

Koło CyberSecurity



Dokumentacja projektowa

Temat pracy: **DOKUMENTACJA Z CUVALLEY HACK 2023**

ObjectNotFound

.....
(drużyna)

Członkowie:

inż. Weronika BURAS
inż. Łukasz CZERNISZEWSKI
inż. Cezary BĄK
inż. Filip ANDREJCZYK
inż. Bartosz BALA

Zamawiający:

KGHM Polska Miedz S.A.

Spis treści

Rozdział I. Opis zlecenia.....	3
I.1. Otrzymane zlecenie.....	3
Rozdział II. Zamodelowane rozwiązanie	4
II.1. Idea rozwiązania	4
II.2. Wymagania funkcjonalne	5
II.3. Wymagania нефunkcjonalne	5
II.4. Model rozwiązania.....	9
II.5. Wybrane środowisko oraz narzędzia	11
II.6. Dokładny opis sieci neuronowej.....	14
II.7. Opis wybranych fragmentów kodu.....	15
II.8. Możliwości rozwoju środowiska	16
Rozdział III. Wygląd aplikacji	16
Podsumowanie	20

Rozdział I. Opis zlecenia

I.1. Otrzymane zlecenie

Stworzenie systemu automatycznej estymacji poziomu wody w rzece

Opis obecnej sytuacji:

Zakładowi Hydrotechnicznemu powierzono odpowiedzialne zadanie prowadzenia eksploatacji i rozbudowy Obiektu Unieszkodliwiania Odpadów Wydobyczych Żelazny Most, gdzie trafiają odpady poflotacyjne z procesu wzbogacania rudy. Składowisko jest umieszczone w naturalnym obniżeniu i otoczone wałami. Do rozbudowy składowiska (podnoszenie korony wału) używa się materiału o najgrubszej frakcji, odseparowanego ze składowanego odpadu. Materiał jest transportowany na składowisko hydrotechniczne, następnie następuje jego naturalna segregacja i sedymentacja. Najdrobniejszy materiał jest składowany w centralnej części składowiska w postaci zawiesiny.

Cel zadania, co chcemy osiągnąć?

Stworzenie modelu/algorytmu, który na bazie danych historycznych oraz prognozy pogody, pochodzących z IMGW, pozwoli prognozować poziom wody rzeki.

Dlaczego to jest ważne dla danego Oddziału KGHM?

Poziom wody w rzece determinuje planowanie gospodarki wodnej w Zakładzie Hydrotechnicznym.

Jakie efekty przyniesie zrealizowanie zadania?

Zrealizowanie zadania pozwoli Zakładowi Hydrotechnicznemu na prowadzenie racjonalnej gospodarki wodnej.

Rozdział II. Zamodelowane rozwiązanie

II.1. Idea rozwiązania

Naszym pomysłem rozwiązania tego problemu zamodelowanie aplikacji w języku Python bazującej na sieciach neuronowych. Sieci neuronowe wydają się idealnym rozwiązaniem do rozwiązywania problemów związanych z uczeniem maszynowym, takich jak analiza danych i predykcja. W Python istnieje wiele bibliotek do tworzenia sieci neuronowych, w naszym rozwiązaniu zdecydowaliśmy się na TensorFlow, która jest najbardziej rozpowszechnionym rozwiązaniem. Warto również pamiętać o odpowiedniej ewaluacji i walidacji modelu, aby upewnić się, że jest on skuteczny. Ważne jest również dobranie odpowiedniego typu sieci neuronowej do konkretnego problemu oraz odpowiednie przygotowanie danych. W przypadku aplikacji bazującej na sieciach neuronowych, ważne jest również uwzględnienie aspektów związanych z wydajnością i skalowalnością, aby aplikacja mogła działać efektywnie nawet przy dużych ilościach danych. W naszym podejściu postawiliśmy na sieć Long Short-Term Memory (LSTM), które zgodnie badaniami przeprowadzanymi przez „Korean institute of communications and information sciences” najlepiej nadają się do tego typu zagadnień.

II.2. Wymagania funkcjonalne

- System będzie automatycznie prognozować poziom wody w rzece na podstawie danych historycznych oraz prognozy pogody pochodzącej z IMGW.
- System będzie zbierał dane historyczne dotyczące poziomu wody w rzece oraz prognozy pogody z IMGW.
- System będzie posiadał interfejs użytkownika pozwalający na przeglądanie prognozowanego poziomu wody w rzece oraz danych historycznych.
- System będzie posiadał funkcję alarmową, która poinformuje użytkownika o przekroczeniu prognozowanego poziomu wody w rzece.
- System będzie zintegrowany z systemami hydrotechnicznymi Zakładu Hydrotechnicznego, umożliwiając automatyczne dostosowanie planów gospodarki wodnej.

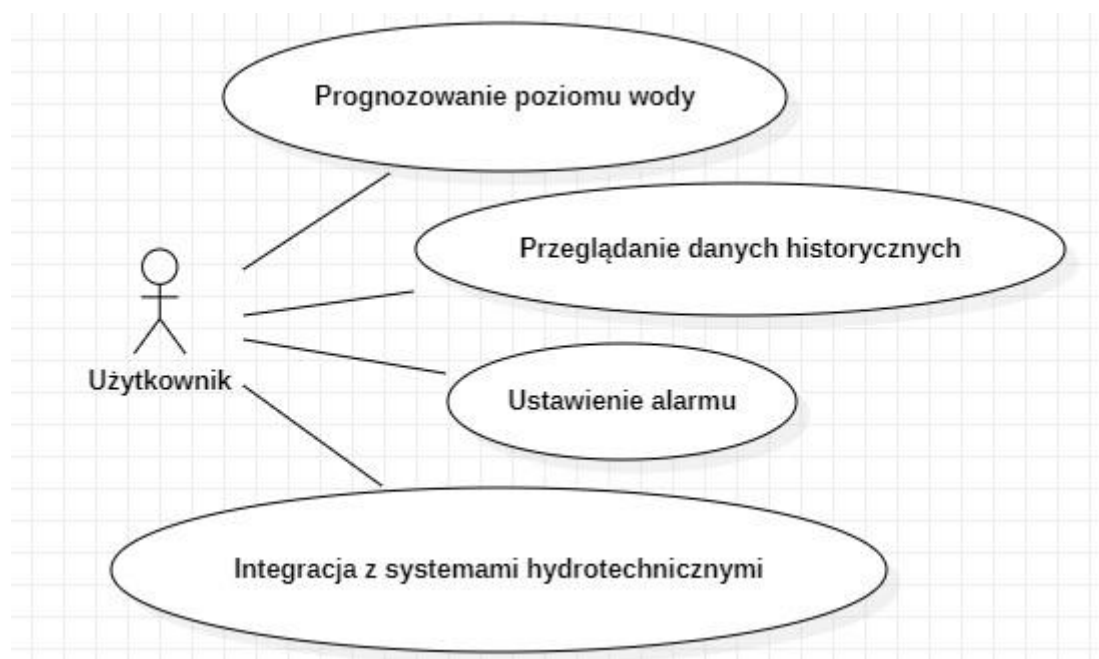
II.3. Wymagania нефunkcjonalne

- System będzie dostępny 24/7 z nie więcej niż 2% przerw w działaniu.
- System będzie wykazywał czas odpowiedzi na poziomie mniejszym niż 5 sekund dla żądania prognozy poziomu wody.
- System będzie skalowalny i będzie mógł obsługiwać rosnące wymagania dotyczące ilości danych i użytkowników.
- System będzie zgodny z normami branżowymi dotyczącymi gospodarki wodnej i bezpieczeństwa.
- System powinien być bezpieczny.

II.3.1. Przypadki użycia

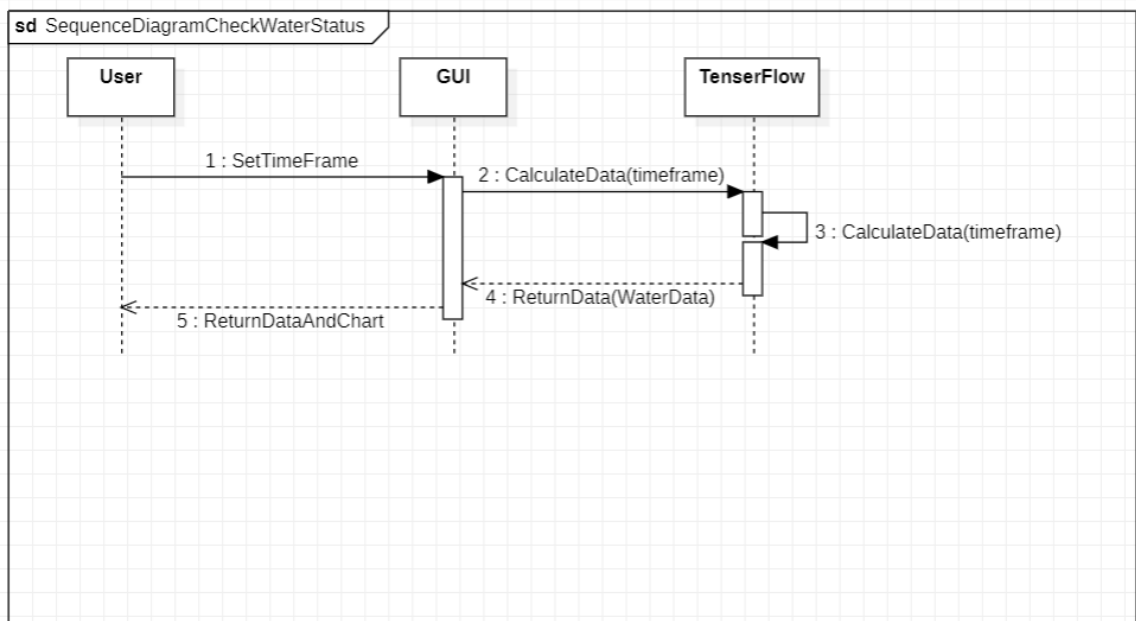
Tabela 1 Przypadki użycia

Nazwa	Działanie
Prognozowanie poziomu wody	Użytkownik wprowadza datę, na którą chce uzyskać prognozę poziomu wody w rzece. System automatycznie generuje prognozę na podstawie danych historycznych oraz prognozy pogody z IMGW.
Przeglądanie danych historycznych	Użytkownik może przeglądać dane historyczne dotyczące poziomu wody w rzece w określonym przedziale czasowym.
Ustawienie alarmu	Użytkownik może ustawić alarm, który poinformuje go o przekroczeniu prognozowanego poziomu wody w rzece.
Integracja z systemami hydrotechnicznymi	Użytkownik może zintegrować system z systemami hydrotechnicznymi Zakładu Hydrotechnicznego, aby automatycznie dostosować plany gospodarki wodnej.

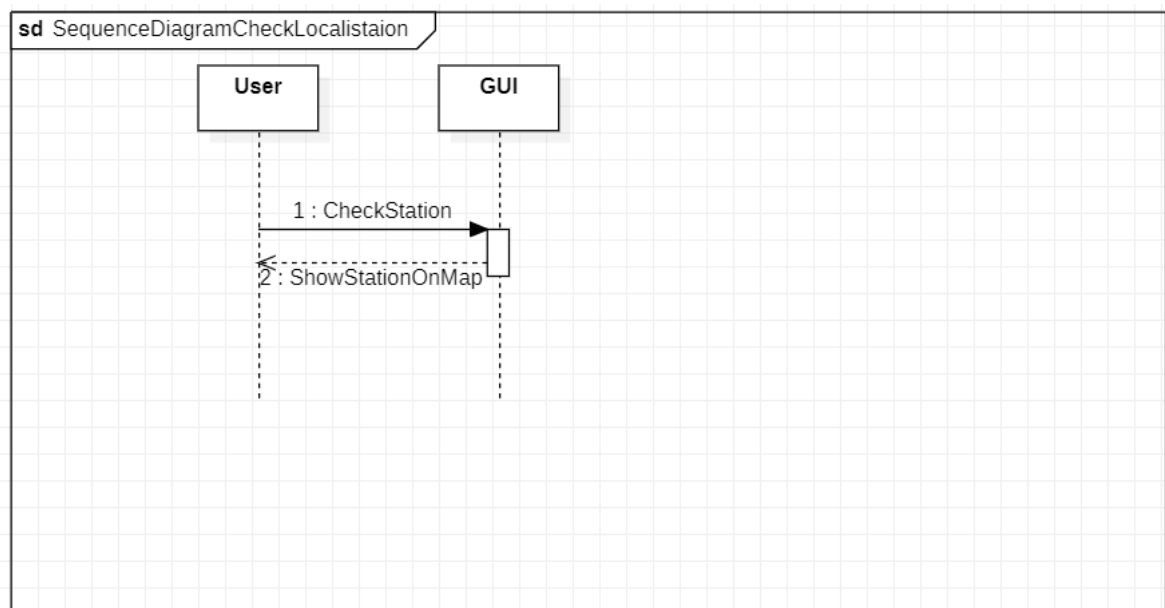


Rys. 1. Diagram przypadków użycia wykonany przy użyciu narzędzia STARUML

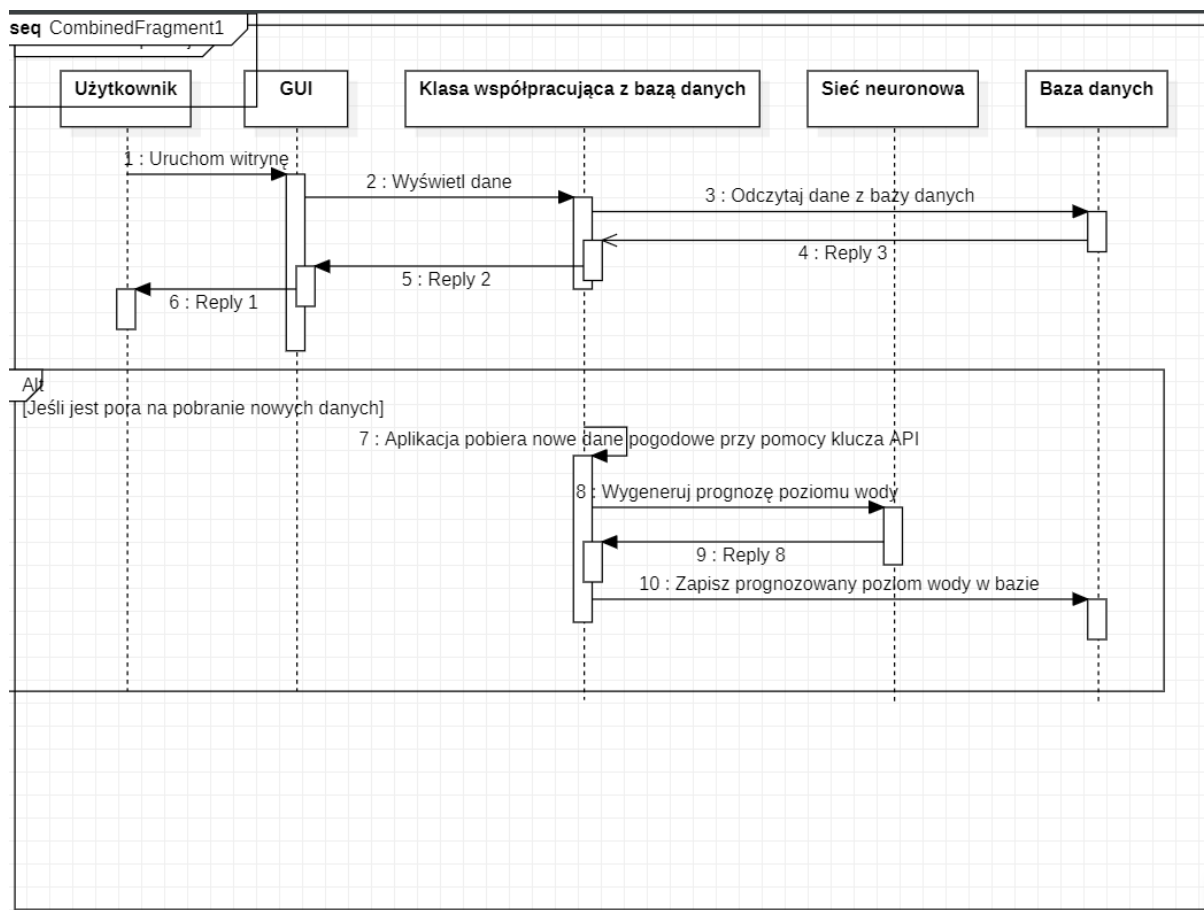
II.3.2. Diagramy sekwencji



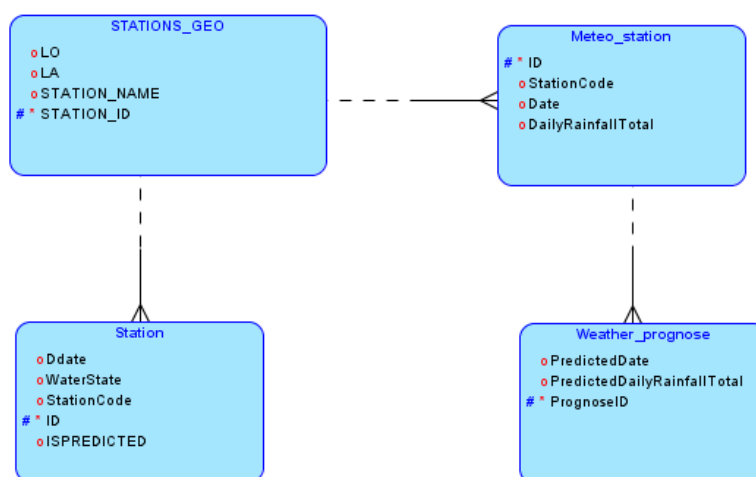
Rys. 2. Diagram sekwencji sprawdzenia stanu wody wykonany przy użyciu narzędzia StarUML



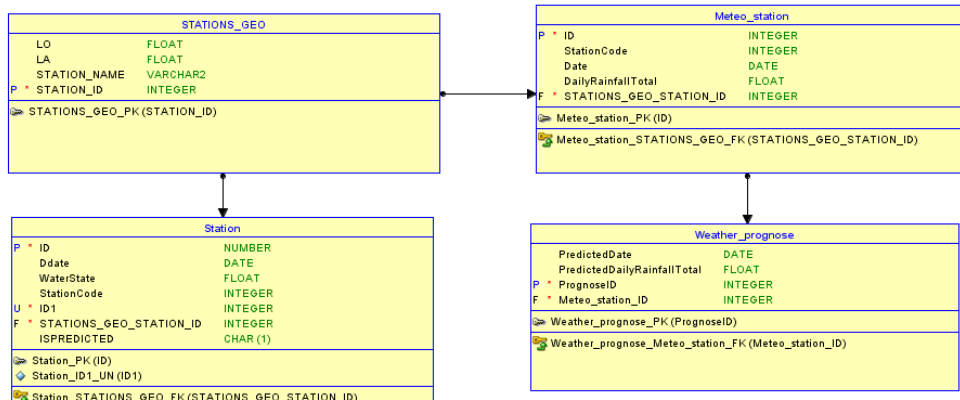
Rys. 3. Diagram sekwencji sprawdzenia lokalizacji stacji wykonany przy użyciu narzędzia StarUML



Rys. 4. Uogólniony diagram sekwencji działania aplikacji wykonany przy użyciu narzędzia StarUML



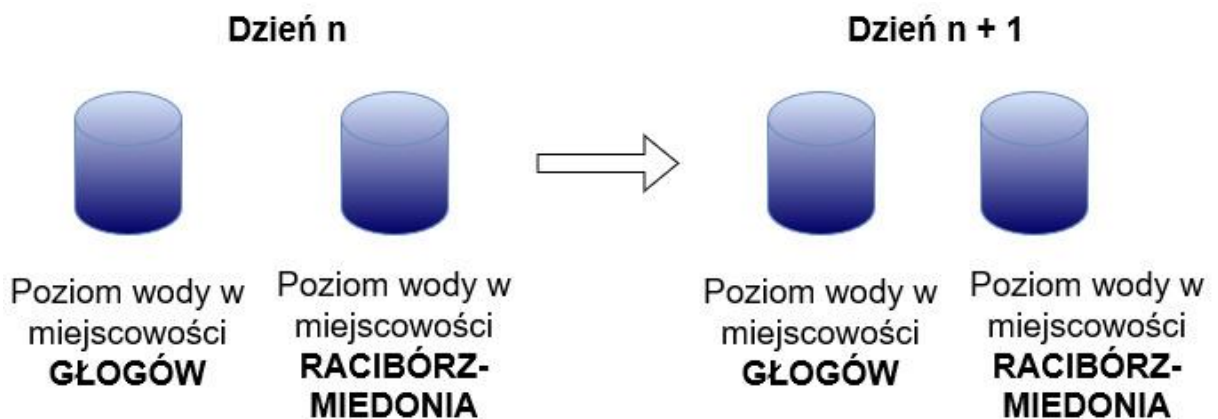
Rys. 5. Logiczny model bazy danych wykonany z wykorzystaniem narzędzia DataModeler



Rys. 6. Fizyczny model bazy danych wykonany z wykorzystaniem narzędzia DataModeler

II.4. Model rozwiązania

Naszym celem jest obliczenie szacunkowego poziomu wody w rzece. Do naszej dyspozycji mamy 2 punkty pomiarowe poziomu wody w rzece oraz 86 punkty meteorologiczne na, których zamierzamy oprzeć nasze przewidywania.



Rys. 7. Schemat

W tym celu zamierzamy zbudować sieć neuronową LSTM opartą na algorytmie:

- t – dzień
- $p_i^t, i \in \{1, 2\}$ – poziom wody w dniu t dla miejscowości {Głogów, Racibórz – Miedonia}
- $s_j^{i,t}, j \in \{1, 2, \dots, 86\}$ – suma prognozowanych opadów w punkcie.
- $x_j^i, j \in \{1, 2, \dots, 86\}$ – wartość odżyłowywania danego punktu na rzekę.

$$p_i^{t+1} = p_i^t + \sum_j^{<1:86>} s_j^{i,t+1} * x_j^i$$

Tak uproszczony model pomimo, braku danych o odległości do rzeki, wysokości na, której punkt się znajduje oraz etc. Jest dokładniejszy ponieważ pozwala zebrać razem wszystkie czynniki o, których nie da się uwzględnić (sztuczne zbiorniki z wodą, bagna, etc.)

W procesie uczenia sieć na podstawie danych historycznych wyznaczy wartości parametrów x dla poszczególnych stacji.

Aby otrzymać prognozowany poziom wody za t dni, należy na wejściu do nauczonej sieci podawać ostatni pomiar poziomu wody oraz prognozowaną sumę opadów w danych punktach. Z rekurencje wywołujemy t razy.

funkcja obliczPoziomWody(p_i^t , lista $< s_j^{i,t} >$):

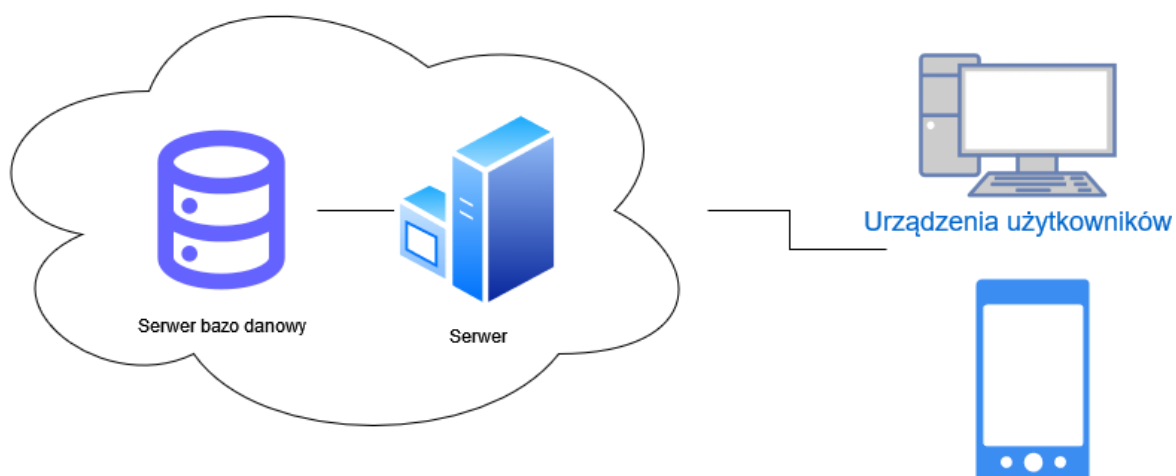
wynik = ... obliczenia w sieci ...

Jeśli został już obliczony poziom wody na wymagany dzień:

Stop

Inaczej:

obliczPoziomWody(wynik, lista $< s_j^{i,t+1} >$)



Rys. 8. Topologia

II.5. Wybrane środowisko oraz narzędzia

II.5.1. Flask

Python Flask – gui aplikacji powstało jako witryna internetowa dzięki temu, jest ona łatwo osiągalna dla zainteresowanych, w ramach potrzeby ograniczenia tylko do wybranych osób można wdrożyć wymaganie logowania lub postawić serwer w sieci wewnętrznej do, której będą miały dostęp tylko wybrane osoby. Flask jest popularnym frameworkiem dla aplikacji webowych.. Pozwala na łatwe i szybkie tworzenie aplikacji webowych, dzięki prostym i modularnym narzędziom. Flask jest w stanie obsługiwać wiele typów aplikacji webowych, od prostych stron internetowych po bardziej zaawansowane aplikacje z interfejsem API. Flask jest dobrym wyborem dla początkujących programistów, ponieważ jest łatwy w użyciu i pozwala na szybkie prototypowanie. Jest też popularny w środowiskach związanych z nauką danych i machine learning, ponieważ jego modułowe podejście pozwala na łatwe integrowanie z innymi narzędziami i bibliotekami.

II.5.2. TensorFlow

TensorFlow to biblioteka open-source do uczenia maszynowego i obliczeń naukowych, która pozwala na budowanie i trenowanie sieci neuronowych. Jest ona szeroko używana w branżach takich jak usługi dla biznesu, rozpoznawanie obrazów, analiza danych, rozpoznawanie mowy i wiele innych.

TensorFlow jest dobry dlatego, że oferuje szeroki zakres funkcji i narzędzi, które umożliwiają łatwą implementację i optymalizację modeli sieci neuronowych. Biblioteka jest bardzo elastyczna i można ją wykorzystać do wielu różnych zastosowań. Ponadto, TensorFlow jest zintegrowany z wieloma innymi bibliotekami i narzędziami do uczenia maszynowego, co pozwala na łatwą integrację z innymi narzędziami i środowiskami.

TensorFlow jest także bardzo dobrze wspierany przez społeczność i stale rozwijany przez Google, co zapewnia regularne aktualizacje i ulepszenia. Dzięki temu jest to narzędzie, które można używać na dłuższą metę, a nie tylko na krótki okres czasu.

Github – kontrola źródeł.

Oracle Autonomous Database jest bezpłatną bazą danych w chmurze, która zapewnia automatyzację i zarządzanie, aby umożliwić szybką i łatwą migrację aplikacji do chmury. Oferuje ona następujące możliwości:

Łatwe tworzenie i skalowanie baz danych: Użytkownicy mogą łatwo utworzyć bazy danych w chmurze bez konieczności konfigurowania sprzętu lub oprogramowania.

Automatyzacja i samodzielne zarządzanie: Baza danych jest zarządzana automatycznie, a procesy takie jak backup, aktualizacja oprogramowania i monitorowanie są zautomatyzowane, co zapewnia niezawodność i niezawodność.

Bezpieczeństwo: Oracle Autonomous Database zapewnia wysokie poziomy bezpieczeństwa, w tym szyfrowanie danych i uwierzytelnianie użytkowników.






Wydajność: Baza danych jest zoptymalizowana pod kątem wydajności, co pozwala na szybką analizę danych i błyskawiczne wykonywanie zadań.


Integracja z innymi usługami chmury: Oracle Autonomous Database jest zintegrowana z innymi usługami chmury, takimi jak Oracle Cloud Infrastructure, co pozwala na łatwe i bezproblemowe łączenie się z innymi usługami.

Oracle Autonomous Database nie oferuje możliwości tworzenia instancji maszyn wirtualnych jako takiej. Jest to rozwiązanie baz danych, które oferuje wysokiej jakości usługi zarządzania bazami danych, takie jak automatyzacja procesów, bezpieczeństwo, skalowalność i wydajność, ale nie oferuje funkcji tworzenia maszyn wirtualnych.

VisualCrossing to platforma, która oferuje narzędzia do analityki geolokalizacyjnej, w tym dostęp do API danych meteorologicznych. Dzięki integracji z różnymi źródłami danych pogodowych, VisualCrossing umożliwia użytkownikom uzyskanie aktualnych informacji dotyczących warunków pogodowych, w tym prognoz pogody, informacji o wietrze, temperaturze, wilgotności, opadach, promieniowaniu słonecznym i wielu innych. Użytkownicy mogą korzystać z tego API, aby integrować dane pogodowe z innymi aplikacjami lub narzędziami, tworzyć własne wizualizacje i raporty, a także automatyzować procesy biznesowe zależne od pogody. W tym projekcie wykorzystano darmową opcję i w ten sposób pozyskiwano w czasie rzeczywistym informacje o pogodzie w danych stacjach meteorologicznych, które z kolei są niezbędne do dalszego przewidywania poziomu Odry.

Pricing

Monthly		Annual 2 months free!		
 Free \$0 /month Get weather data using a free account without any need for a credit card. Choose plan	 Professional \$35 /month For individual users doing planning, data science, personal projects, and prototyping. Choose plan	 Metered \$0.0001 /record The flexible weather data option for apps, teams, and periodic users. Choose plan	 Corporate \$150 /month For teams and small businesses who need to store and share weather data. Choose plan	 Enterprise Contact us For business applications that demand high-availability and unlimited access. Contact us
1000 records/day Single concurrency	10,000,000 records/mo. Single concurrency	Unlimited records/mo. Unlimited concurrency	Unlimited records/mo. Team concurrency	Unlimited records/mo. Unlimited concurrency
<ul style="list-style-type: none">✓ 50 years history✓ 15-day forecast✓ Current conditions	<ul style="list-style-type: none">✓ 50 years history✓ 15-day forecast✓ Current conditions	<ul style="list-style-type: none">✓ 50 years history✓ 15-day forecast✓ Current conditions	<ul style="list-style-type: none">✓ 50 years history✓ 15-day forecast✓ Current conditions	<ul style="list-style-type: none">✓ 50 years history✓ 15-day forecast✓ Current conditions

 Pomoc

Rys. 9. Wybrane możliwości opcji korzystania z VisualCrossing API

JSON

Nieprzetworzone dane

Nagłówki

Zapisz

Kopiuj

Zwiń wszystkie

Rozwiń wszystkie (może to chwilę zająć)

Filtruj JSON

```

queryCost: 1
latitude: 50.8761
longitude: 20.6268
resolvedAddress: "Kielce, Woj. Świętokrzyskie, Polska"
address: "Kielce"
timezone: "Europe/Warsaw"
tzoffset: 1
description: "Similar temperatures con... of snow multiple days."
days:
  0:
    datetime: "2023-01-29"
    datetimeEpoch: 1674946800
    tempmax: -0.7
    tempmin: -3.3
    temp: -1.7
    feelslikemax: -4.7
    feelslikemin: -7
    feelslike: -5.7
    dew: -5.1
    humidity: 78.1
    precip: 0.2
    precipprob: 14.3
    precipcover: 8.33
  preciptype:
    0: "rain"
    1: "snow"
    snow: 0.2
    snowdepth: 0.1
    windgust: 33.5
    windspeed: 15.8
    winddir: 274.2
    pressure: 1023.4
    cloudcover: 91.4
    visibility: 20.7

```

Rys. 10. Fragment otrzymywanego pliku JSON z prognozą pogody

	A	B	C
1			
2	GŁOGÓW (151160060)		RACIBÓRZ-MIEDONIA (150180060)
3	Data	Stan wody [cm]	Stan wody [cm]
4	2011-11-01	199	118
5	2011-11-02	200	118
6	2011-11-03	194	119
7	2011-11-04	195	118
8	2011-11-05	192	116
9	2011-11-06	193	115
10	2011-11-07	203	114
11	2011-11-08	197	113
12	2011-11-09	188	114
13	2011-11-10	179	114
14	2011-11-11	198	117

Rys. 11. Fragment danych z aktualnym poziomem wody podanych przez organizatora

1	BARDO ŚLĄSKIE (250160410)		BIERUTÓW (251170270)		BOLESŁAWÓW (250160610)		BOLKÓW (250160030)		BORÓW (250160070)		
2	Data	Suma opadów [mm]	Status sumy opadów	Suma opadów [mm]	Status sumy opadów	Suma opadów [mm]	Status sumy opadów	Suma opadów [mm]	Status sumy opadów	Suma opadów [mm]	Status sumy opadów
3	01-01-2012	0.2		0.5		0.5		0.7		0.2	
4	02-01-2012			1.1		1		0.8		0	
5	03-01-2012	2.2		0.9		5.6				0	
6	04-01-2012	2.4				12.5					
7	05-01-2012	8.9		7.1		23		10.2		0.5	
8	06-01-2012	1.6		1.9		3.5		3		1.1	
9	07-01-2012	3.8		6.8		2		4.7		4.8	
10	08-01-2012	1.7		3.1		3		4.6		1.4	
11	09-01-2012	3.4		6.1		3.5		6.2		5	
12	10-01-2012	0.1		2.2		2		2.8		0.6	
13	11-01-2012	0.6		0.4		0		1		0	
14	12-01-2012	1.3		2.9		2		3.6		1.8	
15	13-01-2012	2.4		1.9		3.8		1.3		0.4	
16	14-01-2012	0.3		0.3		1.7		1.1		0.5	
17	15-01-2012	0.9		0.3		1.5		0.1		0	
18	16-01-2012	3.9		6.7		3.2		6.3		5.4	
19	17-01-2012	7.8		4.2		10		4.5		5.6	
20	18-01-2012	0.5		2.4		3		1.8		1.3	
21	19-01-2012	2.3		4.4		1		1.4		3.2	
22	20-01-2012	3.2		4.2		0.8		5.6		1.9	

Rys. 12. Fragment danych z aktualnym poziomem wody podanych przez organizatora

II.6. Dokładny opis sieci neuronowej

- Jedna warstwa ukryta składająca się z 64 neuronów
- Funkcja aktywacji dla warstwy wejściowej i ukrytej - relu
- Funkcja aktywacji dla warstwy wyjściowej - linear
- Liczba epok - 300
- Batch size - 32
- Zbiór testowy- 20% całkowitego zbioru wejściowego
- Zostały stworzone dwa modele odpowiednio dla dwóch stacji
- Modele są zapisane i ładowane w razie potrzeb z plików model_{id_stacji}.h5

```

2023-01-29 02:14:19.596116: I tensorflow/core/platform/cpu_feature_guard.cc:193] This TensorFlow binary is optimized with oneAPI Deep Neural Network Library (oneDNN) to use the following CPU instructions in performance-critical operations: AVX AVX2
To enable them in other operations, rebuild TensorFlow with the appropriate compiler flags.
Epoch 1/300
80/80 [=====] - 0s 747us/step - loss: 0.0735
Epoch 2/300
80/80 [=====] - 0s 684us/step - loss: 0.0214
Epoch 3/300
80/80 [=====] - 0s 671us/step - loss: 0.0209
Epoch 4/300
80/80 [=====] - 0s 671us/step - loss: 0.0206
Epoch 5/300
80/80 [=====] - 0s 658us/step - loss: 0.0204
Epoch 6/300
80/80 [=====] - 0s 658us/step - loss: 0.0200
Epoch 7/300
80/80 [=====] - 0s 671us/step - loss: 0.0200
Epoch 8/300
80/80 [=====] - 0s 671us/step - loss: 0.0197
Epoch 9/300
80/80 [=====] - 0s 709us/step - loss: 0.0195
Epoch 10/300
80/80 [=====] - 0s 722us/step - loss: 0.0194

```

Rys. 13. Proces uczenia sieci

```

2023-01-29 02:18:53.415164: I tensorflow/core/platform/cpu_feature_guard.cc:193] This TensorFlow binary is optimized with oneAPI Deep Neural Network Library (oneDNN) to use the following CPU instructions in performance-critical operations: AVX AVX2
To enable them in other operations, rebuild TensorFlow with the appropriate compiler flags.
23/01/28 <class 'datetime.datetime'>
23/01/29 <class 'datetime.datetime'>
1/1 [=====] - 0s 69ms/step
{'ddate': datetime.datetime(2023, 1, 30, 0, 0), 'waterstate': 230.64881563186646, 'stationcode': 150180060, 'ispredicted': True}
{'ddate': datetime.datetime(2023, 1, 30, 0, 0), 'waterstate': 230.64881563186646, 'stationcode': 150180060, 'ispredicted': True}
23/01/30 <class 'datetime.datetime'>
1/1 [=====] - 0s 29ms/step
{'ddate': datetime.datetime(2023, 1, 31, 0, 0), 'waterstate': 300.0054657459259, 'stationcode': 150180060, 'ispredicted': True}
{'ddate': datetime.datetime(2023, 1, 31, 0, 0), 'waterstate': 300.0054657459259, 'stationcode': 150180060, 'ispredicted': True}
23/01/31 <class 'datetime.datetime'>
1/1 [=====] - 0s 29ms/step
{'ddate': datetime.datetime(2023, 2, 1, 0, 0), 'waterstate': 200.4002183675766, 'stationcode': 150180060, 'ispredicted': True}
{'ddate': datetime.datetime(2023, 2, 1, 0, 0), 'waterstate': 200.4002183675766, 'stationcode': 150180060, 'ispredicted': True}

```

Rys. 14. Zapis przewidywanych wyników do bazy

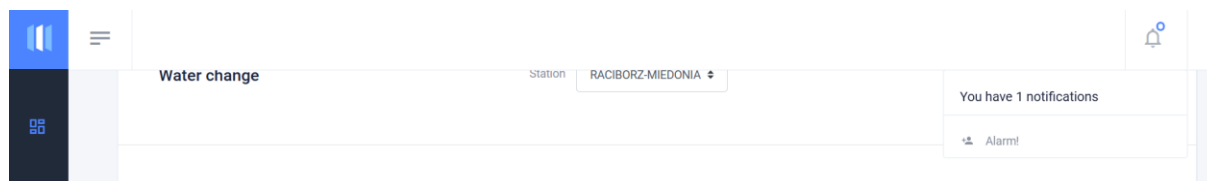
II.7. Opis wybranych fragmentów kodu

Klasa	Funkcja
dbConnect	Współpraca z bazą danych, aktualizowanie, zapisywanie nowych elementów, pobieranie danych
FileWork	Współpraca z plikami csv, xlsx oraz api do prognozy pogody. Klasa ta wczytuje oraz przygotowuje dane, które zostaną następnie wykorzystane
sieciNeuronowe	Generator modeli sieci, odpowiada za przygotowanie danych do modelu, uczenie oraz douczanie modelu sieci neuronowej.
sieciInserToBase	Klasa wykorzystująca opracowany model sieci neuronowej w formacie .h5 to przewidywania stanu wody na podstawie na podstawie wcześniej pobranych danych.
hello	Klasa bazująca na Flasku, służy do prezentacji danych w postaci witryny web.

II.8. Możliwości rozwoju środowiska

W przyszłości należy douczyć się regularnie sieć neuronową na podstawie nowych danych, pozwoli to na zwiększenie jej dokładności.

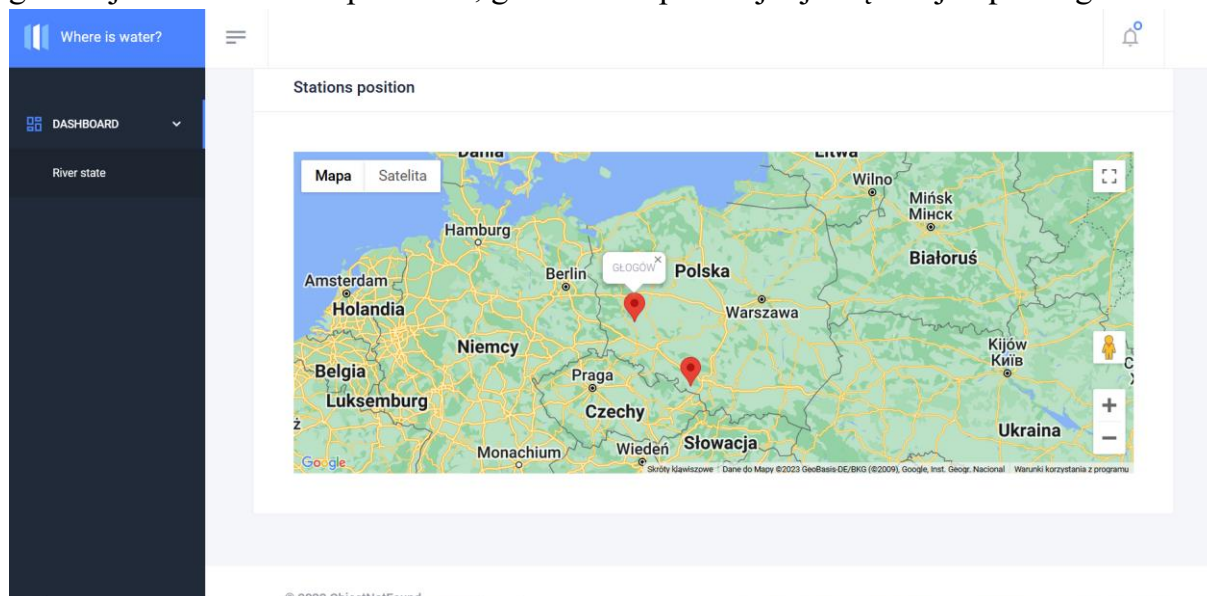
Należy zadbać o możliwość pozyskiwania pogody dla dłuższego odcinka czasu, co pozwoli przewidywać poziom wody dalej w przód. Takie możliwości są, jednak wiąże się to z kosztami. Również możliwe jest w przyszłości wprowadzenie opcji dodawania alarmów, kiedy poziom wody osiągnie zadany stan.



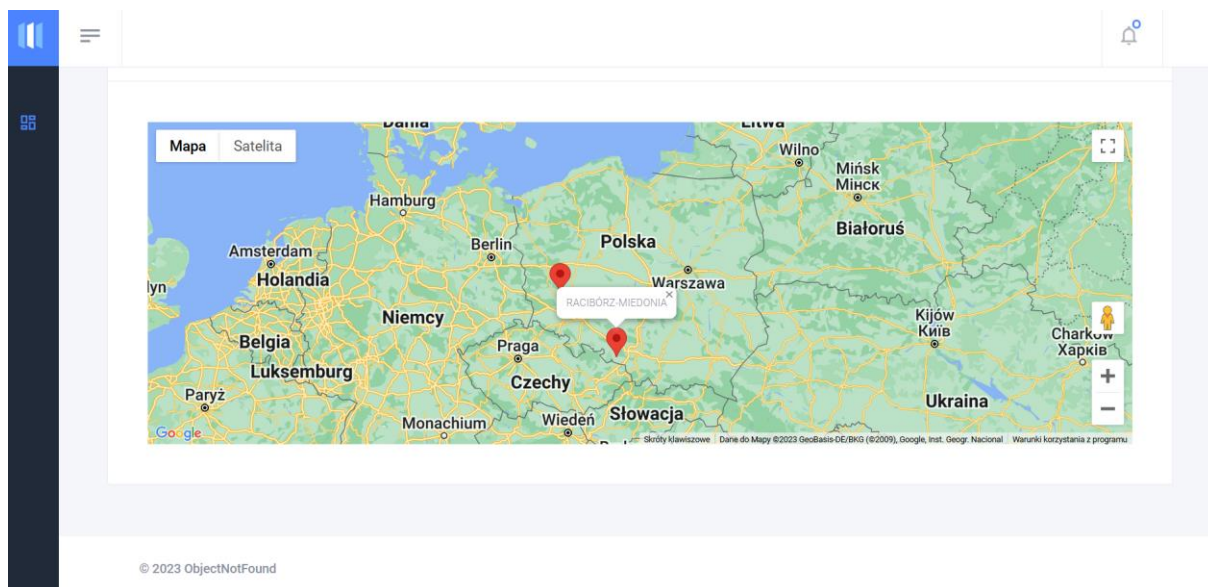
Rys. 15. Możliwość tworzenia w przyszłości alertów

Rozdział III. Wygląd aplikacji

Aplikacja webowa posiada dość intuicyjne menu. Po lewej stronie widnieje wysuwane menu, które można rozbudowywać w późniejszych etapach. Użytkownik na stronie głównej może również sprawdzić, gdzie na mapie znajduje się stacja np. Głogów.

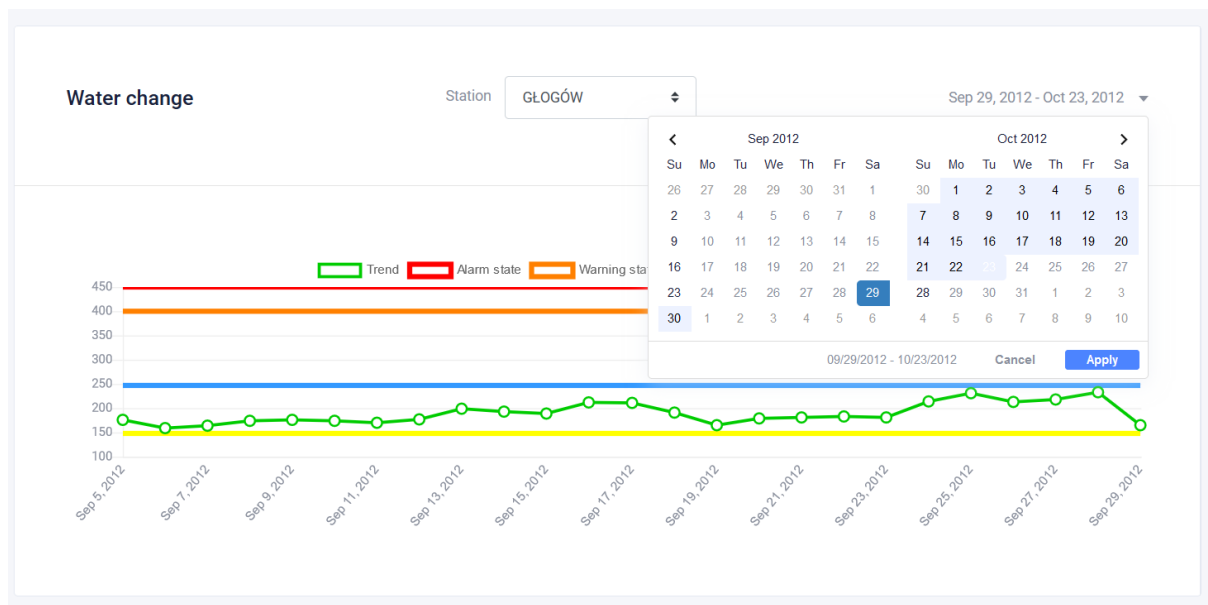


Rys. 16. Mapa ze stacjami

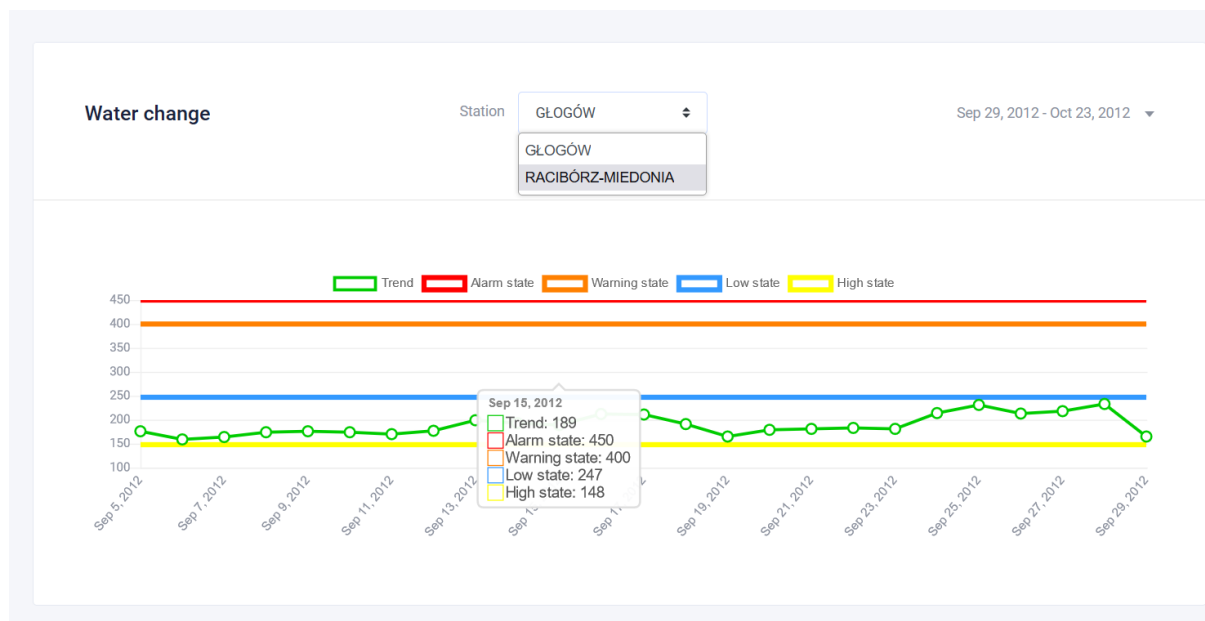


Rys. 17. Mapa ze stacjami

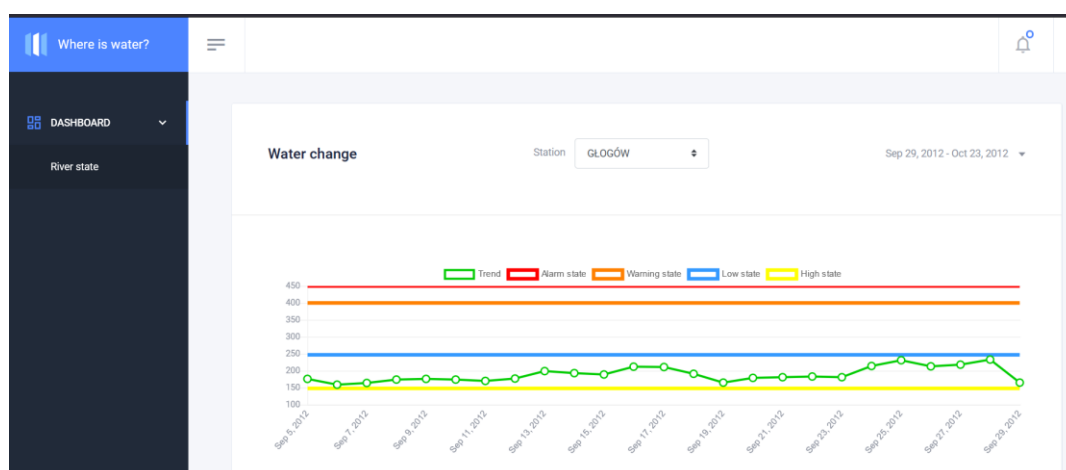
Następnie użytkownik może wybrać zakres dni jaki go interesuje. Możliwy jest również wybór konkretnej stacji z rozwijanej listy. Po najechnaniu na wykres widoczne są dokładne wartości.



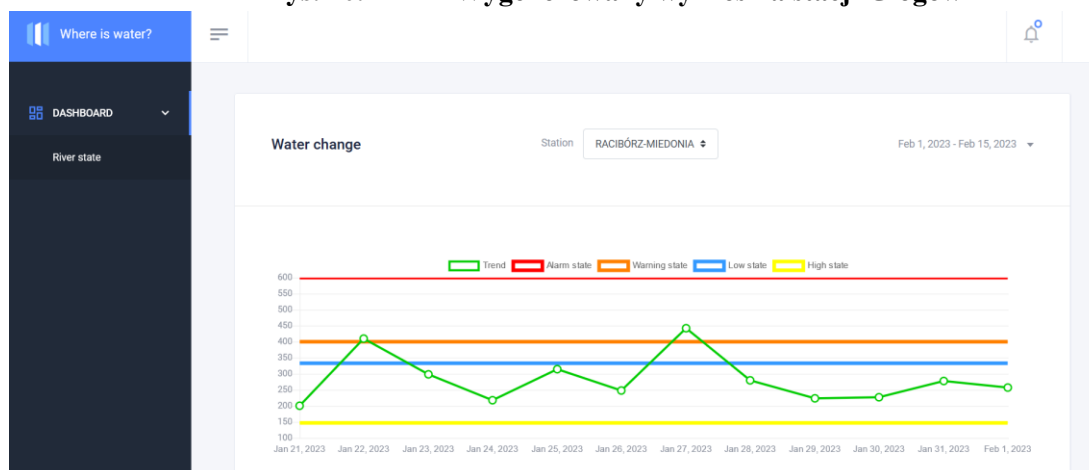
Rys. 18. Możliwość wyboru konkretnych dni w kalendarzu



Rys. 19. Możliwość wyboru stacji

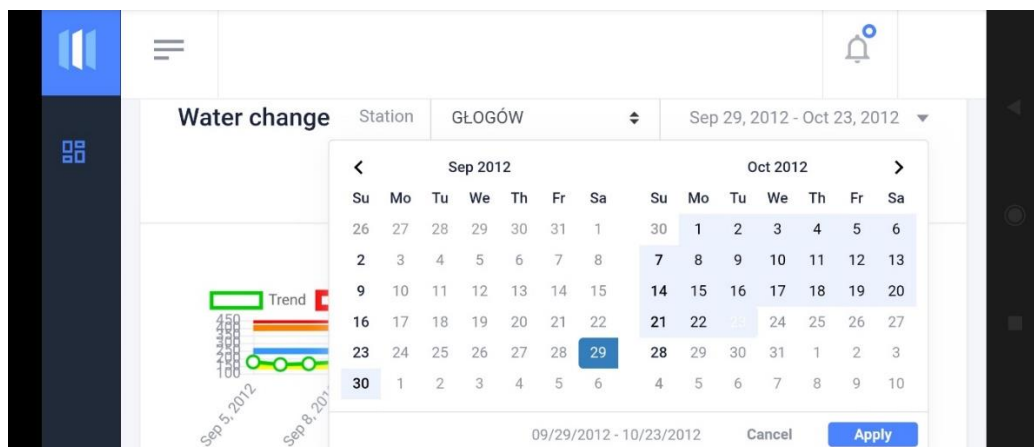


Rys. 20. Wygenerowany wykres na stacji Głogów

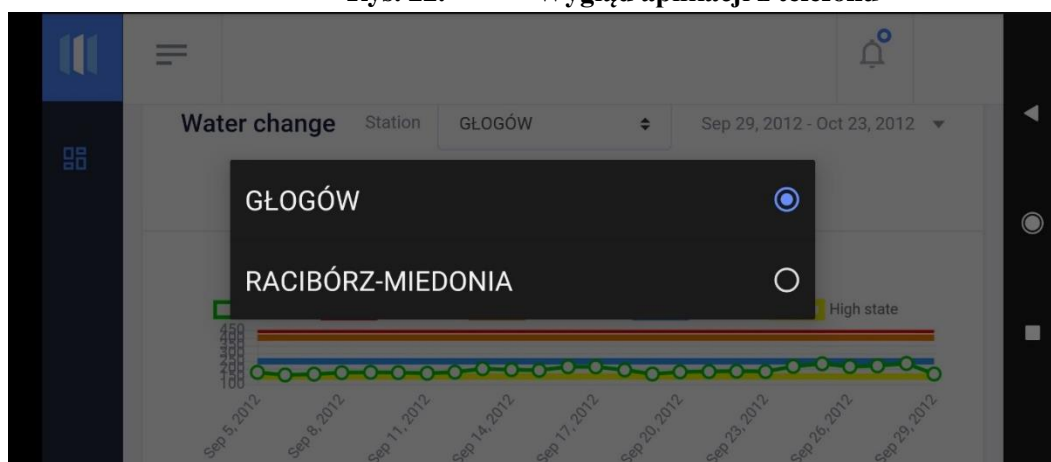


Rys. 21. Wygenerowany wykres dla stacji Racibórz-Miedonia

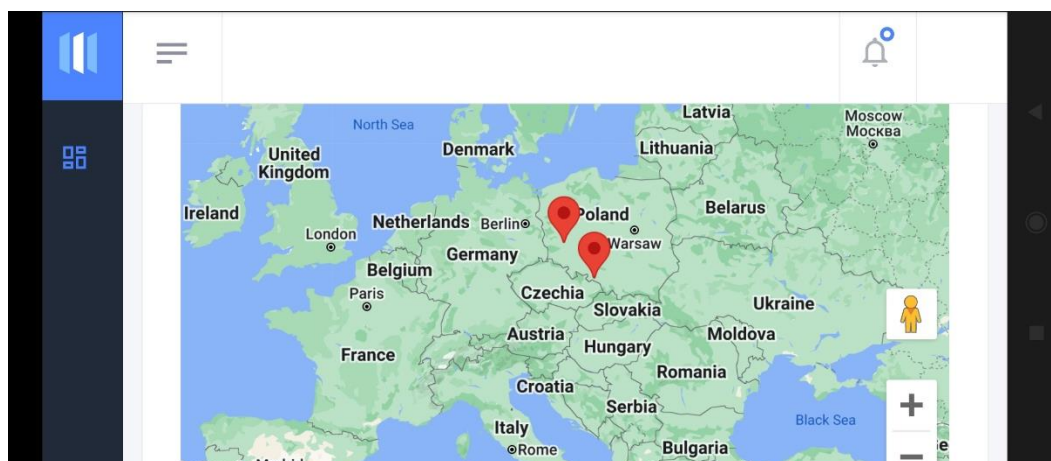
Aplikacja również może być uruchamiana z telefonu mobilnego.



Rys. 22. Wygląd aplikacji z telefonu



Rys. 23. Wygląd aplikacji z telefonu



Rys. 24. Wygląd aplikacji z telefonu

Podsumowanie

Flask jest lekkim frameworkiem do tworzenia aplikacji webowych w języku Python, który jest szybki, łatwy w użyciu i elastyczny. W połączeniu z sieciami neuronowymi, Flask pozwala na stworzenie aplikacji, która będzie w stanie szacować poziom wody w rzece z wysokim prawdopodobieństwem, nawet na bardzo odległy horyzont czasowy.

Aplikacja będzie korzystać z API dostarczającego dane o warunkach pogodowych, takie jak temperatura, opady, ciśnienie atmosferyczne i inne, aby określić, jak będą one wpływać na poziom wody w rzece. Następnie, te dane będą przetwarzane przez sieć neuronową, aby przewidzieć, jak poziom wody będzie się zmieniał w przyszłości.

Flask umożliwia łatwą integrację z sieciami neuronowymi poprzez biblioteki takie jak TensorFlow i Keras. Te biblioteki pozwolą na szybkie i łatwe trenowanie sieci neuronowej i uzyskanie dokładnych wyników.

W aplikacji będzie również zawarty interaktywny kalendarz, który pozwoli użytkownikom wybrać zakres czasowy, w którym chcą zobaczyć poziom wody

w rzece. Po wybraniu zakresu czasowego, aplikacja automatycznie odświeży się i wygeneruje wykres poziomu wody w rzece w wybranym okresie czasu.

Podsumowując, aplikacja webowa z wykorzystaniem Flaska i sieci neuronowej jest skutecznym narzędziem do szacowania poziomu wody w rzece z możliwie wysokim prawdopodobieństwem i z odległym horyzontem czasowym. Aplikacja jest łatwa w użyciu i umożliwia łatwe generowanie wykresów, co ułatwia interpretację danych i podejmowanie decyzji.