpowiednie do prognozowania opadów w oparciu o historyczne dane pogodowe [2] [4] [6] [21]. Porównanie modeli predykcyjnych do prognozowania opadów jest niemal niemożliwe, ponieważ są one dostosowane do konkretnych skal czasowych, klimatu, regionów i pór roku oraz do ich oceny są wykorzystywane różne miary [2]. Stąd, odniesienie się do wyników innych modeli z literatury nie pozwalają na wyciągnięcie jednoznacznych i konkretnych wniosków [2].

Do pomiaru funkcji straty modelu zastosowano błąd MSE. W eksperymencie wykorzystano zbiory: treningowy, walidacyjny i testowy, szczegółowo opisane w rozdziale 5.2. Przed podziałem do wymienionych zbiorów, dane dotyczące opadów deszczu zostały poddane transformacji logarytmu naturalnego. Wynika to z faktu, że opady deszczu mają rozkład skośny,

a dane charakteryzują się dużą rozbieżnością przyjmowanych wartości (od 0,00 mm do 30,99 mm). Transformacja logarytmiczna zmniejsza zakres zmienności tych danych, co pomaga w stabilizacji ich wariancji. Jest to ważne dla algorytmów uczenia maszynowego, które mogą mieć trudności przy dużych skokach i rozbieżnych wartościach. Dodatkowo, zastosowanie transformacji logarytmicznej redukuje wartości skrajne, co pomaga w uzyskaniu bardziej stabilnego i dokładnego modelu. W celu uniknięcia obliczania wartości logarytmu dla zera zastosowano przekształcenie x przedstawione w równaniu 6:

$$F_i' = F_i + \epsilon \tag{6}$$

gdzie F_i' to przekształcona wartość, F_i to oryginalna wartość, a ϵ to bardzo mała liczba nieujemna.

Hiperparametry modelu LSTM wypracowano na podstawie empirycznej analizy. Testowane parametry i ich wartości zostały przedstawione w Tabeli 17.

Tabela 17: Przetestowane wartości hiperparametrów dla modelu LSTM.

Testowany parametr	**Przetestowane wartości**
Liczba warstw LSTM	2, 3
Liczba neuronów w warstwach LSTM	50, 100
Liczba warstw Dropout	0, 1, 2, 3
Współczynnik odrzucenia	0,1, 0,2
epsilon z operacji logarytmu	0,1, 0,5
Liczba pomiarów historycznych	5, 12, 18, 24
Liczba epok	20, 50, 100, 300, 1 000
Funkcja aktywacji	Liniowa, ReLU, LeakyReLU
Ujemne nachylenie LeakyRelu	0,1, 0,3
Współczynnik nauki	10^{-5}, 10^{-3}, 10^{-1}

Zgodnie z informacjami z Tabeli 17 przetestowano liczbę warstw LSTM oraz neuronów w nich występujących, liczbę warstw Dropout wraz ze współczynnikiem odrzucenia, wartości epsilon z operacji logarytmu naturalnego, liczbę pomiarów historycznych, liczbę epok, rodzaj funkcji aktywacji oraz współczynnik nauki. Wartości przedstawionych parametrów były dobierane metodą prób i błędów w celu poprawy wyników modelu. Parametry sieci, które zapewniły najlepsze wyniki dla modelu zostały przedstawione w Tabeli 18.

Tabela 18: Hiperparametry, które zapewniły najlepsze wyniki dla modelu LSTM.

Parametry sieci	**Wartości parametrów**
Liczba warstw LSTM	3
Liczba neuronów w pierwszej warstwie LSTM	50
Liczba neuronów w drugiej warstwie LSTM	50
Liczba neuronów w trzeciej warstwie LSTM	50
Liczba warstw Dropout	1
Współczynnik odrzucenia	0,1
epsilon z operacji logarytmu	0,1
Liczba pomiarów historycznych	5
Liczba epok	100
Funkcja aktywacji	LeakyRelu
Ujemne nachylenia LeakyReLU	0,3
Współczynnik nauki	10^{-5}

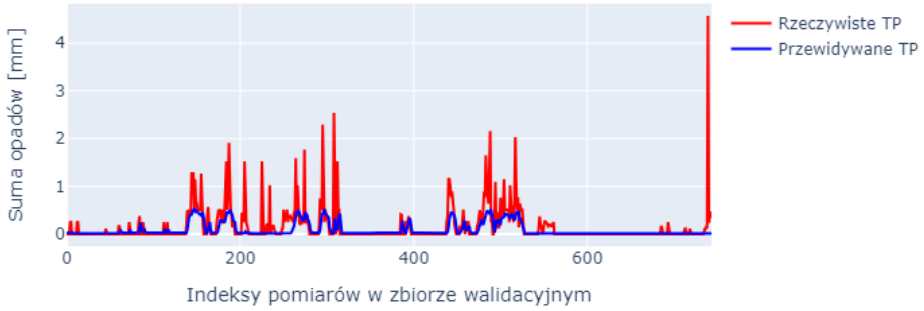
Model został trenowany na 100 epokach w pakietach o wielkości 32. Końcowa architektura rekurencyjnej sieci neuronowej LSTM składała się z trzech warstw LSTM (każda z 50 neuronami), jednej warstwy Dropout ze współczynnikiem odrzucenia równym 0,1, warstwy głębokiej (Dense) z funkcją aktywacji LeakuRelu z ujemnym nachyleniem wynoszącym 0,3, optymalizatora Adam oraz współczynnika nauki równego 10^{-5}. Dane do trenowania modelu zostały przygotowane z pięcioma historycznymi pomiarami oraz z opisaną wcześniej operacją logarytmu naturalnego z epsilonem równym 0,1. Na Rysunku 15 została przedstawiona architektura opisanego modelu rekurencyjnych sieci neuronowych LSTM.

Rysunek 15: Architektura modelu rekurencyjnych sieci neuronowych LSTM.

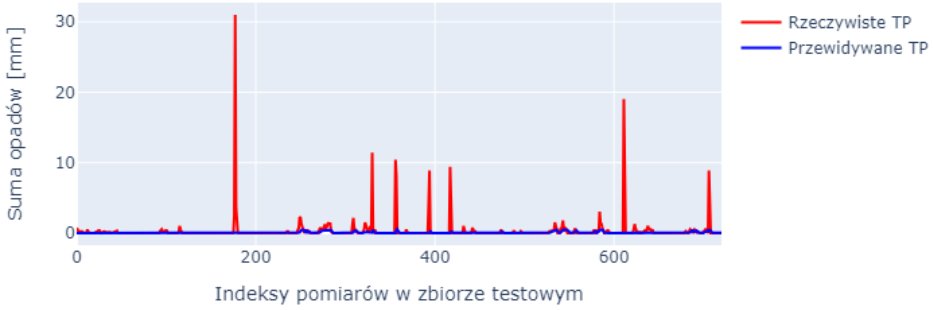
Wyniki MSE, R^2, i MAE na zbiorze walidacyjnym i testowym zostały przedstawione w Tabeli 19 oraz odpowiednio na Rysunku 16 i Rysunku 17. Na zbiorze walidacyjnym osiągnięto MSE o wartości 0,099, MAE wynoszące 0,121 oraz wartość R^2 równą 0,243. Natomiast na zbiorze testowym MSE wyniósł 2,711, MAE osiągnęło wartość 0,279 a R^2 był równy -0,112.

Tabela 19: Wyniki MSE, MAE i R^2 na zbiorze walidacyjnym i testowym dla modelu LSTM z hiperparametrami z Tabeli 18.

Nazwa zbioru	**MSE**	**MAE**	$\boldsymbol{R^2}$
Zbiór walidacyjny	0,099	0,121	0,243
Zbiór testowy	2,711	0,279	-0,112



Rysunek 16: Predykcja modelu LSTM na zbiorze walidacyjnym.



Rysunek 17: Predykcja modelu LSTM na zbiorze testowym.

6 Podsumowanie

Celem projektu było porównanie technik uczenia maszynowego w zagadnieniu predykcji opadów na podstawie informacji historycznych o ciśnienu, temperaturze, wilgotności i sumie opadów. Cel został osiągnięty. Zbadano również, że predykcja na bazie podanych wielkości fizycznych jest w pewnym zakresie możliwa, tzn. modele zazwyczaj dobrze przewidują fakt wystąpienia opadów (wartości większe od 0,00 mm), natomiast podawana przez nie wartość predykcji nie jest dokładna.

Analizując wyniki na zbiorze walidacyjnym, najmniejszy błąd MSE (0,099) wśród modeli uzyskały rekurencyjne sieci neuronowe LSTM. Na kolejnych miejscach znalazł się model LR (0,103), SVR (0,109), RFR (0,117) oraz DT (0,119). Największy współczynnik determinacji R^2 (0,243) uzyskał model rekurencyjnych sieci neuronowych. Wyniki R^2 dla pozostałych modeli wyniosły: LR (0,214), SVR (0,169), RFR (0,105) i DT (0,093). Porównując wyniki MAE, najmniejszy błąd uzyskał model rekurencyjnych sieci neuronowych LSTM (0,121), następnie model SVR (0,159), LR (0,163), DT (0,194) oraz RFR (0,198).

Jeśli chodzi o zbiór testowy, to najmniejszy błąd MSE (2,640) uzyskał model SVR. Błędy MSE dla pozostałych modeli osiągnęły zbliżone wartości: RFR (2,701), LSTM (2,711), DT (2,712) oraz LR (2,713). Na zbiorze testowym wszystkie modele oprócz SVR uzyskały ujemne wartości współczynnika determinacji R^2 tj. RFR (-0,007), DT (-0,012), LR (-0,112). Natomiast model SVR osiągnął wartość dodatnią wynoszącą R^2 (0,015). Najniższe wartości MAE osiągnęły rekurencyjne sieci neuronowe LSTM (0,279). Wartości MAE dla pozostałych modeli były bardzo zbliżone tj. SVR (0,323), LR (0,343), RFR (0,346) oraz DT (0,337).

Reasumując, najlepsze wyniki predykcji na zbiorze walidacyjnym ocenione za pomocą MSE, R^2 i MAE osiągnęły rekurencyjne sieci neuronowe LSTM. Biorąc pod uwagę MSE i R^2 na zbiorze testowym, najlepsze wartośći uzyskał model SVR.

7 Dyskusja i Wnioski

Metryki oceny modeli: MSE, R^2 oraz MAE dla zbioru walidacyjnego są znacząco lepsze niż dla zbioru testowego. Może to wynikać z faktu, że charakterystyki dla poszczególnych okresów w roku różnią się między sobą. Mały zbiór pomiarów spowodował, że modele nie nauczył się zależności danych ze zbioru testowego (zakres dat: od 13 stycznia 2024 do 12 lutego 2024) oraz ze zbioru walidacyjnego (zakres dat: od 13 grudnia 2023 do 13 stycznia 2024). Stąd nie posiadają one zdolności generalizacyjnych oraz są nadmiernie dopasowane do danych treningowych. Różnica w rezultatach między tymi zbiorami może wynikać z faktu, że charakterystyka danych w zbiorze walidacyjnym była zbliżona do tych w zbiorze treningowym (zakres dat: od 20 września 2023 do 13 grudnia 2023).

W celu ulepszenia wyników predykcji modelów należy zgromadzić większy zbiór danych do analizy. W przedstawionym eksperymencie zakres danych obejmował pięć miesięcy, podczas gdy w literaturze często spotyka się znacznie dłuższe okresy pomiarów np. trwające 42 lata [12]. Dodatkowe parametry wejściowe tj. np. prędkość wiatru czy zachmurzenie, stosowane w literaturze np. w [2] [4] [12], udoskonaliłyby modele w zagadnieniu predykcji opadów.

Obecnie, numeryczne modele prognozowania pogody wykorzystują dane meteorologiczne pozyskiwane z różnorodnych źródeł, takich jak satelity oraz systemy obserwacji Ziemi. Te źródła obejmują zarówno automatyczne, jak i załogowe stacje meteorologiczne, samoloty, statki oraz balony meteorologiczne [22].

8 Bibliografia

[1] *Temperature and Humidity Sensor*. URL: https://www.aqara.com/en/product/temperature-humidity-sensor/. (dostęp: 13-05-2024).

[2] Sarmad Dashti Latif et al. "Assessing rainfall prediction models: Exploring the advantages of machine learning and remote sensing approaches". In: *Alexandria Engineering Journal* 82 (2023), pp. 16–25. DOI: https://doi.org/10.1016/j.aej.2023.09.060.

[3] B. Üstün, W.J. Melssen, and L.M.C. Buydens. "Facilitating the application of Support Vector Regression by using a universal Pearson VII function based kernel". In: *Chemometrics and Intelligent Laboratory Systems* 81.1 (2006), pp. 29–40. DOI: https://doi.org/10.1016/j.chemolab.2005.09.003.

[4] Md.Mehedi Hassan et al. "Machine Learning-Based Rainfall Prediction: Unveiling Insights and Forecasting for Improved Preparedness". In: *IEEE Access* 11 (2023), pp. 132196–132222. DOI: 10.1109/ACCESS.2023.3333876.

[5] Suman Markuna et al. "Application of Innovative Machine Learning Techniques for Long-Term Rainfall Prediction". In: *Pure and Applied Geophysics* 180 (2023), pp. 335–363. DOI: 10.1007/s00024-022-03189-4.

[6] Sujan Mondol et al. "A Deep Learning Approach Using Long Short-Term Memory Networks for Enhanced Prediction of Rainfall in the Northeastern Region of Bangladesh". In: 2024.

[7] F. Pedregosa et al. "Scikit-learn: Machine Learning in Python". In: *Journal of Machine Learning Research* 12 (2011), pp. 2825–2830.

[8] The pandas development team. *pandas-dev/pandas: Pandas*. Version latest. Feb. 2020. DOI: 10.5281/zenodo.3509134. URL: https://doi.org/10.5281/zenodo.3509134.

[9] Plotly Technologies Inc. *Collaborative data science*. 2015. URL: https://plot.ly.

[10] François Chollet et al. *Keras*. https://keras.io. 2015.

[11] Charles R. Harris et al. "Array programming with NumPy". In: *Nature* 585.7825 (Sept. 2020), pp. 357–362. DOI: 10.1038/s41586-020-2649-2. URL: https://doi.org/10.1038/s41586-020-2649-2.

[12] Xiaobo Zhang et al. "Annual and Non-Monsoon Rainfall Prediction Modelling Using SVR-MLP: An Empirical Study From Odisha". In: *IEEE Access* 8 (2020), pp. 30223–30233. DOI: 10.1109/ACCESS.2020.2972435.

[13] *Regression Trees, Clearly Explained*. URL: https://www.youtube.com/watch?v=g9c66TUylZ4. (dostęp: 13-05-2024).

[14] *Radom Forest Regressor*. URL: https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestRegressor.html. (dostęp: 13-05-2024).

[15] Carissa Usman et al. “Rainfall prediction model in Semarang City using machine learning”. In: *Indonesian Journal of Electrical Engineering and Computer Science* 30 (2023), p. 1224. DOI: 10.11591/ijeecs.v30.i2.pp1224-1231.

[16] Hanoon M.S., Ahmed A.N., and Zaini N. et al. “Developing machine learning algorithms for meteorological temperature and humidity forecasting at Terengganu state in Malaysia”. In: *Scientific Reports* 11 (2021), pp. 2045–2322. DOI: https://doi.org/10.1038/s41598-021-96872-w.

[17] T. Nield. *Podstawy matematyki w data science*. Helion, 2023, pp. 131–168.

[18] Y. Bengio, P. Simard, and P. Frasconi. “Learning long-term dependencies with gradient descent is difficult”. In: *IEEE Transactions on Neural Networks* 5.2 (Mar. 1994), pp. 157–166. DOI: 10.1109/72.279181.

[19] D. Puchała. *Metody uczenia głębokiego*. 2024.

[20] Y. Bengio, P. Simard, and P. Frasconi. “Learning long-term dependencies with gradient descent is difficult”. In: *IEEE Transactions on Neural Networks* 5.2 (1994), pp. 157–166. DOI: 10.1109/72.279181.

[21] J. M. Frame et al. “Deep learning rainfall–runoff predictions of extreme events”. In: *Hydrology and Earth System Sciences* 26.13 (2022), pp. 3377–3392. DOI: 10.5194/hess-26-3377-2022.

[22] *European Centre for Medium-Range Weather Forecasts*. URL: https://en.wikipedia.org/wiki/European_Centre_for_Medium-Range_Weather_Forecasts. (dostęp: 22-05-2024).