

ОДЕСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ І. І.
МЕЧНІКОВА

(повне найменування закладу вищої освіти)

Факультет математики, фізики та інформаційних технологій

(повне найменування факультету)

Кафедра інформаційних технологій

(повна назва кафедри)

Кваліфікаційна робота

на здобуття ступеня вищої освіти «Магістр»

«Веб-сервіс для прогнозування якості повітря за

допомогою нейромережевих технологій»

(тема кваліфікаційної роботи українською мовою)

**«Web Service for Air Quality Prediction Using Neural
Network Technologies»**

(тема кваліфікаційної роботи англійською мовою)

Виконав: здобувач денної форми навчання
спеціальності 122 Комп'ютерні науки

(код, назва спеціальності)

Освітня програма Комп'ютерні науки

(назва)

Яковенко Максим Олегович

(прізвище, ім'я, по-батькові здобувача)

Керівник к.ф.-м.н, Ткач Т.Б.

(науковий ступінь, вчене звання, прізвище, ініціали)

(підпис)

Рецензент к.т.н., доцент кафедри інженерії ПЗ

національного університету "Одеська

політехніка", Зіноватна Світлана

Леонідівна

(науковий ступінь, вчене звання, прізвище, ініціали)

Рекомендовано до захисту:

Протокол засідання кафедри

Інформаційних технологій

№ від 2024 р.

Завідувачка кафедри

(підпис)

(прізвище, ім'я)

Захищено на засіданні ЕК №

протокол № від 2024 р.

Оцінка / /

(за національною шкалою/шкалою ECTS/ бали)

Голова ЕК

(підпис)

(прізвище, ім'я)

Одеса 2024

АНОТАЦІЯ

У магістерській роботі розглядається розробка веб-сервісу для прогнозування якості повітря за допомогою нейромережових технологій. Метою дослідження є створення інноваційного інструменту, що забезпечує оперативний доступ до даних про стан повітря в реальному часі для міст України. Сервіс сприятиме підвищенню екологічної обізнаності населення, допомагатиме мінімізувати негативний вплив забруднення повітря на здоров'я громадян та підтримувати прийняття обґрунтованих рішень.

Робота включає інтеграцію сучасних технологій збору даних із супутників, сенсорів і метеорологічних джерел, впровадження алгоритмів машинного навчання для аналізу та прогнозування, а також розробку зручного інтерфейсу користувача. Результати дослідження можуть бути корисними як для широкого загалу, так і для організацій, які займаються екологічним моніторингом та управлінням якістю повітря.

ABSTRACT

The master's thesis focuses on developing a web service for air quality forecasting using neural network technologies. The aim of the study is to create an innovative tool that provides real-time access to air quality data for cities in Ukraine. The service is designed to enhance public awareness of environmental conditions, help minimize the adverse effects of air pollution on citizens' health, and support informed decision-making.

The work involves integrating modern data collection technologies from satellites, sensors, and meteorological sources, implementing machine learning algorithms for analysis and forecasting, and developing a user-friendly interface. The research outcomes can benefit both the general public and organizations involved in environmental monitoring and air quality management.

ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ ТА ТЕРМІНІВ	5
ВСТУП	6
1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ.....	8
1.1 Опис предметної області.....	8
1.2 Огляд аналогів.....	9
1.3 Постановка завдання та вимоги до сервісу	11
2 ПРОЄКТУВАННЯ ВЕБ-СЕРВІСУ	14
2.1 Основні компоненти проєкту	14
2.2 Система рекомендацій.....	20
2.3 Динамічне масштабування.....	21
2.4 Діаграма варіантів використання	22
2.5 Проєктування компонентів прогнозування.....	23
2.6 Проєктування бази даних.....	27
2.7 Проєктування архітектури сервісу.....	28
2.8 Проєктування REST API	29
3 РОЗРОБКА ВЕБ-СЕРВІСУ	30
3.1 Інструменти розробки	30
3.2 Основні етапи розробки	32
3.3 Реалізація та навчання моделей прогнозування	40
4 ТЕСТУВАННЯ.....	45
4.1 Тестування веб-сервісу.....	45
ВИСНОВКИ.....	52
ПЕРЕЛІК ВИКОРИСТАНИХ ПОСИЛАНЬ.....	53

ПЕРЕЛІК СКОРОЧЕНЬ ТА ТЕРМІНІВ

AI (Artificial Intelligence) — штучний інтелект, технологія моделювання людського інтелекту в машинах.

AQI (Air Quality Index) — індекс якості повітря, числовий показник, що характеризує рівень забруднення повітря.

PM_{2.5}, PM₁₀ — дрібнодисперсні частинки діаметром до 2.5 мкм та до 10 мкм, відповідно, які є одним з основних забруднювачів повітря.

CNN (Convolutional Neural Network) — згортова нейронна мережа, тип нейромережі, що використовується для обробки просторових даних.

RNN (Recurrent Neural Network) — рекурентна нейронна мережа, модель штучного інтелекту для аналізу часових рядів.

IoT (Internet of Things) — Інтернет речей, мережа підключених пристроїв, що обмінюються даними через інтернет.

API (Application Programming Interface) — програмний інтерфейс, який дозволяє взаємодіяти між різними програмами.

NO_x — оксиди азоту, група хімічних речовин, що забруднюють атмосферу.

CO — чадний газ (оксид вуглецю), один із забруднювачів, шкідливий для здоров'я.

O₃ (Ozone) — озон, газ, що може виступати як забруднювач у приземному шарі атмосфери.

SO₂ — діоксид сірки, забруднювач, що впливає на якість повітря.

Leaflet.js — бібліотека JavaScript для створення інтерактивних карт.

OpenWeather API — сервіс для отримання даних про погоду та якість повітря в реальному часі.

UI (User Interface) — користувацький інтерфейс, візуальне середовище для взаємодії користувача із системою.

ВСТУП

У сучасному світі, де екологічна проблематика займає центральне місце в глобальних дискусіях, якість повітря стала не лише питанням екологічної безпеки, але й безпосередньо впливає на здоров'я та життя людей по всьому світу. Втрати, пов'язані з погіршенням якості повітря, оцінюються мільйонами життів щорічно, роблячи цю проблему однією з найбільш нагальних для розвинених та розвиваючихся країн.

Підвищена концентрація забруднюючих речовин у повітрі може призвести до серйозних наслідків для здоров'я, включаючи респіраторні, серцево-судинні, ракові та інші хронічні захворювання. Крім того, забруднене повітря може впливати на екосистеми, водні ресурси та сільське господарство.

Використання веб-сервісів для прогнозування якості повітря дозволяє не лише забезпечувати громадськість актуальною інформацією, але й впливати на поведінку та рішення людей у реальному часі. Наприклад, інформація про негативні показники якості повітря може спонукати людей до використання захисних масок або обмеження активності на вулицях, що в свою чергу може зменшити вплив забруднення на здоров'я.

Таким чином, веб-сервіси прогнозування якості повітря є важливим інструментом для індивідуальних і колективних дій щодо забруднення повітря, їхня роль у зміцненні здоров'я населення та охорони навколишнього середовища є неоціненною, а подальший розвиток і вдосконалення таких сервісів може мати значний потенційний вплив на якість життя мільйонів людей у всьому світі.

Якість повітря в цілому дуже важлива для здоров'я людини, адже кисень життєво необхідний для функціонування нашого організму. Однак через високий рівень промислового забруднення та індустріалізації, а також наслідки війни, зараз майже неможливо знайти місця зі справді чистим

повітрям. Людський організм здатний самостійно виводити часточки дрібного пилу, але у дуже невеликих обсягах та повільно. PM_{2,5} мають помірно-негативний вплив на здоров'я лише тоді, коли людина не живе в умовах забрудненого атмосферного повітря, а дихає ним лише час від часу.

Не існує спеціальних способів очистити організм від дрібного пилу, можна лише допомогти організму це робити. Важливо періодично перевіряти прогноз погоди та індекс якості повітря (AQI) за шкалою від 0 до 500.

Значення AQI поділяються на рівнів: добрий (0-50), помірний (51-100), шкідливий рівень для чутливих груп і (101-150), шкідливий (151-200), дуже шкідливий (201-300) і небезпечний (301+).

Метою роботи є створення інноваційного веб-сервісу, який забезпечує оперативний доступ до даних про якість повітря в реальному часі для міст України. Цей сервіс має сприяти підвищенню обізнаності громадян щодо екологічного стану їхнього середовища проживання, а також допомагати приймати виважені рішення для зменшення впливу забруднення повітря на здоров'я. У рамках реалізації ставиться завдання об'єднати сучасні технології збору даних із доступними інструментами візуалізації та розробити простий у використанні інтерфейс, який відповідатиме потребам широкої аудиторії, включаючи як пересічних громадян, так і організації, зацікавлені в екологічному моніторингу.

Об'єктом дослідження є процеси аналізу, обробки та прогнозування якості повітря на основі даних та нейромережових алгоритмів.

Предметом дослідження є технології збору, обробки та візуалізації даних про якість повітря, які інтегруються у веб-сервіс.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Опис предметної області

Екологічне прогнозування як напрямок досліджень і розробок стало ключовим інструментом для забезпечення сталого розвитку та запобігання екологічним катастрофам. Основним завданням системи прогнозування якості повітря є моніторинг концентрацій забруднюючих речовин і викидів в атмосферу, таких як дрібнодисперсні частинки (PM2.5 і PM10), оксиди азоту (NOx), озон (O₃) та інші забруднювачі. Відстеження та прогнозування змін у якості повітря дозволяє вчасно вживати заходів для зменшення негативного впливу на здоров'я людей та екосистему [1].

Дані для прогнозування зазвичай збираються через стаціонарні станції моніторингу, супутникові знімки та сенсори IoT, що встановлюються в містах і промислових зонах. Для аналізу отриманих даних застосовуються сучасні методи обробки даних, серед яких ключову роль відіграють нейронні мережі та інші алгоритми штучного інтелекту.

Об'єкт розробки – це система прогнозування якості повітря, побудована на основі нейромережевих технологій. Основним завданням такого сервісу є отримання даних з різних сенсорних мереж та їх обробка для передбачення рівня забруднення в майбутньому. Нейронні мережі дозволяють аналізувати великі обсяги даних, включаючи історичні показники, погодні умови, трафік та інші фактори, щоб надавати точні прогнози.

Така система повинна враховувати широкий спектр факторів, включаючи:

- метеорологічні дані (температура, вологість, швидкість вітру);
- джерела забруднення (промислові викиди, транспортні потоки);
- географічні умови (топографія, розташування міських агломерацій);
- дані моніторингу якості повітря в режимі реального часу.

Така система працює наступним чином:

1. Отримує дані з датчиків, супутників, метеорологічних станцій.
2. Обробляє дані через нейромережі, що здатні виявляти складні залежності між показниками забруднення та іншими факторами.
3. Формує прогнози для різних часових періодів, що можуть охоплювати як короткострокові (години, дні), так і довгострокові перспективи (тижні, місяці).
4. Надає візуалізацію прогнозів у вигляді карт або графіків, доступних через веб-інтерфейс.

1.2 Огляд аналогів

На ринку існують декілька рішень, які спрямовані на моніторинг якості повітря з використанням сучасних технологій.

AirVisual (IQAir) – глобальний веб-сервіс, що надає дані про якість повітря у більш ніж 10 000 містах [2]. Система використовує власні сенсори та метеорологічні дані для формування індексу AQI та прогнозів на основі машинного навчання (рис.1.1).

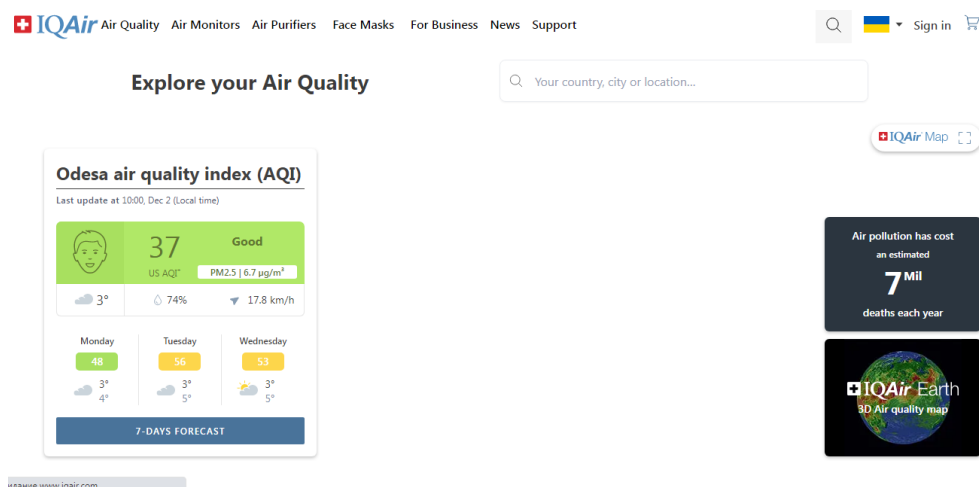


Рисунок 1.1 – Сервіс IQAir

BreezoMeter – платформа, що надає високоточні прогнози якості повітря за допомогою аналізу супутникових даних, даних із сенсорів та метеорологічних джерел. Використовує штучний інтелект для прогнозування впливу на здоров'я людей та попередження про потенційні загрози [3].

BreezoMeter також надає користувачам інтуїтивно зрозумілі мобільні додатки та інтеграції з іншими платформами для відстеження рівня забруднення повітря в реальному часі. Завдяки гнучким налаштуванням, платформа дозволяє адаптувати прогнози під конкретні потреби користувачів та місцеві умови.

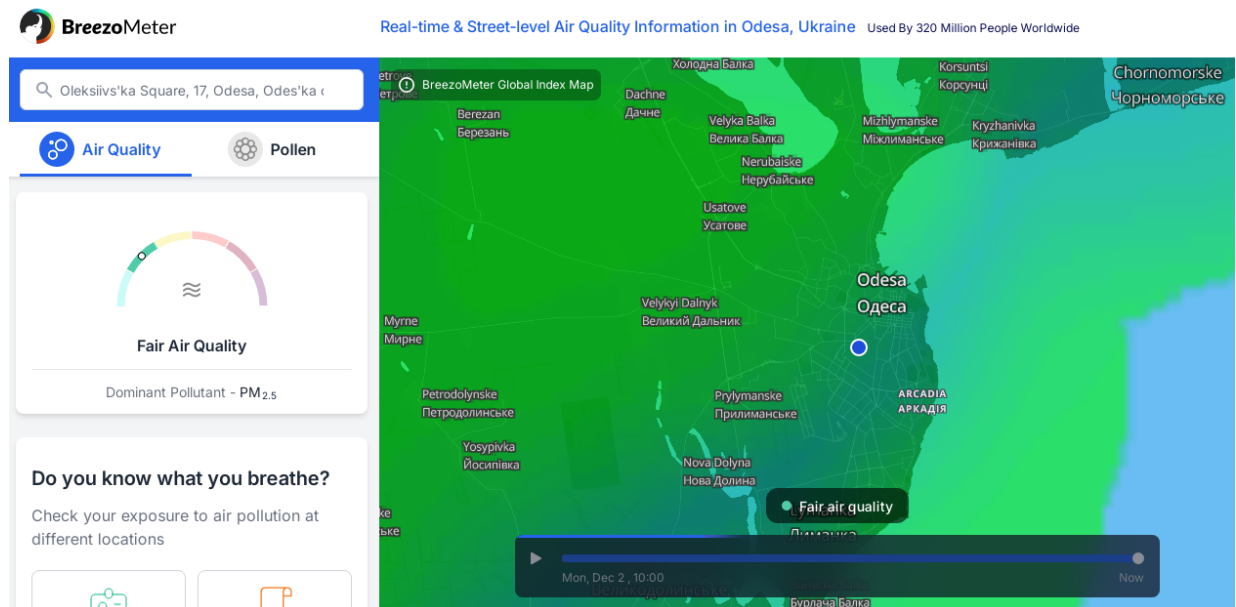


Рисунок 1.2 – Сервіс BreezoMeter

Plume Labs – французький сервіс, який використовує нейромережі для прогнозування забруднення повітря, з фокусом на інтеграцію даних з мобільних пристроїв користувачів, що дозволяє покращити точність прогнозів в режимі реального часу [4].

Plume Labs також пропонує мобільний додаток, який дозволяє користувачам отримувати персоналізовані рекомендації щодо зменшення

впливу забруднення на здоров'я. Сервіс активно співпрацює з дослідницькими установами для підвищення ефективності своїх алгоритмів і поширення знань про екологічні загрози.

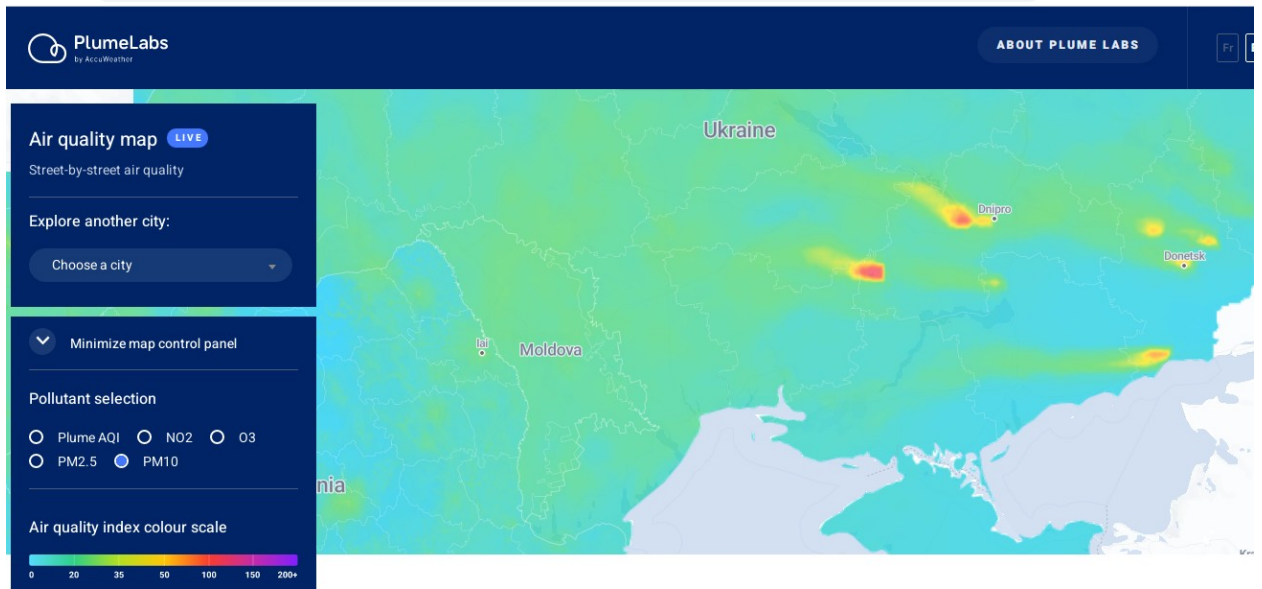


Рисунок 1.2 – Plume Labs

1.3 Постановка завдання та вимоги до сервісу

Основні вимоги до проєкту можна поділити на функціональні та нефункціональні.

Функціональні вимоги:

1. Можливість збору даних з різних джерел: сенсорів, метеорологічних станцій, супутників.
2. Обробка та аналіз даних за допомогою нейронних мереж для прогнозування рівня забруднення.
3. Виведення прогнозу на різні часові інтервали.
4. Надавання користувачам інтуїтивного інтерфейсу для отримання прогнозів у реальному часі.

Нефункціональні вимоги:

1. Надійність системи: вона повинна працювати безперервно і мати високу точність прогнозів.
2. Масштабованість: система має бути готова до розширення кількості точок збору даних і обробки великих обсягів інформації.
3. Швидкодія: користувачі повинні отримувати актуальні дані максимально швидко.
4. Безпека: захист даних користувачів та безпечне зберігання сенсорної інформації.

Вибір нейронної мережі.

Рекурентні нейронні мережі (RNN): добре підходять для обробки часових рядів і передбачення на основі історичних даних. Вони можуть враховувати попередні значення якості повітря і погоди для прогнозування майбутніх показників.

Конволюційні нейронні мережі (CNN): корисні для аналізу просторових даних, таких як карти забруднення чи зображення з супутників.

Гібридні моделі: поєднання RNN і CNN може значно підвищити точність прогнозів, оскільки дозволяє враховувати як часові, так і просторові аспекти проблеми.

Основні функції та можливості веб-сервісу включають:

1. Моніторинг якості повітря в режимі реального часу, з відображенням даних на інтерактивних картах.
2. Прогнозування на майбутнє (години, дні, тижні) з урахуванням різних екологічних і метеорологічних факторів.
3. Сповіщення користувачів про небезпечний рівень забруднення, особливо для людей з хронічними захворюваннями або уразливих груп населення.

4. Аналіз впливу на здоров'я: рекомендації щодо заходів для захисту від забруднення (носіння масок, обмеження активності на відкритому повітрі).

5. Звіти та візуалізації для аналітиків, екологів та муніципальних служб.

Ця система не лише допомагає окремим користувачам, а й може стати важливим інструментом для урядів і організацій, які приймають рішення на основі екологічної безпеки.

Об'єктом дослідження є атмосферне повітря та його якість у міських і приміських зонах України, які найбільше зазнають впливу антропогенних чинників. Особлива увага приділяється концентраціям забруднюючих речовин, таких як PM_{2.5}, PM₁₀, CO, NO₂, SO₂ і O₃, які визначають ступінь безпечності або небезпеки атмосферного повітря для здоров'я населення.

Дослідження також охоплює аналіз факторів, що впливають на рівень забруднення, включаючи інтенсивність автомобільного руху, промислові викиди, погодні умови та сезонні коливання. Зібрані дані дозволяють оцінити екологічну ситуацію в кожному конкретному регіоні, виявити найбільш проблемні райони та надати рекомендації для населення.

Предметом дослідження є технології збору, обробки та візуалізації даних про якість повітря, які інтегруються у веб-сервіс. Вивчаються способи взаємодії з OpenWeather API для отримання актуальної інформації про концентрацію забруднювачів і індекс якості повітря (AQI), алгоритми кольорового кодування даних для їх швидкого сприйняття користувачами, а також методи динамічного оновлення даних у режимі реального часу.

2 ПРОЄКТУВАННЯ ВЕБ-СЕРВІСУ

2.1 Основні компоненти проєкту

Об'єктом розробки є веб-сервіс, що забезпечує моніторинг і прогнозування якості повітря в реальному часі для міст України, зокрема для Одеси. Головною метою такого сервісу є оперативне надання користувачам інформації про екологічний стан, що дозволяє приймати своєчасні заходи для збереження здоров'я та мінімізації впливу забруднень на організм. Основна увага приділяється інтерактивності й доступності, щоб кожен користувач, незалежно від рівня технічних знань, міг швидко зрозуміти екологічну ситуацію у своєму регіоні.

Сервіс базується на сучасних веб-технологіях і використовує інтерактивну карту для візуалізації даних про якість повітря. Дані збираються через API OpenWeather, яке забезпечує доступ до інформації про концентрацію забруднюючих речовин, таких як дрібнодисперсні частинки PM2.5 і PM10, діоксид азоту, озон, чадний газ та інші. Система автоматично аналізує ці показники та визначає індекс якості повітря (AQI), що дозволяє оцінити рівень безпеки атмосфери.

Особливість цього сервісу полягає у його здатності адаптуватися до різних регіонів України, враховуючи специфіку забруднень, кліматичні умови та географічні особливості кожного міста. Наприклад, в Одесі система враховує вплив морського повітря, щільність автомобільного трафіку та промислові викиди, що значно впливають на стан атмосфери. Ця функціональність реалізована завдяки інтеграції сучасних алгоритмів обробки даних, які забезпечують точність прогнозів навіть у складних екологічних умовах.

Крім збору та обробки даних, сервіс надає користувачам рекомендації щодо поведінки в умовах підвищеного забруднення. Інтерактивна карта

дозволяє користувачам обирати міста та переглядати не лише поточні показники якості повітря, але й прогнози на найближчі дні. Це особливо корисно для людей, які планують активності на відкритому повітрі, або для тих, хто належить до груп ризику — людей із хронічними захворюваннями, дітей та літніх осіб.

Важливим елементом є зручність користування. Інтерфейс розроблений так, щоб мінімізувати кількість дій, необхідних для отримання інформації. Усе, що потрібно користувачеві, — це обрати місто на інтерактивній карті. Дані автоматично оновлюються, а кольорове маркування дозволяє одразу зрозуміти рівень забруднення. Наприклад, зелений колір означає, що повітря чисте, жовтий попереджає про незначні забруднення, а червоний сигналізує про небезпеку для здоров'я.

Таким чином, цей веб-сервіс виконує не лише інформаційну, але й соціальну функцію, сприяючи підвищенню екологічної обізнаності населення та мотивуючи до дій, спрямованих на збереження навколишнього середовища.

Об'єкт розробки використовує інтерактивну карту як основний інструмент для візуалізації даних про якість повітря. Карта дозволяє зручно відображати інформацію в реальному часі, забезпечуючи користувачам доступ до ключових показників екологічного стану для обраних міст. На ній розміщені маркери, що позначають міста, кожен із яких кодується кольором відповідно до індексу якості повітря (AQI).

Користувач може клікнути на будь-який маркер, щоб побачити деталізовану інформацію про концентрацію забруднюючих речовин, таких як PM_{2.5}, PM₁₀, NO₂, CO, SO₂ та озон. Інтерактивність карти забезпечується за допомогою бібліотеки Leaflet.js, яка дозволяє не лише масштабувати та переміщати карту, але й додавати спливаючі вікна з детальними даними.

Цей підхід робить систему максимально зрозумілою та зручною для користувачів, незалежно від їхнього рівня технічної підготовки, та дозволяє

швидко отримувати актуальну інформацію про стан повітря в будь-якому з доступних регіонів.

Основною метою розробки веб-сервісу є створення платформи, яка дозволить користувачам отримувати своєчасну та точну інформацію про якість повітря. Це забезпечить можливість не лише оцінювати екологічну ситуацію в конкретних регіонах, але й приймати обґрунтовані рішення для зменшення ризиків для здоров'я. Для досягнення цієї загальної мети були визначені три ключові напрями, які об'єднують технічні, соціальні та інформаційні аспекти проєкту.

Надання доступу до актуальних даних про якість повітря.

Першочерговим завданням є забезпечення точного й оперативного доступу до даних про якість повітря. Сервіс використовує глобальне API, яке надає інформацію про концентрацію основних забруднюючих речовин в атмосфері (PM2.5, PM10, CO, NO₂, SO₂, O₃) і загальний індекс AQI. Дані оновлюються в режимі реального часу, що дозволяє відображати актуальну екологічну ситуацію в містах України.

Завдяки інтеграції інтерактивної карти користувачі мають можливість миттєво отримувати доступ до інформації, обираючи відповідний регіон або точку моніторингу. Це рішення може допомогати як окремим громадянам, так і організаціям, які залежать від об'єктивних екологічних показників.

Підвищення обізнаності громадян про екологічний стан.

Однією з важливих цілей є формування в суспільстві розуміння важливості контролю за якістю повітря. Сервіс не лише надає дані, але й пояснює їхнє значення через прості й зрозумілі інтерфейси.

Це сприяє підвищенню екологічної обізнаності та формуванню культури відповідального ставлення до довкілля, мотивуючи громадян до дій, спрямованих на зменшення впливу шкідливих чинників.

Запобігання негативному впливу на здоров'я шляхом рекомендацій

Одним із важливих елементів сервісу є рекомендації щодо захисту здоров'я в умовах забрудненого повітря. На основі отриманих даних система автоматично генерує поради для користувачів залежно від рівня AQI. Наприклад, при високих значеннях індексу (зони небезпечного чи дуже небезпечного забруднення) сервіс рекомендує зменшити час перебування на відкритому повітрі, використовувати захисні маски або залишатися вдома. Також надаються загальні поради щодо зменшення негативного впливу, такі як встановлення очищувачів повітря у приміщеннях чи уникання активностей поблизу промислових зон. Завдяки цьому громадяни можуть мінімізувати шкоду для здоров'я навіть у складних екологічних умовах.

Концентрації забруднювачів (рис.2.1).

Крім індексу якості повітря (AQI), важливою частиною інформаційної секції є концентрації забруднювачів, які вказують на рівень конкретних шкідливих речовин у повітрі. Для кожного міста система отримує дані про кількість основних забруднюючих компонентів, що можуть впливати на здоров'я людей, таких як:

PM_{2.5} (дрібнодисперсні частинки розміром до 2.5 мкм) – є основними шкідливими елементами забруднення повітря, які здатні проникати в легені і спричиняти серйозні проблеми зі здоров'ям.

PM₁₀ (частинки до 10 мкм) – більші частинки, які також можуть мати негативний вплив, але з меншим проникненням у глибокі шари легенів.

CO (чадний газ) – безбарвний газ, що утворюється при спалюванні органічних матеріалів, який може спричинити отруєння.

NO₂ (діоксид азоту) – газ, що утворюється в результаті діяльності транспорту і промислових підприємств.

SO₂ (діоксид сірки) – продукт спалювання вугілля та інших видів палива, що може викликати дихальні захворювання.

О₃ (озон) – компонент, який утворюється при взаємодії викидів автомобілів і сонячного випромінювання, є небезпечним при високих концентраціях [5].

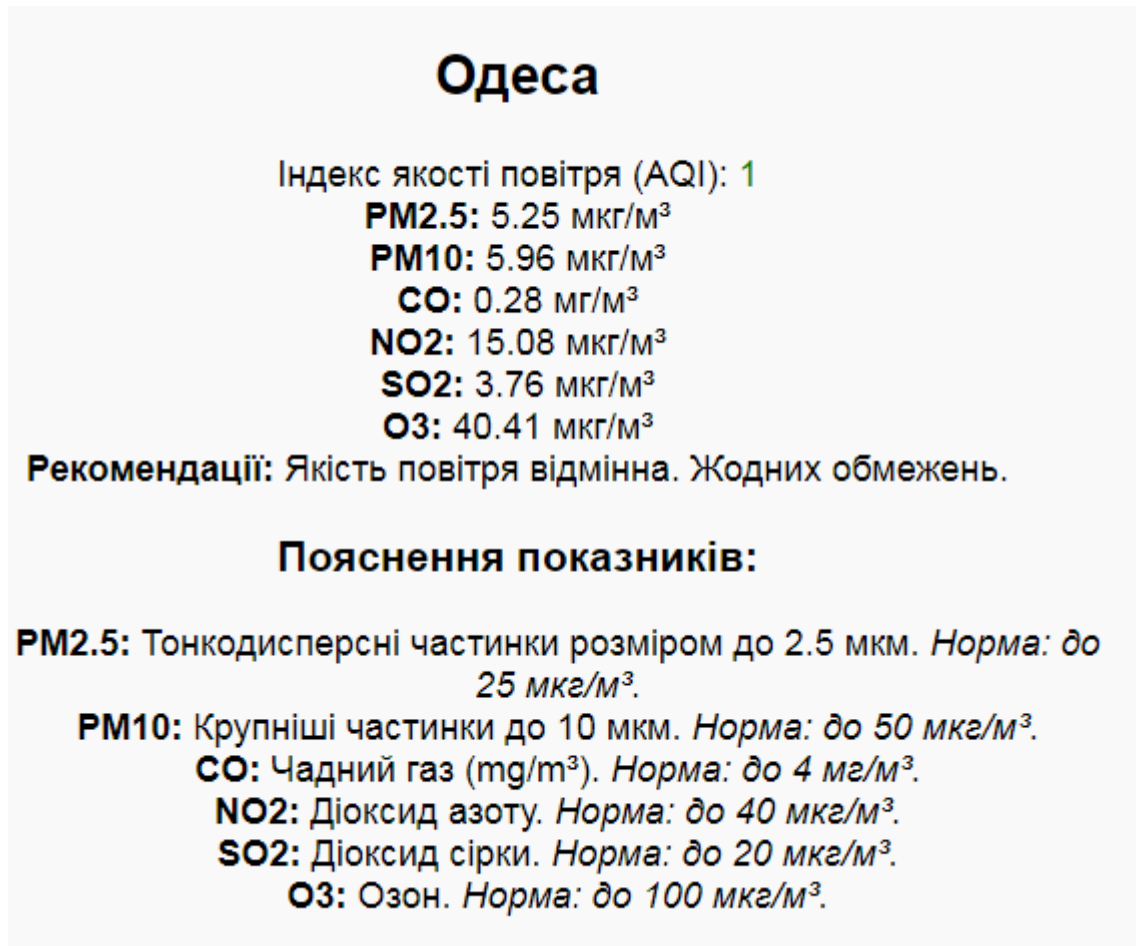


Рисунок 2.1 – Реалізовані показники концентрації забруднюючих речовин

Ці дані відображаються у секції інформації після того, як користувач вибирає місто, і дозволяють йому оцінити, які саме речовини є найбільш небезпечними в даний момент.

Рекомендації для населення (рис.2.2).

Залежно від рівня забруднення повітря, система автоматично генерує рекомендації для населення, що дозволяє зменшити негативний вплив

забрудненого повітря на здоров'я. Для кожного рівня індексу AQI передбачені конкретні поради:

AQI 1 (добрий): Поріг безпеки не перевищено, жодних обмежень немає.

AQI 2 (помірний): Повітря в основному безпечно, але людям з підвищеною чутливістю (наприклад, людям з астмою) варто обмежити перебування на вулиці.

AQI 3 (шкідливий для чутливих груп): Люди з респіраторними захворюваннями повинні обмежити час на відкритому повітрі.

AQI 4 (шкідливий): Варто обмежити фізичну активність на вулиці, особливо людям з хронічними захворюваннями.

AQI 5 (дуже шкідливий): Найгірші умови для здоров'я, рекомендується залишатися вдома, обмежити будь-яку активність на вулиці.

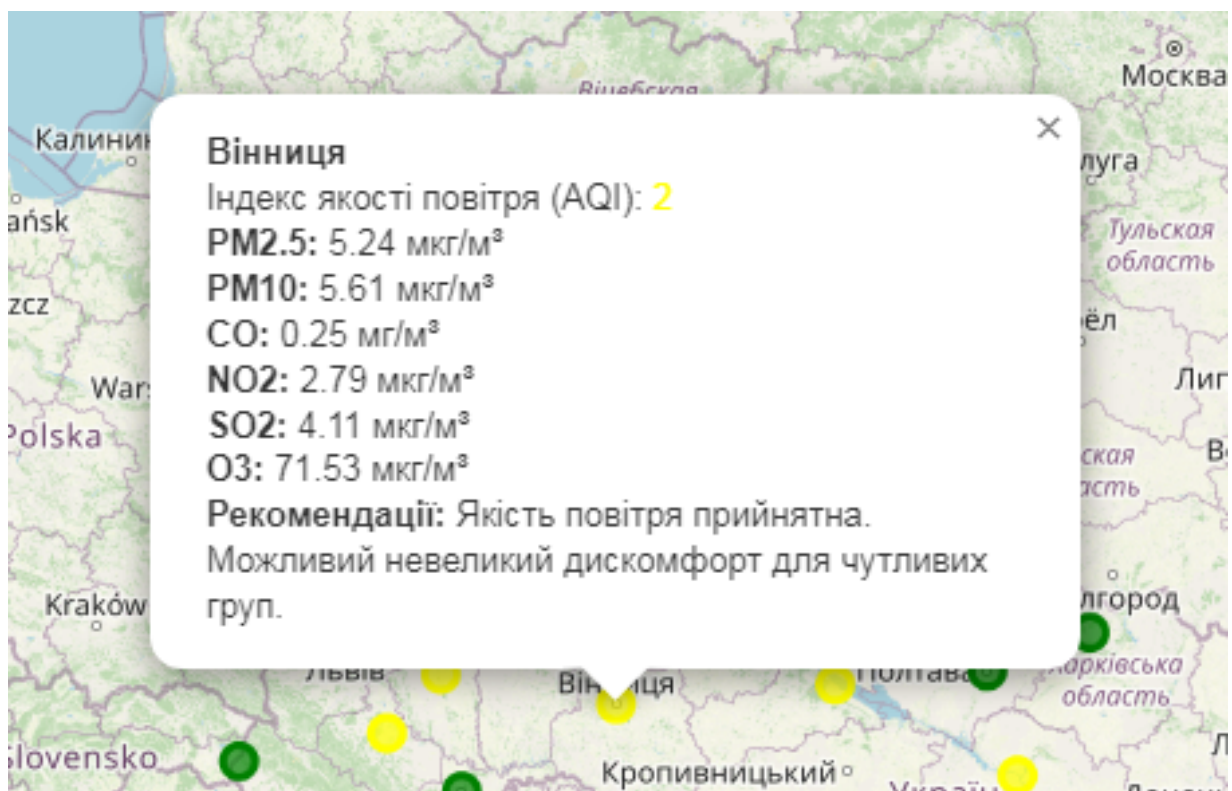


Рисунок 2.2 – Приклад реалізованих рекомендацій для населення

Ці рекомендації надаються автоматично на основі даних, отриманих через API, і відображаються у спеціальному блоці на сторінці. Користувач може отримати поради, що стосуються необхідності носити захисні маски, обмежити фізичні навантаження на свіжому повітрі або використовувати очищувачі повітря вдома.

2.2 Система рекомендацій

Система рекомендацій є важливим елементом веб-сервісу, який забезпечує користувачів корисною інформацією для зменшення негативного впливу забрудненого повітря на здоров'я. Вона працює на основі даних про індекс якості повітря (AQI), що дозволяє автоматично генерувати відповідні поради залежно від рівня забруднення повітря в конкретному місті.

Кожен рівень AQI вказує на певний ступінь небезпеки для здоров'я, що дозволяє персоналізувати рекомендації в залежності від ситуації:

AQI 1 (добрий): Якість повітря в межах норми. Рекомендації: Немає обмежень, можна проводити час на відкритому повітрі.

AQI 2 (помірний): Качество повітря прийнятне для більшості людей, але може бути трохи шкідливим для деяких чутливих груп (діти, літні люди, люди з хронічними захворюваннями). Рекомендації: Обмежити фізичну активність на вулиці, особливо для людей з астмою або іншими респіраторними захворюваннями.

AQI 3 (шкідливий для чутливих груп): люди з хронічними захворюваннями повинні обмежити перебування на відкритому повітрі. Рекомендації: Уникати тривалого перебування на вулиці, скоротити фізичні навантаження, особливо у години пік забруднення.

AQI 4 (шкідливий): забруднення досягає високих рівнів, що може вплинути на здоров'я більшості людей. Рекомендації: Обмежити перебування

на вулиці, використовувати захисні маски, звертатися до лікаря в разі погіршення самопочуття.

AQI 5 (дуже шкідливий): дуже високий рівень забруднення. Рекомендації: Рекомендується залишатися вдома з закритими вікнами, обмежити фізичні навантаження, користуватися спеціальними захисними масками, за необхідності звертатися за медичною допомогою.

Ці рекомендації сприяють зменшенню потенційної шкоди для здоров'я, особливо для вразливих груп населення. Вони автоматично оновлюються на основі отриманих даних про якість повітря та відображаються на сторінці, щоб користувач міг вчасно реагувати на зміни екологічної ситуації.

2.3 Динамічне масштабування

Динамічне масштабування карти – це одна з ключових особливостей системи, яка дозволяє карті автоматично підлаштовуватися під географічне розташування доступних даних, забезпечуючи користувачам максимальний комфорт у використанні ресурсу. Ця функція не лише покращує зручність навігації, але й дозволяє швидко отримувати локальну інформацію про якість повітря в обраному регіоні. Завдяки інтеграції з алгоритмами реального часу, карта може автоматично масштабуватися до зон, де доступні найбільш актуальні дані, та оновлювати їх без потреби додаткових дій з боку користувача.

Функціональність динамічного масштабування включає наступні особливості.

Автоматичне підлаштування карти: при завантаженні сторінки або додаванні нових міст система автоматично масштабує карту, щоб включити всі доступні точки моніторингу, забезпечуючи зручний перегляд усіх маркерів на карті. Це дозволяє користувачам бачити всю географію забруднення повітря без необхідності вручну змінювати масштаб карти.

Масштабування карти залежно від регіону: якщо користувач вибирає конкретне місто або область на карті, система автоматично змінює масштаб так, щоб це місто або область були видимі на екрані. Це особливо зручно, коли потрібно отримати детальну інформацію про конкретний регіон.

Підтримка зручності взаємодії: масштабування також дозволяє користувачам краще орієнтуватися в географічному розташуванні міст, а також зручно переглядати дані про якість повітря в різних точках країни або світу.

Завдяки динамічному масштабуванню карта залишається зручною та легкою у використанні, що забезпечує користувачам максимальний комфорт при перегляді даних про якість повітря, незалежно від того, в якому регіоні вони знаходяться. Крім того, система автоматично визначає масштаб для відображення найважливіших екологічних показників у вибраній локації. Це дозволяє миттєво отримувати детальну інформацію як на локальному рівні, так і в ширшому географічному контексті.

2.4 Діаграма варіантів використання

Для проектування програми було використано мову графічного опису об'єктного моделювання UML. За складеним списком вимог до проектованої системи було побудовано модель взаємодії зовнішнього актора «користувач» із програмною системою у вигляді діаграми варіантів використання. Розроблена діаграма варіантів використання наведена на рис.2.3.

Сценарії дозволяють чітко визначити функціональні межі системи та ключові точки її взаємодії з користувачами. Діаграма також служить основою для подальшого проектування компонентів програмної системи та їхньої інтеграції.



Рисунок 2.3 – Діаграма варіантів використання

2.5 Проектування компонентів прогнозування

Для прогнозування рівня забруднення повітря було вирішено використовувати підхід побудови ансамблю нейронних мереж, де передбачення двох регресійних нейронних мереж усереднюється третьою нейронною мережею.

Тимчасовий компонент. Тимчасовий компонент є повнозв'язковою нейронною мережею з одним прихованим шаром. На вхід нейронної мережі подається вектор, що складається з 25 значень. Перший прихований шар має 30 нейронів. Другий прихований шар має 70 нейронів. Вихідний шар має один нейрон - спрогнозоване значення забруднювача. Кількість шарів та нейронів у шарі було підібрано експериментальним шляхом. Структура нейронної мережі представлена на рис.2.4.

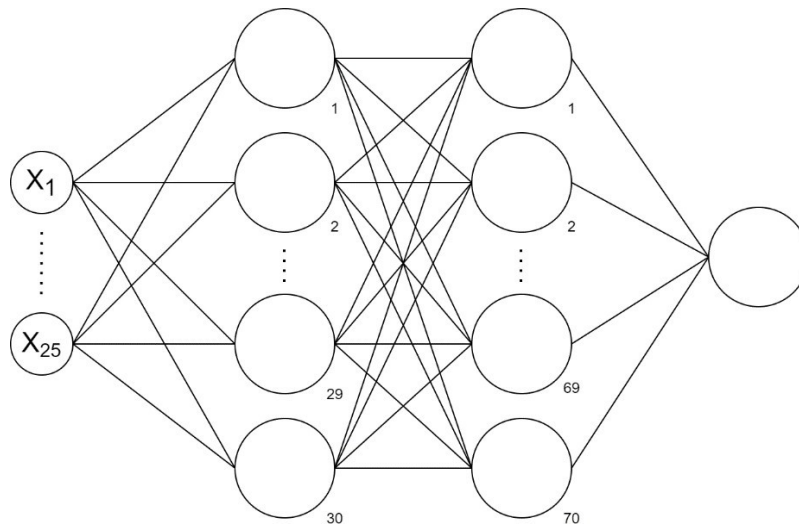


Рисунок 2.4 – Структура нейронної мережі тимчасової моделі

На вхід нейронної мережі подається вектор, що містить набір даних за поточну годину (t) та за наступну годину ($t+1$):

- 1) поточне забруднення (мг/м³);
- 2) день тижня (число від 0 до 6);
- 3) годину прогнозу (число від 0 до 24);
- 4) час доби (d чи n);
- 5) пора року (summer, autumn, winter, spring);
- 6) температура (градуси Цельсія);
- 7) прогноз температури (градуси Цельсія);
- 8) тип погоди (clear, partly-cloudy, cloudy та ін);
- 9) прогноз типу погоди (clear, partly-cloudy, cloudy та ін);
- 10) швидкість вітру (м/с);
- 11) прогноз швидкості вітру (м/с);
- 12) швидкість поривів вітру (м/с);
- 13) прогноз швидкості поривів вітру (м/с);
- 14) напрям вітру (nw, n, ne, e, se, s, sw, w, c);
- 15) прогноз напрямку вітру (nw, n, ne, e, se, s, sw, w, c);
- 16) тиск (мм рт. ст.);

- 17) прогноз тиску (мм рт. ст.);
- 18) вологість повітря (%);
- 19) прогноз вологості повітря (%);
- 20) тип опадів (число від 0 до 4);
- 21) прогноз типу опадів (число від 0 до 4);
- 22) сила опадів (число від 0.0 до 1.0);
- 23) прогноз сили опадів (число від 0.0 до 1.0);
- 24) хмарність (число від 0.0 до 1.0);
- 25) прогноз хмарності (число від 0,0 до 1,0).

Просторовий компонент. Просторовий компонент є повнозв'язковою нейронною мережею з одним прихованим шаром. Для кожної станції на вхід нейронної мережі подається різна кількість даних. Вхідний шар та прихований шар має m і k нейронів відповідно. Параметри m та k були підібрані окремо для кожної станції та забруднювача шляхом перебору можливих структур нейронної мережі та порівняння точності прогнозування. Вихідний шар має один нейрон – спрогнозоване значення забруднювача. Структура нейронної мережі представлена на рис.2.5.

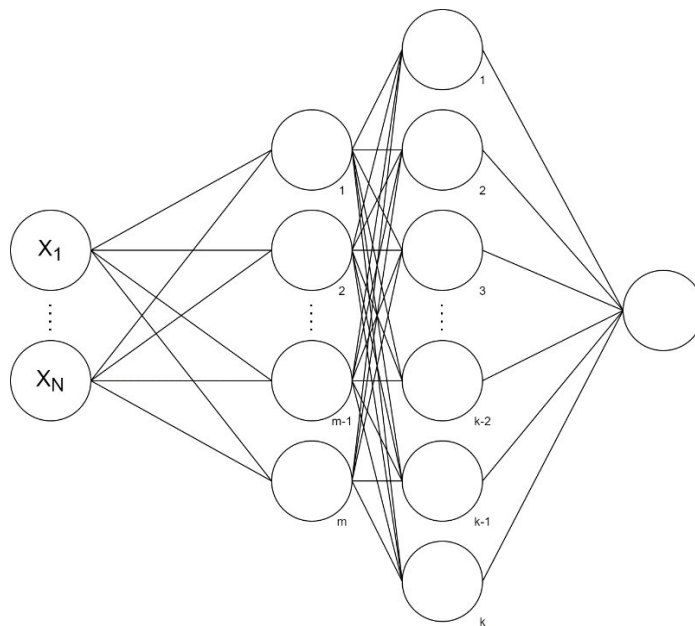


Рисунок 2.5 – Структура нейронної мережі просторової моделі

На вхід нейронної мережі подається вектор, що містить набір показів як самої станції, так і станцій у радіусі 10 кілометрів за поточну годину (t) та за наступну годину ($t+1$). Набір даних аналогічний тимчасовій моделі.

Агрегувальний компонент. Агрегувальний компонент необхідний для об'єднання прогнозу тимчасового та просторового компонента. Агрегувальний компонент має враховувати метеорологічні умови на сусідніх станціях, а також результат роботи тимчасового та просторового компонента. Для цієї мети була спроектована повнозв'язкова нейронна мережа з одним прихованим шаром. Залежно від станції на вхід подається різна кількість даних. Вхідний шар та прихований шар має m та k нейронів відповідно. Параметри m і k були підібрані окремо для кожної станції та забруднювача шляхом перебору можливих структур нейронної мережі та порівняння точності прогнозування. Вихідний шар має один нейрон, оскільки на виході ми отримуємо одне значення забруднювача, що враховує положення сусідніх станцій, метеорологічні умови на станціях, а також результат роботи часового та просторового компонента. Структура нейронної мережі, що агрегує моделі, представлена на рис.2.6.

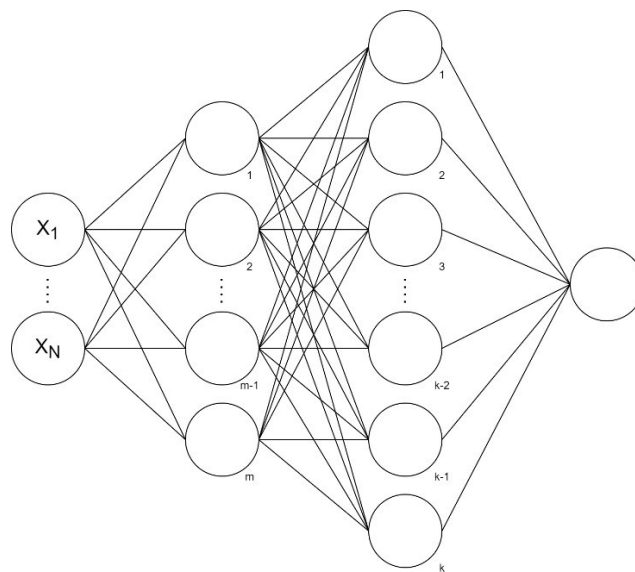


Рисунок 2.6 – Структура нейронної мережі моделі, що агрегує

2.6 Проєктування бази даних

Одним з основних компонентів системи, що розробляється, є база даних. Як базу даних було вирішено використовувати об'єктно-реляційну СУБД PostgreSQL. Для зберігання даних було виділено такі таблиці:

«Station» – таблиця, містить інформацію про станції, де проводяться виміри забруднень;

«Weather» – таблиця, що містить інформацію про погоду;

«WeatherForecast» – таблиця, що містить інформацію про прогноз погоди;

«Pollution» – таблиця, що містить інформацію про забруднення;

«PollutionType» – таблиця, що містить інформацію про типи забруднень;

«PollutionForecast» – таблиця, що містить інформацію про прогноз забруднення.

Схема бази даних, у якій відбито відносини, ключі та поля таблиць представлена на рис.2.7.

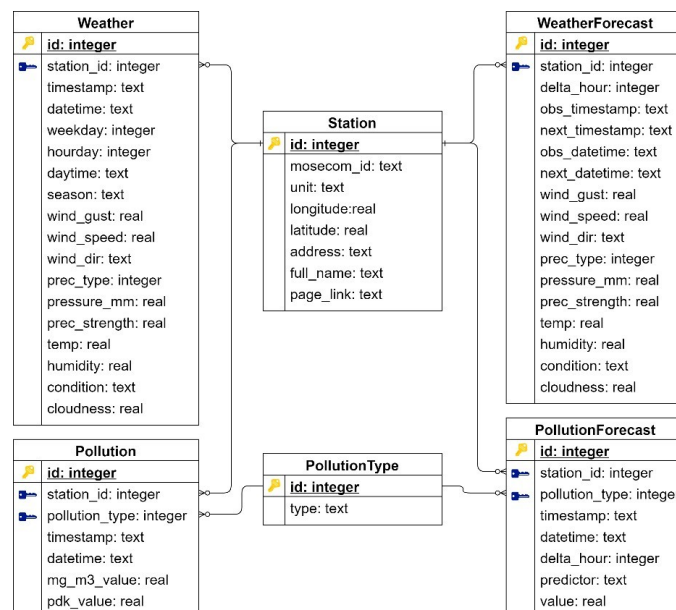


Рисунок 2.7 – Схема бази даних

2.7 Проєктування архітектури сервісу

Система є клієнт-серверною програмою, що складається з наступних частин: веб-сайт, REST API, база даних, модуль прогнозування та модуль збору даних. Кожна з перерахованих частин системи утворює окремий Docker контейнер, що дозволяє розміщувати компоненти незалежно від розташування інших. Модуль прогнозування додатково поділено на наступні компоненти – компонент тимчасового прогнозування, компонент просторового прогнозування, компонент агрегування та компонент донавчання. Для наочного уявлення архітектури системи було побудовано діаграму компонентів (рисунок 2.8).

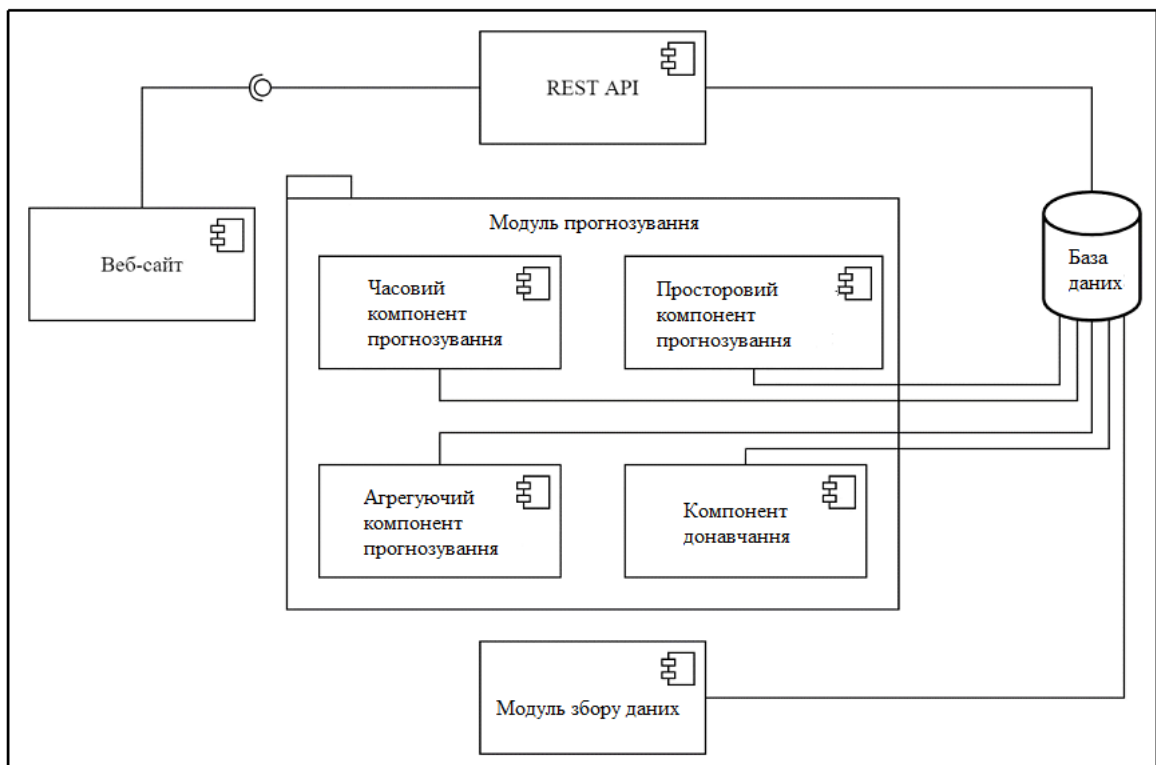


Рисунок 2.8 – Діаграма компонентів

2.8 Проєктування REST API

REST API є набором асинхронних методів REST, що дозволяють будь-яким типам клієнтів взаємодіяти з системою. Завдяки використанню REST API забезпечується незалежність клієнтських додатків від серверної частини, що дозволяє легко масштабувати систему та інтегрувати її з іншими сервісами.

Методи API організовані за принципами REST, що спрощує доступ до ресурсів системи за допомогою стандартних HTTP-запитів (GET, POST, PUT, DELETE).

Таблиця 2 – Список методів REST API модуля

Метод	URL	Опис
GET	/station	Отримання списку всіх станцій
GET	/station/{station_id}/weather	Отримання поточної погоди за станцією ID
GET	/station/{station_id}/metcast	Отримання прогнозу погоди по ID станції
GET	/station/{station_id}/pollution	Отримання поточного забруднення по ID станції та типу забруднювача
GET	/station/{station_id}/pollutions	Отримання всіх актуальних забруднень по ID станції
GET	/station/{station_id}/pollcast	Отримання прогнозу забруднення по ID станції та типу забруднювача
GET	/pollution/types	Отримання списку всіх типів забруднення
GET	/station/{station_id}/is_updated	Перевірка наявності оновлених даних
GET	/api/air-quality?city=Kyiv	Отримання даних про якість повітря для заданої локації
POST	/api/feedback	Надсилання відгуків користувача
PUT	/api/user/settings	Оновлення налаштувань користувача

3 РОЗРОБКА ВЕБ-СЕРВІСУ

3.1 Інструменти розробки

Для реалізації проєкту використовується набір сучасних інструментів, які забезпечують ефективну роботу веб-сервісу та його інтерактивність. Кожен з обраних інструментів виконує свою специфічну функцію, що дозволяє створити зручну, функціональну та масштабовану платформу для моніторингу якості повітря.

Python 3 – високорівнева мова програмування, вибрана для розробки веб-сервісу [6].

PostgreSQL - об'єктно-реляційна СУБД [7].

TensorFlow – open-source платформа для машинного навчання від Google [8].

Keras – бібліотека, що надає високорівневе API до роботи з нейронними мережами [9].

Pandas – бібліотека, що дозволяє обробляти та аналізувати велику кількість даних [10].

Bootstrap Flask – підтримка CSS фреймворку Bootstrap 4 [11].

Scikit-Learn – бібліотека, що містить безліч готових алгоритмів машинного навчання [12].

NumPy – бібліотека, до роботи з матрицями і багатовимірними масивами [13].

Pydantic – бібліотека, призначена для валідації даних [14].

HTML (HyperText Markup Language) є основою для побудови веб-сторінки. У проєкті HTML використовується для створення структури сервісу, зокрема заголовків, інтерактивних елементів, інформаційних блоків та області для відображення інтерактивної карти.

CSS (Cascading Style Sheets) відповідає за візуальне оформлення веб-сторінки. У проєкті CSS використовується для покращення користувацького досвіду за рахунок створення сучасного, привабливого та функціонального дизайну.

Наприклад, карта обрамлена тонким контуром і має закруглені кути для естетичного вигляду, а інформаційні блоки виділяються тінями, що робить їх більш помітними. Крім того, стилізація дозволяє майбутньому реалізувати адаптивний дизайн, який підлаштовується під різні розміри екранів. Кольорове кодування даних (зелене для чистого повітря, червоне для забрудненого) робить інформацію легко зрозумілою навіть для нових користувачів.

JavaScript є основним інструментом для забезпечення інтерактивності та динамічного оновлення веб-сторінки. У рамках цього проєкту JavaScript використовується для взаємодії з API, обробки отриманих даних та їхнього відображення на інтерактивній карті.

Наприклад, при завантаженні сторінки скрипт автоматично надсилає запити до OpenWeather API для отримання актуальної інформації про якість повітря. Після цього JavaScript генерує маркери на карті, які змінюють колір залежно від рівня забруднення. Крім того, клік на маркер викликає оновлення інформаційного блоку з детальними даними про обране місто, такими як показники PM2.5, PM10, NO2, CO, SO2 та озону.

Leaflet.js: бібліотека для роботи з інтерактивною картою.

Leaflet.js – це популярна JavaScript-бібліотека, яка забезпечує зручну роботу з інтерактивними картами. Вона дозволяє легко додавати географічні об'єкти, маркери та спливаючі вікна на карту.

У цьому проєкті Leaflet.js використовується для візуалізації екологічних даних у географічному контексті. Карта автоматично масштабується до області, що охоплює всі обрані міста, а також підтримує функції масштабування й перетягування, що робить її зручною для

користувачів. Завдяки цій бібліотеці реалізовано відображення кольорових маркерів, які миттєво інформують про якість повітря у кожному регіоні.

OpenWeather API є основним джерелом інформації про якість повітря. Цей сервіс надає доступ до даних про рівень забруднюючих речовин у різних регіонах, включаючи концентрації PM2.5, PM10, NO2, CO, SO2, O3, а також загальний індекс якості повітря (AQI). Запити до API виконуються за допомогою JavaScript, після чого отримані дані обробляються та відображаються на карті й у текстовому форматі. Використання OpenWeather API забезпечує високу точність і актуальність інформації, що є критично важливим для сервісу, орієнтованого на роботу в реальному часі.

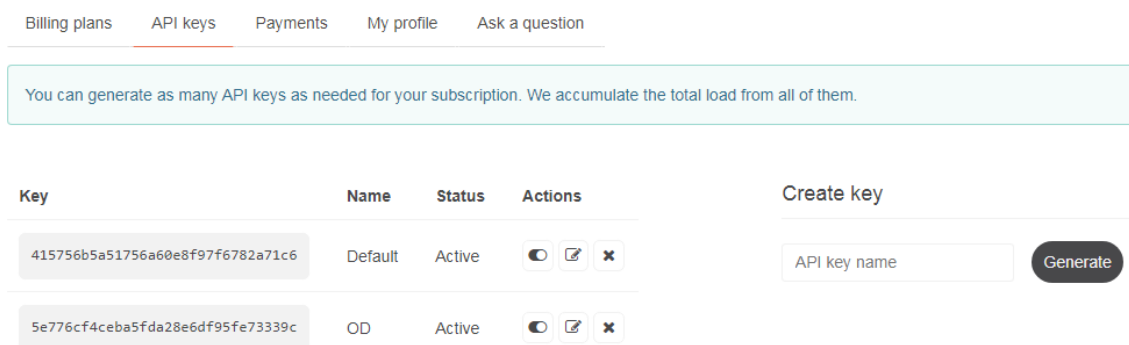


Рисунок 3.1 – Мої API ключі у openweather

3.2 Основні етапи розробки

Розробка веб-сервісу для моніторингу якості повітря включає кілька етапів, кожен з яких має свою специфіку та важливість для загальної реалізації системи. Перший етап — це підготовка, під час якої визначаються основні інструменти, забезпечується доступ до даних через API та створюється основна структура сторінки.

1. Підготовка.

На етапі підготовки закладаються основи для успішної реалізації проєкту. Це критично важливий етап, оскільки правильний вибір інструментів та налаштування інфраструктури безпосередньо впливають на ефективність і швидкість розробки.

Отримання API-ключів.

Для інтеграції з OpenWeather API необхідно отримати API-ключ, що дозволяє безкоштовно або платно отримувати доступ до даних про якість повітря.

На цьому етапі розробник:

1. Реєструється на платформі OpenWeather.
2. Оформлює запит на отримання API-ключа.
3. Зберігає отриманий ключ у конфігураційних файлах або середовищі розробки для використання в коді.
4. Цей ключ використовується для автентифікації запитів до API і доступу до даних про концентрації забруднювачів, індекс якості повітря та інших екологічних показників для заданих координат.

Створення базового каркасу сторінки.

Створення базового каркасу веб-сторінки є першим кроком для формування інтерфейсу користувача. Це включає:

Титульна сторінка: створення основної сторінки з заголовком, карткою та основними елементами для відображення даних.

Структура HTML: розробка контейнерів для карти, інформаційних блоків та елементів взаємодії, таких як кнопки чи панелі для вибору міста.

Початкове налаштування CSS: стилізація базових елементів сторінки, таких як шрифт, колірна схема, розміри елементів та відступи, щоб забезпечити зручність користування.

Створення базових JavaScript-функцій: написання базового коду для інтеграції з API, додавання маркерів на карту та визначення базових функцій взаємодії.

Цей етап дозволяє побудувати початкову версію сторінки, яка містить основні компоненти і готова до подальшої інтеграції з іншими частинами системи, такими як динамічне оновлення даних і покращення інтерфейсу.

Підсумок етапу підготовки:

Цей етап є основою для всього проєкту. Вибір інструментів, налаштування доступу до даних та створення початкового каркасу сторінки забезпечує фундамент, на якому будується весь функціонал веб-сервісу. Успішне завершення цього етапу дозволяє перейти до наступних етапів розробки, таких як інтеграція даних, покращення інтерфейсу та тестування системи.

2. Інтеграція API.

Інтеграція OpenWeather API є важливим етапом у розробці веб-сервісу для моніторингу якості повітря, оскільки саме через це API система отримує актуальні дані про забруднення повітря в реальному часі.

Цей процес включає дві основні складові: формування запитів до API та обробка отриманих даних для подальшого відображення користувачу.

Запити до OpenWeather API.

Перший крок у процесі інтеграції – це надсилання запитів до OpenWeather API для отримання актуальних даних про якість повітря. Запити формуються на основі географічних координат (широти та довготи), що дозволяє визначити, для яких міст або регіонів необхідно отримати інформацію.

Основні етапи для надсилання запиту:

Визначення географічних координат.

Для кожного міста на карті система використовує координати (широту та довготу), що дозволяють точно визначити місце, для якого буде запитана інформація.

Наприклад, для Одеси це координати: широта 46.4825, довгота 30.7233.

Формування запиту до API.

Для отримання даних про якість повітря, запит формується за допомогою URL-адреси OpenWeather API, що включає координати міста та API-ключ для доступу.

Приклад формування запиту (лістинг 3.1):

```
const url =
`https://api.openweathermap.org/data/2.5/air_pollution?lat=${city.lat}&lon=${city.lon}&appid=${API_KEY}`;
```

У цьому запиті:
lat=\${city.lat} – це широта міста.
lon=\${city.lon} – це довгота міста.
appid=\${API_KEY} – це API-ключ, що дає доступ до OpenWeather API.

Лістинг 3.1 – Формування запиту до API

Надсилання запиту.

Запит надсилається через HTTP-запит за допомогою JavaScript-функцій, таких як fetch або XMLHttpRequest.

Наприклад (лістинг 3.2):

```
const response = await fetch(url);
const data = await response.json();
```

Лістинг 3.2 – Надсилання запиту

Обробка отриманих даних.

Після отримання відповіді від API, дані обробляються для подальшого використання в інтерфейсі користувача. Отримані дані містять різні параметри, включаючи рівень забруднення повітря та концентрацію основних шкідливих речовин. Задача розробника полягає в тому, щоб витягнути необхідні показники і правильно їх відобразити.

Аналіз відповіді API: API надає дані у форматі JSON, де можна знайти кілька ключових параметрів:

`main.aqi` — індекс якості повітря (AQI).

`components` — об'єкт, що містить концентрацію основних забруднювачів (PM2.5, PM10, CO, NO2, SO2, O3).

Приклад обробки даних (лістинг 3.3):

```
const aqi = data.list[0].main.aqi; // Індекс якості повітря
const components = data.list[0].components; // Забруднювачі
(PM2.5, PM10 і т.д.)
```

Лістинг 3.3 – Обробка даних

Обробка кожного забруднювача.

Дані по кожному забруднювачу містять числові значення концентрацій, які необхідно обробити та відобразити в інформаційній секції.

Наприклад (лістинг 3.4):

```
const pm25 = components.pm2_5 || 'недоступно'; //
Концентрація PM2.5
const pm10 = components.pm10 || 'недоступно'; //
Концентрація PM10
const co = components.co || 'недоступно'; // Концентрація
CO
const no2 = components.no2 || 'недоступно'; // Концентрація
NO2
const so2 = components.so2 || 'недоступно'; // Концентрація
SO2
const o3 = components.o3 || 'недоступно'; // Концентрація
O3
```

Лістинг 3.4 – Обробка кожного забруднювача

Виведення результатів: після отримання та обробки даних, система оновлює інформаційний блок на веб-сторінці, щоб користувач міг побачити актуальні показники якості повітря.

Це включає:

Виведення індексу AQI.

Відображення концентрацій забруднювачів.

Зміна кольору маркерів на карті залежно від значення AQI.

Приклад оновлення секції інформації (лістинг 3.5):

```
function updateInfo(cityName, airQuality) {
  const infoSection = document.getElementById('info');
  infoSection.innerHTML = `
  <h2>${cityName}</h2>
  <p>Індекс AQI: <span
  style="color:${getAQIColor(airQuality.aqi)}">${airQuality.aqi}</
  span></p>
  <p><b>PM2.5:</b> ${airQuality.pm25} мкг/м³</p>
  <p><b>PM10:</b> ${airQuality.pm10} мкг/м³</p>
  <p><b>CO:</b> ${airQuality.co} мг/м³</p>
  <p><b>NO2:</b> ${airQuality.no2} мкг/м³</p>
  <p><b>SO2:</b> ${airQuality.so2} мкг/м³</p>
  <p><b>O3:</b> ${airQuality.o3} мкг/м³</p>`;}
```

Лістинг 3.5 – Оновлення секції інформації

Підсумок інтеграції API.

Інтеграція з OpenWeather API дозволяє системі отримувати точні дані про якість повітря в реальному часі, автоматично обробляти ці дані та надавати користувачам актуальну інформацію. Завдяки цьому забезпечується швидке та точне оновлення картографічного інтерфейсу і інформаційних блоків на веб-сторінці. REST API реалізовано за допомогою асинхронного

веб-фреймворку FastAPI та бібліотеки Pydantic, що використовується для роботи з моделями даних. Для зв'язку модуля з базою даних використовується високорівнева бібліотека ORM PonyORM. Нижче наведено приклад реалізації REST методу отримання даних станції по її ідентифікатори (лістинг 3.6).

```
@app.get('/station/{station_id}') def
get_station(station_id: int):
    with db_session:
        station = get_station_by_id(station_id) if station:
            return station.to_dict()
        raise HTTPException(status_code=404,
            detail=f'Station with id \'{station_id}\'' not exists')
```

Лістинг 3.6 – Отримання даних станції за ідентифікатором

3. Візуалізація.

Розробка інтерактивної карти за допомогою Leaflet.js.

Одним із основних компонентів веб-сервісу є інтерактивна карта, яка служить основним інтерфейсом для відображення даних про якість повітря в реальному часі.

Для реалізації цієї функціональності використовувалася бібліотека Leaflet.js, яка є популярним інструментом для роботи з картами на веб-сторінках. Leaflet.js дозволяє легко створювати інтерактивні карти з можливістю додавання маркерів, географічних шарів, спливаючих вікон та інших елементів, що сприяють зручному перегляду та взаємодії з картографічними даними.

Кроки реалізації карти.

Ініціалізація карти.

Вибір центру карти та рівня масштабу для відображення необхідних територій. Для цього використовується метод `L.map()`.

Визначення початкових координат для картографічного центру (наприклад, для України або конкретного міста) та рівня масштабу, щоб зручно охопити всю територію.

```
const map = L.map('map').setView([48.3794, 31.1656], 6); //  
Центрування на Україні з масштабом 6
```

Додавання тайлера карти.

Для відображення картографічних даних використовуються «тайли» (мозаїки карт), які можна отримати через відкриті картографічні сервіси, такі як OpenStreetMap.

```
L.tileLayer('https://{s}.tile.openstreetmap.org/{z}/{x}/{y}.  
png', {  
  maxZoom: 19,  
  attribution: '© OpenStreetMap'  
}).addTo(map);
```

Інтерактивність карти.

Користувач може змінювати масштаб карти, перетягувати її для перегляду різних регіонів, що дозволяє зручно переглядати дані про якість повітря в різних містах.

Інтерактивна карта дозволяє користувачу клікати на маркери міст, щоб отримувати додаткову інформацію про якість повітря.

Створення маркерів з кольоровим кодуванням.

Однією з важливих функцій карти є кольорове кодування маркерів для відображення рівня якості повітря. Це дозволяє користувачам швидко зрозуміти стан атмосфери в кожному місті без необхідності детально переглядати числові значення. Для цього використовуються різні кольори маркерів, які відповідають різним рівням індексу AQI (Індексу якості повітря):

Зелений: показує, що якість повітря відмінна (AQI 1).

Жовтий: помірний рівень забруднення (AQI 2).

Помаранчевий: помірно небезпечне повітря для чутливих груп (AQI 3).

Червоний: високий рівень забруднення (AQI 4).

Фіолетовий: дуже небезпечне повітря (AQI 5).

Кроки реалізації маркерів:

Створення маркерів: для кожного міста створюється маркер на карті з певним кольором, який залежить від індексу AQI (лістинг 3.7).

```
const marker = L.circle([city.lat, city.lon], {
  color: getAQIColor(airQuality.aqi), // Визначення кольору
  маркера
  fillColor: getAQIColor(airQuality.aqi),
  fillOpacity: 0.7,
  radius: 20000
}).addTo(map);
```

Лістинг 3.7 – Створення маркерів

Визначення кольору маркера: використовується функція `getAQIColor()`, яка змінює колір маркера в залежності від значення AQI. Вона дозволяє створювати кольорове кодування (лістинг 3.8):

```
function getAQIColor(aqi) {
  if (aqi === 1) return 'green';
  if (aqi === 2) return 'yellow';
  if (aqi === 3) return 'orange';
  if (aqi === 4) return 'red';
  if (aqi === 5) return 'purple';
  return 'gray'; // Якщо дані відсутні}
}
```

Лістинг 3.8 – Визначення кольору маркера

3.3 Реалізація та навчання моделей прогнозування

Підготовка даних для навчання моделей прогнозування.

Дані для навчання збиралися з відкритих джерел OpenWeatherMap, надає дані про поточну погоду, прогноз, історичні дані, а також параметри забруднення повітря (PM2.5, PM10, CO тощо). Для збору даних було розроблено спеціальний парсер, який підключався до джерел даних та витягував необхідну інформацію. Розроблений парсер розміщував отримані дані в документоорієнтованій базі даних MongoDB, після чого всі зібрані дані були експортовані до файлів CSV. Кожен запис даних містить інформацію про концентрацію забруднюючих речовин у повітрі на станції, а також поточні метеорологічні дані та метеорологічний прогноз. Для поділу даних використовувалися Python бібліотеки NumPy та Pandas. Перед навчанням дані були відмасштабовані за допомогою утиліти MinMaxScaler бібліотеки Scikit-Learn.

Конструювання моделей нейронних мереж.

Для навчання моделей прогнозування було розроблено спеціальний клас, який на основі вхідних параметрів конструював нейронну мережу за допомогою фреймворку Keras (лістинг 3.9).

```
class NeuralModel():
    def init (self, layer_sizes: list, in_shape: int): l =
list()
    l.append(layers.Dense(layer_sizes[0],
activation='relu', input_shape=[in_shape]))
    for i in range(1, len(layer_sizes)):
l.append(layers.Dense(layer_sizes[i], activation='relu'))
    l.append(layers.Dense(1))
    self.model = keras.Sequential(l)
    self.optimizer = tf.keras.optimizers.RMSprop(0.0001)
self.model.compile(loss='mse',
optimizer=self.optimizer,metrics=['mae', 'mse'])
    def get_model(self): return self.model
```

Лістинг 3.9 – Клас конструювання нейронної мережі

Ініціалізуючими параметрами класу є список з кількості нейронів на i -му шарі і розмір вхідного вектора. Як оптимізатор використовується RMSprop, а в якості функції активації на кожному шарі використовується relu. різні пов'язані нейронні мережі.

Навчання моделей нейронних мереж.

Для навчання моделей нейронної мережі була розроблена функція `train_nn`, яка приймає як параметри модель нейронної мережі, дані для навчання та ім'я моделі для збереження ваги навченої моделі на диск (лістинг 3.10).

```
def train_nn(model, X, Y, modelname: str):
    x_train, x_test, y_train, y_test = train_test_split (X, Y,
test_size = 0.25)

    early_stopping = EarlyStopping(min_delta=0.0001,
patience=5, monitor='mse')
    history = model.fit(x_train,
y_train, epochs=500, batch_size=32, verbose=0,
callbacks=[early_stopping])

    loss, mae, mse = model.evaluate(x_test, y_test, verbose=0)
model.save(f"models/{modelname}.h5")
```

Лістинг 3.10 – Функція для навчання моделей прогнозування

Як видно з лістингу функція `train_nn` розбиває вхідні дані на навчальну та тестову вибірку з коефіцієнтом 0.25. При навчанні використовується колббек ранньої зупинки, який зупинить тренування нейронної мережі, коли нейронна мережа вийде на плато навчання і не покращуватиме результат прогнозування протягом 5 спроб з різницею в 0,0001.

Реалізація системи автоматичного до навчання.

Система автоматичного донавчання є спеціальним класом, що входить до складу модуля прогнозування розроблюваного веб-сервісу. Цей клас спеціально розроблений для запуску 1 раз на тиждень з метою донавчити моделі на накопичених даних за цей період. Для демонстрації роботи донавчання було змодельовано донавчання на існуючій навченій моделі на фіктивних даних, схожих на реальні. На рис.3.2 показано графік системи донавчання, де подано прогноз нейронної мережі до та після донавчання, а також реальне значення забруднювача.

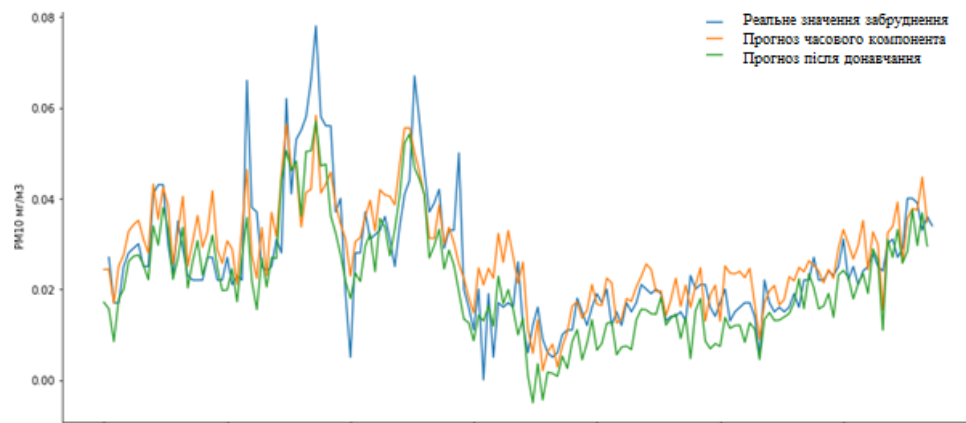


Рисунок 3.2 – Графік системи донавчання

Числові значення максимальної та середньої помилки прогнозування нейронної мережі до та після донавчання представлені в таблиці 3.1.

Таблиця 3.1 – Числові значення максимальної та середньої помилки

Етап	Максимальна помилка	Середня помилка
До донавчання	0.036	0.006
Після донавчання	0.036	0.005

Виходячи з числових значень помилок можна сказати, що донавчання має позитивний вплив на точність прогнозування, середня помилка прогнозування зменшилась, а максимальна залишилася на колишньому рівні. Передбачається, що за більш тривалий термін донавчання максимальна та середня помилка зменшуватимуться.

4 ТЕСТУВАННЯ

4.1 Тестування веб-сервісу

Перевірка роботи веб-сайту полягала у проведенні функціонального тестування. Нижче наведено набір проведених функціональних тестів.

Тест №1. Отримання списку станцій.

Користувач заходить на головну сторінку.

Очікуваний результат: на головній сторінці виводяться картки кожної станції.

Отриманий результат: збігається з очікуваним.

Тест №2. Отримання всіх позначок станцій на карті.

Вхідні дані: користувач заходить на сторінку картки.

Очікуваний результат: на сторінці з картою відображається інтерактивна карта з мітками всіх станцій, пофарбованих у кольори AQI.

Отриманий результат: збігається з очікуваним.

Тест №3. Перехід на сторінку детальною інформацією з картки.

Користувач натискає на будь-яку з міток на карті.

Очікуваний результат: відбувається перехід на сторінку детальної інформації вибраної станції.

Отриманий результат: збігається з очікуваним.

Тест №4. Отримання детальної інформації щодо станції.

Вхідні дані: користувач відкриває сторінку з детальною інформацією станції будь-яким доступним способом.

Очікуваний результат: на сторінці з детальною інформацією по станції відображається банер з інформацією щодо поточного забруднення та таблиці з інформацією про прогнозоване забруднення.

Отриманий результат: збігається з очікуваним.

Тест №5. Автоматичне оновлення даних.

Вхідні дані: користувач відкриває сторінку з детальною інформацією будь-яким доступним способом і протягом однієї години тримає її відкритою.

Очікуваний результат: при надходженні нових даних про забруднення та прогнозі сторінку з детальною інформацією буде автоматично оновлено.

Отриманий результат: збігається з очікуваним.

Тестування REST API.

Перевірка роботи REST API полягала у проведенні функціонального тестування.

Тест №1. Отримання списку станцій.

Вхідні дані: клієнт надсилає GET запит на адресу /station. Очікуваний результат: отримання списку, що складається з JSON об'єктів станцій.

Отриманий результат: збігається з очікуваним.

Тест №2. Отримання станції за ідентифікатором.

Вхідні дані: клієнт надсилає GET запит за адресою /station/{station_id}, де station_id – ідентифікатор станції.

Очікуваний результат: отримання JSON об'єкта станції. Отриманий результат: збігається з очікуваним.

Тест №3. Отримання погоди на станції за ідентифікатором станції. Вхідні дані: клієнт надсилає GET запит за адресою /station/{station_id}/weather, де station_id – ідентифікатор станції.

Очікуваний результат: отримання JSON погодного об'єкта по станції. Отриманий результат: збігається з очікуваним.

Тест №4. Отримання поточного забруднення станції.

Вхідні дані: клієнт надсилає GET запит за адресою /station/{station_id}/pollutions?pollution_type={type}, де station_id – ідентифікатор станції, type – тип забруднення.

Очікуваний результат: отримання JSON об'єкта забруднення по станції.

Отриманий результат: збігається з очікуванням.

Тест №5 Отримання поточних забруднень станції.

Вхідні дані: клієнт надсилає GET запит за адресою /station/{station_id}/pollutions, де station_id – ідентифікатор станції.

Очікуваний результат: отримання списку забруднень станцією.

Отриманий результат: збігається з очікуванням.

Тестування інтерфейсу.

Для тестування використовувалися браузер Chrome, Edge а також пристрої з різними операційними системами. У ході перевірки були ретельно протестовані всі основні функції, включаючи інтерактивність карти, динамічне оновлення даних про якість повітря та коректне відображення маркерів. Усі виявлені недоліки документувалися для подальшого виправлення, що забезпечило всебічне опрацювання та підвищення надійності системи.

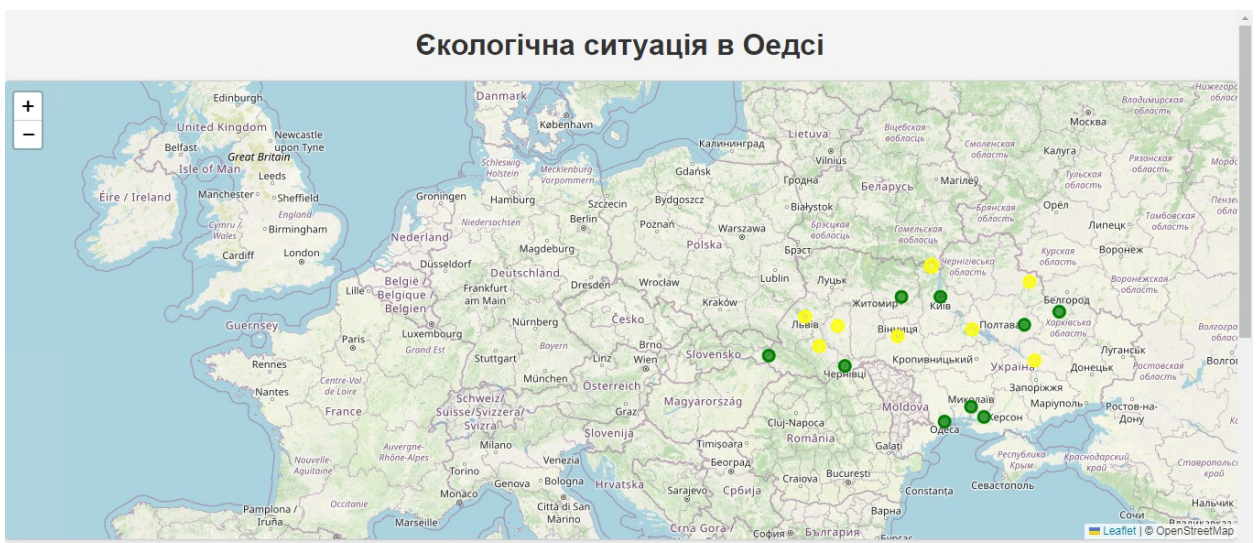


Рисунок 4.1 – Інтерфейс мапи

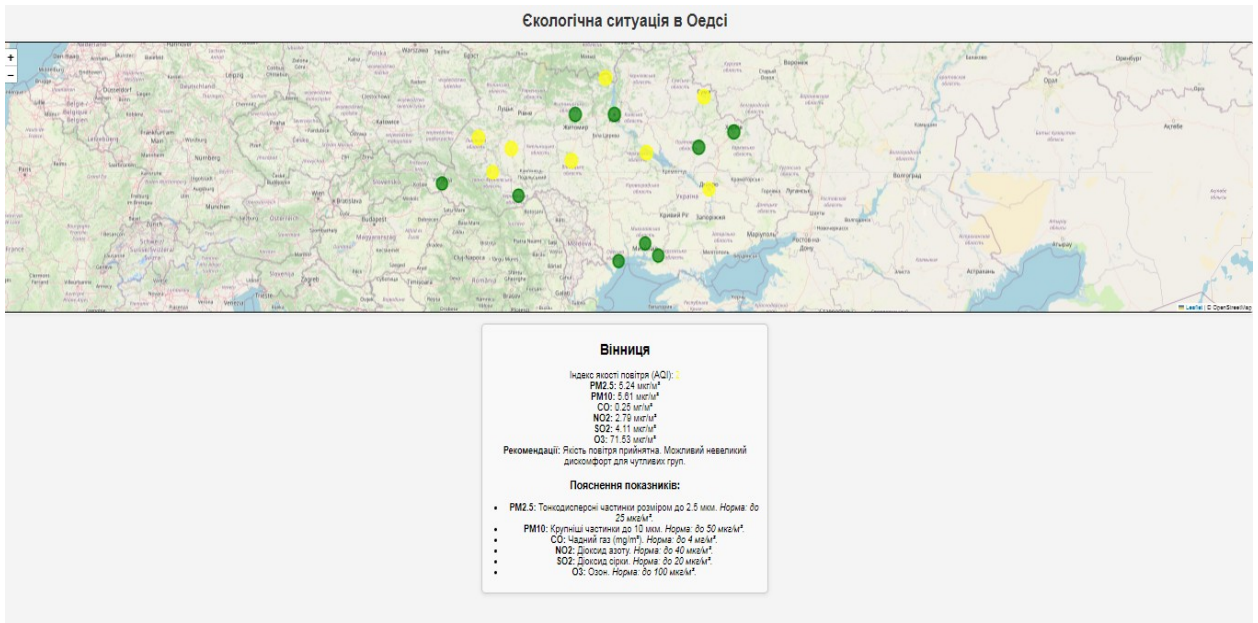


Рисунок 4.2 – Повний інтерфейс (спеціально деформована для наглядного прикладу)

Николаев

Індекс якості повітря: 1 (Отлично)

PM2.5: 8.53 $\mu\text{g}/\text{m}^3$

PM10: 9.81 $\mu\text{g}/\text{m}^3$

CO: 257.02 $\mu\text{g}/\text{m}^3$

NO2: 11.31 $\mu\text{g}/\text{m}^3$

O3: 42.2 $\mu\text{g}/\text{m}^3$

Рисунок 4.3 – Баг, CO через відсутню конвертацію даних у потрібний Європецький формат з Американського, більше ніж в 250 раз більше норми, якщо це і справді було б так ми б всі давно померли від цього. Виявлені помилки та їх вирішення.

У ході тестування були виявлені кілька ключових проблем. Зокрема, було зафіксовано повільну роботу при завантаженні даних через API, що вирішувалося впровадженням механізму кешування. Деякі елементи інтерфейсу неправильно відображалися на пристроях із малими екранами, що виправлялося за допомогою використання адаптивних стилів CSS. Виявлено,

що при некоректних запитах до API відображалися порожні блоки, тому було додано обробку помилок з відповідними повідомленнями для користувачів. У ранніх версіях було відображення на карті лісових пожеж, через деякі помилки почали горіти північний і південний полюси і океанічні льоди. Ці заходи дозволили усунути критичні недоліки та значно покращити загальний рівень роботи сервісу.

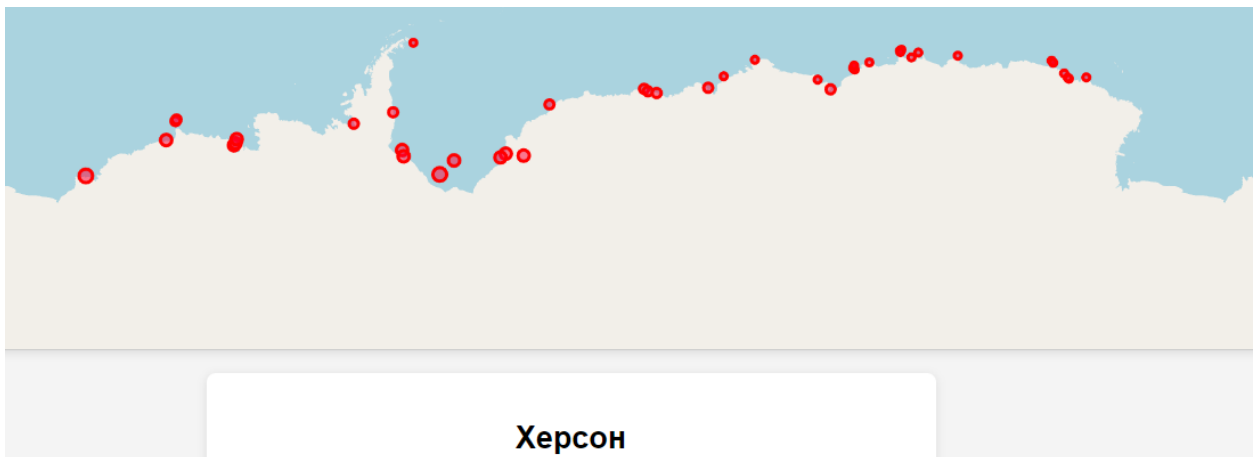


Рисунок 4.4 – Баг: палаючий Південний полюс та океанічні айсберги

Після підбору параметрів та навчання моделей нейронних мереж було проведено тестування якості прогнозування нейронних мереж. Тестування полягало в ініціації прогнозування нейронних мереж на реальних даних за 1 тиждень вимірів, які не брали участь у навчанні. Оскільки моделей нейронних мереж навчено багато, то для оцінки якості взято по 2 моделі однакової станції та однакового забруднення з тимчасових, просторових та агрегуючих моделей. Графік тестового прогнозу моделей прогнозування групи тимчасового компонента представлений рис.4.5.

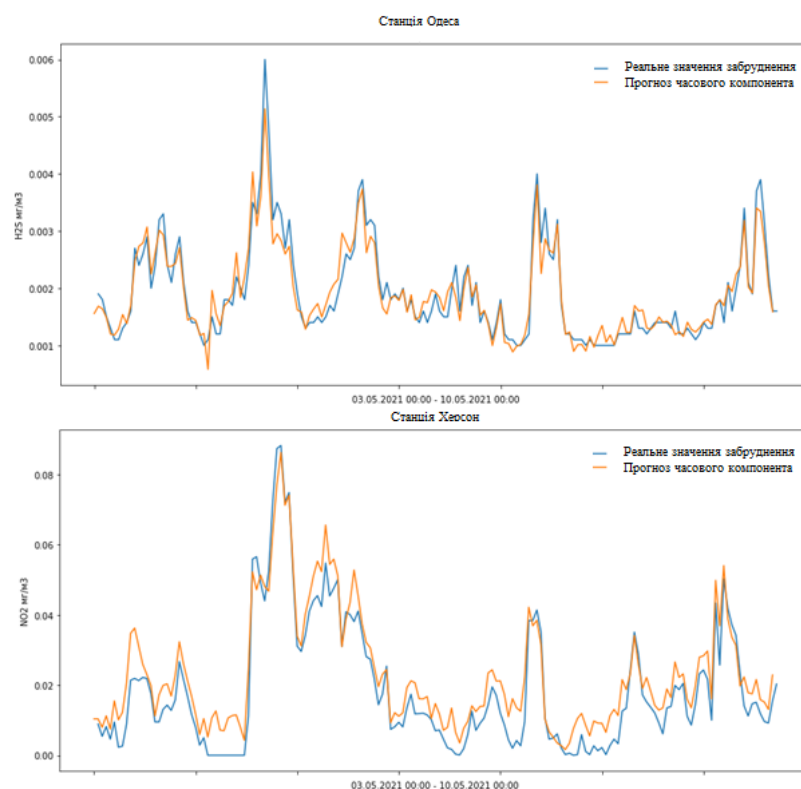


Рисунок 4.5 – Тестовий прогноз тимчасового компоненту

Числові показники максимальної та середньої помилки, а також точності за даними станціями та забруднювачами наведені в таблиці 4.

Таблиця 4 – Помилки та точність тимчасових моделей

Забруднювач	Станція	Максимальна помилка	Середня помилка	Точність, %
H2S	Одеса	0.0023	0.0003	82%
NO2	Херсон	0.0316	0.0073	71%

Опитування людей.

Окрім технічного тестування, я провів опитування серед своїх друзів, щоб отримати зворотний зв'язок про проєкт. Більшість із опитаних зізналися, що практично не цікавляться якістю повітря, а також екологічною ситуацією

у своєму місті, країні чи світі загалом. Для багатьох із них інформація про стан атмосфери не є частиною щоденних звичок або необхідною для ухвалення рішень.

Цей результат підкреслює важливість проєкту, оскільки він не лише забезпечує доступ до актуальних даних, але й має на меті підвищити обізнаність суспільства про важливість контролю за якістю повітря. Реалізовані функції, такі як інтуїтивний інтерфейс, інтерактивна карта та рекомендації щодо дій у разі підвищеного забруднення, спрямовані на те, щоб зробити екологічну інформацію зрозумілою і доступною навіть для тих, хто раніше не замислювався над цими питаннями.

Результати тестування показали, що система відповідає усім заявленим вимогам. Інтерактивна карта працює стабільно, забезпечуючи коректне відображення маркерів і динамічне оновлення даних. Система демонструє високу продуктивність навіть за умов значного навантаження, підтримуючи стабільність при одночасному використанні багатьма користувачами. Інтерфейс було оцінено як інтуїтивно зрозумілий, а адаптивний дизайн забезпечує зручність використання на різних пристроях. Таким чином, проведені тестування дозволили вивести веб-сервіс на рівень, готовий до експлуатації.

ВИСНОВКИ

В рамках даної роботи було розроблено веб-сервіс для прогнозування якості повітря за допомогою нейромережових технологій. При цьому було вирішено такі завдання.

1. Розроблено архітектуру сервісу для прогнозування якості повітря.
2. Розроблено моделі прогнозування якості повітря.
3. Розроблено систему автоматичного донавчання моделей прогнозування якості повітря.
4. Розроблено веб-сервіс, що надає інформацію щодо якості повітря.
5. Розроблено веб-сайт для візуалізації даних щодо якості повітря.
6. Протестовано моделі прогнозування якості повітря.
7. Протестовано веб-сервіс, що надає інформацію про якість повітря.

У майбутньому планується нарощувати функціонал системи та розширювати навчальну вибірку. Розширення навчальної вибірки не лише даними з нових станцій, а й збільшення обсягу накопичених даних вже існуючих станцій підвищить якість прогнозованого забруднення повітря.

ПЕРЕЛІК ВИКОРИСТАНИХ ПОСИЛАНЬ

1. Copernicus Atmosphere Monitoring Service [Електронне видання]: – Режим доступу: <https://atmosphere.copernicus.eu/>
2. IQAir [Електронне ресурс]: – Режим доступу: <https://www.iqair.com>
3. Python [Електронне ресурс]: – Режим доступу: <https://www.python.org/>
4. PostgreSQL [Електронне ресурс]: – Режим доступу: <https://www.postgresql.org/>
5. TensorFlow [Електронне ресурс]: – Режим доступу: <https://www.tensorflow.org/>
6. Keras [Електронне ресурс]: – Режим доступу: <https://keras.io/>
7. Pandas [Електронне ресурс]: – Режим доступу: <https://pandas.pydata.org/>
8. FastAPI [Електронне ресурс]: – Режим доступу: <https://fastapi.tiangolo.com/> Bootstrap Flask [Електронне ресурс]: – Режим доступу: <https://bootstrap-flask.readthedocs.io/en/stable/> (дата звернення: 10.05.2020 р.).
9. NumPy [Електронне ресурс]: – Режим доступу: <https://numpy.org/>
10. Pydantic [Електронне ресурс]: – Режим доступу: <https://pydantic-docs.helpmanual.io/>
11. Chartjs [Електронне ресурс]: – Режим доступу: <https://www.chartjs.org>
12. Bootstrap [Електронне ресурс]: – Режим доступу: <https://getbootstrap.com>
13. Leafletjs [Електронне ресурс]: – Режим доступу: <https://leafletjs.com>
14. Openweathermap [Електронне ресурс]: – Режим доступу: <https://openweathermap.org/api>.