

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Дніпровський національний університет імені Олеся Гончара

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Дніпровський національний університет імені Олеся Гончара

Кваліфікаційна
наукова праця на правах
рукопису

МОЛОДЕЦЬ БОГДАН ВОЛОДИМИРОВИЧ

УДК 004.42

ДИСЕРТАЦІЯ

**РОЗРОБЛЕННЯ ТЕХНОЛОГІЙ ТА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ
ОПЕРАТИВНОГО МОНІТОРИНГУ ЯКОСТІ ПОВІТРЯ**

12 Інформаційні технології

121 Інженерія програмного забезпечення

Подається на здобуття ступеня доктора філософії. Дисертація містить результати власних досліджень. Використання ідей, результатів та текстів інших авторів мають посилання на відповідне джерело

_____ Б.В. Молодець

Науковий керівник:
Байбуз Олег Григорович
доктор технічних наук, професор

Дніпро – 2024

АНОТАЦІЯ

Молодець Б.В. Розроблення технологій та програмного забезпечення оперативного моніторингу якості повітря. – Кваліфікаційна наукова праця на правах рукопису.

Дисертація на здобуття ступеня доктора філософії за спеціальністю 121 Інженерія програмного забезпечення – Дніпровський національний університет імені Олеся Гончара, Дніпро, 2024.

Сучасне суспільство стикається з низкою серйозних екологічних викликів, з якими пов'язані термінові завдання збереження екосистем та охорони здоров'я людей. Один із найактуальніших аспектів цих завдань — це проблема забруднення повітря, яке є серйозною загрозою громадському здоров'ю та стану навколишнього середовища. У зв'язку з цим, розробка та вдосконалення інформаційних систем моніторингу якості повітря набуває особливого значення.

Метою даної дисертації є дослідження, розробка та впровадження інформаційної системи моніторингу якості повітря з метою підвищення ефективності та точності збору та аналізу даних. Дослідження спрямоване на вдосконалення існуючих методів моніторингу та розробку нових технологічних рішень для забезпечення більш точного вимірювання показників якості повітря.

У першому розділі проведено всебічний аналіз існуючих технологій, систем та наукових праць пов'язаних з розробкою технологій та програмного забезпечення оперативного моніторингу якості повітря. Серед інформаційних систем було виділено наступні: AirNow (національна система моніторингу якості повітря в Сполучених Штатах), Європейський індекс якості повітря (EAQI), Світовий індекс якості повітря (WAQI), Національний центр моніторингу навколишнього середовища Китаю (CNEMC), глобальна платформа для моніторингу якості повітря Breezometer та відкрита платформа OpenAQ. Також розглядаються існуючі оцінки якості повітря: Канадський

стандарти якості навколишнього повітря (CAAQS), норми концентрацій визначені ЄС (Європейський союз) та ВООЗ (Всесвітньої організації охорони здоров'я) та показником гранично допустимого забруднення (які використовуються в Україні). В результаті огляду існуючого програмного забезпечення формується перелік недоліків, які мають зникнути. У результаті формуються вимоги до інформаційної системи.

Другий розділ присвячено аналізу моделей оцінки якості повітря, опис їх переваг та недоліків, описується метод покращення роботи моделей оцінок та дані, які будуть використані в процесі моделювання. Для аналізу якості повітря використовуватимуться три види даних: супутникові дані, дані зі станцій наземного моніторингу та метеодані з NOAA (National oceanic and atmospheric administration). Джерелом даних ДЗЗ (Дистанційне зондування землі) слугують супутникові знімки місії Sentinel-5P, які надають вимірювання майже в реальному часі різних забруднюючих речовин у нижніх шарах атмосфери, такі як озон, окис азоту, двоокис сірки, метан, формальдегід. У якості математичних моделей дисперсії газів були розглянуті Гаусова модель, Лагранжева модель, Ейлерова модель та нестационарна модель дисперсії газів CALPUFF. Модель CALLPUFF, яка базується на Лагранжевій моделі, враховує більшу кількість параметрів включно з кліматичною зоною, рівнем висот та іншим. Для уточнення результатів моделювання обрано детерміновані методи інтерполяції, що розраховують результат, ґрунтуючись на вимірних значеннях, які потрапляють в околицю інтерпольованої точки, і на заданих математичних формулах. У результаті аналізу було обрано використовувати модель CALPUFF у поєднанні з методом IDW. Це дозволяє враховувати просторову залежність концентрацій забруднюючих речовин, особливо в областях з обмеженим або відсутнім моніторингом, та підвищити якість оцінки.

У третьому розділі описуються технології та архітектурні рішення, які використовує інформаційна система. Порівнюються протоколи обміну даних, архітектури мережевих протоколів, архітектури програмного забезпечення та провайдери хмарних послуг. В якості хмарних провайдерів розглядаються

Google Cloud Platform, Microsoft Azure, Amazon Web Service. В результаті порівняння було обрано Amazon Web Service, який має найширше географічне охоплення, є інтегрований з різноманітними технологіями та мовами програмування (включаючи й ті, що використовуються в роботі). В якості протокола обміну даних було обрано HTTP (Hypertext Transfer Protocol) через можливість протоколу працювати з захищеним з'єднанням, підтримки механізму кешування та іншого. Розглядається менеджер контейнерів Docker та протокол авторизації OAuth 2.0. Розглядаються та порівнюються архітектурні рішення сервісно-орієнтовну архітектуру, Serverless архітектуру, монолітну архітектуру та мультисервісну архітектуру. Серед архітектур мережевих протоколів розглядаються REST (Representational State Transfer) та GraphQL. Описується обране рішення у вигляді сервісно-орієнтовної архітектури та REST (Representational State Transfer) архітектурою мережевих протоколів.

Четвертий розділ присвячено можливостям практичного застосування розробленої системи. В ньому наведено стек технологій на якому написана система. Серверна частина була реалізована з використання мови програмування Python та Django Rest фреймворком. Цей вибір обумовлений наявністю стандартних модулів, які включають багато зручних інструментів для роботи з файлами, системні виклики, інтерфейси до різних графічних бібліотек і мережевих з'єднань. В якості бази даних обрано реляційну базу даних PostgreSQL з розширенням PostGIS. Клієнтська частина була реалізована за допомогою Angular фреймворку. Описується схеми взаємодії компонентів системи між собою та користувача з системою у вигляді UML-діаграм. Наведена схема баз даних, розглянуті основні таблиці розробленого застосунку. Розписана схема роботи моделі CALPUFF та IDW алгоритму. Ця комбінація може допомогти уточнити модельний прогноз та надати більш повну картину розподілу забруднень у повітрі, що є важливим для оцінки ефектів забруднення на здоров'я та навколишнє середовище. Порівнявши результати прогнозу погоди, що є важливою складовою для оцінки якості повітря, виявлено перевагу Weather Research and Forecasting (WRF) моделі у порівнянні зі стандартними

авторегресійними моделями. Цьому сприяло врахування більшої кількості факторів. Наведені показник відмови, тривалість запиту під час стрес-тестування системи. Вказаний відсоток користувачів, які потенційно зможуть завітати до розробленої системи. Наведені схеми роботи моделей та результати їх роботи.

Висновки підводять підсумки проведеної роботи, виокремлюють основні наукові та практичні здобутки автора у розробці інформаційної системи моніторингу якості повітря. Робота має значний потенціал для подальшого дослідження та розширення у сферах, де важлива автоматизація збору та обробки даних, побудови оцінки про стан якості повітря.

Наукова новизна одержаних результатів полягає у наступному:

- вдосконалена модель оцінки якості повітря шляхом об'єднання: точкової оцінки стану повітря, отриманих зі станцій наземного моніторингу; кількісної оцінки стану повітря (об'єм викидів забрудників) по області використовуючи супутникові дані; якісної оцінки у вигляді інтерполяційної мапи індексу якості повітря побудованим по даним з наземних станцій; доуточнення стану повітря в окремо виділених регіонах за допомогою моделі CALPUFF, що надає комплексну оцінку якості повітря в обраному регіоні;

- вперше запропонована архітектура системи для агрегації та оперативної апробації даних наземних станцій та супутникових знімків, що надало можливість автоматизувати процес збору та серіалізації даних для оцінки якості повітря;

- вперше розроблена інформаційна система, що в реальному часі агрегує дані з станцій наземного моніторингу, метеорологічних даних та даних дистанційного зондування землі, що пришвидшує процес аналізу стану якості повітря.

Практичне значення одержаних результатів полягає у наступних аспектах:

- використовуючи інформаційну систему моніторингу якості повітря громадяни, державні органи та організації зможуть отримувати інформацію про якість повітря та про джерела його забруднення;

- візуалізовані інформаційною системою дані про якість повітря слугують інструментом для прийняття рішень;
- інформаційна система слугує сховищем для даних, для подальшого покращення роботи моделей оцінки якості повітря.

Ключові слова: інформаційні технології, моніторинг якості повітря, архітектура програмного забезпечення, моделювання дисперсії газів, обробка супутникових зображень, багатоспектральне зображення, математична модель, бази даних, API, SOA, геоінформаційна технологія, інформаційна система, сталий розвиток міст, цифрове фотограмметричне зображення.

ABSTRACT

Molodets B.V. Development of technologies and software for operational air quality monitoring. – Qualifying scientific work as a manuscript.

Dissertation for the degree of Doctor of Philosophy in specialty 121 Software Engineering – Oles Honchar Dnipro National University, Dnipro, 2024.

Modern society is facing a number of serious environmental challenges, which are associated with urgent tasks of preserving ecosystems and protecting human health. One of the most pressing aspects of these challenges is the problem of air pollution, which is a serious threat to public health and the environment. In this regard, the development and improvement of air quality monitoring information systems is of particular importance.

The purpose of this work is to investigate, develop, and implement an information system for air quality monitoring in which to improve the efficiency and accuracy of data collection and analysis. The aim of the research is to improve existing monitoring methods and develop new technological solutions to provide more accurate measurement of air quality indicators.

The first section provides a comprehensive analysis of existing technologies, systems, and research studies related to the development of technologies and software for operational air quality monitoring. Existing air quality assessments are also reviewed. The following information systems are considered: AirNow (The National Air Quality Monitoring System in the United States), the European Air Quality Index (EAQI), the World Air Quality Index (WAQI), the China National Environmental Monitoring Centre (CNEMC), the Breezometer global air quality monitoring platform, and the OpenAQ open platform. Existing air quality assessments are also reviewed: Canadian Ambient Air Quality Standards (CAAQS), concentration standards set by the EU (European Union) and WHO (World Health Organisation) and the maximum permissible pollution limit (used in Ukraine). As a result of the review of existing software, a list of issues that need

to be resolved is formed. As a result, the requirements for the information system are formed.

The second section is devoted to the analysis of air quality assessment models, describing their advantages and disadvantages, describing the method of improving the performance of assessment models and the data which will be used in the modeling process. Three types of data are used for air quality analysis: satellite data, data from ground monitoring stations and meteorological data from NOAA (National oceanic and atmospheric administration). The source of remote sensing data is satellite imagery from the Sentinel-5P mission, which provides near real-time measurements of various pollutants in the lower atmosphere, such as ozone, nitrogen oxide, sulphur dioxide, methane, and formaldehyde. The Gaussian model, Lagrangian model, Eulerian model, and the CALPUFF nonstationary gas dispersion model were considered as mathematical models of gas dispersion. The CALLPUFF model, which is based on the Lagrangian model, involves a larger number of parameters, including climate zone, altitude level, etc. To clarify the modelling results, deterministic interpolation methods were chosen, which calculate the result based on measured values that fall in the area around the interpolated point and on specified mathematical formulas. As a result of the analysis, it is chosen to use the CALPUFF model in combination with the IDW method. That allows to consider the spatial dependence of pollutant concentrations, especially in areas with limited or no monitoring, and to improve the quality of the assessment.

The third section describes the technologies and architectural solutions used by the information system. Data exchange protocols, network protocol architectures, software architectures, and cloud service providers are compared. Google Cloud Platform, Microsoft Azure, and Amazon Web Service are considered as cloud providers. As a result of the comparison, Amazon Web Service is chosen, because it has the widest geographical coverage, it is integrated with various technologies and programming languages (including those used in the work). HTTP (Hypertext Transfer Protocol) is chosen as the data exchange

protocol because of the protocol's ability to work with a secure connection, support a caching mechanism, etc. The Docker container manager and the OAuth 2.0 authorization protocol are considered. Architectural solutions are reviewed and compared: service-oriented architecture, Serverless architecture, monolithic architecture and multiservice architecture. Among the network protocol architectures, REST (Representational State Transfer) and GraphQL are considered. The chosen solution is described as a service-oriented architecture and REST (Representational State Transfer) network protocol architecture.

The fourth section is about the practical application of the developed system. It describes the technology stack on which the system is written. The server side is implemented by using the Python programming language and Django Rest framework. This choice is due to the availability of standard modules that include many convenient tools for working with files, system calls, interfaces to various graphic libraries and network connections. PostgreSQL is chosen as database with the PostGIS extension. The client side was released using the Angular framework. It explains the schemes of interaction of the system components with each other and the user with the system by using UML diagrams. The scheme of databases is presented, the main tables of the developed application are considered. The scheme of operation of the CALPUFF model and the IDW algorithm is described. This combination helps to refine the model forecast and provide a more complete picture of the distribution of pollution in the air, which is important for assessing the effects of pollution on health and the environment. After comparing the results of the weather forecast, which is an important component of air quality assessment, the Weather Research and Forecasting (WRF) model is found to be superior to standard autoregressive models. This is due to the inclusion of more factors in the modelling. The failure rate and request duration during stress testing of the system are given. The percentage of users who will be able to access the developed system is indicated. The schemes of the models and the results of their work are presented.

The summary highlights the results of the work that has been carried out, marking out the main scientific and practical achievements of the author in the

development of an air quality monitoring information system. The work has substantial potential for further research and expansion in areas where automation of data collection and processing is important, as well as for building an assessment of air quality.

The scientific novelty of the results obtained is as follows:

- the air quality assessment model is improved by combining: a spot assessment of the air condition obtained from ground monitoring stations; a quantitative assessment of the air condition (pollutant emissions) for the region using satellite data; a qualitative assessment in the form of an interpolation map of the air quality index based on data from ground stations; and a refinement of the air condition in selected regions using the CALPUFF model, which provides a comprehensive assessment of air quality in the chosen region;
- for the first time, a system architecture is proposed for aggregation and operational testing of data from ground stations and satellite images, which makes it possible to automate the process of collecting and serializing data for air quality assessment;
- for the first time, an information system was developed that aggregates data from ground monitoring stations, meteorological data, and remote sensing data in real time, which speeds up the process of analyzing air quality.

The practical significance of the results obtained lies in the following aspects:

- using the air quality monitoring information system, citizens, government agencies, and organizations will be able to receive information on air quality and sources of air pollution;
- the air quality data visualized by the information system serves as a decision-making tool;
- the information system serves as a data repository to further improve the performance of air quality assessment models.

Keywords: information technology, air quality monitoring, software architecture, modeling of gas dispersion, satellite image processing, multispectral images,

mathematical model, database, API, SOA, geoinformation technology, information system, sustainable urban development, digital photogrammetric image.

СПИСОК ОПУБЛІКОВАНИХ ПРАЦЬ ЗА ТЕМОЮ ДИСЕРТАЦІЇ

Статті що індексуються в наукометричній базі Scopus:

1. Molodets, B., Hnatushenko Volodymyr, Boldyriev D., Bulana, T. Information System of Air Quality Assessment Using Data Interpolation from Ground Stations. CEUR Workshop Modern Machine Learning Technologies and Data Science Workshop (MoML&T&DS 2023), Lviv 2023. Vol. 3426. P. 233–245. Режим доступу до ресурсу: <https://ceur-ws.org/Vol-3426/>.
2. Molodets, B., Hnatushenko Volodymyr, Boldyriev D. , Bulana T. Information System of Air Quality Assessment Based of Ground Stations and Meteorological Data Monitoring. CEUR Workshop Intelligent Information Technologies & Systems of Information Security (IntelITSIS 2023). 2023.Vol. 3373, P. 206–216. Режим доступу до ресурсу: <https://ceur-ws.org/Vol-3373/>.

Статті у наукових фахових виданнях України:

3. Молодець Б.В, Тарасов В.П., Булана Т.М., Байбуз О.Г. Аналіз існуючих технологій моніторингу повітря. Регіональний міжвузівський збірник «Системні Технології», м. Дніпро, 2021, Випуск 3(134). С. 67–78. doi: <https://www.doi.org/10.34185/1562-9945-3-134-2021-08>. Режим доступу до ресурсу: <https://journals.nmetau.edu.ua/index.php/st/article/view/748>.
4. Молодець Б.В, Байбуз О.Г. Використання методу обернено зважених відстаней з моделлю CALPUFF для інтерполяції концентрацій забруднюючих речовин у повітрі. Збірник наукових праць «Актуальні проблеми автоматизації та інформаційних технологій», м. Дніпро, 2024, С. 156–163. doi: <https://www.doi.org/10.15421/432315>. Режим доступу до ресурсу: <https://actualproblems.dp.ua/index.php/APAIT/article/view/240>.
5. Молодець Б.В, Булана Т.М. Аналіз існуючих архітектур для розробки системи оцінки якості повітря. Регіональний міжвузівський збірник «Системні Технології», м. Дніпро, 2024, Випуск 3(152). С. 128–139 doi: <https://www.doi.org/10.34185/1562-9945-3-152-2024-13>. Режим доступу до ресурсу: <https://journals.nmetau.edu.ua/index.php/st/article/view/1702>

Наукові праці, які засвідчують апробацію матеріалів дисертації:

6. Молодець Б.В. Створення системи агрегування даних за допомогою Celery. VII Всеукраїнську науково-практичну конференцію «Перспективні напрямки сучасної електроніки, інформаційних і комп'ютерних систем». м. Дніпро, 2022. С.48-49. Режим доступу до ресурсу: <http://meics.dnure.dp.ua/files/MEICS-2022.pdf>
7. Молодець Б.В., Болдирєв Д.О., Децик В.О., Булана Т.М., Грабовець Д.М. Розробка модульного навчального додатку з використання гнучкої архітектури docker контейнерів. Міжнародна науково-технічна конференція Інформаційні технології в металургії та машинобудуванні, м. Дніпро, 2022. С. 298-300 doi: <https://doi.org/10.34185/1991-7848.itmm.2022.01.059> Режим доступу до ресурсу: <https://journals.nmetau.edu.ua/index.php/itmm/article/view/1175>
8. Молодець Б.В, Болдирєв Д.О, Булана Т.М. Аналітичне рішення задач перенесення забруднень у двовимірному просторі. Перспективні напрямки сучасної електроніки, інформаційних і комп'ютерних систем (MEICS-2021), м. Дніпро, 2021. С. 21-22. Режим доступу до ресурсу: <http://meics.dnure.dp.ua/files/MEICS-2021.pdf>
9. Болдирєв Д.О., Молодець Б.В., Булана Т.М. Розробка програмного додатку для вирішення задачі перенесення забруднень у двовимірному просторі. XIX міжнародна науково-практична конференція Математичне та програмне забезпечення інтелектуальних систем (МПЗІС-2021), м. Дніпро, 2021. С. 21-22. Режим доступу до ресурсу: <http://mpzis.dnu.dp.ua/wp-content/uploads/2021/12/mpzis-2021.pdf>

ЗМІСТ

Перелік умовних позначень.....	16
ВСТУП	17
РОЗДІЛ 1. АНАЛІЗ ІСНУЮЧИХ ТЕХНОЛОГІЙ ОЦІНКИ ЯКОСТІ ПОВІТРЯ	21
1.1 Аналіз існуючого програмного забезпечення оцінки якості повітря.....	22
1.1.1 Аналіз прикладного програмного забезпечення для інтернета речей.....	23
1.1.2 Програмне забезпечення для моделювання дисперсії газів з точкового джерела	24
1.2 Оцінки якості повітря норми і стандарти	26
1.3 Індекс якості повітря	30
1.4 Європейський індекс якості повітря	32
1.5 Аналіз Американських стандартів оцінки якості повітря	34
1.6 Огляд наукових праць пов'язаних з оцінкою якості повітря.....	37
1.7 Висновок	39
РОЗДІЛ 2. МАТЕМАТИЧНІ МОДЕЛІ ОЦІНКИ ЯКОСТІ ПОВІТРЯ	41
2.1 Супутникові знімки місії Sentinel-5P	42
2.2 Опис даних зі станції наземного моніторингу	44
2.3 Опис метеоданих з NOAA (National oceanic and atmospheric administration).....	46
2.4 Моделювання дисперсії газів за допомогою Ейлерової моделі.....	56
2.5 Моделювання дисперсії газів за допомогою Гаусових моделей.....	57
2.6 Моделювання дисперсії газів за допомогою Лагранжевих моделей.....	60
2.7 Нестационарна модель дисперсії газів CALPUFF	61
2.8 Метод інтерполяції для постобробки результатів моделювання.....	63
2.9 Висновок	65
РОЗДІЛ 3. ТЕХНОЛОГІЇ МОДЕЛІ ДЛЯ РОЗРОБКИ АРХІТЕКТУРИ ІНФОРМАЦІЙНОЇ СИСТЕМИ.....	67
3.1 Порівняння провайдерів хмарних послуг	67

3.1.1	Аналіз платформи Google Cloud Platform	68
3.1.2	Аналіз платформи Microsoft Azure.....	69
3.1.3	Аналіз платформи Amazon Web Services	70
3.2	Різновиди збору даних для інформаційної системи моніторингу якості повітря.....	72
3.3	Аналіз протоколів для обміну даних в системі	74
3.3.1	Можливості протоколу обміну даними UDP	74
3.3.2	Можливості протоколу обміну даними TCP	76
3.3.3	Можливості протоколу обміну даними HTTP	77
3.4	Віртуальна контейнеризація інформаційної системи.....	78
3.5	Авторизаційний фреймворк для надання обмеженого доступу в систему.	79
3.6	Аналіз архітектури мережесх протоколів	80
3.7	Аналіз архітектурних рішень для створення інформаційної системи.....	85
3.8	Висновок	94
РОЗДІЛ 4. РЕЗУЛЬТАТИ РОБОТИ СИСТЕМИ		96
4.1	Моделювання розповсюдження забруднення використовуючи модель CALPUFF та IDW алгоритму.....	97
4.2	Інформаційна технологія прогнозу погоди для оцінки якості повітря.....	100
4.3	Інформаційна технологія парсингу та агрегації даних в системі	103
4.4	Інформаційна технологія взаємодії системи з користувачем	112
4.5	Тестування інформаційної системи оцінки якості повітря	123
4.6	Висновок	129
ВИСНОВКИ.....		131
СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ.....		133
ДОДАТКИ		145

Перелік умовних позначень

AWS – Amazon Web Services;

AQI – Air quality index;

EAQI – European Air Quality Index;

WAQI – World Air Quality Index;

EPA – Environmental Protection Agency;

FTP – File transfer protocol;

UDP – User datagram protocol;

TCP – Transmission Control Protocol;

HTTP – Hypertext Transfer Protocol;

MSA – Microservice architecture;

SOA – Serviceoriented architecture;

REST – Representational State Transfer;

ВООЗ – Всесвітня організація охорони здоров'я;

ГДЗ – Гранично допустиме забруднення;

MTV – Model template view архітектура;

SOLID – Single responsibility principle, Open/closed principle, Liskov substitution principle, Interface segregation principle, Dependency inversion principle;

ВСТУП

Обґрунтування вибору теми дослідження. Сучасне суспільство стикається з низкою серйозних екологічних викликів, з якими пов'язані термінові завдання збереження екосистем та охорони здоров'я людей. Один із найактуальніших аспектів цих завдань — це проблема забруднення повітря, яке є серйозною загрозою громадському здоров'ю та стану навколишнього середовища. У зв'язку з цим, розробка та вдосконалення інформаційних систем моніторингу якості повітря набуває особливого значення.

Значущі зміни в кліматі, швидке індустріалізації та зростання автотранспортного руху спричиняють накопичення токсичних речовин у атмосфері, що призводить до погіршення якості повітря та підвищує ризик розвитку різноманітних захворювань у населення. У цьому контексті, інформаційні системи моніторингу стають необхідним інструментом для ефективного виявлення, контролю та зменшення рівнів забруднення повітря.

В сучасному світі, де виробництво та технології необоротно впливають на довкілля, питання забезпечення високої якості повітря є ключовим завданням для збереження здоров'я населення та екосистем. Забруднення повітря стає серйозною загрозою для громадського здоров'я, і ефективні механізми моніторингу та управління якістю повітря стають невід'ємною частиною стратегій збереження середовища.

Інформаційні системи моніторингу є важливим інструментом у боротьбі з забрудненням повітря. Сучасні технології дозволяють в режимі реального часу збирати та аналізувати дані про якість повітря, що надає можливість оперативно реагувати на забруднення та вживати заходів для його усунення.

Зв'язок роботи з науковими програмами, планами, темами. Дисертаційна робота виконана в рамках науково-дослідної роботи № ФПМ-79-19 «Розробка програмного комплексу аналізу та прогнозування часових рядів», 2019-2021 рр. № держреєстрації 0119U101056 та науково-дослідної роботи №

ФПМ-2-22 «Розроблення програмного забезпечення аналізу та кластеризації часових рядів» 2022-24 рр. № держреєстрації 0122U001465.

Мета і завдання дослідження. Метою даної дисертації є дослідження, розробка та впровадження інформаційної системи моніторингу якості повітря з метою підвищення ефективності та точності збору та аналізу даних. Дослідження спрямоване на вдосконалення існуючих методів моніторингу та розробку нових технологічних рішень для забезпечення більш точного вимірювання показників якості повітря.

Об'єкт дослідження – процес зміни якості повітря.

Предмет дослідження – інформаційна система моніторингу якості повітря.

Актуальність даної роботи полягає в тому, що в умовах зростаючого рівня промислового виробництва та автотранспорту, питання якості повітря стає насущним для розвинених та країн які розвиваються. Впровадження ефективних інформаційних систем моніторингу дозволить оперативно реагувати на негативні зміни та забезпечить створення умов для здорового та безпечного середовища для мешканців.

Для досягнення мети пропонується здійснити комплексний підхід, що включає в себе кілька ключових завдань, наведених нижче.

- проаналізувати існуючі системи та рішення стосовно моніторингу якості повітря;
- дослідити математичні моделі для оцінки забруднення повітря;
- розробити архітектурне рішення для агрегації наземних та супутникових даних, та розрахунку якості повітря;
- реалізувати побудову інформаційної системи з моніторингу за розробленою архітектурою;
- провести апробацію результатів діагностики якості повітря, проведеної за допомогою розроблених алгоритмів та моделей;

— розробити інформаційну систему оперативного моніторингу якості повітря на основі розроблених моделей та алгоритмів для ефективного контролю та управління рівнями забруднення атмосферного повітря.

Методи дослідження: методи та технології інженерії програмного забезпечення, баз даних, математичне моделювання якості повітря.

Наукова новизна одержаних результатів полягає у наступному:

— вдосконалена модель оцінки якості повітря шляхом об'єднання: точкової оцінки стану повітря, отриманих зі станцій наземного моніторингу; кількісної оцінки стану повітря (об'єм викидів забрудників) по області використовуючи супутникові дані; якісної оцінки у вигляді інтерполяційної мапи індексу якості повітря побудованим по даним з наземних станцій; доуточнення стану повітря в окремо виділених регіонах за допомогою моделі CALPUFF, що надає комплексну оцінку якості повітря в обраному регіоні;

— вперше запропонована архітектура системи для агрегації та оперативної апробації даних наземних станцій та супутникових знімків, що надало можливість автоматизувати процес збору та серіалізації даних для оцінки якості повітря;

— вперше розроблена інформаційна система, що в реальному часі агрегує дані з станцій наземного моніторингу, метеорологічних даних та даних дистанційного зондування землі, що пришвидшує процес аналізу стану якості повітря.

Практичне значення одержаних результатів полягає у наступних аспектах:

— використовуючи інформаційну систему моніторингу якості повітря громадяни, державні органи та організації зможуть отримувати інформацію про якість повітря та про джерела його забруднення;

— візуалізовані інформаційною системою дані про якість повітря слугують інструментом для прийняття рішень;

— інформаційна система слугує сховищем для даних, для подальшого покращення роботи моделей оцінки якості повітря.

Особистий внесок здобувача. Аналіз літературних даних, оцінка математичних моделей та розробки програмного забезпечення, обробка отриманих результатів здійснені безпосередньо автором. Постановка мети і завдань дослідження, а також аналіз і узагальнення отриманих результатів проводились спільно з науковим керівником д.т.н., проф. О. Г. Байбузом.

Апробація результатів дисертації. Основні положення та результати дисертаційної роботи доповідалися й обговорювалися на 5-му семінарі з сучасних технологій машинного навчання та науки про дані (MoMLeT & DS 2023) та 4-му Міжнародному семінарі з інтелектуальних інформаційних технологій та систем інформаційної безпеки (IntelITSIS-2023).

Публікації. Основні положення й результати дисертаційної роботи опубліковано у 9 роботах: 2 статті що індексуються в наукометричній базі Scopus, 2 статті у наукових фахових виданнях України категорії Б та 6 тез доповідей у збірниках матеріалів наукових конференцій.

Зв'язок роботи з науковими програмами, планами, темами. Дисертаційна робота виконувалась у відповідності з індивідуальним планом підготовки аспіранта кафедри математичного забезпечення ЕОМ Дніпровського національного університету імені Олеся Гончара.

Структура та обсяг дисертації. Робота містить вступ, 4 розділи, висновки та список використаних джерел, що містить 104 найменувань. Загальний обсяг дисертації – 145 сторінок, обсяг основного тексту – 114 сторінки. Робота містить 46 рисунків та 14 таблиць.

РОЗДІЛ 1. АНАЛІЗ ІСНУЮЧИХ ТЕХНОЛОГІЙ ОЦІНКИ ЯКОСТІ ПОВІТРЯ

Аналіз існуючих інформаційних систем моніторингу якості повітря може включати огляд різних аспектів, таких як технічні можливості, методології вимірювань, покриття мережі, точність даних, доступність та ефективність. Нижче наведено декілька ключових пунктів, які можна розглядати під час аналізу існуючих інформаційних систем моніторингу якості повітря:

1. Сенсори та Обладнання. Тип сенсорів: оцінка типів сенсорів, які використовуються для вимірювання забруднень у повітрі. Важливо враховувати точність, чутливість та діапазон вимірювань. Кількість та розміщення: перевірка кількість сенсорів у системі та їх розміщення для забезпечення адекватного покриття території.

2. Дані та їхнє збирання: Точність даних: оцінка точності вимірювань та якість даних, що збираються системою. Інтервал вимірювань: перевірка, як часто проводяться вимірювання та як це впливає на точність моніторингу.

3. Методологія та стандарт. Використані методики - визначення, які методики використовуються для оцінки якості повітря та чи вони відповідають встановленим стандартам. Співставлення з міжнародними стандартами - перевірка чи система відповідає міжнародним стандартам та протоколам;

4. Доступність та Публічна Інформація. Доступ до даних – оцінка доступності отриманих даних для громадськості та інших зацікавлених сторін. Інтерактивність та візуалізація – чи є інтерактивні інтерфейси та засоби візуалізації інструментом для полегшення розуміння даних.

5. Система попередження та реагування. Механізми попередження – визначення, як система попередження про якість повітря функціонує та як швидко реагує на забруднення. Системи взаємодії з громадськістю – перевірка, що інформація легко доступна для громадськості та що існують механізми залучення громадськості до процесу моніторингу.

6. Забезпечення конфіденційності та безпеки. Конфіденційність даних – визначення заходів, які вживаються для забезпечення конфіденційності

особистих даних. Захист від кіберзагроз – визначення, які заходи безпеки встановлені для захисту системи від кіберзагроз.

Під час аналізу важливо брати до уваги контекст місцевих умов, потреб громади та специфічні аспекти, що впливають на якість повітря в конкретній локації.

1.1 Аналіз існуючого програмного забезпечення оцінки якості повітря

Існує багато інформаційних систем моніторингу якості повітря, які використовуються по всьому світу. Деякі приклади таких систем включають:

1. AirNow (США) – це національна система моніторингу якості повітря в Сполучених Штатах. Вона забезпечує реальний час інформацією про рівні забруднення повітря та рекомендаціями для громадян.

2. European Air Quality Index (EAQI). Європейський індекс якості повітря – це інформаційна система для моніторингу якості повітря в Європейському союзі. Система використовує інформацію з різних джерел та визначає загальний індекс якості повітря для кожної області.

3. World Air Quality Index (WAQI). Опис: Світовий індекс якості повітря – це глобальна інформаційна система, яка надає дані про забруднення повітря з різних джерел. Використовується для моніторингу та порівняння якості повітря в різних місцях по всьому світу.

China National Environmental Monitoring Centre (CNEMC) – це національний центр моніторингу навколишнього середовища Китаю. Забезпечує інформацією про якість повітря та інші екологічні показники в режимі реального часу.

5. Breezometer. – це глобальна платформа для моніторингу якості повітря. Характеристики: Використовує дані з сенсорів, метеорологічні дані та моделі для визначення рівня забруднення повітря та надає інтерактивні карти для користувачів.

6. OpenAQ. – Опис глобальна платформа, що надає відкритий доступ до даних про якість повітря. Вона збирає дані з різних джерел і надає API для розробників та дослідників.

Ці системи відіграють важливу роль у наданні інформації громадянам, владам та науковцям для прийняття рішень щодо якості повітря та заходів для її покращення.

1.1.1 Аналіз прикладного програмного забезпечення для інтернету речей

Для пристроїв Aeroqual[1], таких як AQM 65, AQS 1, AQY 1, Dust Sentry або Dust Profiler розроблено вбудоване ПЗ. Доступ за допомогою спеціального інтерфейсу програмного забезпечення надає користувачам доступ до даних про якість повітря в режимі реального часу. Це також дозволяє надавати технічну підтримку та обслуговування. Програмне забезпечення базується на вебтехнологіях. Також наявний хмарний сервіс, що має такі можливості:

- підключення до датчика через WIFI, Ethernet або стільниковий телефон;
- перегляд датчиків на мапі;
- перегляд та завантаження даних у режимі реального часу;
- відображення даних на графіках;
- вибір періоду усереднення – 1 хв, 30 хв, 1 год, 2 год, 4 год, 8 год, 12 год, 24 год;
- використання рози вітру для визначення джерел забруднення;
- налаштування SMS та електронних повідомлень у відповідь на екстремальні події;
- надання даних через FTP або електронною поштою безпосередньо вибраним одержувачам;
- калібрування приладу віддалено (використовуючи вбудовану опцію калібратора).

Програмне забезпечення BL-VIEW генерує онлайн візуальне зображення висоти конвективного граничного шару, що забезпечує відображення місцевих умов. Дані про висоту граничного шару можна інтегрувати у зовнішні програми та прогнозування. Наявна можливість перегляду, аналізу та відображення даних у режимі реального часу. Інформація, що надається програмою BL-View[2], допомагає покращити моніторинг та прогнозування якості повітря та має такі особливості:

- доступ до технічного обслуговування до хмароміра за допомогою внутрішньої термінальної програми;
- підтримка моделей хмаромірів V31 та CL51;
- відображення стану хмароміра та якості зв'язку на головному екрані, щоб можна було дослідити робочі попередження та помилки;
- якість даних незалежно від погоди;
- використання алгоритму, що враховує можливі опади та хмарність, для забезпечення надійної звітності за будь-яких погодних умов;
- автоматичне аналізування даних граничного шару та зберігання результатів у файлах журналів, які можуть виступати як вхідні дані до чисельних моделей прогнозування погоди.

1.1.2 Програмне забезпечення для моделювання дисперсії газів з точкового джерела

Для дослідження впливу розподілу забруднення від конкретного джерела або групи джерел для оцінки якості повітря та впливу на здоров'я людини для малих просторових масштабів приватними компаніями розробляється спеціальне ПЗ. Окрім моделювання вони можуть включати оцінку прогнозу якості повітря відповідно до стандартів та граничних значень, включаючи стандарти ВООЗ, ЄС, Великобританії, США або Китаю.

BREEZE – компанія яка розробляє програмне забезпечення для дисперсійного моделювання. Включає в себе розробку та керування складними

наборами геофізичних та метеорологічних даних та інтеграцію таких наукових моделей як:

- AERMOD;
- CALPUFF;
- DEGADIS;
- SLAB;
- AFTOX;
- INPUFF;
- CAMx;
- CMAQ.

Програми, дотримуються стандартів Microsoft, упорядковують та візуалізують складні аналізи моделювання дисперсії, а також забезпечують, що всі пов'язані з ними вхідні та вихідні файли моделювання на 100% сумісні з нормативними документами EPA[3].

Lakes Environmental займається розробкою ПЗ для дисперсійного моделювання повітря з використанням інформації про джерела забруднення та метеорологічні умови для обчислення переміщення та концентрацій забруднювача в атмосфері.

AERMOD View – це найчастіше використовувана модель повітряної дисперсії і застосовується для широкого діапазону викидів в радіусі 50 км. Застосування розраховане для складного рельєфу та міської забудови.

AERSCREEN View – це діагностична модель на основі американської моделі AERMOD, яка оцінює найгірші концентрації рівня забруднення для одного джерела забруднення.

CALPUFF View – система, що використовує нестационарну Гаусову модель, яка підходить для моделювання транспортування часток на великі відстані, атмосферної хімії, видимості та в складних умовах атмосфери прибережних зон.

ADMS - Urban – це комплексна система моделювання якості повітря у великих міських районах та містах. Це єдина практична модель якості міського повітря, яка включає найновіші наукові розробки, представляє повний спектр

типів джерел забруднення, що зустрічаються в міській місцевості, враховує складну міську морфологію, включаючи вуличні каньйони, та забезпечує масштабування від вуличного до регіонального.

Основним недоліком цих програм є висока вартість, необхідність задання великої кількості параметрів, детальних метеорологічних та даних рельєфу.

1.2 Оцінки якості повітря норми і стандарти

Всесвітня організація охорони здоров'я (ВООЗ) видала настанови щодо рівня забруднення оточуючого повітря, які широко використовуються у всьому світі як орієнтири для встановлення стандартів та цілей управління якістю повітря.

Забруднення повітря у містах по всьому світу має серйозні наслідки для здоров'я. Це спричинило передбачувані 4,2 мільйона передчасних смертей у 2016 році. $PM_{2.5}$ – дрібнодисперсні речовини діаметром 2,5 мкм або менше є найнебезпечнішим забруднювачем, оскільки він може проникати через легеневий бар'єр та потрапляти в систему крові, викликаючи серцево-судинні, респіраторні та ракові захворювання. Отже за рахунок зниження рівня забруднення повітря країни можуть зменшити кількість захворювань [4]. Також існують три проміжні цільові стандарти, що наведені в таблиці 1.1:

Таблиця 1.1 Норми концентрацій $PM_{2.5}$

Середньорічна	Середньодобова
15 мкг/м ³	37,5 мкг/м ³
25 мкг/м ³	50 мкг/м ³
35 мкг/м ³	75 мкг/м ³

В керівництві ВООЗ зазначено, що середньорічні концентрації $PM_{2.5}$ не повинні перевищувати 10 мкг/м^3 , тоді як середньодобова не більш 25 мкг/м^3 більше трьох разів на рік.

Аналогічно розроблені стандарти ВООЗ щодо інших забруднювачів в таблиці 1.2:

Таблиця 1.2 Стандарти забруднювачів

Забруднювач	Концентрація
PM_{10}	річна концентрація 20 мкг/м^3 т середньодобова 50 мкг/м^3
O_3	100 мкг/м^3 , середня 8 годин.
NO_2 ,	40 мкг/м^3 середньорічна і 200 мкг/м^3 в середньому на 1 годину.
SO_2	20 мкг/м^3 середньодобова та 500 мкг/м^3 10 хвилинна.

Діяльність міських, національних та регіональних рівнів, що працюють в галузі транспорту, енергетики, поводження з відходами, містобудування та сільського господарства у регулюванні та впровадженні стандартів якості, є життєво важливими для чистого повітря [5].

Канадські стандарти якості навколишнього повітря (CAAQS) для дрібних твердих часток ($PM_{2.5}$) та озону виникли в результаті інтенсивного дослідження і встановлення нормативів у співпраці федеральних та територіальних урядів під егідою Ради міністрів Канади (CCME) з початку 2010-х років. Цей процес визначення стандартів відіграв ключову роль у формуванні стратегій з охорони довкілля та забезпеченні громадського здоров'я. Встановлені CAAQS для $PM_{2.5}$ та озону на 2015 та 2020 роки визначають максимально допустимі рівні цих забруднювачів у повітрі, які вважаються безпечними для здоров'я людини та екосистем [6].

Ці стандарти базуються на великій кількості наукових доказів щодо впливу PM_{2.5} та озону на здоров'я людини, включаючи дослідження епідеміологічних та експериментальних даних. Ці стандарти описані в таблиці 1.3.

Таблиця 1.3 Канадські стандарти якості для PM_{2.5} та Озону

Забруднювач	Час усереднення	Концентрації		Метрика
		2015	2020	
PM _{2.5}	24 год	28 µg/m ³	27 µg/m ³	Середнє значення річних концентрацій за 3 роки
PM _{2.5}	річний	10.0 µg/m ³	8.8 µg/m ³	Середнє значення річних концентрацій за 3 роки
Озон	8 год	63 ppb	62 ppb	Середні значення середніх значень чотирьох найвищих добових максимумів за 8 годин за 3 роки

У свою чергу Європейський Союз (ЄС) розробив широкий законодавчий акт, який встановлює стандарти та цілі щодо ряду забруднюючих речовин у повітрі.

Директиви ЄС щодо якості повітря (Директива 2008/50/ЄС про якість навколишнього повітря та очищення повітря для Європи та Директива 2004/107/ЄС про важкі метали та поліциклічні ароматичні вуглеводні в навколишньому повітрі) встановлюють порогові концентрації забруднюючих речовин, які не повинні перевищуватись у заданий проміжок часу. У разі перевищення, органи влади повинні розробити та впровадити плани управління якістю повітря. Ці плани повинні мати на меті довести концентрацію забруднюючих речовин до рівнів нижче граничних та цільових значень.

Вибрані стандарти ЄС та настанови Всесвітньої організації охорони здоров'я (ВООЗ) узагальнені в таблиці нижче. Вони застосовуються протягом

різних періодів часу, оскільки від цього залежить їх вплив на здоров'я людини. Значення ВООЗ встановлюються для охорони здоров'я і, як правило, суворіші порівняно з узгодженими стандартами ЄС (таблиця 1.4).

Таблиця 1.4 Норми концентрацій визначені ЄС та ВООЗ

Директива якості повітря ЄС			Норми ВООЗ
Забруднювач	Період усереднення	Концентрація	Концентрація
PM _{2.5}	погодинний		25 мг/м ³
PM _{2.5}	Річний	25 мг/м ³	10 мг/м ³
PM ₁₀	погодинний	50 мг/м ³	50 мг/м ³
PM ₁₀	річний	40 мг/м ³	20 мг/м ³
O ₃	Максимальне середнє за 8 годин	120 мг/м ³	100 мг/м ³
NO ₂	погодинний	200 мг/м ³	200 мг/м ³
NO ₂	річний	40 мг/м ³	40 мг/м ³

В Україні оцінка фактичного або прогнозного (розрахункового) рівня забруднення атмосферного повітря проводиться шляхом співставлення показника забруднення однією речовиною або сумарного показника забруднення (СПЗ) сумішшю речовин з показником гранично допустимого забруднення (ГДЗ). Допустимим визнається рівень, що не перевищує ГДЗ [5, 7].

Основою оцінки забруднення атмосферного повітря населених місць є гігієнічні нормативи допустимого вмісту в ньому хімічних, біологічних речовин (чи агентів) та допустимого впливу фізичних факторів. Ці правила стосуються оцінки забруднення атмосферного повітря лише хімічними та і біологічними речовинами (або агентами). (“Державні санітарні правила охорони атмосферного повітря населених місць”). Для оцінки забруднення атмосферного повітря використовуються значення концентрацій забруднюючих речовин, одержаних при розрахунках очікуваного забруднення атмосферного повітря за діючою

методикою, або значення концентрацій фактичного забруднення атмосферного повітря, одержані при лабораторних дослідженнях у відповідності з вимогами державного стандарту 17.2.3.01-86 «Охрана природы. Атмосфера. Правила контроля якості повітря населених пунктів» та «Керівництво по контролю забруднення атмосфери» (РД 52.04.186-89) на стаціонарних, маршрутних або підфакельних постах. При оцінці забруднення атмосферного повітря населених місць допустимим та безпечним для здоров'я людей приймається рівень, при якому концентрації окремих забруднюючих речовин а також сумарні показники забруднення не перевищують встановлені гігієнічні нормативи допустимого вмісту (ГДК – гранично допустима концентрація, ОБРД – орієнтовні безпечні рівні дії, ГДЗ – гранично допустимого забруднення).

Оцінка фактичних і прогнозних рівнів забруднення атмосферного повітря повинна проводитися у всіх видах передпроектної та проектної документації (в тому числі у локальних і зведених томах ГДВ – гранично допустимих викидів) з метою обґрунтування заходів, які будуть забезпечувати додержання гігієнічних нормативів [7].

1.3 Індекс якості повітря

AQI – це індекс для звітування про якість щоденного повітря. AQI зосереджується на впливі на здоров'я, які ви можете відчувати протягом декількох годин або днів після вдихання забрудненого повітря. ЕРА розраховує AQI для п'яти основних забруднювачів повітря: наземний озон, забруднення РМ, оксид вуглецю, діоксид сірки та діоксид азоту.

Щоб полегшити розуміння ЕРА розробила власний стандарт AQI, який ділиться на шість категорій, кожна з яких має власний вплив на здоров'я. Для кожної з цих забруднюючих речовин ЕРА встановила національні стандарти якості повітря для захисту здоров'я населення. Озонові та частинки повітря, які містяться в повітрі, є двома забруднювачами, які представляють найбільшу загрозу здоров'ю людини в цій країні.

Рівні індексу представлені у таблиці 1.5 [8]:

Таблиця 1.5 Рівні індексу якості повітря за стандартом ЕРА

Назва рівню	Діапазон значень	Пояснення
Хороший	від 0 до 50	Якість повітря вважається задовільною, а забруднення повітря не представляє мало ризику або не має ніякого ризику
Помірний	від 51 до 100	Якість повітря прийнятна; однак для деяких забруднювачів може спостерігатися помірне занепокоєння щодо здоров'я дуже незначної кількості людей.
Нездоровий для чутливих груп	від 101 до 150	Люди з уразливої групи можуть мати проблеми зі здоров'ям
Нездоровий	від 151 до 200	Кожен може почати відчувати деякі негативні наслідки для здоров'я, а члени уразливих груп можуть відчувати більш серйозні наслідки
Дуже нездоровий	від 201 до 300	Кожен може зазнати більш серйозних наслідків для здоров'я
Небезпечний	більше 300	Може постраждати все населення

Чим вище значення AQI, тим більший рівень забруднення повітря та більша стурбованість здоров'ям. Наприклад, значення AQI 50 являє собою хорошу якість повітря з невеликим потенціалом впливу на здоров'я, тоді як значення AQI понад 300 представляє небезпечну якість повітря. Мета AQI – допомогти зрозуміти, яким чином місцева якість повітря впливає на ваше здоров'я.

1.4 Європейський індекс якості повітря

Європейський індекс якості повітря дозволяє користувачам більше розуміти якість повітря там, де вони живуть, працюють або подорожують. Відображаючи інформацію для Європи, користувачі можуть отримати розуміння якості повітря в окремих країнах, регіонах та містах. Індекс базується на значеннях концентрації п'яти основних забруднюючих речовин, включаючи:

- тверді частинки (PM10);
- дрібнодисперсні речовини (PM2.5);
- озон (O3);
- діоксид азоту (NO2);
- діоксид сірки (SO2).

Діоксид азоту та діоксид сірки, які головним чином утворюються внаслідок викидів з транспортних засобів і промислових процесів, можуть викликати подразнення дихальних шляхів і знижувати функцію легенів. Чадний газ, що утворюється при неповному згорянні вуглеводнів, може блокувати транспортування кисню в організмі, що особливо небезпечно для людей із серцево-судинними захворюваннями.

Цей індекс зазвичай представляється у вигляді числових значень або кольорової шкали, де кожне значення або кольорова категорія відповідає певному рівню якості повітря. Інтерпретація цих кольорових категорій дозволяє громадянам та органам влади зрозуміти, які заходи можуть бути необхідними для захисту здоров'я та навколишнього середовища в залежності від рівня забруднення повітря.

Він відображає потенційний вплив якості повітря на здоров'я, обумовлений забруднювачем, концентрація якого найбільша [9].

Інформація про концентрацію приведена в таблиці 1.6.

Таблиця 1.6 Індекс на основі концентрацій забруднювачів

Забруднювач	Рівень індексу				
	Добре	Непогано	Помірно	Погано	Дуже погано
PM _{2.5}	0-10	10-20	20-25	25-50	50-800
PM ₁₀	0-20	20-35	35-50	50-100	100-1200
NO ₂	0-40	40-100	100-200	200-400	400-1000
O ₃	0-80	80-120	120-180	180-240	240-600
SO ₂	0-100	100-120	200-350	350-500	500-1250

В таблиці 1.7 представлені повідомлення про вплив на здоров'я людини.

Таблиця 1.7 Пов'язані із індексом повідомлення про вплив на здоров'я людини

Індекс якості повітря	Все населення	Уразливі групи населення
Добре	Якість повітря добра	Якість повітря добра
Непогано	Насолоджуйтесь звичайними видами активного відпочинку	Насолоджуйтесь звичайними видами активного відпочинку
Помірно	Насолоджуйтесь звичайними видами активного відпочинку	Насолоджуйтесь звичайними видами активного відпочинку
Погано	Подумайте про зменшення інтенсивних занять на свіжому повітрі, якщо у вас з'являються такі симптоми, як біль в очах, кашель або біль у горлі	Подумайте про зменшення фізичних навантажень, особливо на відкритому повітрі, особливо якщо у вас симптоми.
Дуже погано	Зменшіть фізичні навантаження на свіжому повітрі	Уникайте фізичних занять на свіжому повітрі

За допомогою індексу користувачі можуть планувати свою діяльність, обираючи найбільш сприятливі моменти для перебування на свіжому повітрі, особливо у разі підвищеного рівня забруднення. Це особливо важливо для вразливих груп населення, таких як діти, літні люди та особи з хронічними захворюваннями.

1.5 Аналіз Американських стандартів оцінки якості повітря

Для забезпечення роботи мережі відповідно до умов, вона повинна бути розроблена з урахуванням різних типів ділянок моніторингу. Вони повинні бути здатні інформувати про піковий рівень забруднення повітря, типові рівні в населених пунктах, забруднення повітря, що транспортуються до міста та регіону та за його межами, та рівні забруднення повітря поблизу конкретних джерел. Тож можна виділити шість загальних типів ділянок, які потребують контролю[10]:

- ті, які визначають найвищі концентрації, які очікуються в зоні, що охоплена мережею;
- розміщені для вимірювання типових концентрацій у районах з високою щільністю населення;
- які визначають вплив джерел забруднення або категорій джерел на якість повітря;
- які визначають загальний рівень фонові концентрації;
- для визначення ступеня транспорту забруднюючих речовин серед населених пунктів;
- які розміщені для вимірювання впливу забруднення повітря на видимість, рослинний покрив чи інші впливи на оточуюче середовище.

Для уточнення природи зв'язку між загальними цілями моніторингу, типами ділянок та фізичним розташуванням конкретного монітора визначається концепція просторового масштабу. Метою розміщення моніторів є правильна відповідність просторового масштабу, представленого вибіркою

спостережуваного повітря, та просторового масштабу, найбільш відповідного до типу моніторингової ділянки, забруднювача повітря, що вимірюється, та цілі моніторингу.

Таким чином, масштаб описується з точки зору фізичних розмірів повітряного об'єму, найближчого до місця спостереження, яке має досить однакові фактичні концентрації забруднюючих речовин. Масштаби, що представляють найбільший інтерес для типів моніторингу представлені у таблиці 1.8 [10]:

Таблиця 1.8 Просторові масштаби

Назва масштабу	Опис
Мікро масштаб	Визначає концентрації в об'ємах повітря, пов'язані з розмірами площі від кількох метрів до приблизно 100 метрів
Середній масштаб	Визначає концентрацію, характерну для районів розміром до декількох міських кварталів з розмірами від 100 метрів до 0,5 кілометра
Середній масштаб	Визначає концентрацію, характерну для районів розміром до декількох міських кварталів з розмірами від 100 метрів до 0,5 кілометра
Районний масштаб	Визначає концентрації в межах деякої розширеної площі міста, яка має відносно рівномірне використання земель із розмірами в межах від 0,5 до 4 кілометрів
Міський масштаб	Визначає концентрації в межах міста, порядку від 4 до 50 кілометрів

Регіональний масштаб	Визначає, як правило, сільську територію з досить однорідною географією без великих джерел забруднення і простягається від десятків до сотень кілометрів
Національний та глобальний масштаби	Представляють концентрації, що характеризують територію країни та земну кулю в цілому

Правильне розміщення приладу вимагає уточнення мети моніторингу, типів ділянок, необхідних для досягнення мети, а потім бажаного просторового масштабу. Наприклад, розглянемо випадок, коли метою є визначення відповідності NAAQS (National Ambient Air Quality Standards) [11] шляхом визначення максимальних концентрацій озону для даної території. Такі ділянки, швидше за все, будуть відповідати приміському житловому району, де люди, ймовірно, проводять більше часу на відкритому повітрі. Ділянки, розташовані в цих районах, швидше за все, представляють міський масштаб. У цьому прикладі фізичне розташування визначали, враховуючи схеми викидів, активність населення та метеорологічні характеристики, що впливають на утворення та дисперсію озону. Таким чином, просторовий масштаб не використовувався в процесі відбору, а був результатом розташування ділянки.

Фізичне розташування ділянки визначається за рахунок спільного розгляду як основної цілі моніторингу, так і типу місця моніторингу. Наприклад, для визначення концентрацій $PM_{2.5}$ (particulate matter), які характерні для географічної області, що має відносно високі концентрації $PM_{2.5}$, більш доцільною є ділянка районного масштабу. Такий майданчик, ймовірно, буде розташований у житловому чи бізнес-районі, що має високу загальну щільність викидів $PM_{2.5}$, але не в безпосередній близькості від жодного окремого домінуючого джерела. Зауважимо, що в цьому прикладі бажаний масштаб був

важливим фактором при визначенні фізичного розташування ділянки моніторингу.

В будь-якому випадку класифікація мережі за її типом та просторовим масштабом є необхідною і сприятиме інтерпретації даних моніторингу (наприклад, публічна звітність, відповідність NAAQS або підтримка досліджень) [12].

1.6 Огляд наукових праць пов'язаних з оцінкою якості повітря

На сьогоднішній день проведено багато досліджень щодо методів оцінки якості повітря та факторів, які найбільше впливають на якість повітря. Існує багато досліджень, які намагаються вирішити проблему нестачі оперативних даних [13-18]. У роботі [19] автори порівняли локальні та глобальні методи розрахунку індексу якості повітря на прикладі міста Дакка. Результати цього дослідження свідчать про те, що локальний AQI більш точно відображає якість повітря у відповідній досліджуваній зоні за певних реальних сценаріїв. У роботі [20] автори класифікували та проаналізували дані моніторингу викидів шкідливих речовин у Львівській області та створили серію екологічних карт на основі даних моніторингу атмосферного повітря у Львівській області. У роботі [21] автори пропонують метод прогнозування концентрацій забруднення повітря на неспостережуваних територіях з використанням інструментів просторової інтерполяції в ГІС. Це дає цінну інформацію про дисперсію забруднення повітря. У дослідженні [22] представлені нещодавні просторово-часові зміни в ІЗА та якості повітря в головному міському районі Чунціна. Повний набір даних було реконструйовано з використанням нового методу (LRTC-TNN) для інтерполяції відсутніх значень. У роботі [23] автори порівнюють модель, яка використовує найвищий суб-індекс, що залежить від стандартів забруднювачів USEPA, з моделлю, яка включає ваги всіх забруднювачів, як модель агрегованого індексу якості повітря (AAQI). AAQI є комплексним показником якості повітря і є більш корисним в управлінні навколишнім середовищем, оскільки він представляє

декілька забруднювачів якості повітря. У роботі [24] автори зосереджуються на стратегії управління екологічною екосистемою за допомогою алгоритму штучного інтелекту (ШІ) та досліджують кореляцію між якістю повітря та метеорологією. Експериментальний аналіз показує, що середня температура має позитивну кореляцію з AQI, тоді як відносна вологість і швидкість вітру мають негативну кореляцію з AQI. Крім того, помилка прогнозування AQI за допомогою запропонованої моделі RF + BP + GA не перевищує 0,32, що свідчить про відмінну відповідність фактичних значень AQI. У роботі [25] автори досліджують, що сільськогосподарський сектор має значний вплив на індекс якості повітря, використовуючи систему AgrImOnIA (Agriculture Impact On Italian Air). Ця система оцінює роль тваринницького сектору в якості повітря в регіоні Ломбардія і дозволяє проводити порівняння з іншими європейськими регіонами.

Існує багато глобальних служб моніторингу якості повітря, які розроблені та функціонують по всьому світу, але більшість з них не призначені для збору даних, необхідних для оцінки впливу населення на чутливі забруднювачі повітря. Проблема моніторингу якості повітря є предметом багатьох наукових досліджень. У роботі [17] автори надають всебічний огляд сучасного стану справ у сфері якості міського повітря та висвітлюють відкриті питання та дослідницькі виклики у вирішенні проблеми забруднення повітря у містах. Деякі дослідження застосовують машинне навчання для картографування та прогнозування якості повітря. У роботі [26] автори застосували техніку машинного навчання з часовими рядами записів забруднення твердими частинками для прогнозування та розробки карти вразливості до забруднення твердими частинками. Випадковий ліс був застосований для перевірки та візуалізації взаємозв'язків між твердими частинками та різними незалежними змінними. У роботі [27] зроблено спробу створити мультимодальну модель машинного навчання для прогнозування показників якості повітря там, де немає станцій моніторингу. У статті [15] пропонується вдосконалений метод оцінки якості міського повітря, який об'єднує просторово-часові дані з різних джерел. Цей метод інтегрує метеорологічні дані з даними про соціальну активність у місті для формування

комплексного набору екологічних даних. У дослідженні [18] досліджуються зміни індексу якості повітря під час різних локдаунів, спричинених COVID-19, з використанням супутникових даних. Порівняння інструментів моделювання [28, 29] описує можливості та переваги використання CALPUFF та інших моделей, таких як AERMOD та CDF. У статті [30] автори наводять результати моделювання якості повітря за допомогою програми AERMOD (програмне забезпечення BREEZE AERMOD): результати моделювання показали, що в досліджуваному районі спостерігається висока концентрація CO. У роботі [31] показано розсіювання чотирьох забруднюючих речовин (SO₂, CO, NO₂ та PM_{2.5}), що викидалися з нафтопереробного заводу Даура в Багдаді влітку 2013 року та взимку 2013-2014 років за допомогою системи моделювання CALPUFF: були розраховані погодинні обсяги викидів забруднюючих речовин на основі фактичних обсягів палива, спожитого на нафтопереробному заводі.

Однак більшість запропонованих методологій моніторингу якості повітря потребують подальшої оцінки та перегляду, перш ніж їх можна буде використовувати на практиці в інших містах.

1.7 Висновок

В результаті проведеного аналізу існуючих стандартів розглянуті індекси якості повітря такі як американські, європейські та норми ВООЗ. Були розглянуті розроблені програмні забезпечення для пристроїв моніторингу такі як Aeroqual, BL-VIEW, та програмне забезпечення для моделювання розподілу заводів: BREEZE, AERMOD View, CALPUFF View, ADMS-Urban.

Основним недоліком цього програмного забезпечення є висока вартість, необхідність задання великої кількості параметрів, детальних метеорологічних та даних рельєфу. Більшість з наведеного ПЗ дає лише інформацію про локальний стан якості повітря або збирають дані лише з власних станцій моніторингу. Також інтерфейс цих програм є неінтуїтивним.

В зв'язку з цим виникає потреба в створенні інформаційної система, яка має відповідати наступним критеріям:

- легко розгортатися на різних типах машин;
- збирати та серіалізувати різні типи даних;
- запускати завдання у фоновому режимі, в іншому процесі;
- візуалізувати результати.

Також необхідно сформулювати математичну модель для оцінки якості повітря.

РОЗДІЛ 2. МАТЕМАТИЧНІ МОДЕЛІ ОЦІНКИ ЯКОСТІ ПОВІТРЯ

Математична модель якості повітря включає в себе різні параметри та залежності, щоб кількісно оцінити рівень забруднення атмосферного повітря. Оцінка якості повітря зазвичай враховує концентрації різноманітних забруднюючих речовин, таких як оксиди азоту (NO_x), сірковуглець (SO_x), тверді частки PM₁₀ і PM_{2.5}, бензапірен, вуглеводні та інші. Модель повинна враховувати фізичні та хімічні процеси, такі як дифузія, реакція в атмосфері та осадження забрудників, що впливають на їх розподіл та стан. Невід'ємним фактором впливу є метеорологічні умови, такі як швидкість вітру, температура, вологість та інші, що можуть впливати на розповсюдження забруднюючої речовини.

Результатом роботи математичної моделі якості повітря є надання оцінки впливу концентрації забрудників на здоров'я людини у вигляді індексу якості повітря.

Однією з базових математичних моделей може бути рівняння зваженої суми концентрацій різних забрудників, де кожен забрудник має свій ваговий коефіцієнт, що відображає його вплив на загальну якість повітря. Така модель може мати вигляд:

$$AQI = \omega_1 \cdot C_1 + \omega_2 \cdot C_2 + \dots + \omega_n \cdot C_n, \quad (2.1)$$

де:

- AQI – індекс якості повітря (Air Quality Index);
- C_1, C_2, \dots, C_n – концентрації різних забрудників;
- $\omega_1, \omega_2, \dots, \omega_n$ – вагові коефіцієнти, що відображають важливість кожного забрудника.

Цей підхід дозволяє враховувати вагомість різних забрудників, але вимагає визначення вагових коефіцієнтів на основі екологічних та медичних даних.

Однак для більш точних прогнозів і аналізу може використовуватися більш складна модель, що враховує характеристики атмосферного руху, взаємодію різних забрудників та їхні впливи на здоров'я людини. Такі моделі можуть

включати у себе різні диференціальні рівняння та статистичні методи для аналізу даних.

2.1 Супутникові знімки місії Sentinel-5P

Супутникові знімки допомагають визначити великі райони забруднення, спричинені пожежами, пиловими або піщаними бурями, виверженнями вулканів, великими промисловими джерелами або транспортуванням техногенного забруднення з інших регіонів. Менші джерела, такі як невеликі підприємства чи місцеві дороги, не будуть видимі на супутникових знімках [32].

Супутникове дистанційне зондування забезпечує дані про якість повітря. Просторові розриви ресурсів наземного моніторингу неминучі, а супутникові дані здатні заповнювати інформацію про якість повітря в районах без наземних станцій. Існують численні супутникові дані, які можуть вказувати на критерії забруднювачів повітря (наприклад, $PM_{2.5}$ та NO_2) та парникові гази (наприклад, CH_4 та CO_2). Вибір супутникових даних для вирішення конкретних питань якості повітря залежить від точності даних та просторового масштабу [33].

Одним із джерелом даних для оцінки якості є дані з супутникових знімків. Моніторинг якості повітря за допомогою супутникових знімків є ефективним методом, який дозволяє отримувати глобальну та об'єктивну інформацію про розподіл забруднюючих речовин у повітрі. Для прикладу, Sentinel-5P — це спутник, який належить до програми Copernicus Європейського космічного агентства (ESA). Запущений у 2017 році, він обладнаний інструментом Tropomi (TROPOspheric Monitoring Instrument), який вимірює різноманітні гази в атмосфері, такі як діоксид азоту, оксид сірки, метан, формальдегід та інші, що дозволяє в реальному часі визначати якість повітря. Більш раніше (у 2004 році) Національною аеронавтичною і космічною адміністрацією США (NASA) запустила місію Aura, яка є частиною програми Earth Observing System (EOS). Aura обладнана інструментами, які дозволяють визначати склад атмосфери

Землі, включаючи вимірювання озону, оксидів азоту, сірки, вуглецю та інших газів, які впливають на якість повітря.

Ось опис та переваги цього методу:

- супутники обладнані спеціальними датчиками, які реєструють різні види електромагнітного випромінювання, що виходить від Землі. Це включає видиме світло, інфрачервоне та мікрохвильове випромінювання;

- за допомогою спектрального аналізу супутникові системи можуть розрізняти різні типи поверхонь та визначати характеристики атмосферних газів та аерозолів;

- супутниковий моніторинг дозволяє проводити ретроспективний аналіз, оцінювати тенденції забруднення повітря та виявляти зони особливого інтересу для подальших досліджень;

- супутникові системи забезпечують можливість моніторингу якості повітря в будь-якій точці світу. Це особливо корисно для виявлення забруднення на великих територіях або у віддалених районах;

- спостереження в реальному часі. Деякі супутники, такі як ті, які входять до системи Copernicus в рамках програми Global Monitoring for Environment and Security (GMES), можуть надавати дані в режимі реального часу, що важливо для оперативного реагування на екологічні події;

- дистанційний моніторинг. Можливість моніторингу з великої відстані дозволяє уникнути проблем, пов'язаних із здійсненням вимірювань на місці, особливо у важкодоступних або небезпечних регіонах;

- карта розподілу забруднюючих речовин. Супутниковий моніторинг надає можливість створення деталізованих карт розподілу забруднюючих речовин, визначення їхніх джерел та динаміки розповсюдження.

Хоча супутниковий моніторинг має свої переваги, важливо враховувати, що він може бути менш точним у порівнянні з наземними станціями щодо конкретних параметрів, таких як концентрація забруднюючих речовин у нижніх шарах атмосфери. Також, використання обох методів у поєднанні може забезпечити комплексний підхід до моніторингу якості повітря.

2.2 Опис даних зі станції наземного моніторингу

Станції наземного моніторингу якості повітря – це спеціально обладнані місця, розташовані у різних частинах міста або країни, де здійснюється систематичний збір даних про забруднення повітря. Ці станції зазвичай вимірюють рівні різних забруднювачів, таких як оксиди азоту, оксиди сірки, вуглеводні, тверді частки та інші шкідливі речовини.

Надання даних про якість повітря може здійснюватися через різноманітні джерела, як державні (національні системи моніторингу якості повітря, управляють якими державні органи), так і приватні (приватні компанії мають свої власні станції моніторингу та надавати доступ до цих даних через вебсайти або мобільні додатки). Це може бути здійснено напрямку зі спеціалізованих станцій моніторингу повітря або шляхом доступу до відповідних сайтів екомоніторингу. Розглянемо найбільші джерела та дані які ними надаються. Нижче наведено приклад таких джерел даних.

World Air Quality Index (WAQI) надає дані про якість повітря у вигляді індексу якості повітря (AQI) для різних міст та регіонів по всьому світу. Дані про AQI зазвичай включають наступну інформацію: Місце та час вимірювання: Це включає географічні координати місця, де проводилося вимірювання, а також дату та час, коли дані були зібрані. Рівні забруднювачів: WAQI надає інформацію про рівні різних забруднювачів повітря, таких як PM_{2.5} (тверді частки з діаметром менше 2,5 мікрометрів), PM₁₀, NO₂ (діоксид азоту), SO₂ (діоксид сірки), O₃ (озон), одиниці вимірювання мкг/м³. Індекс якості повітря (AQI): AQI – це числове значення, яке відображає загальну якість повітря на основі рівнів різних забруднювачів. Це зазвичай числове значення від 0 до 500, де нижчі значення вказують на кращу якість повітря, а вищі – на гіршу. Категорії якості повітря: AQI також може бути відображений у вигляді категорій якості повітря, таких як «Добре», «Задовільно», «Незадовільно», «Погано», «Дуже погано» та «Небезпечно». Ці категорії допомагають людям зрозуміти, як впливає якість повітря на їхнє здоров'я [34].

Таблиця 2.1 Таблиця рівнів забруднень WAQI, наслідки
для здоров'я та застереження

Індекс якості повітря	Вразливі групи населення	Рекомендації
Добре (0 - 50)	Якість повітря вважається задовільною	Якість повітря добра
Помірно (51-100)	Якість повітря прийнятна; Крім людей з надзвичайно чутливих до забруднення повітря	Насолоджуйтесь звичайними видами активного відпочинку
Шкідливо для чутливих груп (101 – 150)	Члени чутливих груп можуть відчувати наслідки для здоров'я	Насолоджуйтесь звичайними видами активного відпочинку
Нездорове (151-200)	Кожна людина може почати відчувати наслідки для здоров'я	Зменшення фізичних навантажень, особливо на відкритому повітрі
Дуже шкідливо для здоров'я (201-300)	Попередження про невідкладні стани для здоров'я. Частіше страждає все населення	Необхідне обмеження або повне уникнення навантаження на відкритому повітрі
Небезпечно (300+)	Кожна людина відчуватиме більш серйозні наслідки для здоров'я	Необхідно уникати навантажень на свіжому повітрі

Кожна така платформа надає обмежений перелік забруднювачів. Наприклад, одним із джерел моніторингу радіаційної обстановки на території України використовується Головний Центр Спеціального контролю [35]. Організація надає інформацію про радіаційний стан в 6-х точках з інтервалом у 6 годин починаючи з 2017 року.

Серед переваг використання наземних станцій є:

- висока точність вимірювань: Станції моніторингу можуть забезпечити точні дані про рівні забруднення повітря в конкретних місцях, оскільки вони знаходяться безпосередньо на місці;

- реальний час: Вимірювання проводяться в реальному часі, що дозволяє оперативно реагувати на зміни та вживати заходів для поліпшення якості повітря;

- детальні дані: Станції можуть забезпечувати детальні дані про різні типи забруднень та їх джерела.

В той же час існують і певні недоліки:

- обмежена просторова покриття: Мережа станцій моніторингу може бути обмежена географічно, що робить важко визначення якості повітря віддалених або важкодоступних місць;

- високі витрати на обслуговування: Підтримка та обслуговування станцій моніторингу може бути дорогими, особливо при необхідності постійного моніторингу великої кількості параметрів;

- чутливість до впливу зовнішніх факторів: Результати моніторингу можуть бути спотворені впливом погодних умов або інших зовнішніх факторів.

2.3 Опис метеоданих з NOAA (National oceanic and atmospheric administration)

Для отримання початкових даних про погоду було обрано систему GFS.

Global Forecast System (GFS) — глобальна система чисельного прогнозування погоди, що містить глобальну комп'ютерну модель та варіаційний аналіз, що виконується Національною метеорологічною службою США (NWS).

Математична модель запускається чотири рази на день і дає прогнози на 16 днів наперед, але зі зменшенням просторової роздільної здатності через 10 днів. Майстерність прогнозування зазвичай зменшується з часом (як і в будь-якій чисельній моделі прогнозування погоди). Це одна з провідних синоптичних

моделей середнього діапазону загального користування. Модель GFS використовується для прогнозування різних погодних явищ, включаючи температуру повітря, вологість, опади, вітер і інші. Ці прогнози надають цінну інформацію для планування дій у великому діапазоні галузей, від сільського господарства до авіації, від енергетики до туризму.

Модель GFS (Global Forecast System) представляє собою спектральну модель з горизонтальною роздільною здатністю приблизно 13 км протягом перших 10 днів та 27 км від 240 до 384 годин (16 днів). Ця модель охоплює широкий простір, здатний передбачати погодні умови на глобальному рівні. По вертикалі, модель розділена на 127 шарів і простягається від поверхні землі до мезопаузи, що приблизно на висоті 80 км від поверхні землі.

До 2015 року модель GFS відставала від точності інших глобальних моделей погоди, що стало особливо помітно під час передбачення вихід урагану Сенді в море. Модель GFS неправильно передбачила рух урагану до чотирьох днів до його наближення до берега, у той час як модель Європейського центру середньострокових прогнозів погоди вдало передбачила сходження на сушу через 7 днів. Ця помилка була обумовлена обмеженнями обчислювальних ресурсів Національної метеорологічної служби (NWS). У відповідь на це NWS придбала нові суперкомп'ютери, що значно збільшило обчислювальну потужність з 776 терафлопс до 5,78 петафлопсів. Таким чином, обчислювальні можливості моделі GFS значно покращилися.

Станом на 19 липня 2017 року модель GFS була оновлена, а нова версія, на відміну від попередньої, більш точно враховує фактори, такі як коливання Меддена – Джуліана та повітряний шар Сахари. У 2018 році обчислювальна потужність знову була збільшена, до 8,4 петафлопсів, що дозволило моделі GFS ще точніше передбачати погоду в різних регіонах світу. Нещодавно NWS також випробувала потенційну модель заміни FIM у пошуках ще кращих результатів, але відмовилася від неї приблизно в 2016 році через відсутність істотних поліпшень порівняно з моделлю GFS.

У 2019 році внаслідок недавнього десятикратного збільшення обчислювальної потужності було заплановано оновлення моделі GFS, що дозволить збільшити її горизонтальну роздільну здатність до 9 км (5,6 миль) та 128 шарів до 16 днів у порівнянні з поточним пробігом 13 км (8,1 миль) і 64 шари за 10 днів.

22 березня 2021 року NOAA оновила модель GFS, поєднавши її з глобальною хвильовою моделлю WaveWatch III, яка збільшить роздільну здатність GFS з 64 до 127 вертикальних рівнів, одночасно розширивши вікно прогнозування з 10 до 16 днів. Це залишило деяких метеорологів сподіватися, що оновлення GFSv16 буде достатньо, щоб скоротити розрив у точності за допомогою моделі ECMWF, яка вважалася найточнішою глобальною моделлю погоди на той час [35].

Шари GFS 0.25 відносяться до вертикальної структури атмосфери, яку модель GFS використовує для прогнозування погоди. Нижче наведений опис основних шарів, що використовуються в GFS 0.25 та полів, з яких вони складаються.

Погода значною мірою визначається барометричним тиском [37]. Ось список шарів атмосферного тиску на різних висотах, які можна знайти в моделі GFS:

- поверхневий шар (Surface Layer): Атмосферний тиск на рівні поверхні Землі;
- 1000 гектопаскаль(гПа) або мілібар (мб): Це рівень близької до морської рівнини;
- 925 гектопаскаль: Висота близько 750-800 метрів над рівнем моря;
- 850 гектопаскаль: Висота близько 1-1,5 кілометрів над рівнем моря;
- 700 гектопаскаль: Висота близько 3-3,5 кілометрів над рівнем моря;
- 500 гектопаскаль: Висота близько 5-5,5 кілометрів над рівнем моря;
- 300 гектопаскаль: Висота близько 8-9 кілометрів над рівнем моря.
- 200 гектопаскаль: Висота близько 10-12 кілометрів над рівнем моря;

- 150 гектопаскаль: Цей рівень розташований на висоті приблизно 16 км над рівнем моря. Нижче цього рівня починається стратосфера;
- 100 гектопаскаль: Цей рівень розташований на висоті близько 17-18 км над рівнем моря. Це вже верхній шар стратосфери;
- 50 гектопаскаль: Цей рівень розташований на висоті приблизно 24-25 км над рівнем моря. Це знаходиться в мезосфері;
- 20 гектопаскаль: Цей рівень розташований на висоті близько 30-32 км над рівнем моря. Це вже досить високо в мезосфері;
- 10 гектопаскаль: Цей рівень розташований на висоті близько 40-42 кілометрів над рівнем моря. Це вже вище верхньої межі стратосфери;
- 5 гектопаскаль: Цей рівень розташований на висоті приблизно 46-48 кілометрів над рівнем моря. Це знаходиться в мезосфері;
- 1 гектопаскаль: Цей рівень розташований на висоті близько 52-54 кілометрів над рівнем моря. Це вже високо в мезосфері;
- 0.1 гектопаскаль: Цей рівень розташований на висоті близько 80-82 кілометрів над рівнем моря. Це вже вище межі мезосфери та в межах термосфери;
- 0.01 гектопаскаль: Цей рівень розташований на висоті близько 100-102 кілометрів над рівнем моря. Це вже дуже високо в термосфері.

Зазначимо, що атмосферний тиск на рівнях нижче 200 гектопаскаль рідко використовується для метеорологічного аналізу та прогнозування, оскільки ці рівні знаходяться в верхній частині стратосфери та мезосфери, де атмосферний тиск дуже низький, а погодні явища майже відсутні. Атмосферний тиск на рівнях нижче 20 гектопаскаль знаходиться в дуже високих шарах атмосфери, в межах мезосфери та верхньої стратосфери [38]. На таких висотах атмосферний тиск надзвичайно низький, і погодні явища майже відсутні. Атмосферний тиск на рівнях нижче 1 гектопаскаль знаходиться в екстремально високих шарах атмосфери, де атмосфера стає дуже розрідженою і практично відсутня [39]. На таких висотах вже немає практичної атмосфери для більшості метеорологічних

явищ. В таблиці 2.2 вказані параметри, що вимірюються та розраховуються для данного діапазону шарів.

Таблиця 2.2 Таблиця полів шарів з рівнем атмосферного тиску від 0.01 гектопаскаль до 70 гектопаскаль

Параметр	Опис	Одиниці вимірювання
HGT	Geopotential Height, Геопотенціальна висота	gpm
TMP	Temperature, Температура	К
RH	Relative Humidity, Відносна вологість	%
SPFH	Specific Humidity, Питома вологість	кг/кг
VVEL	Vertical Velocity (Pressure), Вертикальна швидкість (тиск)	Па/с
DZDT	Vertical Velocity (Geometric), Вертикальна швидкість (геометрична)	м/с
UGRD	U-Component of Wind, U-компонент вітру	м/с
VGRD	V-Component of Wind, V-компонент вітру	м/с
ABSV	Absolute Vorticity, Абсолютна завихреність	1/с
O3MR	Ozone Mixing Ratio, Коефіцієнт змішування озону	кг/кг

Поверхневий шар (Surface Layer) – це верхній шар атмосфери, що безпосередньо знаходиться над поверхнею Землі. Поверхневий шар (Surface Layer) в моделі GFS (Global Forecast System) представляє собою перший шар атмосфери, що знаходиться безпосередньо над поверхнею Землі. Цей шар є особливо важливим для прогнозування погоди, оскільки в ньому відбуваються багато ключових метеорологічних процесів та фізичних явищ. Основні характеристики поверхневого шару включають:

— температура повітря. Поверхневий шар відображає температуру повітря на рівні поверхні землі. Це важлива характеристика для прогнозування температурних умов та комфорту для людей і тварин;

— вологість. Вологість повітря в поверхневому шарі визначає, наскільки насиченим водяною парою є повітря. Вологе повітря може призвести до утворення хмар, туману та опадів;

— вітер. Напрямок та швидкість вітру на рівні поверхні землі важливі для прогнозування погодних умов, мікроклімату та розповсюдження забруднювачів атмосфери;

— атмосферний тиск. Тиск, який вимірюється на рівні поверхні землі, відображає ступінь стиснення або розтягнення повітря в атмосфері, що може впливати на погоду та стійкість атмосфери;

— опади. Поверхневий шар також включає в себе інформацію про опади, такі як дощ, сніг, град, які можуть впливати на погоду та кліматичні умови.

Перелік параметрів що включає в собі поверхневий шар описані в таблиці 2.3.

Таблиця 2.3 Таблиця полів поверхневого шару (Surface Layer)

Параметр	Опис	Одиниці вимірювання
PRES	Pressure, Тиск	Па
HGT	Geopotential Height, Геопотенціальна висота	gpm
PRATE	Precipitation Rate, Кількість опадів	кг/м ² /с
VEG	Vegetation, Рослинність	%
SOTYP	Soil Type, Тип ґрунту	-
LFTX	Surface Lifted Index, Індекс підняття поверхні	К
CAPE	Convective Available Potential Energy, Конвективна доступна потенційна енергія	Дж/кг
CIN	Convective Inhibition, Конвективне гальмування	Дж/кг

Поверхневий шар є ключовим елементом для прогнозування погоди на короткий термін і він є основою для подальшого аналізу та прогнозування погодних умов на більш великих висотах та глибині атмосфери.

Рівень 1000 мб (мілібар) або 1000 гПа (гектопаскаль) в моделі GFS (Global Forecast System) відповідає атмосферному тиску, який близький до середнього тиску на рівні морської поверхні. Вважається, що на рівні морської рівнини тиск становить близько 1013.25 гектопаскаль (стандартне атмосферне тиск). Однак, у моделі GFS значення тиску на рівні 1000 гектопаскаль може відрізнятися від цього стандартного значення через метеорологічні умови в даний момент часу та розташування. На практиці, значення тиску на рівні 1000 гектопаскаль в моделі GFS використовується як основна вихідна точка для аналізу та прогнозування погоди. Воно дозволяє встановити базовий рівень атмосферного тиску для подальшого аналізу вертикальних змін тиску на різних висотах та прогнозування погодних умов. Отже, рівень 1000 гектопаскаль в моделі GFS допомагає аналізувати та передбачати погодні умови, враховуючи базовий рівень атмосферного тиску, який близький до середнього значення на рівні морської рівнини.

Рівень 925 мб (мілібар) або 925 гПа (гектопаскаль) в моделі GFS (Global Forecast System) відповідає атмосферному тиску на висоті приблизно 750-800 метрів над рівнем моря. Цей рівень використовується для аналізу та прогнозування погоди на висотах над поверхнею землі. Атмосферний тиск зменшується з висотою через менше навантаження повітря від верхніх шарів атмосфери. Таким чином, тиск на рівні 925 гектопаскаль є значно меншим, ніж на рівні морської рівнини. Він може варіюватися в залежності від метеорологічних умов та географічного розташування. На практиці, значення тиску на рівні 925 гектопаскаль в моделі GFS використовується для аналізу та передбачення погоди на певних висотах над рівнем моря, де відбуваються важливі атмосферні процеси. Це може включати зміни в погодних умовах, температурі повітря, вітрових умовах та інших факторах, які впливають на погоду у цьому регіоні.

Рівень 850 мб (мілібар) або 850 гПа (гектопаскаль) в моделі GFS (Global Forecast System) відповідає атмосферному тиску на висоті приблизно 1-1,5 км над рівнем моря. Цей рівень використовується для аналізу та прогнозування погоди

на середніх висотах над поверхнею землі. На цій висоті атмосферний тиск значно менше, ніж на рівні морської рівнини, оскільки повітряна маса зменшується з висотою. Він використовується для аналізу вертикального розподілу тиску та взаємодії повітряних мас на середніх висотах над поверхнею землі. Значення тиску на рівні 850 гектопаскаль може бути важливим для прогнозування погоди, оскільки на цій висоті зазвичай розташовані ключові шари атмосфери, в яких відбуваються важливі метеорологічні процеси. Наприклад, верхні частини хмар, рівні переходу температури повітря від плюсових до мінусових значень, та різні зони вітрів можуть бути визначені на цій висоті. Отже, рівень 850 мб допомагає аналізувати та передбачати погодні умови на середніх висотах над поверхнею землі, що важливо для розуміння метеорологічних явищ та їх впливу на погоду в конкретному регіоні.

Рівень 700 мб (мілібар) або 700 гПа (гектопаскаль) в моделі GFS (Global Forecast System) відповідає атмосферному тиску на висоті приблизно 3-3,5 км над рівнем моря. Цей рівень використовується для аналізу та прогнозування погоди на середніх висотах над поверхнею землі. На цій висоті значення атмосферного тиску вже досить низьке, що свідчить про подальше зменшення атмосферного тиску зі зростанням висоти. Рівень 700 гектопаскаль є важливим для аналізу динаміки атмосфери на середніх висотах та вивчення вертикальних процесів, таких як утворення хмар, розподіл вологості, вертикальний рух повітря та розвиток погодних систем. Аналіз значень тиску на рівні 700 гектопаскаль може допомогти в розумінні руху повітря та динаміки атмосфери на середніх висотах, що в свою чергу впливає на погодні умови. Наприклад, вертикальні атмосферні процеси на цій висоті можуть впливати на формування та рух погодних фронтів, грозових хмар, а також впливати на літаки, що перебувають у повітрі. Отже, рівень 700 гектопаскаль в моделі GFS є важливим для аналізу та передбачення погоди на середніх висотах над поверхнею землі та допомагає зрозуміти вертикальну структуру атмосфери та її вплив на погодні явища.

В шарах з рівнем атмосферного тиску від 100 гектопаскаль до 1000 гектопаскаль збираються та розраховуються характеристики що наведені у таблиці 2.4.

Таблиця 2.4 Таблиця полів шарів з рівнем атмосферного тиску від 100 гектопаскаль до 1000 гектопаскаль

Параметр	Опис	Одиниці вимірювання
TMP	Temperature, Температура	К
RH	Relative Humidity, Відносна вологість	%
SPFH	Specific Humidity, Питома вологість	кг/кг
VVEL	Vertical Velocity (Pressure), Вертикальна швидкість (тиск)	Па/с
DZDT	Vertical Velocity (Geometric) , Вертикальна швидкість (геометрична)	м/с
UGRD	U-Component of Wind, U-компонент вітру	м/с
VGRD	V-Component of Wind, V-компонент вітру	м/с
ABSV	Absolute Vorticity , Абсолютна вихровість	1/с
CLWMR	Cloud Mixing Ratio, Коефіцієнт перемішування хмар	кг/кг
ICMR	Ice Water Mixing Ratio , Коефіцієнт змішування льоду та води	кг/кг
RWMR	Rain Mixing Ratio, Коефіцієнт змішування дощу	кг/кг
SNMR	Snow Mixing Ratio, Коефіцієнт перемішування снігу	кг/кг
GRLE	Graupel , Граупель (утворюється, коли сніг в атмосфері стикається з переохолодженою водою)	кг/кг
O3MR	Ozone Mixing Ratio , Коефіцієнт змішування озону	кг/кг

Шар 1 Hybrid Level у моделі GFS 0.25 є одним з ключових компонентів, який надає важливу інформацію для аналізу та прогнозування погодних умов на великій висоті в атмосфері.

Діапазон висот у шарі 1 Hybrid Level моделі GFS 0.25 зазвичай охоплює велику вертикальну область атмосфери, починаючи з нижніх шарів стратосфери і закінчуючи верхніми шарами тропосфери. Точний діапазон може змінюватися в залежності від конкретної реалізації моделі та обраного розділення по вертикалі. Однак зазвичай висоти у шарі 1 Hybrid Level можуть бути приблизно від 10 до 70 кілометрів над рівнем моря. Точні межі діапазону можуть варіюватися в залежності від конфігурації моделі та специфікацій модельного рану. Враховуючи, що модель GFS 0.25 має просторове розділення 0.25 градуса, шар 1 Hybrid Level може мати різний вертикальний розділ в залежності від цілей моделювання та потреб дослідження.

Основними параметрами, що включаються в шар 1 Hybrid Level, є геопотенціальна висота і атмосферний тиск. Всі параметри наведені в таблиці 2.5. Ця комбінація дозволяє отримати комплексну інформацію про стан атмосфери на великій висоті, враховуючи як вертикальний розподіл тиску, так і енергію повітря в геопотенціальних метрах.

Таблиця 2.5 Таблиця полів шару 1 Hybrid Level

Параметр	Опис	Одиниці вимірювання
CLWMR	Коефіцієнт змішування хмари	кг/кг
ICMR	Коефіцієнт змішування крижаної води	кг/кг
RWMR	Коефіцієнт змішування дощу	кг/кг
SNMR	Коефіцієнт змішування снігу	кг/кг
GRLE	Graupel, Граупель (утворюється, коли сніг в атмосфері стикається з переохолодженою водою)	кг/кг

Ці дані розташовані у вільному доступі, а математична модель WRF має спеціальні бібліотеки для обробки саме файлів з моделі GFS. Саме тому вони найбільше підходять у рамках задачі.

2.4 Моделювання дисперсії газів за допомогою Ейлерової моделі

Основна ідея ейлерових моделей полягає в чисельному розв'язанні рівняння атмосферного транспорту в фіксованих координатах:

$$\frac{\partial c}{\partial t} = -\nabla \cdot (c\vec{v}) + S_c + \nabla \cdot (D_c \nabla c), \quad (2.2)$$

де \vec{v} – вектор вітру, S_c – умови джерела, а D_c – коефіцієнт дифузії. Математично, це часткове диференціальне рівняння другого порядку, вирішення якого при наявності відповідних початкових та граничних умов надає просторово-часову еволюцію концентрації частки.

У випадку одного джерела при стаціонарних (тобто $\frac{\partial c}{\partial t} = 0$) викидах і метеорологічних умовах, член джерела зазвичай розглядається як гранична умова навітряного вітру. Для середньої швидкості вітру $\bar{u}(x, y, z)$, що дме в напрямку невід'ємної осі x , у навітряній площині (y, z) застосовується наступна гранична умова:

$$c = \frac{Q}{\bar{u}(0, y_s, z_s + \Delta h)} \delta(z - z_s - \Delta h) \delta(y - y_s), \quad (2.3)$$

де Q – швидкість викиду забруднювача, z_s – фізична висота джерела, розташованого в точці $(0, z_s)$, Δh – висота підйому факела, а δ – оператор Кронекера [40].

Ейлерові моделі можуть бути ефективними з точки зору обчислень, оскільки вони використовують глобальну сітку для представлення атмосфери та забруднюючих речовин. Це дозволяє швидко обчислювати розподіл забруднюючих речовин на великій території. Для широкомасштабних прогнозів забруднення повітря Ейлерові моделі можуть бути досить точними, оскільки вони дозволяють враховувати широкомасштабні потоки повітря та їхні взаємодії з забруднюючими речовинами. Також вони бути корисними для аналізу трендів забруднення повітря на великих просторових масштабах та в довготривалій перспективі.

Але у цих моделей є також певні недоліки серед яких недостатня деталізація локальних процесів - моделі можуть не давати достатньої деталізації локальних процесів, таких як розподіл забруднюючих речовин вузьких узорах або вулицях. Варто зазначити, що Ейлерові моделі часто не враховують ефекти дифузії від конкретних джерел забруднення, що може призводити до недооцінки концентрацій забруднюючих речовин у місцях найближчих до джерел забруднення. Для точного моделювання забруднення повітря Ейлерові моделі часто потребують великого обсягу вхідних даних про метеорологічні умови та викиди забруднюючих речовин.

Узагальнюючи, Ейлерові моделі можуть бути корисними для прогнозування та аналізу загальних трендів забруднення повітря на великих просторових масштабах, але вони можуть бути менш ефективними для деталізованого моделювання локальних процесів та взаємодій.

2.5 Моделювання дисперсії газів за допомогою Гаусових моделей

Більшість короткодисперсних моделей засновані на припущенні, що Метеорологічні умови просторово однорідні і мало змінюються з часом протягом періоду, який зазвичай становить одну годину. Це еквівалентно говорячи, що часова шкала, яка керує варіаціями в метеорології, більша, ніж час проходження між джерелом і рецептором.

Гауссівська модель забруднення повітря – це математична модель, яка використовує гаусівські (нормальні) розподіли для апроксимації розподілу концентрацій забруднюючих речовин в повітрі. Ця модель часто використовується в атмосферній науці та інженерії для прогнозування розподілу забруднень та визначення їх впливу на навколишнє середовище[41].

Основні характеристики Гауссівської моделі забруднення повітря включають:

- середня концентрація. Модель визначає середні значення концентрацій забруднюючих речовин у різних точках або регіонах;

— стандартне відхилення. Гауссівська модель також враховує ступінь розподілу концентрацій відносно середнього значення за допомогою стандартного відхилення;

— дифузія. Враховується дифузія забруднень в атмосфері, тобто їх поширення вітром та іншими атмосферними процесами;

— висотний профіль. Враховується висотний розподіл концентрацій забруднюючих речовин, оскільки вони можуть розташовуватися на різних висотах у повітряному шарі;

— випадковість. Гауссівська модель може враховувати випадкові елементи, які впливають на концентрації забруднюючих речовин.

Модель базується на простій формулі, яка описує тривимірне поле концентрації, що генерується точковим джерелом за стаціонарних метеорологічних умов та умов викидів. Модель гауссового факелу зображена на Рисунок 2.1.

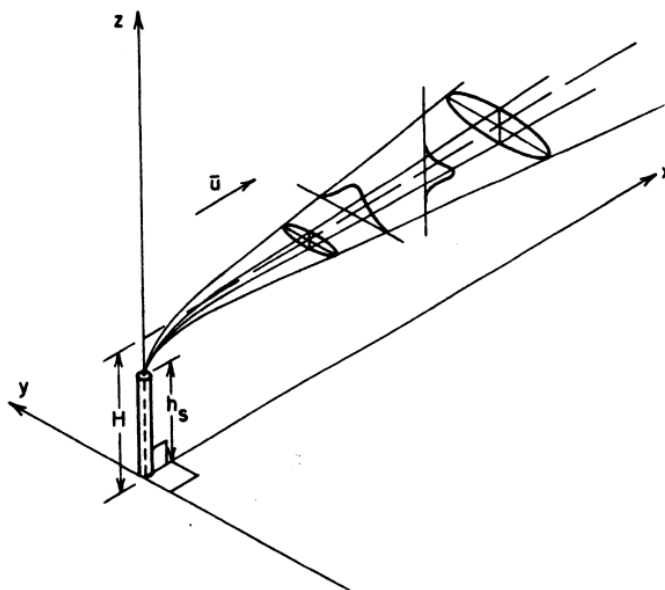


Рисунок 2.1 Модель розповсюдження забруднювача в повітрі вздовж осі x, координата джерела (0, 0, H) [40]

Для простоти факел рухається в напрямку додатної осі x. У загальній системі відліку формула гауссового факела має вигляд [42]:

$$c = \frac{Q}{2\pi\sigma_h\sigma_z|\bar{u}|} \exp \left[-\frac{1}{2} \left(\frac{\Delta_{cw}}{\sigma_h} \right)^2 \right] \cdot \left[-\frac{1}{2} \left(\frac{z_s + \Delta h - z_r}{\sigma_z} \right)^2 \right], \quad (2.4)$$

де $c(s,r)$ – це концентрація в точці $r = (x_r, y_r, z_r)$, обумовлена викидами в точці $s = (x_s, y_s, z_s)$, Q – це інтенсивність викидів; $\sigma_h(j_h, d)$ та $\sigma_z(j_z, d)$ – це стандартне відхилення (горизонтальна і вертикальна) просторового розподілу концентрації факела (часто σ_h , позначається як σ_y); j_h і j_z – горизонтальний і вертикальний стани турбулентності (більш детально обговорюються нижче); Δh – це підйом факела викидів, який є функцією параметрів викидів, метеорології та відстані з навітряного боку d . d – відстань з навітряного боку від приймача до джерела, де

$$d = [(r - s) \cdot \bar{u}] / |\bar{u}|, \quad (2.5)$$

де \bar{u} – середній вектор швидкості вітру $(\bar{u}_x, \bar{u}_y, \bar{u}_z)$ на висоті викиду; Δ_{cw} – відстань між приймачем і джерелом (тобто, між приймачем і центральною лінією факелу), де

$$\Delta_{cw} = (|r - s|^2 - d^2)^{1/2} \quad (2.6)$$

Рівняння 2.4 застосовується за умови $d > 0$; якщо $d \leq 0$, то $c = 0$.

З рівняння 2.4 видно, що воно відноситься до стаціонарного стану (не залежить часу), використовує метеорологічні умови (стан вітру та турбулентності), які повинні вважатися однорідними та стаціонарними в модельованій області (тобто, між s та r), і не може працювати в умовах штилю, тобто коли \bar{u} прямує до нуля. Однак простота гауссівського підходу, його відносна легкість використання з легко вимірюваними метеорологічними параметрами і, стимулювали дослідження, спрямовані на усунення деяких обмежень гауссівської теорії в обробці складних ситуацій реального світу[40].

Рівняння 2.4 в загальному випадку записується у вигляді

$$c = \frac{Q}{2\pi\sigma_y\sigma_z|\bar{u}|} \exp \left[-\frac{1}{2} \left(\frac{y_r}{\sigma_y} \right)^2 \right] \cdot \left[-\frac{1}{2} \left(\frac{h_e - z_r}{\sigma_z} \right)^2 \right], \quad (2.7)$$

де \bar{u} – середня горизонтальна швидкість вітру, h_e – ефективна висота викиду (тобто $h_e = z_s + \Delta h$).

Ця модель дозволяє аналізувати розподіл забруднень в просторі та часі, допомагаючи у визначенні можливих впливів на здоров'я людей та навколишнє

середовище. Вона може використовуватися для розробки стратегій зменшення забруднень та для ефективного управління якістю повітря в міських та промислових областях.

2.6 Моделювання дисперсії газів за допомогою Лагранжевих моделей

Моделі Лагранжа обчислюють траєкторії забруднення повітря на основі детермінованих та стохастичних ефектах. Ці траєкторії обчислюються на основі звичайних диференціальних рівнянь замість використання часткових диференціальних рівнянь. Рівняння траєкторії для однієї частинки є звичайним диференціальним рівнянням:

$$\frac{d\vec{r}}{dt} = \vec{v} + \vec{v}_1, \quad (2.8)$$

де \vec{r} – положення частинки, \vec{v} – детермінована швидкість частинки (що складається з адвекції та відстоювання), а \vec{v}_1 – вектор турбулентного коливання вітру.

Моделі Лагранжа можуть враховувати такі фактори, як вітер, турбулентність атмосфери, фізичні і хімічні властивості забруднюючих речовин, а також взаємодію з різними географічними об'єктами (наприклад, містами, гірськими ланцюгами тощо). Вони можуть бути корисними для прогнозування рівнів забруднення повітря від різних джерел, таких як промислові установки, транспорт, пожежі тощо, а також для оцінки ефективності заходів контролю забруднення та розробки стратегій зменшення впливу забруднення на здоров'я людей та навколишнє середовище. Однак для моделювання на великі відстані потрібно обчислити велику кількість одиничних траєкторій, що призводить до збільшення вартості обчислення [43].

2.7 Нестационарна модель дисперсії газів CALPUFF

CALPUFF — це багатовидова нестационарна модель дисперсії газів, яка моделює вплив різних метеорологічних умов у часі та просторі, транспортування, трансформацію та видалення забруднюючих речовин. CALPUFF призначений для використання на вагах від десятків метрів від джерела до сотень кілометрів. Він включає в себе алгоритми для ефектів ближнього поля, таких як спуск верхівки стека, низхід будівлі, перехідний підйом плавучої та імпульсної шлейфу, ефекти дощової шапки, часткове проникнення шлейфу, ефекти сітки масштабу місцевості та берегових взаємодій, а також вплив на місцевість, а також ефекти на більшій відстані, такі як видалення забруднюючих речовин за рахунок вологого очищення та сухого осадження, хімічної трансформації, впливу вертикального зсуву вітру, транспортування над водою, фумігації шлейфу та впливу концентрації твердих частинок на видимість [44].

CALPUFF підходить для моделювання розповсюдження газів на великій відстані (відстань джерело-рецептор від 50 до кількох сотень кілометрів) викидів від точкових, об'ємних, районних та лінійних джерел. Метеорологічні вхідні дані мають бути повністю охарактеризовані за допомогою тривимірного вітру та метеорологічних умов, що змінюються у часі та просторі, за допомогою CALMET. CALPUFF також може використовуватися в кожному конкретному випадку, коли модель більше підходить для конкретного застосування. Перевагою вибору системи моделювання CALPUFF, є повне врахування застою, зміни вітру, а також часових і просторових коливань метеорологічних умов при транспортуванні та розсіювання.

Окрім 3D-метеорологічних полів, розроблених діагностичною метеорологічною моделлю CALMET, CALPUFF може використовувати метеорологічні дані однієї станції, що зберігаються у форматі, який використовується іншими моделями дисперсії (ISC3ST, AUSPLUM, CTDMPLUS). Однак метеорологічні файли однієї станції не дозволяють

CALPUFF скористатися своїми можливостями для обробки просторово змінюваних метеорологічних полів [45].

CALPUFF створює файли погодинних концентрацій для кожного змодельованого виду, потоків вологого осадження, потоків сухого осадження, а для додатків видимості — коефіцієнтів вимирання. Процес та потік інформації, необхідний для проведення дослідження дисперсії повітря [46].

Формула адвективного переносу в моделі CALPUFF визначає швидкість переміщення забруднювачів у напрямку основного потоку повітря. Ця швидкість враховується у рівняннях руху повітря та використовується для прогнозування розподілу забруднювачів в атмосфері. Формула адвективного переносу має вигляд:

$$\vec{V} = \vec{u} + \vec{V}_t, \quad (2.9)$$

де \vec{V} — загальна швидкість переносу (адвективна + турбулентна), \vec{u} — адвективна швидкість (швидкість основного потоку повітря), \vec{V}_t — турбулентна компонента швидкості.

Турбулентна компонента швидкості в моделі CALPUFF зазвичай враховується як додатковий внесок, який може бути визначений на основі статистичних даних або вимірювань турбулентності в атмосфері. Загальна швидкість переносу визначає, як швидко забруднювачі переміщуються в атмосфері та як вони розповсюджуються в напрямку основного потоку повітря. Ця швидкість враховується при моделюванні розподілу забруднювачів у просторі та часі.

Нижче вказана формула, що використовується для моделювання розподілу викидів в часі та просторі в моделі CALPUFF описується наступним чином:

$$Q(x, y, z, t) = Q_0 \cdot f(x, y, z) \cdot g(t), \quad (2.10)$$

де $Q(x, y, z, t)$ — інтенсивність викиду у точці з просторовими координатами x, y, z у момент часу t , Q_0 — початкова інтенсивність викиду, $f(x, y, z)$ — функція, що визначає просторовий розподіл викиду залежно від координат x, y, z , $g(t)$ — функція, що визначає розподіл викиду в часі.

Для моделювання просторового розподілу викидів забруднювачів в атмосфері У CALPUFF використовується Гаусівська модель. Головна ідея полягає в тому, що концентрація забруднювачів змінюється за Гаусівським законом відносно відстані від джерела викиду. Це означає, що концентрація найвища неподалік від джерела викиду і поступово зменшується зі збільшенням відстані[46].

Гаусівський розподіл викидів в CALPUFF виражений такою формулою:

$$C(x, y, z) = \frac{1}{2\pi\sigma_x\sigma_y} \cdot e^{-\frac{(x-x_0)^2}{2\sigma_x^2}} \cdot e^{-\frac{(y-y_0)^2}{2\sigma_y^2}} \cdot h(z), \quad (2.11)$$

де $C(x, y, z)$ – концентрація забруднювача в точці з координатами x, y, z , Q – інтенсивність викиду, x_0, y_0 – координати джерела викиду, σ_x, σ_y – стандартні відхилення Гаусівського розподілу вздовж горизонтальних осей, $h(z)$ – функція, що визначає вертикальний розподіл викиду, наприклад, експоненціальний зважений зниження з висотою.

Ця модель може використовуватися для розрахунку концентрації забруднювачів в будь-якій точці атмосфери, враховуючи розташування джерела викиду та метеорологічні умови. Головна перевага Гаусівської моделі полягає у здатності точно відображати просторові властивості розподілу забруднювачів в атмосфері.

2.8 Метод інтерполяції для постобробки результатів моделювання

Інструменти інтерполяції надають можливість створити безперервну (або передбачувану) поверхню за значенням з опорних точок. Вимірювання висоти, величини або концентрації спостережуваних об'єктів і явищ у кожній точці досліджуваної території, як правило, є складним або дуже дорогим. Замість цього система може вимірювати показники в опорних точках, розподілених по поверхні, і прогнозувати значення, які можуть бути присвоєні всім іншим. Вхідні точки можуть бути розташовані як у вигляді регулярної сітки, так і випадковим чином[47].

Існує цілий ряд способів отримати для кожної точки прогнозоване значення; кожен метод розглядається як модель. Для кожної моделі існує цілий ряд припущень, які впливають з даних; деякі моделі краще підходять для конкретних типів даних. Кожна модель прогнозує значення з використанням різних обчислень. Інструменти інтерполяції зазвичай діляться на детерміновані і геостатистичні методи[48].

Детерміновані методи інтерполяції розраховують результат, ґрунтуючись на вимірних значеннях, які потрапляють в околицю інтерпольованої точки, і на заданих математичних формулах, які визначають згладженість результуючої поверхні. Детерміновані методи включають алгоритми IDW (метод обернено зважених відстаней), Nearest Neighbor (метод найближчого сусіду), Moving Average (ковзного середнього) і Linear (лінійної інтерполяції) [47].

Метод обернено зважених відстаней – це середньозважений інтерполятор. Ви повинні надати в якості вхідних значення розсіяних даних, включаючи координати кожної точки даних та геометрію вихідної сітки. Функція буде обчислювати інтерпольоване значення для заданої позиції у вихідній сітці. Для кожного вузла сітки отримане значення Z буде обчислюватися за формулою:

$$Z = \frac{\sum_{i=1}^n \frac{Z_i}{r_i^p}}{\sum_{i=1}^n \frac{1}{r_i^p}}, \quad (2.12)$$

де Z_i – це відоме значення точки i , r – це відстань від вузла сітки до точки i , p – це степінь зваження, n – кількість точок, що входить до вікна пошуку. В цьому методі фактор зважування w це $w = \frac{1}{r^p}$.

В описаних алгоритмах фігурує поняття вікна пошуку. Вікно пошуку в алгоритмах створення сітки, зазначено у формі обертового еліпса. Він описується трьома параметрами:

- перший радіус (вісь x , якщо кут повороту дорівнює 0);
- другий радіус (вісь y , якщо кут повороту дорівнює 0);
- кут повороту еліпса пошуку.

Для обчислення будуть використовуватися лише точки, розташовані всередині еліпса (включаючи його межу).

2.9 Висновок

Проаналізувавши математичні моделі якості повітря, поставлені вимоги до моделі: модель повинна враховувати фізичні та хімічні процеси, такі як дифузія, реакція в атмосфері та осадження забрудників, що впливають на їх розподіл та стан. Невід'ємним фактором впливу є метеорологічні умови, такі як швидкість вітру, температура, вологість та інші, що можуть впливати на розповсюдження забруднюючої речовини.

Для аналізу якості повітря використовуватимуться три види даних:

- супутникові дані (супутникові знімки місії Sentinel-5P);
- дані зі станцій наземного моніторингу;
- метеодані з NOAA (National oceanic and atmospheric administration).

Після проведення аналізу класичних моделей таких як Ейлерова модель, Гаусова модель та Лагранжева модель, були виявлені недоліки в їх роботі такі як не деталізація локальних процесів, застаріла параметризація моделей, на великій відстані потрібно обчислити велику кількість одиничних траєкторій, що призводить до збільшення вартості обчислення.

На противагу розглянута модель CALLPUFF, яка базується на Лагранжевій моделі, але враховує більшу кількість параметрів включно з кліматичною зоною, рівнем висот та іншим.

Для уточнення результатів моделювання обрано детерміновані методи інтерполяції, що розраховують результат, ґрунтуючись на вимірних значеннях, які потрапляють в околицю інтерпольованої точки, і на заданих математичних формулах, а саме метод обернено зважених відстаней.

Поєднання методу IDW з моделлю CALPUFF дозволяє поєднати переваги обох підходів. IDW доповнює модель CALPUFF, дозволяючи враховувати просторову залежність концентрацій забруднюючих речовин, особливо в

областях з обмеженим або відсутнім моніторингом. Таке поєднання поліпшує точність та достовірність результатів прогнозування розподілу забруднень у повітрі.

РОЗДІЛ 3. ТЕХНОЛОГІЇ МОДЕЛІ ДЛЯ РОЗРОБКИ АРХІТЕКТУРИ ІНФОРМАЦІЙНОЇ СИСТЕМИ

У розділі «Технології для розробки інформаційної системи» був проведений аналіз сучасних інструментів, методів і технологій, які використовуються для створення інформаційних систем. Цей розділ відіграє важливу роль у визначенні оптимального підходу до розробки системи, забезпечуючи її ефективність, масштабованість та надійність. Ми розглянемо різноманітні архітектури програмного забезпечення, такі як клієнт-серверні, веборієнтовані, мікросервісні тощо, та вивчимо їх переваги та недоліки залежно від конкретних потреб інформаційної системи. Крім того, ми розглянемо різні мови програмування, фреймворки, бази даних та інші інструменти, які використовуються для розробки систем. Основна мета – не лише ознайомитися з технологіями, але й визначити оптимальний набір інструментів та методів для успішної реалізації конкретної інформаційної системи з урахуванням її вимог і особливостей. Цей розділ допоможе нам зрозуміти, як використовувати технології та досягнути поставлених цілей ефективно і з високою якістю.

3.1 Порівняння провайдерів хмарних послуг

Використання хмарних провайдерів для систем моніторингу якості повітря може мати кілька переваг і стати ефективним рішенням для таких систем. Нижче перераховано кілька аргументів на користь використання хмарних послуг у цьому контексті:

Гнучкість та Масштабованість. Хмарні послуги надають можливість масштабування системи моніторингу в залежності від потреб. Ви можете легко збільшувати чи зменшувати обсяги ресурсів в залежності від обсягу даних та завдань обробки.

Висока Доступність. Провайдери хмарних послуг, такі як AWS (Amazon Web Services), GCP (Google Cloud Platform) або Azure, забезпечують високу

доступність та надійність своїх послуг, що дозволяє підтримувати неперервну роботу системи моніторингу.

Зберігання та Обробка Даних. Хмарні платформи пропонують широкий вибір служб для зберігання та обробки даних, включаючи бази даних, аналітичні інструменти та інші рішення. Це може полегшити управління та аналіз великого обсягу даних про якість повітря.

Інтеграція з IoT-пристроями. Багато хмарних провайдерів мають служби для Інтернету Речей (IoT) [49], що дозволяє легко інтегрувати дані з сенсорів і пристроїв, які використовуються в системах моніторингу якості повітря.

Безпека та Захист Даних. Провайдери хмарних послуг зазвичай вкладають величезні зусилля у забезпечення безпеки даних. Вони надають інструменти для шифрування, аутентифікації та контролю доступу, що допомагає захищати конфіденційні дані про якість повітря.

Оптимізація Витрат. Використання хмарних послуг може дозволити оптимізувати витрати, оскільки ви платите тільки за використані ресурси. Це може бути особливо корисним для проектів з обмеженим бюджетом.

Широкий Вибір Інструментів. Хмарні провайдери надають різноманіття інструментів для моніторингу та управління системами, що дозволяє ефективно відслідковувати та управляти якістю повітря.

Хоча є чимало переваг, важливо також враховувати певні виклики, такі як можливість виникнення затримок у доступі до даних через мережу або питання щодо приватності та безпеки даних. Перед вибором конкретного хмарного провайдера слід ретельно оцінити вимоги вашого проекту та перевірити, який провайдер найкраще відповідає цим вимогам.

3.1.1 Аналіз платформи Google Cloud Platform

Google Cloud Platform (GCP) – це хмарна платформа від компанії Google, яка надає широкий спектр хмарних послуг для розробки, тестування, розгортання та управління програмним забезпеченням та інфраструктурою.

Для розробки інформаційної системи могли б знадобитись такі інфраструктурні рушення як: Computed Engine, який надає віртуальні машини в хмарному середовищі з можливістю масштабування за потреби, Kubernetes Engine – керований сервіс контейнерів, що дозволяє розгортати, керувати та масштабувати контейнеризовані застосунки з використанням Kubernetes, App Engine – платформа для розробки та розгортання вебзастосунків та мікрослужб без необхідності управління інфраструктурою [50].

В якості сховища можна було б використати Google Cloud Storage – об'єктне сховище в хмарі, яке надає надійне та масштабоване сховище для зберігання будь-яких об'єктів даних, таких як файли, зображення, відео тощо. Для збереження повідомлень відкладених завдань або ж для збору помилок чи аналітики системи добре підійшов би Cloud Datastore – NoSQL-база даних, яка надає масштабовану та надійну інфраструктуру для зберігання даних[51].

3.1.2 Аналіз платформи Microsoft Azure

Microsoft Azure – це хмарна платформа від компанії Microsoft, яка надає широкий спектр хмарних послуг та інфраструктури для розробки, тестування, розгортання та управління програмним забезпеченням та інфраструктурою.

Ця платформа надає наступні інфраструктурні послуги (IaaS – Infrastructure as a Service): Azure Virtual Machines – надає можливість орендувати віртуальні машини для різних завдань та роботи з різними операційними системами, Azure Kubernetes Service (AKS) – керований сервіс контейнерів, який дозволяє розгортати та керувати контейнеризованими застосунками з використанням Kubernetes. Azure Functions – сервіс обчислення без сервера, який дозволяє запускати код у відповідь на події без необхідності управління інфраструктурою[52].

Azure може слугувати платформа для розробки, розгортання та масштабування вебзастосунками та служб без управління інфраструктурою. Azure SQL Database – це керована послуга реляційної бази даних, яка дозволяє легко створювати, керувати та масштабувати бази даних в хмарі.

До хмарні послуги (SaaS – Software as a Service) можна віднести Microsoft 365 – хмарний сервіс, який надає комплексні рішення для офісних робіт, такі як електронна пошта, текстовий процесор, спільна робота над документами тощо. Dynamics 365 – це хмарна платформа для автоматизації діяльності підприємства, включаючи управління відносинами з клієнтами, управління виробництвом, фінанси тощо[53].

Для аналітики та складних розрахунків можна використати Azure Machine Learning – платформу для створення, навчання та розгортання моделей машинного навчання. Облікова система аналітики платформи об'єднує обробку та аналіз даних у реальному часі [54].

Також платформа надає набір інструментів для розробки, тестування та розгортання програмного забезпечення з інтеграцією систем контролю версій, засобів автоматизації та спільної роботи.

3.1.3 Аналіз платформи Amazon Web Services

Amazon Web Services (AWS) – це найбільший та найпопулярніший хмарний провайдер у світі, що надає широкий спектр послуг для обчислення, зберігання даних, аналізу, штучного інтелекту, Інтернету Речей (IoT), машинного навчання та інших областей.

Amazon Web Services надає масштабовані віртуальні сервери для обчислення, які можна налаштовувати за потреби Amazon EC2 (Elastic Computed Cloud). EC2 пропонує різні типи інстанцій, кожен з яких призначений для конкретних вимог щодо обчислювальних, мережевих ресурсів та ресурсів сховища. Наприклад, інстанції типу «t2» призначені для загальних завдань, тоді як «c5» – для високопродуктивних обчислень [55]. Кожна інстанція EC2 може використовувати різні типи сховища, такі як Amazon Elastic Block Store (EBS) для постійного сховища або локальні інстанції для тимчасового зберігання. EC2 дозволяє масштабувати інстанції вертикально або горизонтально в залежності від обсягу роботи. Користувачі можуть використовувати інструменти автоматизації, такі як Auto Scaling, для динамічного змішування чи зменшення

обчислювальних ресурсів, що вплине на ціну за використання ресурсів [56]. Вартість використання EC2 визначається на основі різних факторів, таких як тип інстанції, кількість використаних ресурсів, тривалість роботи та обсяг сховища. Це дає користувачам гнучкість у витрачанні коштів в залежності від їхніх потреб.

Для зручного керування та оренди контейнерів можна використати керовану службу контейнерів у хмарному середовищі Amazon Web Services – Amazon ECS (Amazon Elastic Container Service). Він підтримує контейнери Docker і надає можливість орендувати, масштабувати їх на основі інфраструктури AWS.

Для швидких обчислень запропоновано скористуватись Amazon Lambda. це сервіс обчислення у хмарному від Amazon Web Services (AWS), який надає можливість виконувати код без необхідності управління серверами. AWS Lambda використовується для створення серверних функцій, які можуть відповідати на події, спричинені іншими службами AWS або зовнішніми джерелами. Він підтримує різні мови програмування, включаючи Node.js, Python, Ruby, Java, Go та .NET. Це дозволяє розробникам використовувати свою улюблену мову для написання функцій.

В якості об'єктного сховища Amazon Web Services надає Amazon Simple Storage Service (Amazon S3), яке призначене для зберігання та витягування будь-якого обсягу даних у будь-який момент. S3 є однією з основних інфраструктурних послуг AWS і забезпечує надійне, масштабоване та доступне сховище [57].

Крім цього, Amazon Web Services ще багато різних сервісів таких як: AWS IoT (управління та забезпечення безпеки пристроїв Інтернету Речей), Amazon CloudFront (служба CDN для швидкого і надійного розповсюдження контенту), Amazon RDS (управління реляційними базами даних, такими як MySQL, PostgreSQL, Oracle та іншими), Amazon CloudWatch (сервіс моніторингу та логування для ресурсів та застосунків у реальному часі) та багато іншого.

У порівнянні з Google Cloud Platform та Azure Amazon Web Services має більший спектр послуг, інтегрований з більшою кількістю технологій. Також він включає велику кількість сервісів, що підтримують широкий спектр вимог.

Однак на відміну від інших розглянутих має складну політику ціноутворення, але ця проблема відома і має вирішитись з плином часом.

Беручи до уваги всі характеристики і поставлену мету отримали наступні результати: Amazon Web Services є лідером за розміром та має найширше географічне охоплення, є представленим в багатьох регіонах та дата-центрах, є інтегрованим з різноманітними технологіями та мовами програмування, має найширший вибір спеціалізованих послуг та інновацій, має потужну глобальну мережу. Отже в якості провайдера хмарних послуг та провайдера серверів було обрано саме Amazon Web Services.

3.2 Різновиди збору даних для інформаційної системи моніторингу кості повітря

Агрегатор і парсер є двома різними поняттями, пов'язаними з обробкою та збором інформації з інтернет-ресурсів. Давайте розглянемо їх визначення та відмінності:

Агрегатор – це інструмент або сервіс, який збирає та об'єднує інформацію з різних джерел у одному місці. Наприклад, новинний агрегатор може збирати заголовки та статті з різних новинних сайтів та представляти їх користувачам на одній платформі. Агрегатори дозволяють користувачам отримувати доступ до різних джерел інформації без необхідності відвідувати кожен сайт окремо. Вони спрощують процес споживання контенту та роблять його більш доступним та зручним [58].

Парсер – це програма або скрипт, який аналізує структуру даних на вебсайті та витягує певні елементи або інформацію. Наприклад, парсер може сканувати вебсторінку та вилучати дані, такі як ціни на товари, назви статей, або будь-яку іншу інформацію з HTML-коду сторінки. Парсери використовуються для автоматизації процесу вилучення даних з вебсайтів. Їхні функції можуть включати аналіз HTML або інших мов розмітки, витягування текстової

інформації, інтерпретацію структурованих даних та зберігання отриманих даних у бажаному форматі.

Існує кілька видів парсерів, які використовуються для витягування інформації з вебсайтів. Кожен вид парсера має свої характеристики і використовується в різних ситуаціях. Ось деякі загальні види парсерів:.

HTML – парсери призначені для аналізу та вилучення даних з HTML-коду вебсторінок. Вони можуть використовувати бібліотеки, такі як BeautifulSoup для мов програмування Python або jsoup для Java. HTML - парсери дозволяють аналізувати структуру вебсторінок та вилучати текст, посилання, таблиці та інші елементи [59].

XML – парсери використовуються для обробки та витягування інформації з документів у форматі XML (eXtensible Markup Language). Вони можуть бути корисними для аналізу даних, що використовуються в різних вебслужбах та обміні даними. XML-парсери можуть витягувати дані з XML-документів та перетворювати їх для подальшого використання.

JSON-парсери спеціалізуються на обробці та витягуванні даних з формату JSON (JavaScript Object Notation). JSON - парсери допомагають розбирати та витягувати об'єкти та дані, представлені у форматі JSON. Вони широко використовуються в веброзробці для взаємодії між вебсерверами та клієнтами, а також для обробки даних в API.

Регулярні вирази (Regex) використовують регулярні вирази для пошуку та вилучення шаблонів тексту з документів. Regex-парсери дозволяють визначати певні патерни або шаблони в тексті та вилучати відповідні фрагменти[60].

Спеціалізовані парсери. Інші парсери можуть бути розроблені для конкретних задач або форматів даних. Наприклад, парсери для обробки RSS-стрічок, парсери PDF-документів тощо. Спеціалізовані парсери застосовуються для витягування конкретних типів даних або обробки конкретних форматів.

3.3 Аналіз протоколів для обміну даних в системі

Протокол обміну даними – це формалізований набір правил і процедур, які визначають спосіб, у який пристрої взаємодіють між собою для передачі інформації. Він служить як мова комунікації, яка дозволяє різним пристроям, програмам або системам спілкуватися і обмінюватися даними незалежно від їх типу, марки або розташування. Протоколи обміну даними визначають такі аспекти комунікації, як формати даних, механізми аутентифікації, шифрування, перевірку цілісності даних, реакцію на помилки та інші аспекти, які необхідні для ефективного і надійного обміну інформацією. Ці протоколи можуть бути загальними або спеціалізованими. Загальні протоколи, такі як HTTP (Hypertext Transfer Protocol) для передачі вебсторінок, SMTP (Simple Mail Transfer Protocol) для електронної пошти або FTP (File Transfer Protocol) для передачі файлів, використовуються для різноманітних цілей і можуть бути використані в різних сценаріях. З іншого боку, спеціалізовані протоколи можуть бути розроблені для конкретних завдань або областей, таких як мережева безпека, керування мережею або передача мультимедійних даних. У кінцевому підсумку, протоколи обміну даними є основою для ефективної і надійної комунікації в комп'ютерних системах та мережах, дозволяючи різним пристроям і програмам спілкуватися між собою навіть у різних середовищах та умовах.

3.3.1 Можливості протоколу обміну даними UDP

UDP (User Datagram Protocol) – це простий протокол обміну даними, який працює в мережевому стеку інтернет-протоколів, що забезпечує відправку даних без забезпечення надійності або порядку доставки. Основна його ідея полягає у тому, що він передає невеликі блоки даних, відомі як датаграми, між відправником і одним або кількома отримувачами. Датаграма – це фундаментальна одиниця передачі даних в мережевому протоколі, зокрема у протоколі UDP (User Datagram Protocol). Вона представляє собою невеликий блок даних, який включає в себе корисну інформацію для передачі, а також додаткові метадані для адресації та маршрутизації [61].

Основні характеристики UDP: Ненадійність: UDP не гарантує доставку даних. Це означає, що якщо датаграма не досягає призначеного отримувача через мережеві затримки або втрату пакетів, UDP не буде намагатися перевідправити датаграму. Висока швидкодія: Оскільки UDP не має механізмів перевірки цілісності даних або відновлення втрачених пакетів, він працює швидше за TCP (Transmission Control Protocol), що забезпечує надійну передачу. Придатність для реального часу: UDP ідеально підходить для застосунків, де важлива низька затримка, таких як голосова та відеоконференції або онлайн-ігри. Втрата деяких кадрів або пакетів може призвести до невеликих відхилень у якості, але намагаючись відновити втрачені дані, може виникнути додаткова затримка, що робить UDP менш придатним для таких застосувань. Мультикастинг і широкомовність: UDP підтримує мультикастинг та широкомовність, що дозволяє відправляти датаграми групі приймачів або всім пристроям у мережі [62]. Простота і низький рівень обробки: У порівнянні з TCP, UDP має менше накладних витрат, оскільки він не вимагає зберігання стану з'єднання і не виконує складних алгоритмів контролю потоку або керування перевантаженням. UDP використовується в різних застосунках, таких як DNS (Domain Name System), DHCP (Dynamic Host Configuration Protocol), протоколи потокової передачі даних (наприклад, RTP – Real-time Transport Protocol) та багато інших, де низька затримка і простота важливіше, ніж надійність передачі.

Звичайна схема роботи UDP може бути подана так:

1. Створення датаграми: Відправник створює датаграму, додавши до неї дані для передачі, а також вказуючи IP-адресу та порт призначення. Відправка датаграми: Датаграма відправляється на вказану IP-адресу та порт призначення через мережний інтерфейс. Ця операція може виконуватися безпосередньо операційною системою або за допомогою програмного забезпечення, що працює на рівні застосунку.
2. Передача через мережу: Датаграма маршрутизується через мережу і може проходити через кілька маршрутизаторів та комутаторів, перш ніж досягне призначеного вузла.

3. Прийом датаграми: Призначений вузол отримує датаграму, перевіряє її заголовок, щоб визначити, чи призначена вона саме йому, та розпочинає обробку отриманих даних.

4. Обробка даних: Отримані дані обробляються на приймаючому вузлі згідно з логікою застосунку, що працює на цьому вузлі.

5. Відповідь (необов'язково): у випадку взаємодії між клієнтом і сервером або просто у випадку необхідності, приймаючий вузол може відправити відповідь на відправлену датаграму, використовуючи таку ж саму схему роботи [63].

Ця схема дозволяє взаємодіяти з віддаленими вузлами шляхом надсилання коротких повідомлень без створення постійного з'єднання, що робить UDP ефективним для великої кількості застосувань, де швидкість передачі даних є важливішою за надійність. Однак це також означає, що програмам, що використовують UDP, потрібно вирішувати проблеми з надійністю, відновленням даних та контролем стану передачі самостійно.

3.3.2 Можливості протоколу обміну даними TCP

Transmission Control Protocol (TCP) – це протокол на вищому рівні мережевого стеку, який забезпечує надійний обмін даними між комп'ютерами у мережі Інтернет. Основна його мета – гарантувати доставку даних у правильному порядку і без втрат. Основні характеристики TCP: Надійність: Одна з ключових властивостей TCP – це його надійність. Він включає в себе механізми перевірки цілісності даних, відновлення втрачених або пошкоджених пакетів та повторну передачу випадково втрачених пакетів. Це робить TCP ідеальним для застосувань, де важлива точність і повнота передачі даних, таких як передача файлів, електронна пошта тощо. Контроль потоку: TCP включає в себе механізми контролю потоку для регулювання швидкості передачі даних між відправником і отримувачем. Це дозволяє уникнути перевантаження отримувача даними, які він не може обробити на даний момент. Механізми відновлення з'єднання: TCP встановлює та підтримує з'єднання між відправником і отримувачем передачі

даних, а також має механізми для відновлення цих з'єднань у випадку їх розриву [64]. Відновлення порядку: TCP гарантує, що дані будуть доставлені в правильному порядку, навіть якщо вони були розділені на окремі пакети для передачі по мережі. Висока накладна витрата: У порівнянні з UDP, TCP має більшу накладну витрату через додаткові механізми контролю потоку, відновлення з'єднання та перевірки цілісності даних. TCP використовується в багатьох важливих мережових протоколах та застосунках, таких як HTTP для вебсайтів, SMTP для електронної пошти, FTP для передачі файлів, SSH для безпечного з'єднання з віддаленими комп'ютерами тощо. Його надійність та механізми контролю зробили його одним з найбільш широко використовуваних протоколів у мережевому світі[65].

3.3.3 Можливості протоколу обміну даними HTTP

HTTP (Hypertext Transfer Protocol) – це протокол на вищому рівні мережевого стеку, який використовується для передачі гіпертекстових документів у Всесвітній павутині. Основним призначенням HTTP є забезпечення способу взаємодії між клієнтськими програмами (наприклад, веббраузерами) та вебсерверами для отримання вебсторінок та ресурсів. Основні характеристики HTTP: Простота використання: HTTP використовує прості текстові повідомлення для взаємодії між клієнтами і серверами. Це робить його легким для розуміння та відлагодження. Безстанційність: Кожен запит HTTP обробляється сервером незалежно від попередніх запитів. Це означає, що сервер не зберігає жодної інформації про стан попередніх запитів від клієнта. Методи запитів: HTTP визначає різні методи запитів, такі як GET, POST, PUT, DELETE, що визначають тип операції, яку клієнт хоче виконати над ресурсом на сервері. Статус коди відповіді: HTTP використовує статус коди відповіді для позначення результату виконання запиту. Наприклад, 200 OK показує успішне виконання запиту, 404 Not Found – відсутність ресурсу, а 500 Internal Server Error – помилка на сервері [66]. Можливість кешування: HTTP підтримує механізми кешування, що дозволяють клієнтам та проксі – серверам зберігати копії ресурсів локально

для швидшого доступу в майбутньому. Підтримка TLS/SSL: HTTP може працювати з захищеними з'єднаннями за допомогою протоколів TLS (Transport Layer Security) або SSL (Secure Sockets Layer), забезпечуючи конфіденційність і цілісність передачі даних. HTTP використовується як основний протокол для передачі вебсторінок, зображень, аудіо- та відеофайлів, а також для взаємодії з вебсерверами через API. Його простота та широке застосування роблять його одним з найбільш важливих протоколів в мережевому світі [67, 68].

3.4 Віртуальна контейнеризація інформаційної системи

Для ефективного розгортання багатьох моделей одночасно та оптимізації витрат на серверну інфраструктуру, моделі були упаковані у власні контейнери Docker. Кожній моделі відповідає окремий контейнер, а для обміну даними між ними було використано спільний том.

Docker – це платформа для створення, доставки та запуску контейнеризованих додатків. Вона дозволяє легко створювати контейнери, автоматизувати їх запуск та керувати їх життєвим циклом. Docker дозволяє запускати безліч контейнерів на одній фізичній машині.

Контейнери – це спосіб упакувати програму разом з її залежностями в один образ. Цей образ запускається в ізольованому середовищі, що не впливає на основну операційну систему. Вони дозволяють відокремити додаток від інфраструктури, що спрощує розгортання та зменшує можливість конфліктів між залежностями [69].

Контейнеризація схожа на віртуалізацію, але вона більш легковагома. У віртуальному середовищі запускається окрема операційна система, тоді як у контейнері – використовується ядро основної операційної системи. Це дозволяє контейнерам ефективніше використовувати ресурси. Контейнери дозволяють спростити розгортання програм, оскільки всі необхідні залежності включаються в контейнер. Також вони полегшують розгортання на кількох серверах, оскільки не потрібно повторювати одні й ті самі дії для кожного сервера.

Docker volumes – це зручний спосіб обміну даними між контейнерами. Docker автоматично створює директорії для зберігання даних і дозволяє різним контейнерам обмінюватися даними через ці директорії [70].

Використання Docker дозволяє легко масштабувати моделі та максимально використовувати переваги хмарних серверів у майбутньому.

3.5 Авторизаційний фреймворк для надання обмеженого доступу в систему

Протокол OAuth 2.0 (Open Authorization 2.0) є важливим стандартом для делегування авторизації в розподілених системах, особливо в сучасному Інтернеті та вебзастосунках. Ось переваги та недоліки протоколу OAuth 2.0:

Переваги протоколу OAuth 2.0:

- спрощення авторизації: OAuth 2.0 дозволяє користувачам давати дозвіл на обмежений доступ до своїх ресурсів без використання паролю та без передачі чутливих облікових даних;
- декларативність прав доступу: Дозволяє визначати різні рівні доступу для різних частин ресурсів, надаючи більшу гнучкість та контроль;
- сумісність з різними ідентифікаторами: Дозволяє використовувати різні механізми аутентифікації, такі як OpenID Connect, для підтримки різних сценаріїв;
- спрощений процес рефрешу токенів: Використовуючи рефреш-токени, OAuth 2.0 дозволяє оновлювати токени без необхідності повторної аутентифікації користувача [71];
- підтримка для сторонніх застосунків: Дозволяє стороннім застосункам безпечно отримувати доступ до обмежених ресурсів за допомогою авторизації користувача;
- широке застосування в Інтернеті: Використовується для авторизації та доступу до API, соціальних мереж, календарів, електронної пошти та інших онлайн-ресурсів;

- легка інтеграція: Є популярним та широко підтримується, що полегшує інтеграцію з різними сервісами та системами.

Недоліки протоколу OAuth 2.0:

- відсутність стандарту для аутентифікації: OAuth 2.0 сам по собі не надає стандартного механізму для аутентифікації користувачів. Для цього часто використовуються інші протоколи, такі як OpenID Connect;

- потенційні проблеми з безпекою: Некоректне використання або налаштування може призвести до проблем з безпекою, таких як неналежна обробка токенів чи вразливості в реалізації;

- складність для розробників: Для деяких розробників може виявитися важким зрозуміти та правильно реалізувати різні типи грантів та їх взаємодію. Необов'язкова шифрування: OAuth 2.0 не вимагає використання шифрування для обміну токенами, хоча за рекомендацією шифрування може бути використано для забезпечення безпеки;

- відсутність вбудованого механізму для валідування токенів: Перевірка валідності токенів, їх терміну дії та повноважень може вимагати додаткової реалізації на рівні клієнта.

Незважаючи на ці недоліки, OAuth 2.0 залишається одним з ключових стандартів для делегування авторизації в Інтернеті, і його переваги в багатьох випадках переважають недоліки. Застосування вірного гранту та правильна конфігурація дозволяють забезпечити безпеку та зручність взаємодії між різними застосунками та сервісами[72].

3.6 Аналіз архітектури мережевих протоколів

Перед розробкою системи необхідно розглянути ключові аспекти вибору між різними архітектурними підходами при розробці мережевих протоколів для сучасних систем і застосунків. Правильний вибір архітектури може визначити ефективність, безпеку, масштабованість та зручність вашої системи для

користувачів і розробників. У цьому розділі ми детально розглянемо такі ключові аспекти:

Для початку познайомимось з архітектурним стилем Representational State Transfer (REST). Це архітектурний стиль, що використовується для розробки розподілених систем, таких як вебзастосунки. Він базується на принципах, які забезпечують стабільність, масштабованість, модульність та простоту взаємодії між компонентами системи [73].

У Representational State Transfer, дані системи представлені у вигляді ресурсів, які можуть бути, наприклад, об'єктами, колекціями або сервісами. Кожен ресурс має унікальний ідентифікатор (URI), який використовується для доступу до нього. Ресурси у REST можуть мати різні представлення, наприклад, JSON, XML, HTML тощо. Для прикладу формат протоколу Simple Object Access Protocol в якості формату передачі використовує XML формат. Клієнт може вибирати представлення, яке йому потрібно, шляхом вказання відповідного заголовка запиту [74, 75].

На відміну від Simple Object Access Protocol який не накладає обмеження на протокол транспортування даних (для передачі даних можна використати як Web протокол HTTP, так й MQ.) Representational State Transfer використовує HTTP методи для взаємодії з ресурсами. Найпоширенішими методами є GET (для отримання ресурсу), POST (для створення нового ресурсу), PUT (для оновлення існуючого ресурсу) та DELETE (для видалення ресурсу)[77].

Коли сервер отримує запит від клієнта, він обробляє його відповідно до методу та URI, що містяться у запиті. Наприклад, якщо клієнт виконав GET-запит на певний URI, сервер може відповісти з ресурсом, що відповідає цьому URI. Це може бути, наприклад, HTML-сторінка, зображення, файл або будь-який інший контент, який сервер може надати. Після обробки запиту сервер формує HTTP-відповідь, яка містить статус-код, що показує результат обробки запиту. Цей статус-код може бути, наприклад, 200 OK, що вказує на успішну обробку запиту, або 404 Not Found, який означає, що ресурс за вказаним URI не знайдений. Крім того, HTTP-відповідь може містити дані, пов'язані з відповіддю. Це може бути

вміст, що відповідає на запит клієнта, такий як ресурси або повідомлення про помилку[78]. Клієнт отримує HTTP-відповідь від сервера і обробляє її відповідно до своїх потреб. Наприклад, якщо клієнт виконав GET-запит на отримання даних, він може відобразити ці дані у вебінтерфейсі або використати їх у своєму застосунку, що зображено на рисунку 3.1.

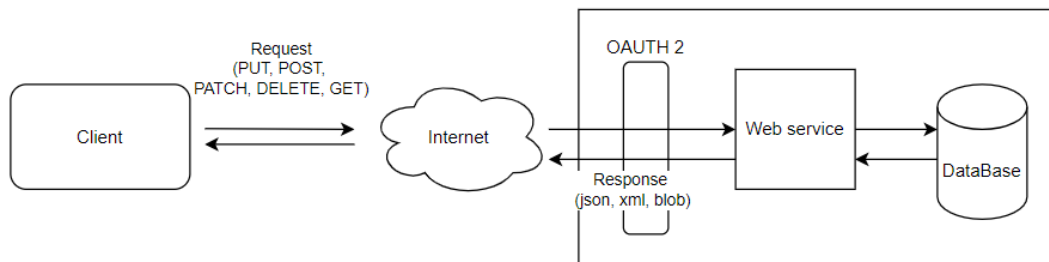


Рисунок 3.1 Схема роботи Representational State Transfer

Взаємодія може продовжуватися, якщо клієнт або сервер має ще дій, що вони хочуть виконати. Наприклад, клієнт може виконати PUT-запит для оновлення даних на сервері або DELETE-запит для видалення ресурсу, що показано на рисунку 3.2.

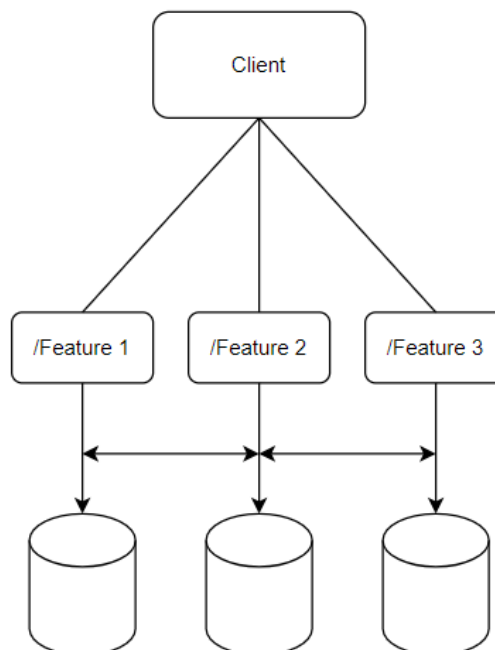


Рисунок 3.2 Схема взаємодії з даними за використання Representational State Transfer (REST)

Альтернативою Representational State Transfer може стати GraphQL – мова запитів та середовище виконання для розробки вебзастосунків. GraphQL відрізняється від REST основними концепціями гнучкості та ефективності передачі даних між клієнтом та сервером. Замість того, щоб використовувати фіксовані ендпоінти, як це робить REST, GraphQL дозволяє клієнтам визначати точно ті дані, які їм потрібні, шляхом створення запитів зі специфікованою структурою. [79]. На відміну від REST, де ресурси представлені фіксованою структурою URL-адреси, GraphQL використовує один єдиний ендпоінт, через який клієнт надсилає свої запити. Це дозволяє зменшити кількість запитів до сервера та спростити структуру вебзастосунків. Відмінність підходів представлено на рисунку 3.3.

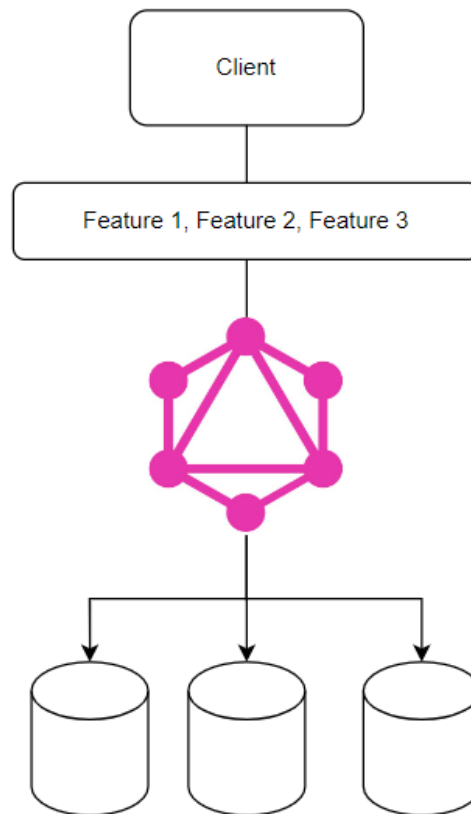


Рисунок 3.3 Схема взаємодії з даними за використання GraphQL

Клієнти отримують всю інформацію, що пов'язана з ресурсом, за одним запитом. GraphQL використовує власний типізований мовний формат запитів та відповідей, що називається GraphQL schema, а у Representational State Transfer

дані передаються у форматі, визначеному сервером (часто JSON або XML). У REST ресурси представлені за допомогою URI, і взаємодія з ними здійснюється через CRUD-операції (створення, читання, оновлення, видалення)[80]. У GraphQL ресурси представлені графом об'єктів, які можуть мати зв'язки між собою. Клієнт може запитувати лише ті дані, які йому потрібні, та у вигляді, який йому зручний. REST підтримує стандартні механізми кешування HTTP, такі як заголовки Cache-Control і ETag. GraphQL не має вбудованих механізмів кешування, але може використовувати сторонні бібліотеки або інструменти для кешування даних. Документація REST API зазвичай створюється окремо від коду, що може викликати невідповідність між документацією та реальними можливостями API. GraphQL автоматично генерує документацію API (GraphiQL), що дозволяє клієнтам досліджувати та тестувати API безпосередньо в браузері.

З цього випливають такі переваги GraphQL:

- гнучкість запитів: GraphQL дозволяє клієнтам визначати, які саме дані вони хочуть отримати, шляхом вказання потрібних полів у запиті. Це забезпечує гнучкість та ефективність передачі даних;
- зменшення кількості запитів: Завдяки можливості запитувати лише потрібні дані, GraphQL допомагає зменшити кількість запитів до сервера та оптимізує мережевий трафік;
- типізація даних: GraphQL використовує типи даних для визначення структури та формату даних, що передаються через API, що сприяє документації та робить API більш прозорим для розробників;
- автоматична документація: GraphQL автоматично генерує документацію API (GraphiQL), що дозволяє клієнтам досліджувати та тестувати API безпосередньо в браузері;
- підписка на зміни (Subscriptions): GraphQL підтримує механізм підписок, що дозволяє клієнтам отримувати реальний час оновлення даних від сервера, що особливо корисно для реал-тайм застосунків.

Але в цього підходу є й певні недоліки у порівнянні з Representational State Transfer [81]:

- складність реалізації: Реалізація GraphQL може бути складнішою, особливо для великих або складних проєктів, порівняно з Representational State Transfer, який має просту структуру та підходи;
- навантаження на сервер: GraphQL може призводити до збільшення навантаження на сервер у випадку великої кількості складних запитів, оскільки клієнти можуть запитувати різні дані з одного запиту;
- безпека: GraphQL може бути вразливим до підтверджених запитів (англ. «query over-fetching»), коли клієнт отримує більше даних, ніж потрібно. Це може призвести до можливих проблем з безпекою та конфіденційністю даних;
- кешування: GraphQL не має вбудованих механізмів кешування, що може призвести до неефективного використання ресурсів у випадку частих запитів на одні й ті ж дані.

Проаналізувавши недоліки та переваги обидвох підходів було вирішено обрати Representational State Transfer черкз те що:

- запити повертатимуть великий обсяг даних по станціям тощо, що будуть запитуватися 100% «клієнтів» розробленої системи, тому існує запит на підключення кешування без втрати даних;
- відсутня необхідність у гнучких запитів на користь швидколії сервера;
- наявність «query over-fetching», може невілювати переваги використання протоколу OAuth2.

3.7 Аналіз архітектурних рішень для створення інформаційної системи

При проєктуванні будь-якої інформаційної системи або програмного продукту, одним з ключових аспектів є вибір відповідної архітектури. Архітектура системи визначає спосіб організації компонентів, взаємодії між

ними та загальну структуру системи. Правильний вибір архітектури може значно вплинути на продуктивність, масштабованість, безпеку та інші характеристики системи.

Вибір архітектури програмного забезпечення є ключовим етапом у процесі розробки будь-якого проекту. Коректна обрана архітектура визначає ефективність, гнучкість та масштабованість системи. У цьому розділі ми розглянемо різні архітектурні підходи та їх особливості, щоб зробити обґрунтований вибір для нашого проекту. Перш за все, варто визначити основні вимоги до системи, такі як швидкість розробки, масштабованість, надійність, зручність у використанні тощо.

На основі цих вимог ми можемо зрозуміти, яка архітектура буде найбільш відповідати нашим потребам. Ми можемо розглянути різні архітектурні підходи, такі як монолітна архітектура, мікросервісна архітектура, архітектура, орієнтована на сервіси (SOA), або навіть serverless архітектуру. Кожен з них має свої переваги та недоліки, і вибір залежить від конкретних вимог та умов проекту. Після аналізу цих архітектурних підходів ми зможемо прийняти обґрунтоване рішення, яке буде найкращим з точки зору наших потреб та обмежень. Важливо також пам'ятати, що архітектура може еволюціонувати протягом часу, тому варто мати можливість адаптувати її до змін у вимогах та умовах проекту.

Serverless архітектура – це підхід до розробки програмного забезпечення, де розробник не має безпосереднього контролю над серверами, на яких запускається програмне забезпечення. Замість того, щоб керувати інфраструктурою, розробник працює з окремими функціями (функціями як сервісом, Function as a Service – FaaS), які викликаються відповідно до потреб системи [82].

Розробник пише і розгортає окремі функції, які викликаються лише при необхідності, без необхідності управління інфраструктурою. Можна розрізнити кілька парадигм безсерверних обчислень:

- функція як послуга (FaaS), реалізована, наприклад, за допомогою AWS Lambda (Hendrickson, et al., 2016);

- програмне забезпечення як послуга (SaaS) через Google Cloud;
- база даних як послуга, як у випадку з Microsoft Azure для PostgreSQL.

FaaS можна розглядати як гібрид платформи як послуги (PaaS) та програмного забезпечення як послуги (SaaS): інфраструктура, а також інформація повністю контролюється хмарним постачальником, навіть якщо програмне забезпечення управляється клієнтами або розробниками (Korhonen, 2017) (Kumari, Sahoo, Behera, Misra, & Sharma, 2021).

Системи FaaS або «Функція як послуга» дозволяють розробникам створювати та розміщувати застосунки (що складаються з однієї функції або набору функцій), які динамічно масштабуються.

Ці застосунки призначені для виконання однієї функції на вимогу, а потім відкладають запуск до наступного разу, коли вона знадобиться. Такі системи забезпечують величезну економію коштів для розробників застосунків, дозволяючи їм платити лише за ті екземпляри, на яких виконуються їхні безсерверні застосунки [83].

Так як для розробки системи було обрано платформу Amazon Web Services то для реалізації цієї архітектури можна було б скористатися наступним:

- AWS Lambda – це сервіс для виконання коду без необхідності управління інфраструктурою. Можна завантажувати свою функцію до Lambda та викликати її відповідно до потреб застосунку,
- Amazon API Gateway – сервіс для створення, публікації, підтримки, моніторингу та захисту API (можна використовувати API Gateway для створення HTTP або WebSocket API для виклику ваших Lambda функцій через інтернет),
- Amazon DynamoDB – повністю керована NoSQL база даних, яка забезпечує швидкий та надійний доступ до даних для застосунку (можна використовувати DynamoDB для зберігання та керування даними, які обробляються вашими Lambda функціями) та інші (рисунок 3.4).

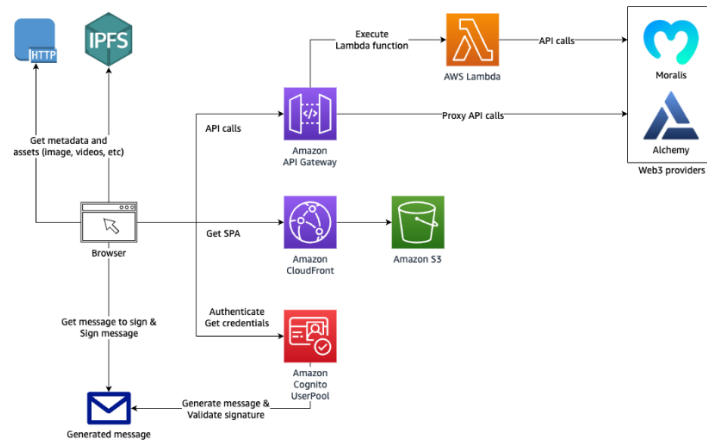


Рисунок 3.4 Схema Serverless архітектури для розгортання на Amazon Web Services [84]

Використання цієї архітектури надає наступні переваги:

- масштабованість – Serverless дозволяє автоматично масштабувати ресурси в залежності від навантаження, що робить його ідеальним для проєктів зі змінним навантаженням;
- швидкість розгортання – Serverless дозволяє швидко розгорнути функції без необхідності управління інфраструктурою.

Але у Serverless архітектури є свої недоліки:

- обмеження функцій – існують обмеження на тривалість виконання функцій, розмір пам'яті та інші обмеження, які можуть бути несумісними з деякими типами задач (в тому числі задач по розрахунку якості повітря);
- складність відлагодження: Відлагодження serverless застосунків може бути складнішим через їх розподілений характер.

Так як в нашої системи регулярно працюватиме збір даних з різних ресурсів та є велика кількість складних розрахунків (таких як запуск моделі CALLPUFF) [85] всі переваги цієї архітектури нівелюються.

Альтернативою Serverless архітектури може стати мультисервісна архітектура. Мікросервісна архітектура – це підхід до розробки програмного забезпечення, де програмний застосунок розбивається на невеликі, самостійні модулі, відомі як мікросервіси. Кожен мікросервіс виконує конкретну функцію

та має власний набір інтерфейсів API для взаємодії з іншими мікросервісами. Мікросервіси взаємодіють між собою через API, що дозволяє їм співпрацювати та обмінюватися даними. Кожен мікросервіс може використовувати власний стек технологій, що дозволяє використовувати найкращі інструменти для конкретних завдань. Якщо один мікросервіс відмовляє, це не призводить до відмови всього застосунку, а лише до проблем в окремій функціональності [86].

Для розгортання мультисервісної архітектури на платформі Amazon Web Services можна скористатися наступним: сервісами контейнеризації Amazon Elastic Compute Cloud (EC2) або Amazon Elastic Kubernetes Service (EKS) , які дозволяють запускати та керувати контейнерами для ваших мікросервісів, Amazon Simple Storage Service (S3) в якості сховища для зберігання об'єктів , такі як файли, зображення та інше, Amazon CloudFront в якості швидкому та ефективному доставці вмісту вебзастосунку, Amazon ALB для балансування навантаження, Amazon DynamoDB та Amazon Aurora як нереляційну та реляційну бази даних відповідно та Amazon ElastiCache для підвищення продуктивності та масштабованість застосунку шляхом кешування даних в оперативній пам'яті та забезпечуючи швидкий доступ[87]. Схема зображена на рисунку 3.5.

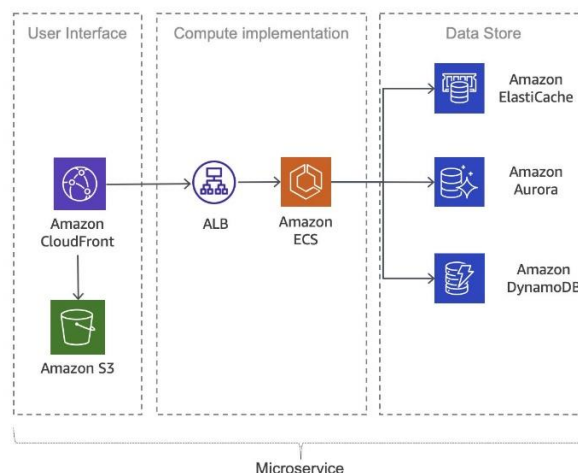


Рисунок 3.5 Схема мультисервісної архітектури для розгортання на Amazon Web Services [88]

До переваг мультисервісної архітектури можна віднести наступне:

- гнучкість: Мультисервісна архітектура дозволяє розділити застосунок на невеликі, незалежні сервіси, що надає більшу гнучкість у розробці, тестуванні та впровадженні нових функцій;

- швидкість розгортання: Можливість незалежного розгортання кожного сервісу дозволяє прискорити процес впровадження змін та оновлень;

- масштабованість: Кожен сервіс може бути масштабований незалежно, що дозволяє оптимізувати ресурси та витрати.

До недоліків мультисервісної архітектури можна віднести наступні пункти:

- складність управління конфігурацією: З більшою кількістю сервісів у системі стає складніше керувати конфігураціями та налаштуваннями. Доведеться вирішити питання версіонування, розгортання та управління налаштуваннями кожного сервісу окремо;

- складність моніторингу та відлагодження: З більшою кількістю сервісів також ускладнюється моніторинг їх роботи та відлагодження випадків неполадок. Доведеться впроваджувати системи моніторингу та журналювання для кожного сервісу окремо;

- системні проблеми із залежностями: Більше сервісів означає більше залежностей між ними. Якщо один сервіс стає недоступним або працює некоректно, це може вплинути на інші сервіси, що може призвести до каскадного ефекту збоїв;

- проблема інтеграції та управління: Інтеграція між сервісами може вимагати значних зусиль у вирішенні питань сумісності, забезпеченні безпеки та забезпеченні спільного управління;

- обмін даними. Зменшується швидкодія системи через накладні розходи на мережеві виклики між сервісами (в тому числі при обміні даними).

Ще одною не менш популярною з точки зору використання в проектах можна назвати монолітну архітектуру. Монолітна архітектура – це тип архітектури програмного забезпечення, де весь програмний продукт розробляється як єдине ціле, зазвичай у вигляді одного великого застосунку. Основна ідея полягає в тому, що всі компоненти застосунку взаємодіють один з

одним без зовнішніх інтерфейсів або середовищ виконання. Основні характеристики монолітної архітектури включають:

- цілісність – усі компоненти застосунку розглядаються як єдине ціле, розробка, впровадження та масштабування ведуться разом;
- внутрішня взаємодія – всі частини застосунку спілкуються одна з одною напрямку, без використання мережових запитів або сервісних викликів;
- монолітний стек технологій – у монолітній архітектурі використовується єдиний стек технологій для всіх компонентів;
- простота розгортання і масштабування – так як застосунок складається з одного компонента, розгортання і масштабування може бути простішим порівняно з розподіленими системами.

До недоліків можна віднести складність масштабування. Монолітні застосунки можуть бути важко масштабувати, оскільки всі їх компоненти взаємодіють тісно між собою. При збільшенні навантаження доводиться масштабувати всю архітектуру, навіть якщо лише деякі компоненти потребують додаткових ресурсів [89].

Альтернативним рішенням, яке було обрано для розробки інформаційної системи, стала сервісно-орієнтовна архітектура (SOA). Це архітектурний підхід до розробки програмного забезпечення, в якому функціональність застосунків розглядається як набір послуг, які можуть бути використані та взаємодіяти між собою за допомогою стандартних протоколів. Основна ідея SOA полягає в тому, щоб розділити функціональність застосунків на окремі сервіси, які можуть бути реалізовані та підтримувані незалежно один від одного [90].

У сервісно-орієнтовній архітектурі функціональність застосунків розбивається на окремі сервіси, які можуть бути незалежними один від одного. Кожен сервіс виконує конкретну функцію і надає зовнішній інтерфейс для взаємодії з іншими сервісами. Кожен сервіс має ясно визначені межі та відокремлений від інших сервісів. Це дозволяє сервісам бути більш автономними та менш залежними від змін у інших частинах системи. Сервісно-орієнтовна архітектура дозволяє легко інтегрувати нові сервіси з існуючими системами, що

дозволяє побудувати застосунки з високим рівнем гнучкості та розширюваності. Завдяки розділенню функціональності на окремі сервіси, сервісно-орієнтовна архітектура дозволяє краще масштабувати систему, оскільки можна масштабувати тільки ті сервіси, які потребують додаткових ресурсів [91].

Основними відмінностями сервісно-орієнтовна архітектура від мікросервісної архітектури є наступні пункти:

- сервісно-орієнтовна архітектура зазвичай створює більш великі та важкі сервіси, які можуть містити різноманітну функціональність; Мікросервісна архітектура, навпаки, розглядає функціональність як набір дрібних, відокремлених сервісів, які керуються принципом «один сервіс – одна функція»;

- у сервісно-орієнтовній архітектурі, великі сервіси можуть мати складні залежності один від одного, що може робити їх важко масштабувати та підтримувати. У мікросервісній архітектурі, кожен сервіс незалежний і може бути масштабованим та підтримуваним окремо;

- у сервісно-орієнтовній архітектурі сервіси можуть використовувати різні протоколи та стандарти для взаємодії між собою. У мікросервісній архітектурі, зазвичай використовуються легкі та стандартні протоколи, такі як HTTP або REST, для взаємодії між сервісами;

- у сервісно-орієнтовній архітектурі керування багатьма великими сервісами може бути складним завданням. У мікросервісній архітектурі, хоча кількість сервісів може бути більшою, кожен сервіс є меншим і простішим у розгортанні та управлінні. Замість того, щоб мати один великий додаток, який включає в себе всі функції, мікросервісна архітектура дозволяє розділити додаток на дрібніші компоненти, які називаються мікросервісами.

Більше відмінностей сервісно-орієнтовної архітектури з мікросервісною архітектурою відображені в таблиці 3.1.

Таблиця 3.1 Таблиця відмінностей сервісно-орієнтовної архітектури від мікросервісної архітектури

	Сервісно-орієнтовна архітектура	Мікросервісна архітектура
Реалізації	різні сервіси зі спільними ресурсами	самостійні та спеціалізовані невеликі сервіси
Підключення	ESB використовує кілька протоколів обміну повідомленнями, таких як SOAP, AMQP і MSMQ	API, Java Message Service, Pub/Sub
Сховище даних	спільне зберігання даних.	незалежне зберігання даних.
Розгортання	впровадження змін вимагає повної перебудови	кожен мікросервіс можна зберігати в контейнерах
Можливість багаторазового використання	повторно використовує сервіси через спільні ресурси	кожен сервіс має свої незалежні ресурси; мікросервіси можна повторно використовувати за допомогою їх API
Гнучке управління	узгоджене управління даними у всіх сервісах.	різні політики керування даними для кожного сховища.

Хоча мікросервісна архітектура набула широкого визнання в останні роки, сервісна орієнтована архітектура має свої переваги, особливо в певних випадках та для поточної інформаційної системи:

— більша взаємодія сервісів: У SOA, сервіси зазвичай взаємодіють між собою більш глибоко та в тіснішому співробітництві, оскільки вони можуть мати більшу кількість спільних елементів та функціональних можливостей;

- більша єдність даних та стандартизація: SOA сприяє розробці спільних стандартів для обміну даними між сервісами, що дозволяє забезпечити більшу єдність даних та керування ними в різних частинах системи;
- більш просте управління сервісами: У SOA, управління меншою кількістю великих сервісів може бути простішим у порівнянні з управлінням великою кількістю дрібних мікросервісів, які потребують більшої уваги до деталей;
- менша складність інфраструктури: SOA може мати меншу загальну складність інфраструктури порівняно з мікросервісною архітектурою, оскільки кількість сервісів може бути меншою, але кожен сервіс може бути складним та функціонально багатшим.

3.8 Висновок

Проаналізовано існуючі провайдери хмарних послуг таких як: Google Cloud Platform, Microsoft Azure та Amazon Web Services. Незважаючи на недоліки з політикою ціноутворення Amazon Web Services є лідером за розміром та має найширший географічний охоплення, представлений в багатьох регіонах та дата-центрах, є інтегрований з різноманітними технологіями та мовами програмування, має найширший вибір спеціалізованих послуг та інновацій, має потужну глобальну мережу. Тому в якості провайдера хмарних послуг та провайдера серверів є саме Amazon Web Services.

Були розглянуті протоколи обміну даними UDP , TCP та HTTP. HTTP є основним протокол для передачі вебсторінок, зображень, аудіо- та відеофайлів, а також для взаємодії з вебсерверами через API. HTTP може працювати з захищеними з'єднаннями та підтримує механізми кешування, що є важливим для інформаційної системи.

Через наявність великої кодової бази, написаної на різних мовах програмування підтримка та розширення системи унеможливорюється. Для

полегшення розробку інформаційну систему необхідно обернути в Docker. Ця платформа вирішує проблему шляхом ізоляції компонентів в окремі контейнери всередині одної фізичної машини.

Проаналізувавши архітектури мережових протоколів, для інформаційної системи обрано Representational State Transfer (REST) архітектуру. Вона базується на принципах, які забезпечують стабільність, масштабованість, модульність та простоту взаємодії між компонентами системи. Важливою перевагою Representational State Transfer архітектури є те що ресурси можуть мати різні представлення, наприклад, JSON, XML, HTML тощо. Так як інформаційній системі не потрібні гнучкі запити (які надає GraphQL), але потрібна швидкодія на користь сервера швидкодії сервера шляхом додавання кешування запитів, яке легше реалізувати саме з REST.

Проаналізувавши існуючі архітектури програмного забезпечення для інформаційної системи було обрано сервісно-орієнтовну архітектуру (SOA). Серед її переваг є те що:

- це архітектурне рішення сприяє розробці спільних стандартів для обміну даними між сервісами, що дозволяє забезпечити більшу єдність даних та керування ними в різних частинах системи;
- управління меншою кількістю великих сервісів може бути простішим у порівнянні з управлінням великою кількістю дрібних мікросервісів, які потребують більшої уваги до деталей;
- SOA може мати меншу загальну складність інфраструктури порівняно з мікросервісною архітектурою, оскільки кількість сервісів може бути меншою, але кожен сервіс може бути складним та функціонально багатшим.

РОЗДІЛ 4. РЕЗУЛЬТАТИ РОБОТИ СИСТЕМИ

Інформаційна система моніторингу якості повітря – це комплексний інструмент, який призначений для збору, аналізу, візуалізації та поширення даних про якість повітря в певній місцевості. Ця система має на меті забезпечити спостереження за станом довкілля та надати користувачам доступ до інформації, необхідної для прийняття обґрунтованих рішень щодо захисту здоров'я та навколишнього середовища.

На етап парсингу та агрегації даних в інформаційній системі моніторингу якості повітря включає в себе обробку та конвертацію даних з різних джерел в єдиний формат, а також комбінування цих даних для отримання комплексної інформації. Для даних з супутникових знімків це тайлінг даних та їх складання до сховища. Для даних з наземних станцій – це конвертація всіх показників забруднення до єдиних одиниць вимірювань. Після обробки дані в залежності від типу складаються до бази даних або до хмарного сховища. Схема взаємодія експерта з інформаційною системою зображена на рисунку 4.1.

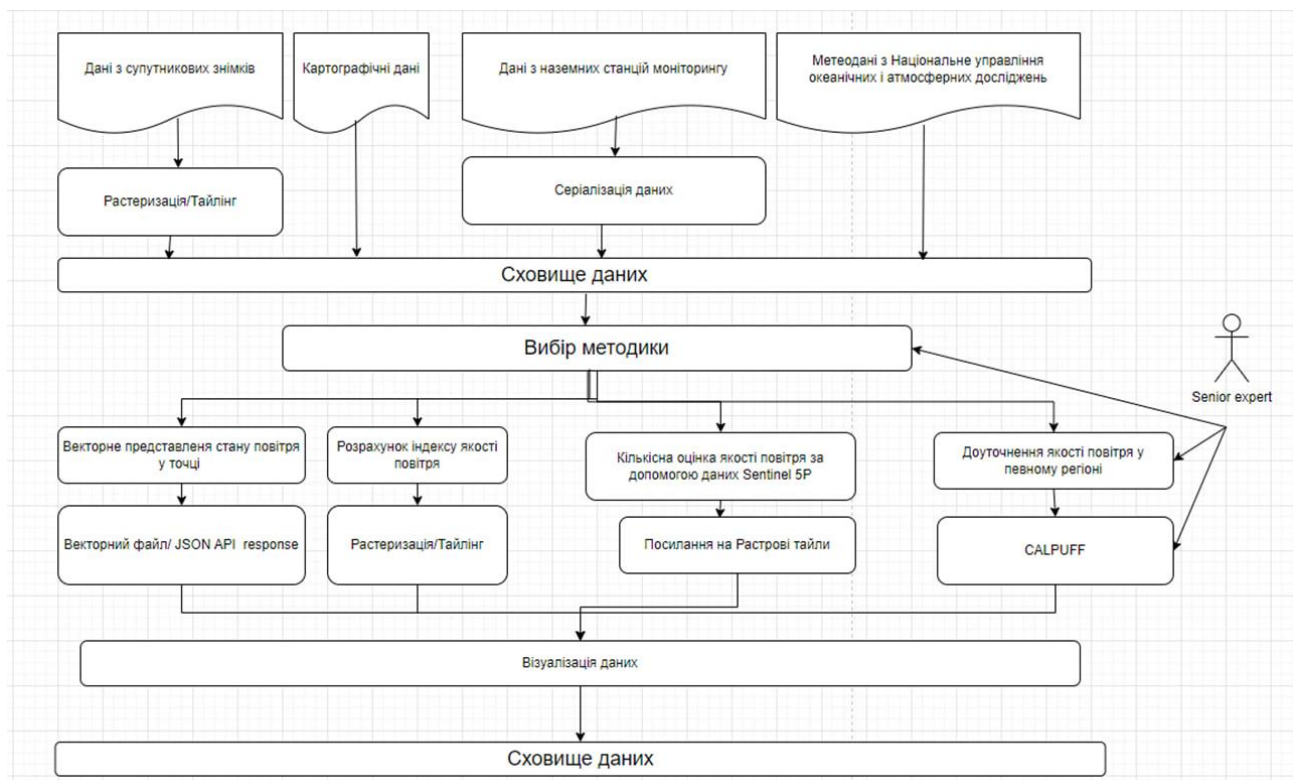


Рисунок 4.1 Схема агрегації та постобробки даних інформаційною системою

На етапі постобробки експерт обирає методику оцінки якості повітря серед яких: векторне представлення стану повітря у точці, розрахунок індексу якості повітря, кількісна оцінка якості повітря за допомогою даних з місії Sentinel 5P або доуточнення якості повітря у певному регіоні. В залежності від обраної методології формує відповіді у вигляді векторного файлу (формату GeoJson) що віддається засобами REST API, посилання на розраховані тайли та розраховану та доуточнену інтерполяційну сітку відповідно. Дані надаються експерту. За необхідністю перезберігаються до бази даних або хмарного сховища.

4.1 Моделювання розповсюдження забруднення використовуючи модель CALPUFF та IDW алгоритму

У поєднанні з IDW, модель CALLPUFF може використовувати дані з мережі точок вимірювання концентрацій забруднень, але використовувати алгоритм IDW для оцінки значень концентрацій в тих місцях, де немає прямих вимірів. Це може бути корисно для створення більш повного образу розподілу забруднень у повітрі та для визначення впливу цих забруднень на конкретні області.

Основні кроки використання IDW в поєднанні з моделлю CALLPUFF можуть виглядати наступним чином:

- збір даних: Збір вихідних даних про концентрації забруднень у повітрі з мережі вимірювальних станцій;
- очистка даних: Перевірка даних на наявність відсутніх значень, аномальних даних або помилок. Видалення або виправлення неправильних значення;
- виконання інтерполяції з використанням IDW: Застосування метод IDW до набору даних для створення контурних карт концентрацій забруднюючих речовин у повітрі;
- використання IDW-інтерпольованих даних у моделі CALPUFF: Введення IDW-інтерпольованих даних як початкові умови або вхідні дані для

моделі CALPUFF. Це може бути зроблено вручну або за допомогою спеціального програмного забезпечення;

- запуск моделі CALPUFF: Запуск моделі CALPUFF з використанням введених IDW-інтерпольованих даних;
- аналіз результатів: Оцінка результати моделювання CALPUFF, порівняння їх з фактичними даними та іншими методами моделювання;
- виправлення та уточнення: Внесення коректив в дані чи параметри моделі для поліпшення точності та надійності результатів .

На рисунку 4.2 зображена схема отримання доуточненого прогнозу забруднення з використанням даної моделі.

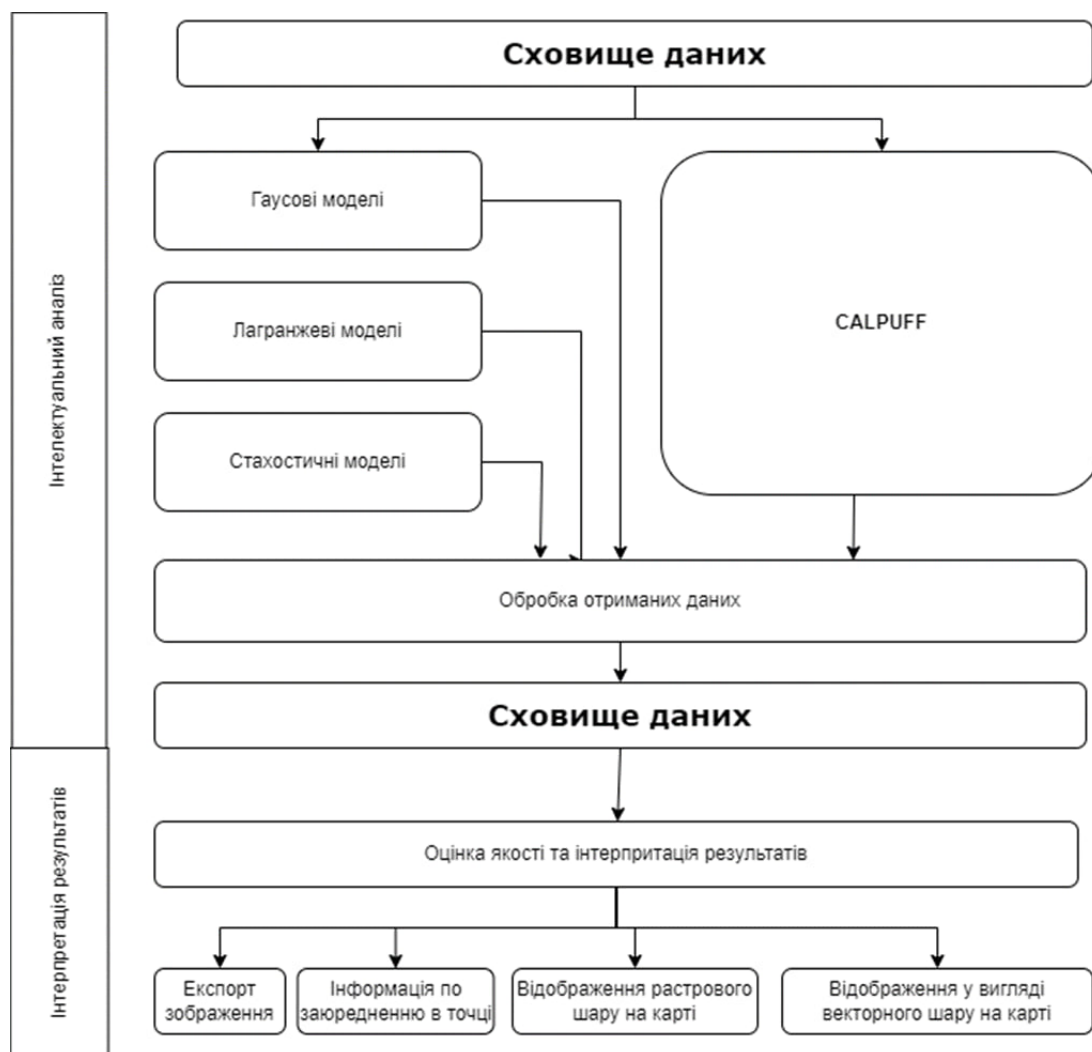


Рисунок 4.2 Схема роботи моделей доуточнення якості повітря у певному регіоні

Ця комбінація може допомогти уточнити модельний прогноз та надати більш повну картину розподілу забруднень у повітрі, що є важливим для оцінки ефектів забруднення на здоров'я та навколишнє середовище.

Отриманий результат було растерізовано для швидкості відображення та була зроблена маска для отримання шлейфу, шляхом фільтрація нульових значень. Результат зображений на рисунку 4.3.

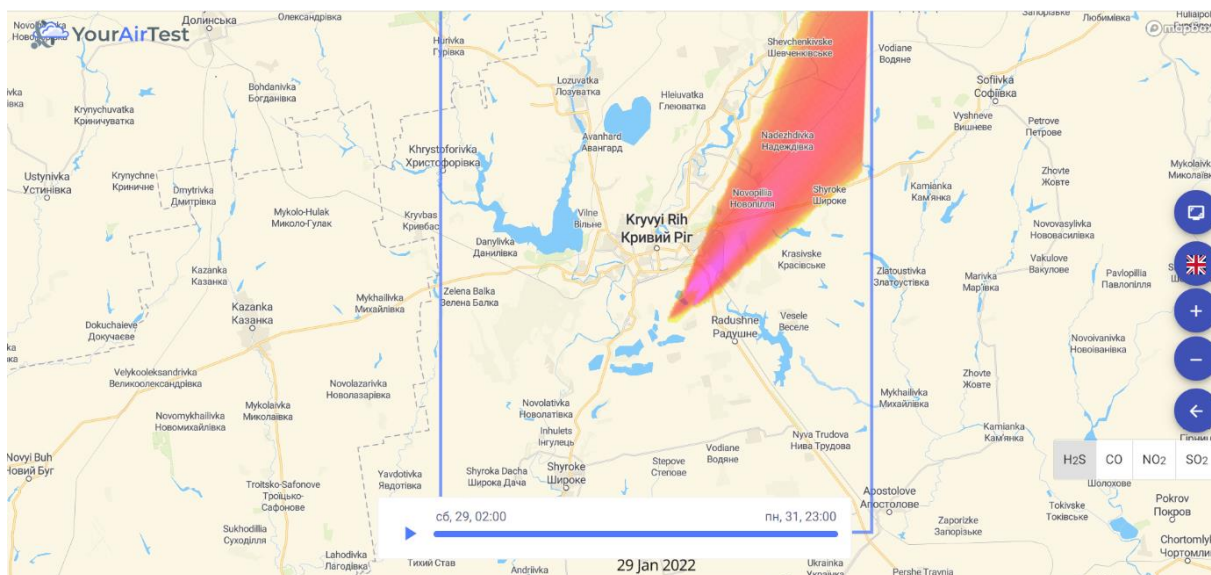


Рисунок 4.3. Результати моделювання з використанням CALPUFF та IDW алгоритму

Для того, щоб дізнатися значення індексу забруднення в точці, відмінній від точки розташування станції, яка надала інформацію про забруднюючі речовини, було обрано алгоритм інтерполяції IDW, а саме його реалізацію в GDAL. GDAL – це бібліотека-транслятор, яка може читати та записувати геопросторові дані як у растровому, так і у векторному форматах. Вона забезпечує уніфіковану модель даних для цих форматів, що дозволяє застосункам працювати з різними форматами геопросторових даних через єдиний API. Окрім бібліотеки, GDAL також включає низку утиліт командного рядка для маніпулювання та обробки геопросторових даних [92].

4.2 Інформаційна технологія прогнозу погоди для оцінки якості повітря

Всі операції з векторними та растровими даними виконуються у віртуальній пам'яті бібліотеки, що дозволило уникнути запису файлів на жорсткий диск. Передавши полігон та тип індексу забруднення для розрахунку, в результаті отримуємо растр з результатами інтерполяції.

В якості моделі прогнозування погоди було обрано Weather Research and Forecasting (WRF) – систему чисельного прогнозування погоди наступного покоління, розроблену для атмосферних та оперативних прогнозів. WRF може виконувати моделювання на основі фактичних атмосферних умов або ідеалізованих умов. WRF пропонує гнучку та обчислювально ефективну платформу для оперативного прогнозування, яка відображає останні досягнення у фізиці, чисельних рішеннях та асиміляції даних, зроблених розробниками з широкої спільноти дослідників [93]. Схему WRF візуалізовано на рисунку 4.4.

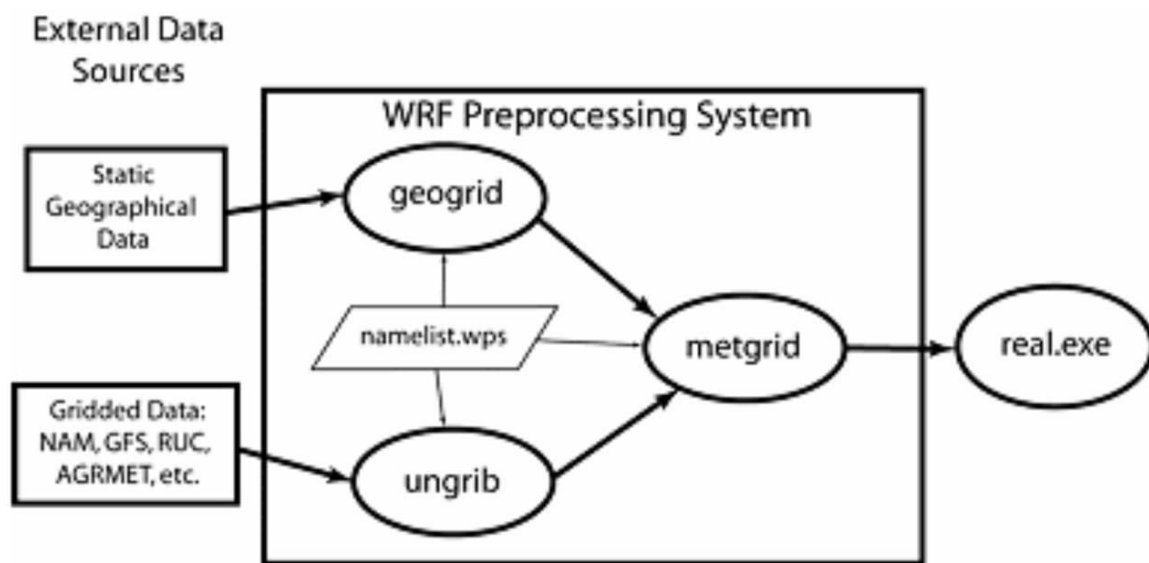


Рисунок 4.4: Схеми роботи WRF

Для коригування моделі дані її прогнозів за минулі періоди порівнювалися з реальними показаннями державних метеостанцій України. За допомогою мови програмування Python між цими точками була проведена інтерполяція. Ці моделі порівнювалися з даними реальних датчиків за такими параметрами, як Температура (2 м від землі); вітер U; вітер V; вологість; тиск[94].

Було вирішено порівняти результати прогнозу моделі з прогнозом авторегресійних моделей, які використовуються для опису стаціонарних стохастичних процесів. Модель ARMA (Autoregressive moving average) є комбінацією моделей AR та MA. AR(p) – модель, що описує досліджуваний процес в момент часу, який залежить від значень p попередніх часових інтервалів [95]:

$$y_t = \beta_1 t_{t-1} + \beta_2 t_{t-2} + \dots + \beta_p t_{t-p} + \varepsilon_t \quad (4.1)$$

де $\beta_1, \beta_2, \dots, \beta_p$ – константи, ε_t – випадкова похибка.

Модель MA представляє стаціонарний процес як лінійну комбінацію послідовних значень білого шуму. Ця модель корисна як доповнення до моделей авторегресії для більш детального опису шумової складової. Модель виражається наступним рівнянням:

$$y_t = \varepsilon_t - \gamma_1 \varepsilon_{t-1} - \gamma_2 \varepsilon_{t-2} - \dots - \gamma_q \varepsilon_{t-q}, \quad (4.2)$$

де $\gamma_1, \gamma_2, \dots, \gamma_q$ – параметри моделі.

На графіку автокореляції на Рисунку 4.4 видно, що дані не мають тенденції, що дозволяє зробити висновок про стаціонарність досліджуваного ряду (стаціонарний ряд повинен мати постійне середнє значення і коливатися навколо цього середнього значення з постійною дисперсією).

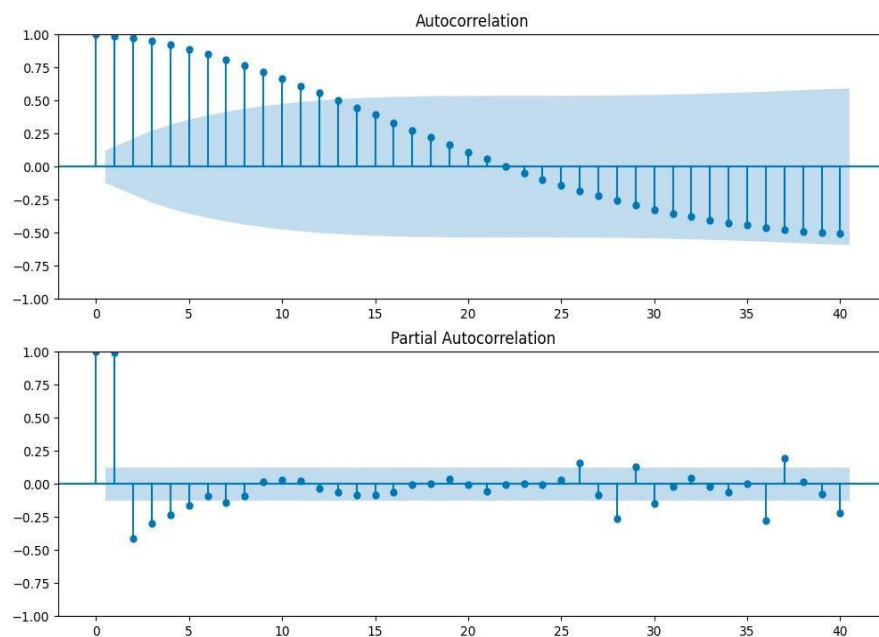


Рисунок 4.4 Графік автокореляції та часткової автокореляції

Оскільки часовий ряд є стаціонарним, модель ARIMA (Авторегресійне інтегроване ковзне середнє) має бути використана зі ступенем диференціації 0. Вона використовує різницю початкових спостережень для того, щоб зробити часовий ряд стаціонарним. ARIMA має 3 вхідні параметри: p – порядок моделі авторегресії, d – ступінь диференціювання, q – порядок моделі ковзного середнього.

Для порівняння ми використали модель ARIMA(3, 0, 3) для прогнозування температури і порівняли її з WRF.

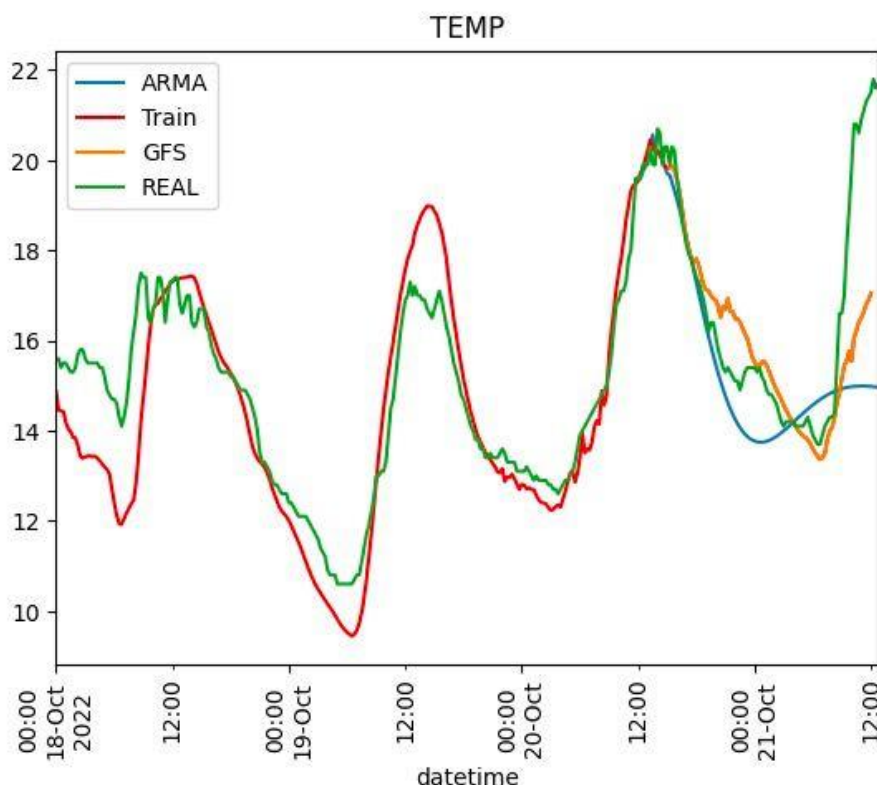


Рисунок 4.5 Порівняння результату прогнозу

На рисунку 4.5 показано, що різниця між WRF та ARIMA є меншою за один градус, тому для оцінки якості прогнозу були розраховані середньоквадратичні помилки: WRF-GFS модель має приблизно 1,2858, тоді як ARIMA модель має лише 1,33, тобто в середньому WRF є більш точною, ніж ARIMA модель, але її можна покращити за допомогою постобробки, наприклад, Multi-sensor Advection Diffusion, що використовує супутникові спостереження для виявлення хмар, а також для виявлення та використання хмар [96].

4.3 Інформаційна технологія парсингу та агрегації даних в системі

Для відображення проміжних результатів та форматування вихідних файлів системи було обрано мову програмування Python у версії 3.7. Використання Python надає переваги завдяки можливості розбивати програми на модулі, що дозволяє їх легко використовувати в інших програмах. Однією з переваг Python є його велика бібліотека стандартних модулів, що становить чудову основу для створення нових застосунків. Крім того, широка спільнота Python сприяє легкості вивчення та вирішенню проблем. Стандартні модулі містять різноманітні зручні інструменти, включаючи засоби для роботи з файлами, системні виклики, інтерфейси до різних графічних бібліотек та мережових з'єднань, що робить Python потужним і гнучким інструментом для розробки програмного забезпечення.

Django – це високорівневий вебфреймворк на основі Python, який має на меті допомогти розробникам створювати вебзастосунки якомога ефективніше, від концепції до завершення. Один з основних фокусів Django – безпека, і він надає розробникам інструменти, які допомагають уникнути поширених пасток і вразливостей безпеки. Відповідає архітектурній схемі моделі шаблонних представлень (MTV). Основна мета Django – полегшити створення складних, керованих базами даних вебсайтів. Фреймворк робить акцент на повторному використанні та «зв'язності» компонентів, меншій кількості коду, низькому рівні зв'язності, швидкій розробці та принципах SOLID. Django також надає додатковий адміністративний інтерфейс для створення, читання, оновлення та видалення даних, який динамічно генерується шляхом самоаналізу та налаштовується за допомогою моделей адміністратора.

Django – це фреймворк, який гнучко і зручно поєднується з різними вебсерверами, що робить його ідеальним вибором для будь-якого вебпроєкту. Зокрема, Django сумісний з вебсерверами, такими як Apache, Nginx (з використанням WSGI), Unicorn, або Cherokee (з використанням flup). Це

означає, що ви можете вибрати вебсервер, який найкраще підходить для вашого проєкту або відповідає вашим потребам у продуктивності та безпеці. Крім того, Django можна легко налаштувати для роботи з сервером FastCGI, який підтримує такі платформи, як Lighttpd або Hiawatha, а також з іншими WSGI-сумісними вебсерверами. Ця гнучкість дозволяє легко і швидко інтегрувати Django у вашу існуючу інфраструктуру або вибрати оптимальне рішення для вашого проєкту. Django, який є надійним і потужним фреймворком для веброзробки, забезпечує підтримку для різних систем управління базами даних (СУБД). Основними базами даних, які підтримуються Django, є PostgreSQL (одна з найпопулярніших об'єктно-реляційних баз даних з відкритим вихідним кодом, яка володіє розширеними можливостями SQL і забезпечує надійне зберігання та оптимізований доступ до даних), MySQL (СУБД, яка широко використовується веброзробниками, особливо в поєднанні з PHP), SQLite (легковажна вбудовувана СУБД, ідеальна для розвитку прототипів або простих проєктів) та Oracle (високоєфективна корпоративна база даних, яка зазвичай використовується в великих підприємствах). Крім цих основних СУБД, Django також має підтримку для Microsoft SQL Server за допомогою django-mssql, а також може працювати з іншими базами даних, такими як IBM Db2, SQL Anywhere та Firebird, за допомогою зовнішніх інструментів. Ця широка підтримка баз даних робить Django гнучким і варіативним фреймворком для будь-яких потреб у веброзробці [97].

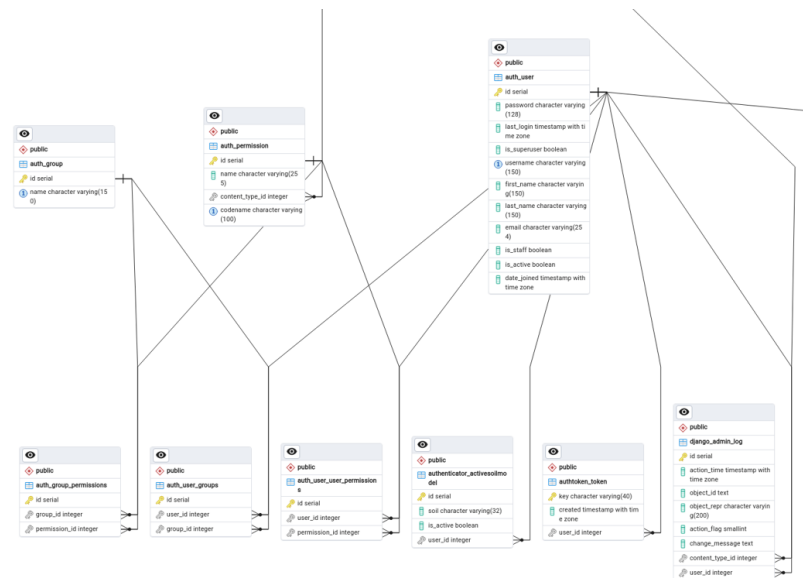
PostgreSQL широко відома своєю міцною архітектурою, високим рівнем надійності та здатністю підтримувати цілісність даних. Її набір функцій також вважається дуже надійним. Як результат, PostgreSQL заслужила гарну репутацію в галузі. PostgreSQL працює на всіх основних операційних системах, сумісна з ACID з 2001 року і має потужні застосунки, такі як популярний розширювач геопросторових баз даних PostGIS. Не дивно, що PostgreSQL стала реляційною базою даних з відкритим вихідним кодом, яку обирають багато людей і організацій [98]. PostgreSQL оснащена багатьма функціями, які допомагають розробникам створювати застосунки, адміністраторам захищати цілісність даних

і створювати відмовостійкі середовища, а також допомагають керувати вашими даними, незалежно від того, наскільки великим чи малим є набір даних. PostgreSQL намагається відповідати стандарту SQL, коли така відповідність не суперечить традиційним функціям або може призвести до поганих архітектурних рішень. Багато функцій, що вимагаються стандартом SQL, підтримуються, хоча іноді вони мають дещо інший синтаксис або функції.

PostGIS – це розширення просторової бази даних для PostgreSQL, яке дозволяє зберігати просторові дані, такі як точки, лінії та полігони, та маніпулювати ними. Він розширює PostgreSQL новими типами (такими як геометрія, географія та растр), функціями, операторами та індексами, які спеціально розроблені для просторових даних. За допомогою PostGIS користувачі можуть виконувати просторові запити, маніпулювати просторовими даними та виконувати складний просторовий аналіз за допомогою SQL. Він широко використовується в географічних інформаційних системах (ГІС), вебкартографічних застосунках і сервісах, що базуються на визначенні місцезнаходження. До переліку функцій PostGIS 2+ входять:

- обробка та аналіз як векторних, так і растрових даних для сплайсингу, морфінгу, перекласифікації та збору/об'єднання за допомогою SQL;
- просторове відтворення SQL-функцій, викликаних як векторними, так і растровими даними;
- підтримка імпорту та експорту векторних даних з файлів ESRI (Environmental Systems Research Institute) за допомогою командного рядка та графічного інтерфейсу, а також підтримка інших форматів за допомогою сторонніх інструментів з відкритим вихідним кодом;
- командний рядок для імпорту растрових даних багатьох стандартних форматів: GeoTiff, NetCDF, PNG, JPG;
- функції візуалізації та імпорту векторних даних для стандартних текстових форматів, таких як KML, GML, GeoJSON, GeoHash та WKT (Well-known text) з використанням SQL;

Після того, як користувач успішно пройшов авторизацію через сторонній сервіс за допомогою OAuth2, Django застосунок отримає токен доступу від цього сервісу. Токен доступу – це унікальний ключ, який дозволяє вашому додатку звертатися до API стороннього сервісу в ім'я автентифікованого користувача. Отримавши інформацію про користувача від стороннього сервісу, ваш Django додаток зберігає дані до бази даних, що відображено на рисунку 4.7.



Для оптимізації процесу збору та аналізу даних про якість повітря було розроблено агрегатори, які призначені для стандартизації та узагальнення методів отримання та обробки інформації з різних джерел. Це дозволяє забезпечити однаковий формат даних незалежно від джерела їх походження та спростити подальший аналіз.

107

забезпечуючи ефективне збирання, стандартизацію та обробку інформації. Основні джерела даних про якість повітря можуть включати в себе інформацію від датчиків, метеостанцій, даних від місцевих влад та інших джерел. Кожне з цих джерел може мати власний формат даних та спосіб їх отримання. Тому агрегатори створені з метою об'єднання різноманітної інформації в єдиний стандартизований формат для подальшого аналізу.

Системи збирає данні з відкритих API (Application programming interfaces). Це реалізовано за допомогою функцій парсеру та агрегатора. За них відповідають воркери Celery, які забезпечують збір даних з понад тисячі датчиків протягом години.

У Celery, використовується модуль `celery.beat`, щоб запускати періодичні або cron-подібні задачі. Цей дозволяє планувати виконання задач на основі різних розкладів часу. Парсери запускаються на кроні і ставляться в чергу, яка реалізовано за допомогою сервісу RabbitMQ [99]. Схема зображена на рисунку 4.8.

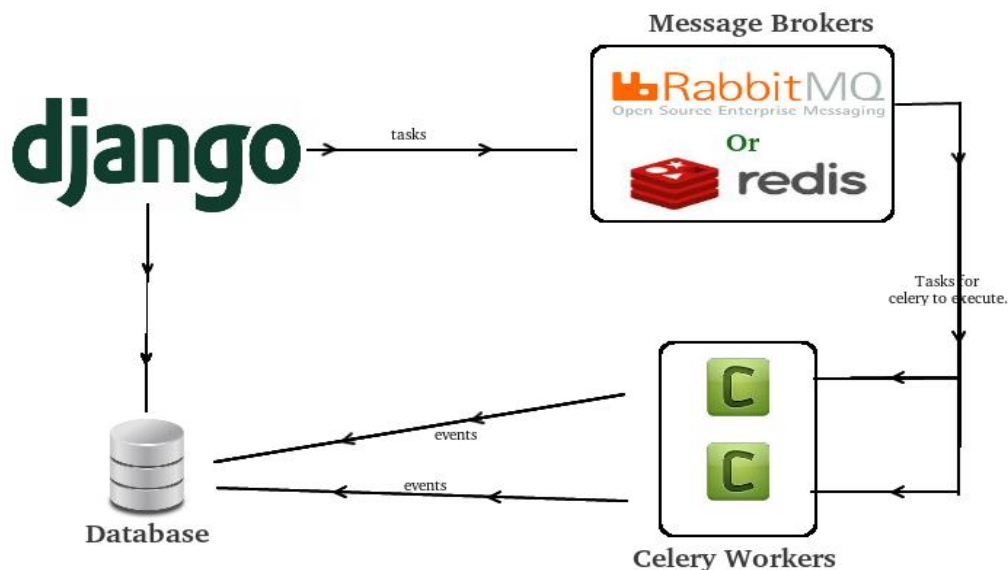


Рисунок 4.8 Схема системи з таск-менеджером Celery [99]

На рисунку 4.9 зображена схема задачі, що зберігається в базі даних.

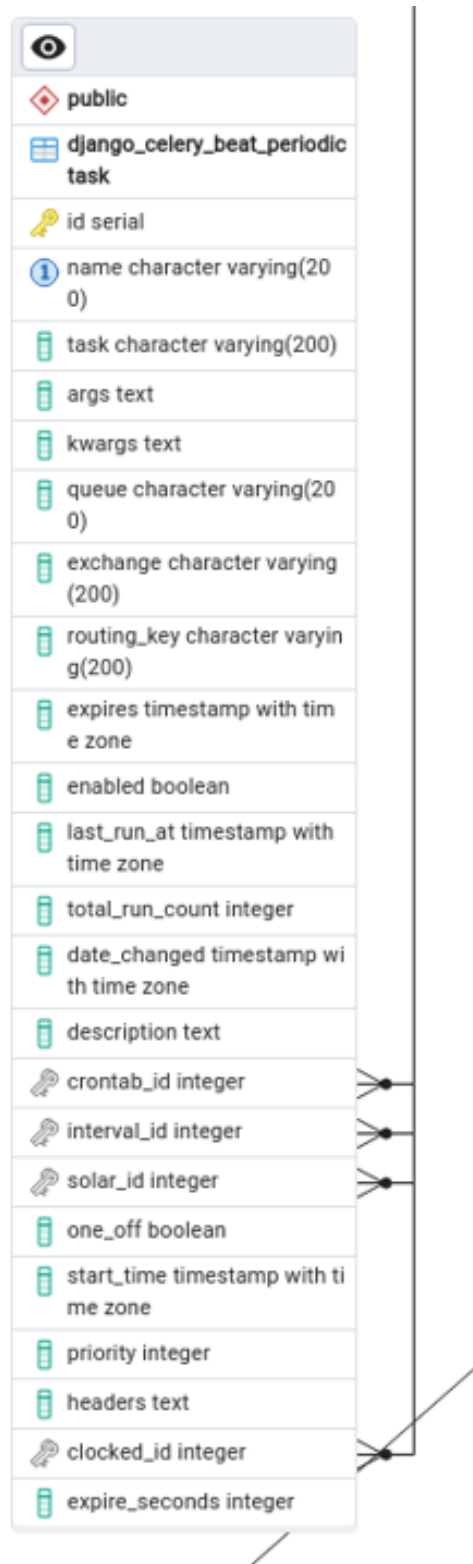


Рисунок 4.9 Модель задачі Celery

Зі схеми можна побачити, що задача Celery можна запускати в таких режимах:

- crontab розклад (потужним інструментом для визначення періодичності виконання задач на основі виразів cron);

- годинний розклад (визначає точний час для виконання задач, які повинні бути виконані в певний момент часу);
- періодичний розклад (задає періодичність виконання задачі за заданим інтервалом часу);
- Сонячний розклад (специфічний тип розкладу, який дозволяє визначати періодичність виконання задач на основі позиції сонця).

На рисунку 4.10 зображені схеми розкладів та параметри для їх конфігурація.



Рисунок 4.10 Схеми розкладу Celery Beat задач

Як спосіб передачі даних використано архітектурний підхід REST, що базується на протоколі HTTP і дозволяє передавати дані у форматах, відмінних від XML. В системі реалізована система кешування даних, що надає можливість прискорити видачу інформації з сервера. Система зберігає всю поведінку користувача, положення карти, обраний режим відображення інформації тощо.

Для забезпечення безпеки передачі даних використовується API-токен, який є унікальним для кожної станції і прикріплюється до всіх запитів.

Запит надсилається потім розбирається парсером, і дані заносяться в поля таблиці бази даних. Оскільки станції є мобільними і надсилають свої GPS координати, парсер розроблений таким чином, що якщо локація станції перемістилася більш ніж на 50 м, то вона розпізнається як нова.

На рисунку 4.11 зображена кількість унікальних станцій зібраних за рік в період з 2020 по 2024 роки.



Рисунок 4.11 Графік збору даних станцій моніторингу якості повітря з 2020 по 2024 роки

Дані по кожній станції накопичуються протягом однієї години, після чого вираховується середній показник по станції. Використовуючи ці значення можна розрахувати значення AQI для станції. За допомогою бібліотеки Gdal створюється Tiff-файл, в який вносяться значення AQI станцій та інтерполяція між ними між ними. Далі команда Gdalwarp створює зображення у форматі Png на якому кольори показують стан якості повітря. Після цього з картинки вирізаються фрагменти, що відповідають доменам країн з картинки. З цих даних генеруються тайли – картинки для різних масштабів сайту. Коли генерація тайлів завершилась, створюється папка на поточну годину для конкретного AQI, а потім ця папка завантажується в Bucket S3 (Amazon Simple Storage Service).

Крім того, результати моделі CALPUFF були збережені у вигляді хмари точок (з мінімальним часовим інтервалом 1 година) у форматі GeoJSON на Amazon S3 (Amazon Simple Storage Service) та відображені у вигляді векторного шару на карті за допомогою бібліотеки Mapbox GL. Для візуалізації представлення даних користувач може перемикатися між шарами за допомогою повзунка часового потоку, а також може вибрати забруднювач. Система дозволяє користувачеві експортувати вибраний шар як растрове зображення на локальний комп'ютер.

4.4 Інформаційна технологія взаємодії системи з користувачем

Програмне забезпечення має тенденцію до зростання, і нові члени команди, які мають різні рівні знань, повинні мати можливість приєднатися до команди і почати працювати над однією частиною програми, навіть якщо вони не знайомі з усією архітектурою.

Ці вимоги передбачають, що програма повинна мати такі основні функції:

- модульна конструкція;
- односпрямований потік даних
- передбачуване управління станом;
- комунікаційний рівень для асинхронних запитів;
- рівень представлення не залежить від базового рівня.

Розвиток модульного дизайну та декомпозиція проекту на менші частини (модулі) зумовлена необхідністю мати можливість легко підключати та виключати наші модулі за будь-якої потреби, не витрачаючи багато часу та не порушуючи структуру проекту, покривати кожну частину програми окремими юніт-тестами та давати можливість уникати конфліктів під час розробки, збільшуючи при цьому чисельність команди. Таким чином, такий підхід розділяє застосунок на блоки функціональності та полегшує його підтримку і тестування.

В якості фреймворку для розробки клієнтської частини було вирішено обрати Angular. Аналізуючи результати дослідження, можна зробити висновок,

що швидкість роботи цих двох фреймворків дуже схожа. Ймовірно, це пов'язано з тим, що Vue.js починав своє існування як відгалуження Angular, і він написаний в дуже схожому стилі з Angular. Основною перевагою Vue.js є його продуктивність та низький поріг входження.

Основними факторами, які впливають на вибір Angular як фреймворку, є його структура та наявність підтримки типізації вищеописаних компонентів з-під коробки.

Тому не варто використовувати Angular при розробці «посадочних сторінок», сервісів з «тонким» клієнтом, невеликих проектів [100].

Було перевірено статистику користування браузерів. У 2023 році користувачі користувалися браузерами в наступному співвідношенні:

- Chrome – 65%;
- Safari – 18.55%;
- Edge – 5.03%;
- Firefox – 2.8%;
- Samsung Internet – 2.63%;
- Opera – 2.53%;
- інші – 3,45.

Дані відображені на рисунку 4.12 [101].

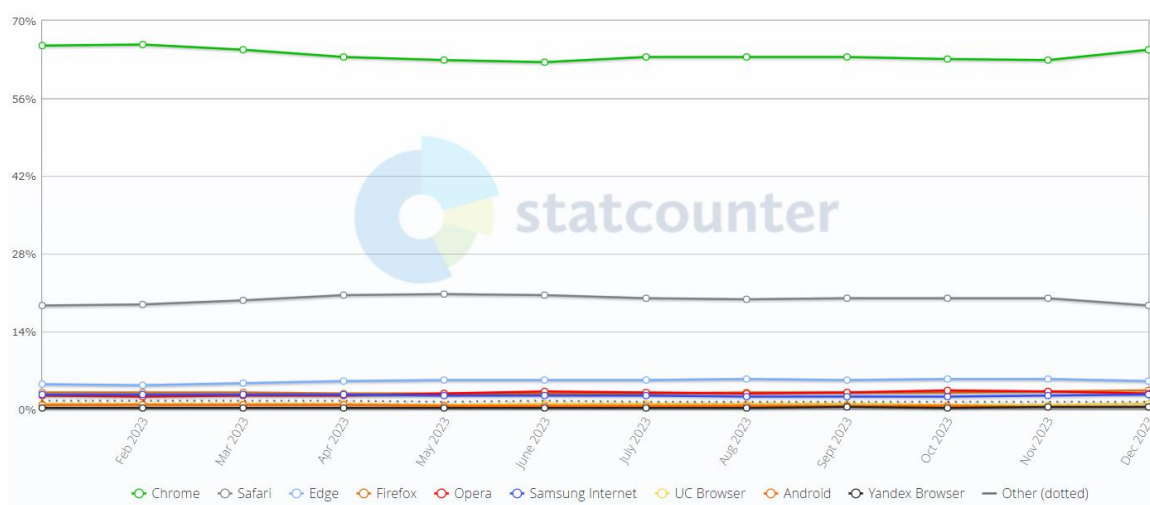


Рисунок 4.12 Співвідношення користувачів браузерів за 2023 рік [101]

Щоб напевно переконатися в можливості відобразити нашу систему користувачами, було азято стаистику найбільш використовуваних версій браузера на 2023 рік, які зображені на рисунку 4.13.

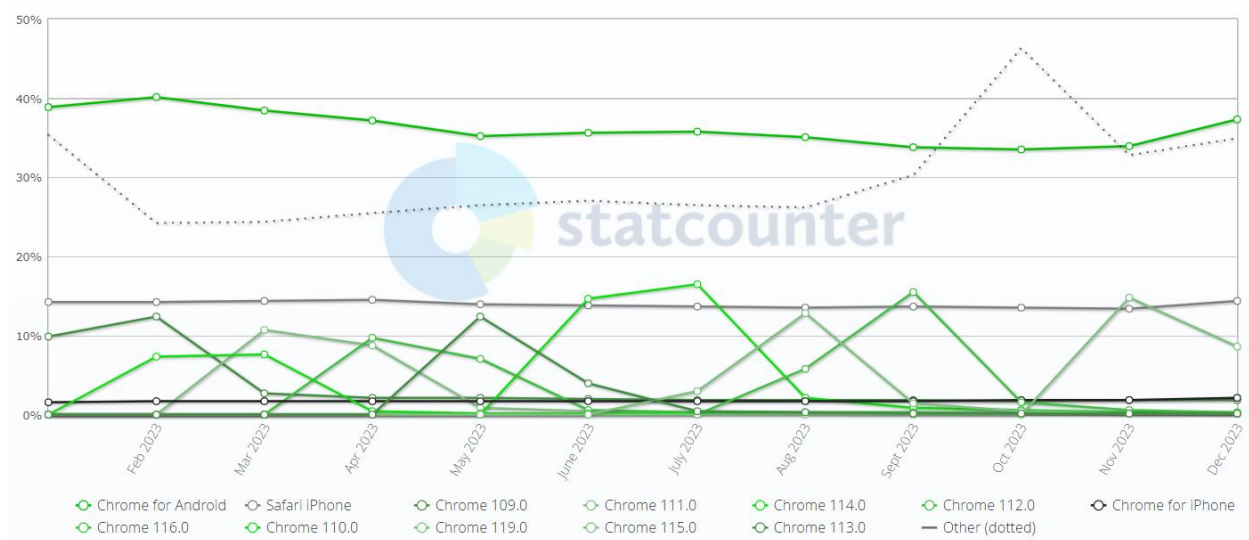


Рисунок 4.13 Співвідношення версій браузерів якими користувалися у 2023 році [102]

З цього графіку видно що левову частку (41,46%) мають мобільна версія Chrome для пристроїв з операційними системами Android та IOS, 16,33 % займають 2 LTS (Long-term support) версії Chrome, 14,52% – Safari iPhone та 4,2 % – Edge версії 122 [102].

Виходячи з цих даних та даними про підтримку браузерів фреймворком Angular [103] можна казати про 93% покриття нашої системи. В 7% входять браузери які теж підтримуються розробленою системою через V8 engine, на якому написані більшість сучасних браузерів.

Застосунок розділений на три шари – презентаційний, реферативний та основний. Презентаційний шар використовується для відображення HTML і обробки взаємодії з користувачем. Абстрактний рівень відповідає за взаємозв'язок між презентаційним та основним рівнями. Основний шар містить логіку роботи ядра програми, наприклад, маніпуляції з даними, зв'язок з API тощо. Кожен рівень поділяється на модулі, впорядковані за функціями. Модулі – це окремі блоки коду, які можна підключати та виключати за потреби. Кожен блок

був відокремлений модулем. Асинхронні сервіси (служби) та управління даними було вирішено винести в окремі модулі, оскільки вони виконують різні завдання. Те ж саме стосується і модулів представлення. Модулі представлення – це окремі блоки, які не залежать від інших модулів. Вони включають необхідні залежності (наприклад, конфігурацію, утиліти тощо) від ядра програми, і роблять це через рівень абстракції. Рівень абстракції забезпечує доступ до ядра програми, завдяки чому можна розділити презентаційний та основний шари.

Вебзастосунок було розроблено з використанням підходу реактивного програмування [104]. Підхід базується на роботі з асинхронними потоками даних (послідовності подій, впорядкованих за часом). Прослуховування потоку називається підпискою. Функції, які ми оголошуємо, називаються спостерігачами. Потік – це об'єкт спостереження. Потоки – це асинхронні HTTP-запити до системи та глобального репозиторію, який виступає посередником між презентаційною частиною (ієрархією вебкомпонентів) та даними, що надходять від сервера. Слухачі – це компоненти верхнього рівня (smart) презентаційного рівня, які відображають дані або передають їх далі по ієрархії компонентів.

На рисунку 4.14 зображена діаграму стану роботи клієнтської частини застосунку після завантаження в браузері.

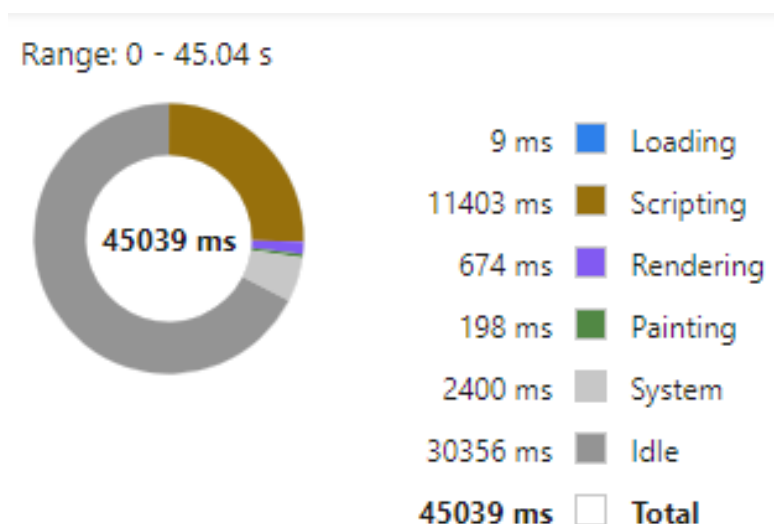


Рисунок 4.14 діаграму стану роботи клієнтської частини застосунку після завантаження в браузері.

Клієнтською частиною інформаційної системи є SPA (Single Page Application), що надає наступні переваги:

- не потрібно кожного разу заново ініціалізувати сторінку;
- дані кешуються в браузері;
- передаються дані, а не шаблони.

Клієнтська частина реалізована з використанням фреймворку Angular, що дало їй наступні переваги: ієрархічна структура модулів [24], розбиття клієнтської частини на 3 шари: презентаційний (відповідає за відображення даних в системі), абстрактний (спосіб взаємодії основного та презентаційного шарів) та основний (відповідає за роботу з даними), оскільки вебзастосунок – SPA управляє передачею даних, а не уявлень.

В системі реалізовано кешування даних, що дає можливість прискорити обмін даними між клієнтом та сервером.

Самі компоненти поділяються на «розумні» та «дампові»: одна частина компонентів може виконувати певні маніпуляції з даними, звертатися до API тощо, а інша – лише відображати дані з батьківського компонента. Використання такого підходу дало наступні переваги:

- можливість повторного використання завжди вважається головною перевагою більшості підходів до програмування;
- дотримання принципу D.R.Y (Don't Repeat Yourself) означає, що ви можете швидко та ефективно додавати однакову функціональність у різні області;
- рефакторинг частини або всього застосунку вимагає змін лише в меншій кількості місць;
- читабельність;
- полегшує покриття тестами.

На рівні компонентів налаштовується односпрямований потік даних (вниз по дереву компонентів надсилаються дані, вгору – події, викликані користувачем під час взаємодії з системою), який показано на рисунку 4.15.

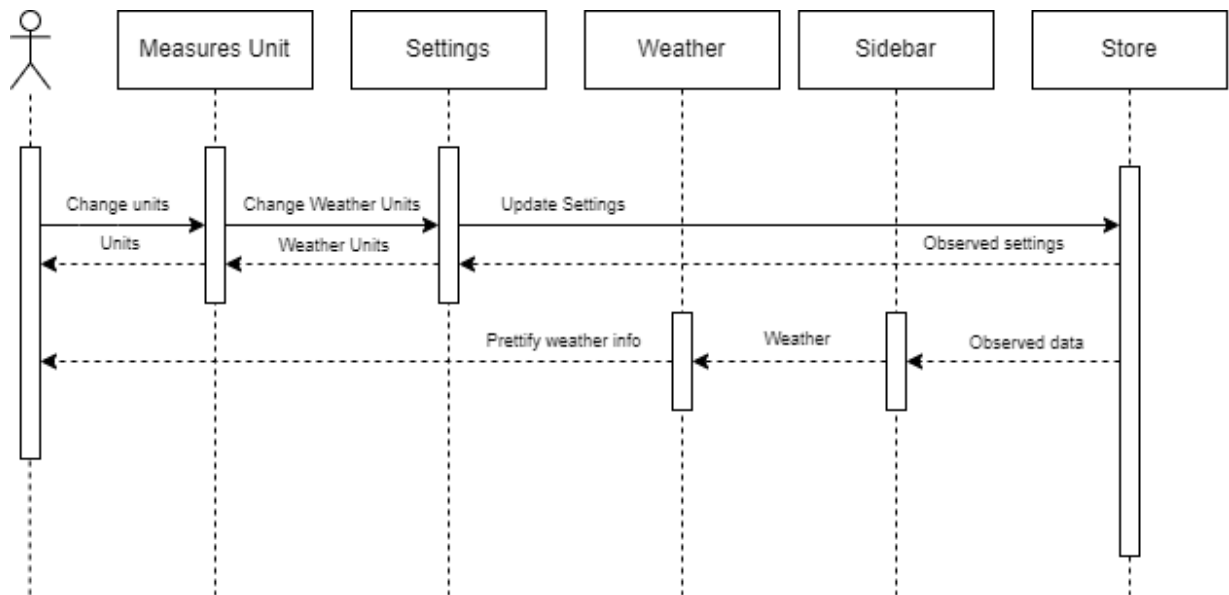


Рисунок 4.15 Поведінкова діаграма користувача

На вихід інформаційної системи подаються такі дані: растрові тайли побудованої інтерполяції індексу якості повітря, що зберігаються на Amazon S3, дані про станції моніторингу якості повітря, передані через HTTP-запити.

Для ефективного контролю за поточним станом додатка розробники можуть використовувати бібліотеку NgRx в середовищі Angular. NgRx забезпечує можливість реактивно відстежувати зміни стану додатка та керувати ними. Ця бібліотека використовує Redux, який базується на концепції одностороннього потоку даних. Вона включає в себе такі ключові елементи, як:

- дії (actions);
- редуктори (reducers);
- ефекти (effects).

Вони спільно допомагають управляти станом застосунку.

Схема менеджера стану NgRx може зображена на відповідному рисунку 4.16. Це дозволяє чітко розуміти, як працює менеджер стану та як кожен його елемент взаємодіє з іншими для забезпечення консистентності та ефективності застосунку.

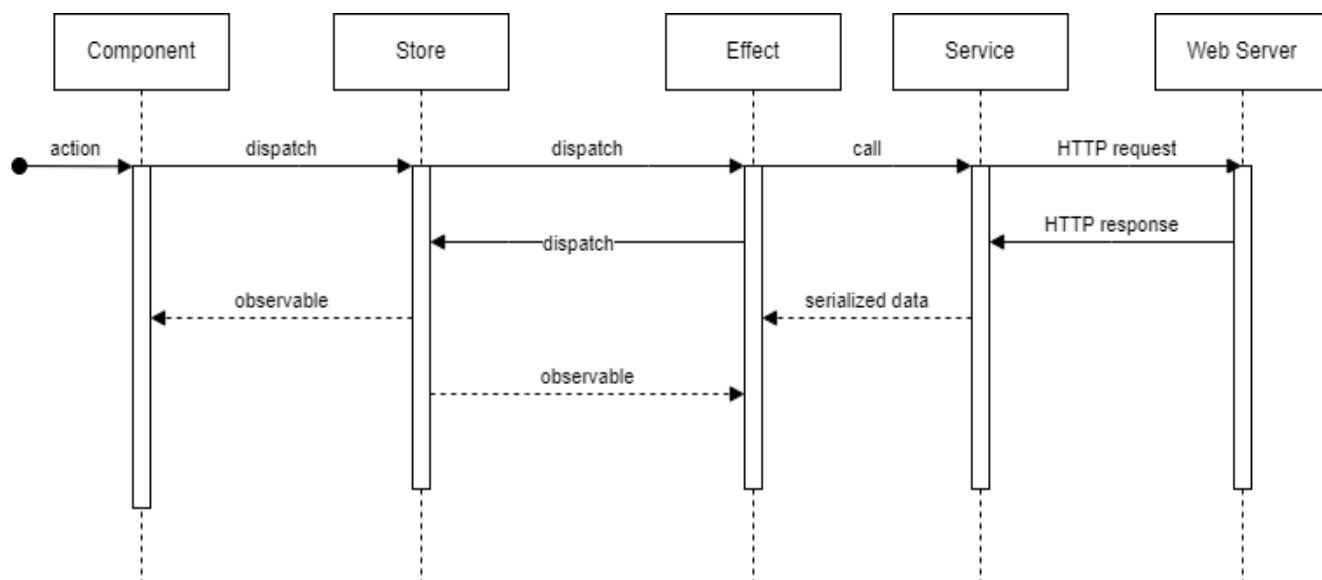


Рисунок 4.16 Схема менеджера стану показана

Система складається з наступних модулів:

- модуль сторінки (використовується для реалізації лінивого завантаження);
- загальний модуль (реалізує фундаментальну функціональність фреймворку, включаючи директиви, сервіси визначення місцезнаходження, що використовуються в маршрутизації, HTTP-сервіси, підтримку локалізації і т.д.)
- модулі з функціональністю (модуль, що містить компоненти, декоратори тощо, які виконують певну функціональність і можуть бути імпортовані в будь-який модуль сторінки).

Система складається з наступних сторінок:

- загальна карта;
- карта лісових пожеж в Україні;
- карта викидів забруднюючих речовин від підприємства (Кривий Ріг).

На загальній карті користувач може: переглянути поточну карту радіаційного забруднення по станціях (понад 4000 станцій) на рисунку 4.17.

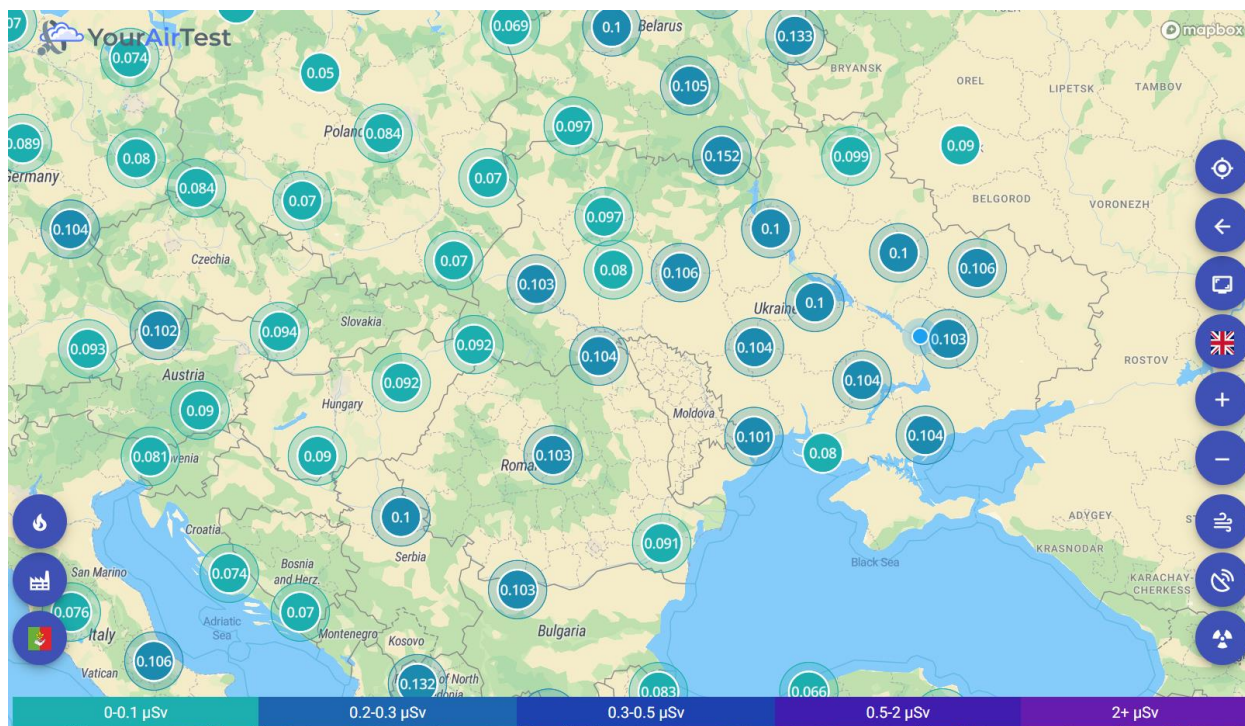


Рисунок 4.17 Мапа станцій що вимірюють радіаційне забруднення

На рисунку 4.18 розрахований індекс якості повітря США AQI по станціях (понад 12000 станцій) та розраховане растрове представлення цього індексу для Європи, Північної та Південної Америки та деяких країн Азії та Океанії на рисунку.

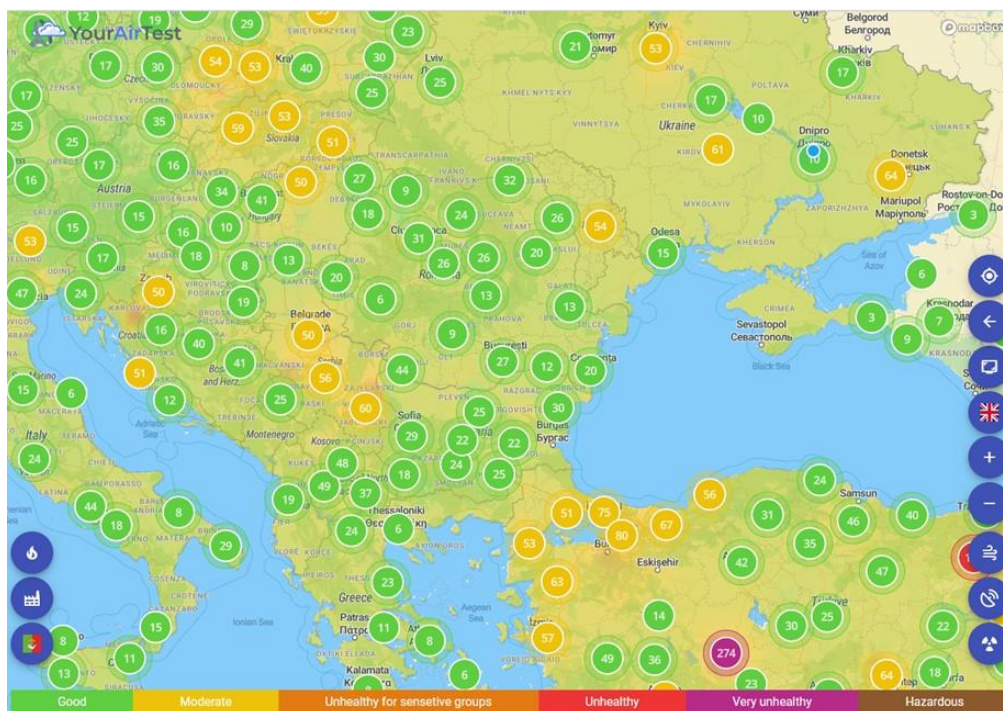


Рисунок 4.18 Мапа станцій що вимірюють радіаційне забруднення

Натиснувши на конкретну обрану станцію користувач системи отримає звіт за тиждень моніторингу (за дні в котрі був проведений забір) та відображений на графіку максимального мінімального та середньодобового значення, що відображено на 4.19. рисунок

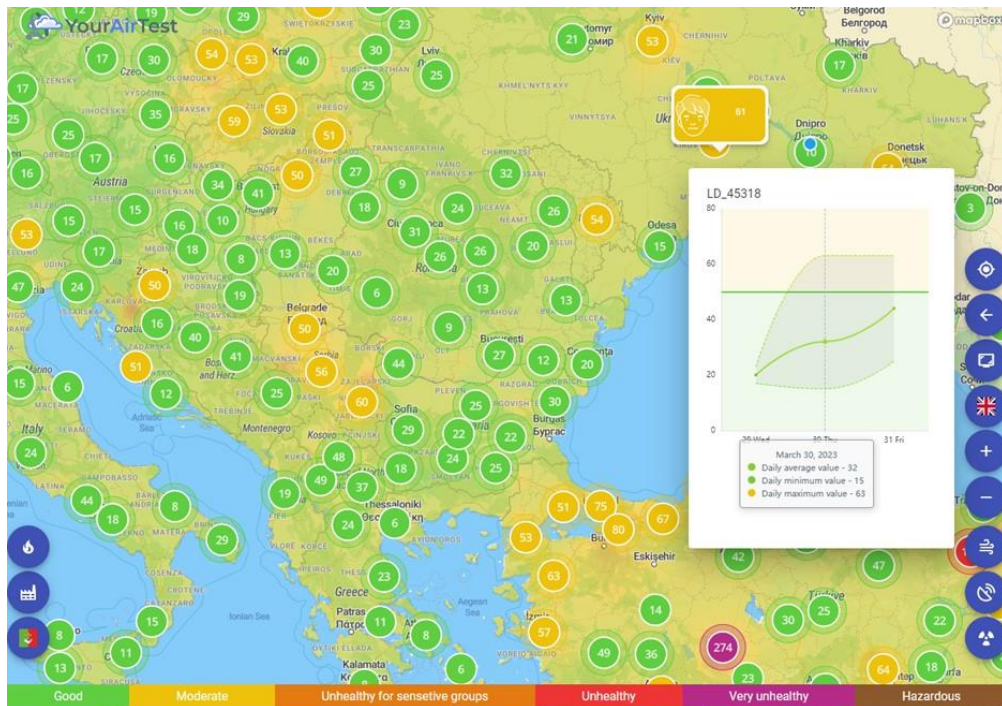


Рисунок 4.19 Мапа станцій що вимірюють радіаційне забруднення

Карта пожеж в Україні (поточний період березень-квітень 2022 року), на якій користувач може: переглянути на карті пожежі за обраною ним датою.

Вибравши пожежу (користувач повинен натиснути на маркер пожежі), користувач відкриває модальний діалог і бачить:

- детальну інформацію про пожежу (площа ураження, місце виникнення пожежі, температура в точці, потужність пожежі, концентрація та кількість викидів речовин у повітря);
- симуляцію розповсюдження пожежі;
- порівняти супутникові знімки до, під час та після пожежі;
- експортувати дані у растровий файл;
- порівняти супутникові зображення до та після пожежі.

На рисунках 4.20 показано карту з пожежами за добу на території України.

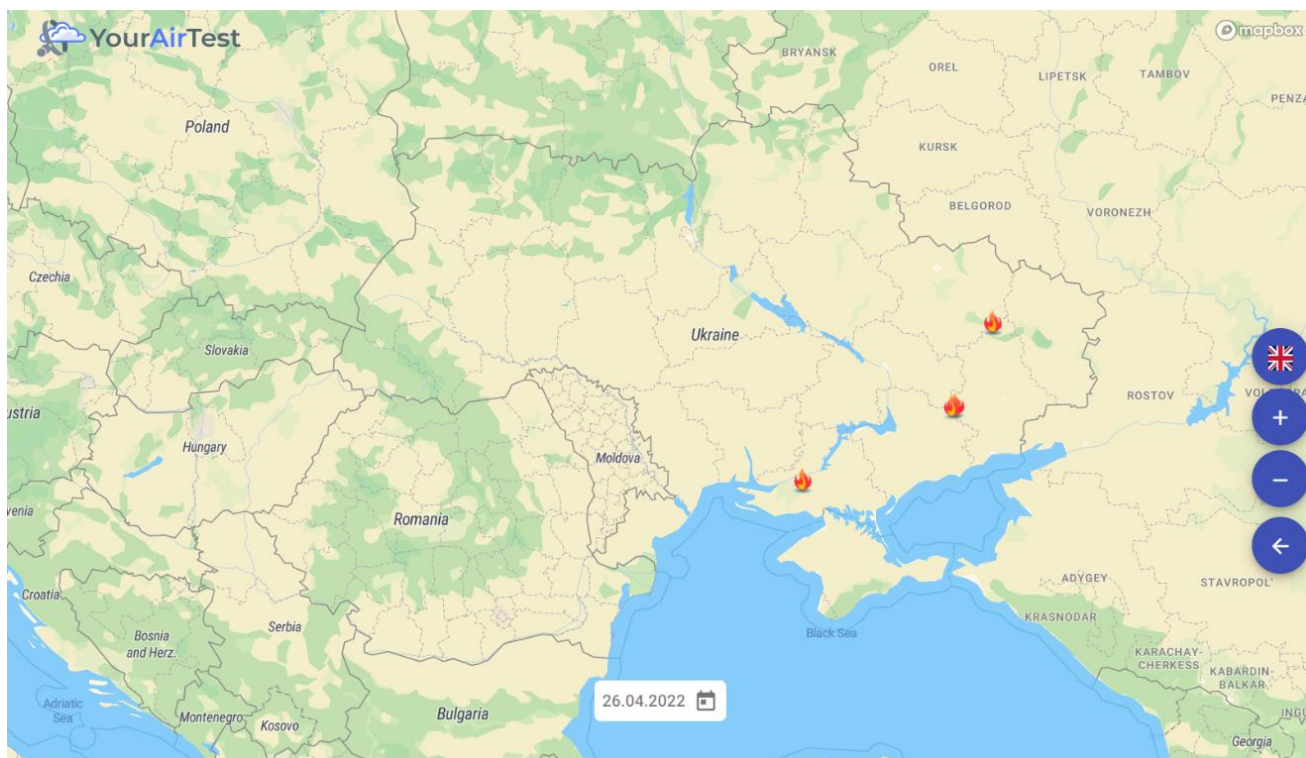


Рисунок 4.20 Мапа пожеж

Діалогове вікно з детальною інформацією про пожежу відображено на рисунку 4.21.

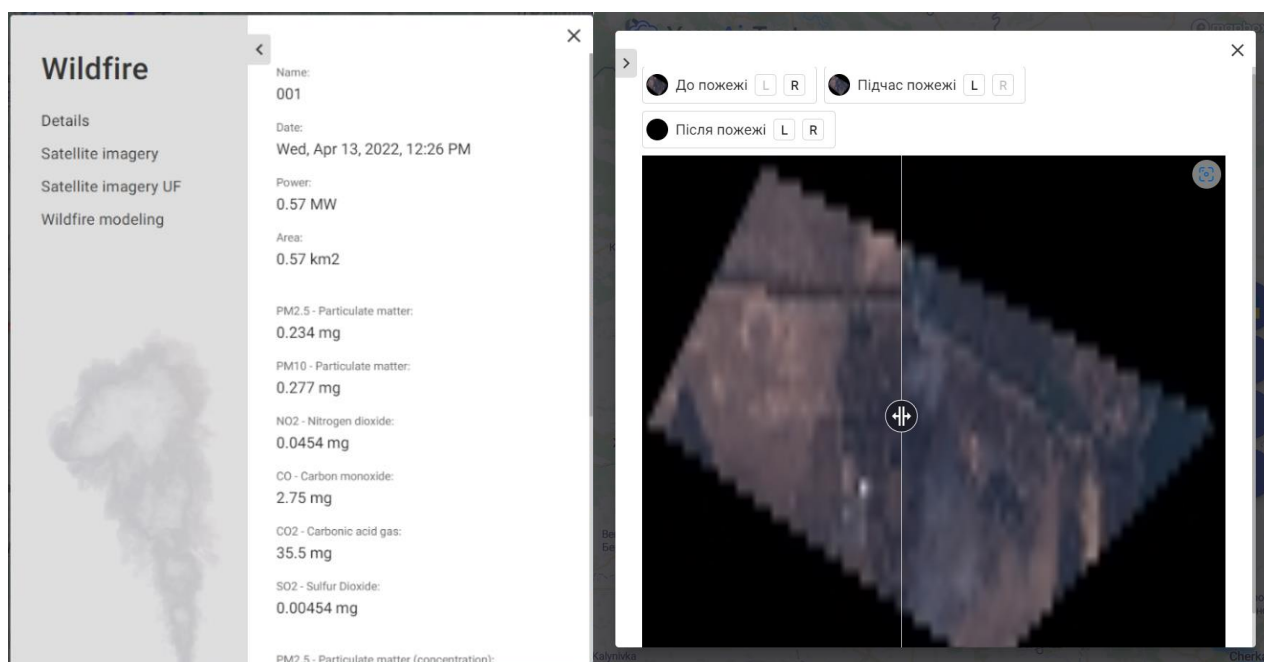


Рисунок 4.21 Панель пожеж: а) загальна інформація та обсяг забруднення; б) порівняння стану місцевості за допомогою супутникових знімків

Для порівняння стану території було обрано супутникові знімки з NASA FIRMS.

Карта викидів забруднюючих речовин у Кривому Розі візуалізує результати використання прогнозної моделі WRF та моделі багатошарового розсіювання з інтегрованою системою моделювання Лагранжа CALPUFF.

Користувач може

- переглянути історію викидів забруднення за допомогою програвача (2 дні, з інтервалом в 1 годину, доступне автоматичне відтворення);

- експортувати обраний етап у вигляді растрового зображення.

Можна вибрати забруднювач, який візуалізується. Можна перемикатися між 4 газами:

- H_2S – сірководень, токсичний газ 3 класу небезпеки, який діє безпосередньо на нервову систему;

- CO – чадний газ, впливає на серцево-судинну систему та центральну нервову систему;

- NO_2 – діоксид азоту, змінює склад крові, знижує вміст гемоглобіну;

- SO_2 – діоксид сірки, призводить до серцево-судинних та онкологічних захворювань, а також завдає непоправної шкоди природі.

На рисунку 4.22 зображений супутниковий знімок місії Sentinel 5P



Рисунок 4.22 Знімки місії Sentinel 5P, що характеризують забруднення повітря частками з діаметром менше 2,5 мікрометрів

Постоброблений знімок включає в себе фільтрацію даних для видалення шуму та непотрібних артефактів, удосконалення контрастності та різкості зображення, а також відображення даних на картах для легшого визначення географічного розподілу забруднення. Знімки можуть бути використані дослідниками, органами влади для моніторингу стану довкілля, оцінки впливу забруднення на здоров'я людини та екосистеми, а також для прийняття рішень щодо заходів з покращення якості повітря та зменшення забруднення атмосфери.

4.5 Тестування інформаційної системи оцінки якості повітря

Інформаційна система моніторингу якості повітря повинна працювати в реальному часі та знаходитися у відкритому доступі тому.

Для тестування продуктивності система необхідно емітувати одночасний вхід N користувачів.

Для розуміння кількості користувачів системи була зібрана інформація про кількість унікальних користувачів у системі за добу.

Дані, що відображаються на рисунку 4.23, було зібрано за допомогою Google analytics.

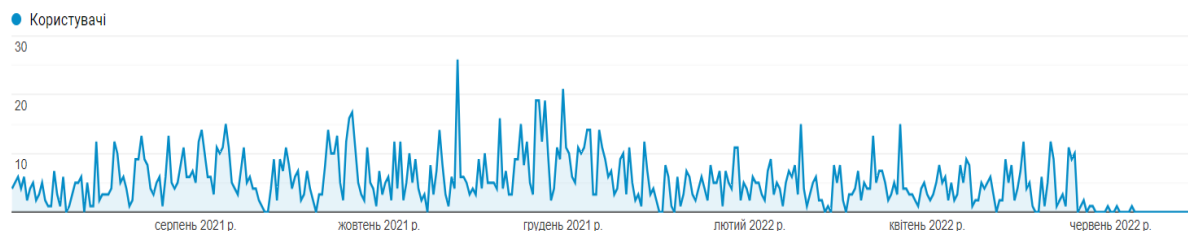


Рисунок 4.23 Кількість відвідувань інформаційної системи за добу в період з 01.06.2021 по 30.06.2022

Можна побачити що добова кількість відвідувачів не перебільшувала 30 користувачів. Тому було вирішено зробити імітацію для випадку коли в систему зайшло 10, 20, 30 та 50 користувачів.

Для початку було вирішено застосувати стратегію поступового навантаження: до системи поступово (майже один за одним) під'єднуються 10 ботів, які починають взаємодіяти з системою. Графік з запланованою кількістю ботів зображений на рисунку 4.24.

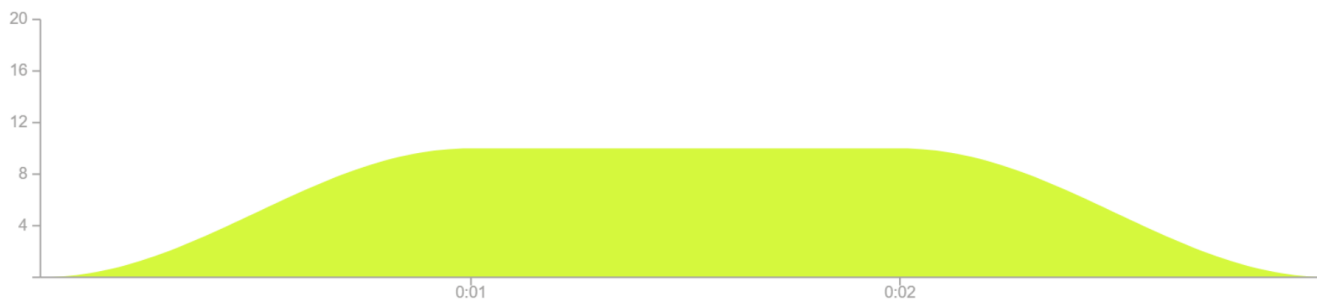


Рисунок 4.24 Графік запланованого навантаження системи під час тестування (кількість ботів – 10)

На рисунку 4.25 зображено фактичну кількість ботів у кожний проміжок часу тестування.

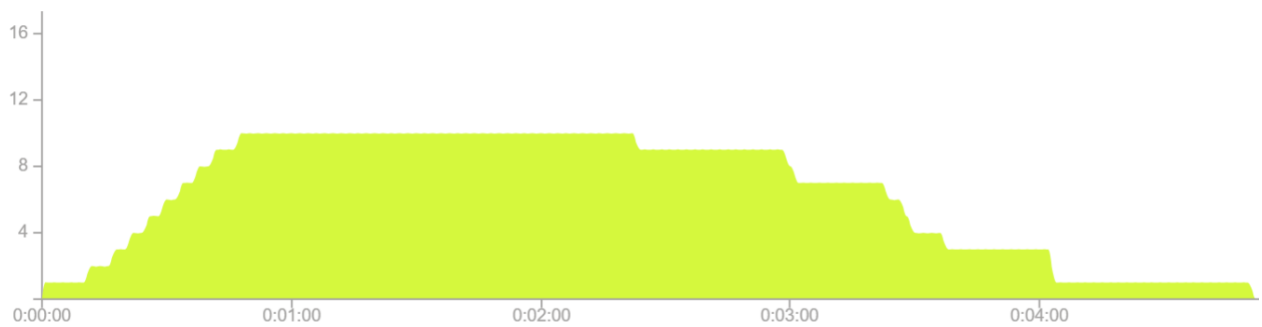


Рисунок 4.25 Графік фактичної кількості ботів у кожен проміжок часу (кількість ботів – 10)

На графіку 4.25 зображено середній час завантаження даних за певний проміжок часу.

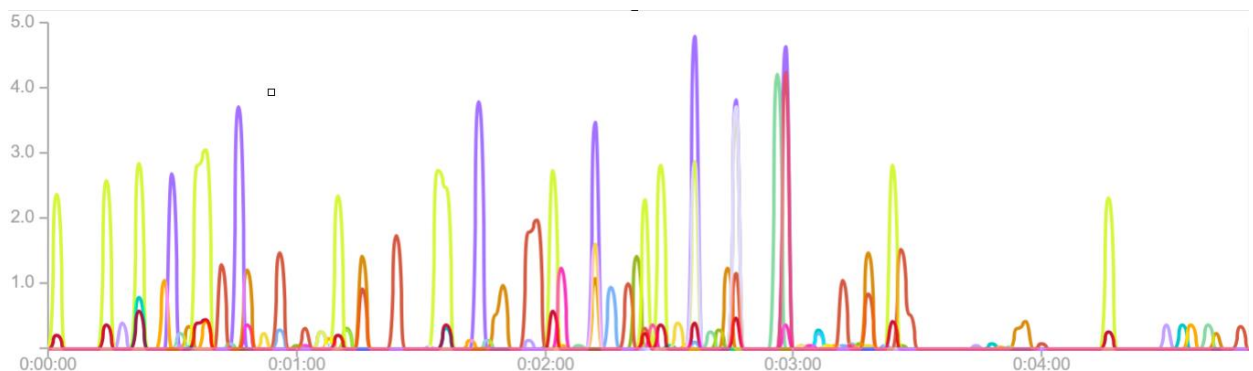


Рисунок 4.25 Середній час завантаження даних (для 10 ботів)

В якості результату була отримана наступне:

- середній час відповіді складав 0.681 секунд;
- квантиль 95 для середнього часу відповіді станове 2.105 секунд;
- квантиль 50 для середнього часу відповіді станове 0.315 секунд

(Рисунок 4.26).

Percentile	Max (s)	Avg (s)	Overall (s)
p50	4,218	0,315	0,033
p80	8,541	1,528	1,002
p90	9,083	2,008	2,497
p95	9,595	2,105	3,625
p99	9,595	2,105	8,389

Рисунок 4.26 Таблиця квантилів (кількість ботів – 10)

Для початку другого експерименту було вирішено застосувати стратегію поступового навантаження та 20 ботів. Графік з запланованою кількістю ботів зображений на рисунку 4.27.

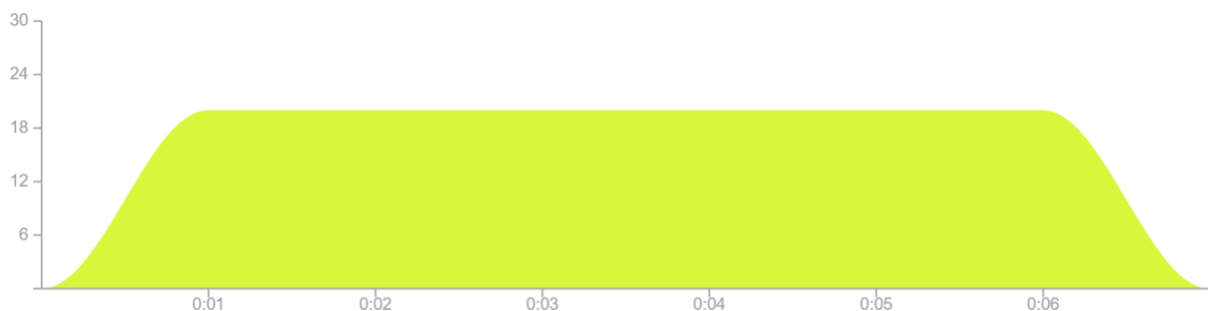


Рисунок 4.27 Графік навантаження системи під час тестування (для 20 ботів)

На рисунку 4.28 зображено фактичну кількість ботів у кожний проміжок часу тестування.



Рисунок 4.28 Графік фактичної кількості ботів у кожен проміжок часу (кількість ботів – 20)

В якості результату була отримана наступне:

- середній час відповіді склав 1.49 секунд;
- квантиль 95 для середнього часу відповіді становить 5.453 секунд;
- квантиль 50 для середнього часу відповіді становить 0.446 секунд

(Рисунок 4.29).

Percentile	Max (s)	Avg (s)	Overall (s)
p50	3,491	0,446	0,045
p80	17,405	3,381	1,983
p90	19,339	5,077	5,429
p95	19,634	5,453	8,955
p99	19,634	5,453	16,13

Рисунок 4.29 Таблиця квантилів (кількість ботів – 20)

Під час тестування 7 запити не пройшли 30 секундний таймауту, що зображено на рисунку 4.30

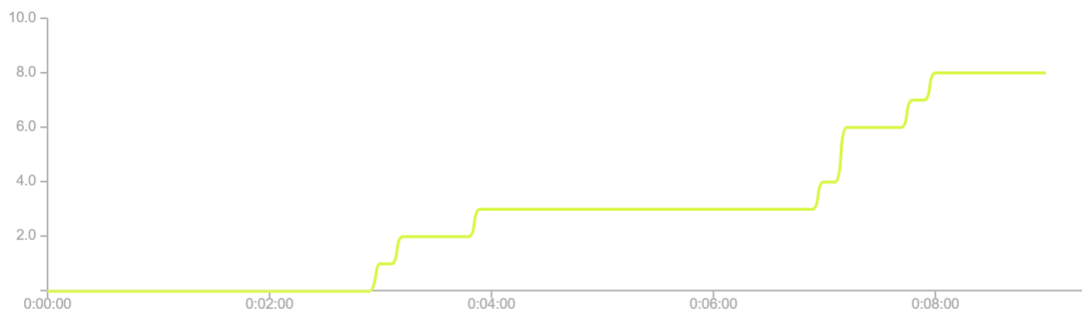


Рисунок 4.30 Графік відмов (кількість ботів – 20)

Для тестування з 30 ботами застосували агресивну стратегію (всі боти починають взаємодію з інформаційною системою майже одночасно). Графік з запланованою кількістю ботів зображений на рисунку 4.31.



Рисунок 4.31 Графік навантаження системи під час тестування (кількість ботів – 30)

На рисунку 4.32 зображено фактичну кількість ботів у кожний проміжок часу тестування.



Рисунок 4.32 Графік фактичної кількості ботів у проміжок часу (для 30 ботів)

На графіку 4.33 зображено середній час завантаження даних за певний проміжок часу.

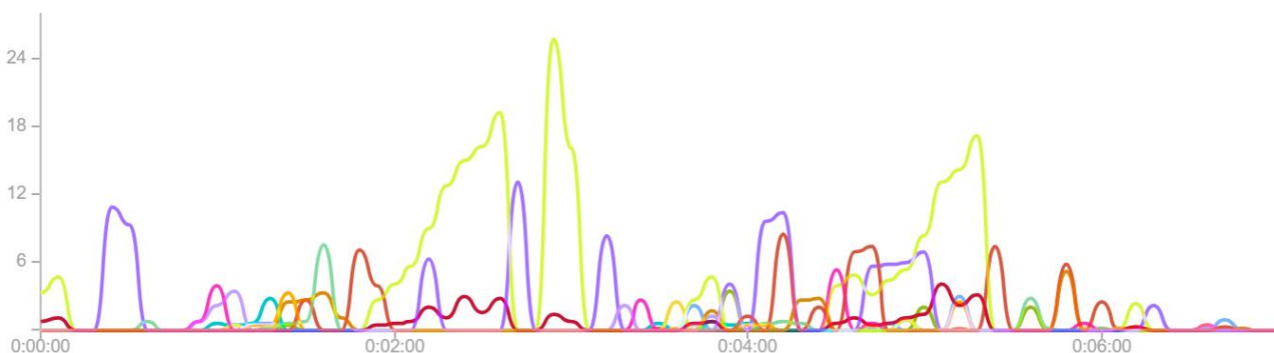


Рисунок 4.33 Середній час завантаження даних за проміжок часу (для 30 ботів)

В якості результату була отримана наступне:

- середній час відповіді складав 2.59 секунд;
- квантиль 95 для середнього часу відповіді станове 6.92 секунд;
- квантиль 50 для середнього часу відповіді станове 1.661 секунд

(Рисунок 4.34).

5 of 5 series visible

Percentile	Max (s)	Avg (s)	Overall (s)
p50	13,295	1,661	0,269
p80	26,108	5,555	4,024
p90	26,732	6,691	9,699
p95	26,779	6,92	14,968
p99	26,779	6,92	22,735

Рисунок 4.34 Таблиця квантилів (кількість ботів – 30)

Під час тестування 33 запити не пройшли 30 секундний таймауту, та 2 запити впали з іншої причини, як зображено на рисунку 4.35.

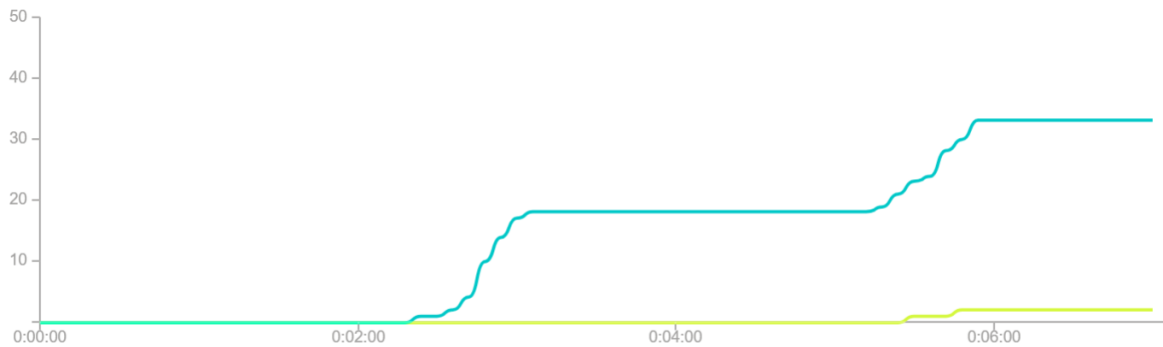


Рисунок 4.35 Графік відмов (кількість ботів – 30)

Для фінального тесту було вирішено запустити 50 ботів. Графік з запланованою кількістю ботів зображений на рисунку 4.36.

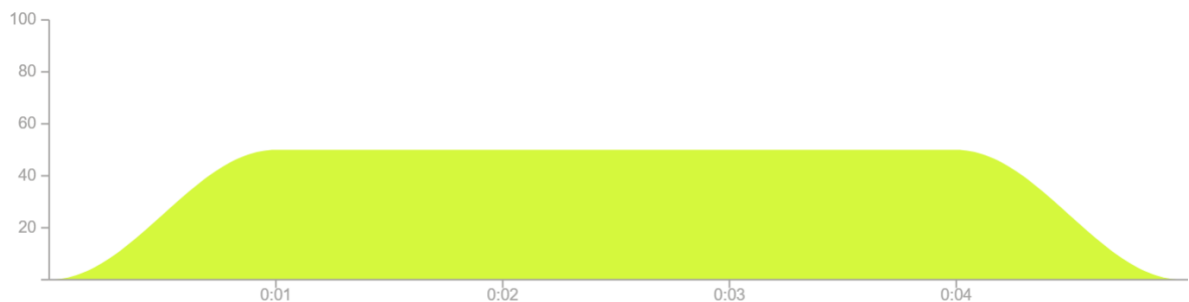


Рисунок 4.36 Графік навантаження системи під час тестування
(кількість ботів – 50)

На рисунку 4.37 зображено фактичну кількість ботів у кожний проміжок часу тестування.



Рисунок 4.37 Графік фактичної кількості ботів у кожний проміжок часу (кількість ботів – 50)

В якості результату була отримана наступне:

- середній час відповіді складав 1.77 секунд;
- квантиль 95 для середнього часу відповіді становив 5.073 секунд;
- квантиль 50 для середнього часу відповіді становив 0.976 секунд

(Рисунок 4.38).

5 of 5 series visible

Percentile

p50

p80

p90

p95

p99

Max (s)

Avg (s)

Overall (s)

9,453

0,976

0,089

17,006

3,84

2,334

17,558

4,9

6,169

18,661

5,073

10,843

18,762

5,075

19,889

Рисунок 4.38 Таблиця квантилів (кількість ботів – 50)

Під час тестування 46 запити не пройшли 30 секундний таймауту, та 14 запити впали з іншої причини, як зображено на рисунку 4.39.

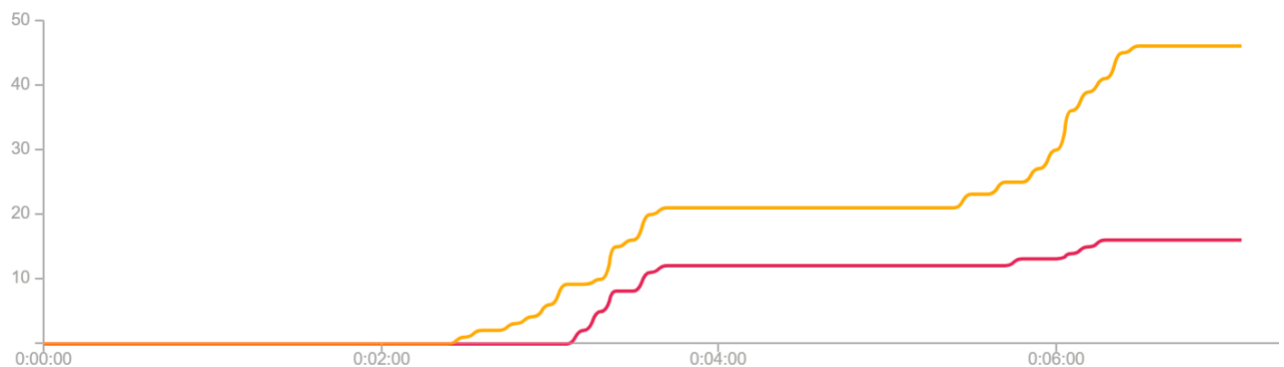


Рисунок 4.39 Графік відмов (кількість ботів – 50)

Для вирішення цієї проблеми (проблема швидкодії при збільшенні кількості підключених користувачів) можна підвищити інстанс AWS EC2, що пришвидшить роботу системи (але збільшить витрати) або ж збільшити кількість кешованих запитів.

4.6 Висновок

Описавши інформаційну систему, було обґрунтовано вибір стеку технологій для реалізації клієнтської та серверної частини.

Серверна частина була реалізована з використання мови програмування Python та Django Rest фреймворком. Цей вибір обумовлений наявністю

стандартних модулів, які включають багато зручних інструментів для роботи з файлами, системні виклики, інтерфейси до різних графічних бібліотек і мережевих з'єднань. В якості бази даних обрано реляційну базу даних PostgreSQL з розширенням PostGIS. Це поєднання дало змогу виконувати просторові запити, маніпулювати просторовими даними та виконувати складний просторовий аналіз за допомогою SQL.

Клієнтська частина була реалізована за допомогою Angular фреймворку через його структуру та наявність підтримки типізації вищеописаних компонентів з-під коробки. Сама клієнтська частина розроблена як односторінковий застосунок. Швидкість оновлення та розширення системи з точки зору клієнтського застосунку забезпечується модульністю та компонентно-орієнтовним підходом.

Запропонована комбінація моделі CALPUFF та IDW алгоритму. може допомогти уточнити модельний прогноз та надати більш повну картину розподілу забруднень у повітрі, що є важливим для оцінки ефектів забруднення на здоров'я та навколишнє середовище. Отриманий результат було растеризовано для швидкості відображення та була зроблена маска для отримання шлейфу, шляхом фільтрація нульових значень.

Порівнявши результати прогнозу погоди, що є важливою складовою для оцінки якості повітря, виявлено перевагу Weather Research and Forecasting (WRF) моделі у порівнянні зі стандартними авторегресійними моделями. Цьому сприяло врахування більшої кількості факторів. Незважаючи на більш точний результат, цю модель можна вдосконалити шляхом постобробки, наприклад, Multi-sensor Advection Diffusion.

ВИСНОВКИ

Сучасне суспільство стикається з низкою серйозних екологічних викликів, з якими пов'язані термінові завдання збереження екосистем та охорони здоров'я людей. Один із найактуальніших аспектів цих завдань — це проблема забруднення повітря, яке є серйозною загрозою громадському здоров'ю та стану навколишнього середовища. У зв'язку з цим, розробка та вдосконалення інформаційних систем моніторингу якості повітря набуває особливого значення.

Провівши аналіз класичних моделей таких як Ейлерова модель, Гаусова модель та Лагранжева модель, були виявлені недоліки в їх роботі такі як не деталізація локальних процесів, застаріла параметризація моделей, на великій відстані потрібно обчислити велику кількість одиничних траєкторій, що призводить до збільшення вартості обчислення. На противагу розглянута модель CALLPUFF, яка базується на Лагранжевій моделі, але враховує більшу кількість параметрів включно з кліматичною зоною, рівнем висот та іншим. Для уточнення результатів моделювання обрано детерміновані методи інтерполяції, що розраховують результат, ґрунтуючись на виміряних значеннях, які потрапляють в околицю інтерпольованої точки, і на заданих математичних формулах, а саме метод обернено зважених відстаней.

Проаналізовано існуючі технології та інструменти для розробки інформаційної системи серед яких: провайдери хмарних послуг, протоколи обміну даних, система управління віртуальними контейнерами, протоколи доступу, архітектуру мережевих протоколів та архітектуру програмного забезпечення. В результаті була формована сервісно-орієнтовна архітектура, що сприяє розробці спільних стандартів для обміну даними між сервісами, що дозволяє забезпечити більшу єдність даних та керування ними в різних частинах системи та в той же час є відносно простою в управлінні власних складових.

Описавши інформаційну систему, було обґрунтовано вибір стеку технологій для реалізації клієнтської та серверної частини. Серверна частина була реалізована з використання мови програмування Python та Django Rest фреймворком. Цей вибір обумовлений наявністю стандартних модулів, які

включають багато зручних інструментів для роботи з файлами, системні виклики, інтерфейси до різних графічних бібліотек і мережевих з'єднань. В якості бази даних обрано реляційну базу даних PostgreSQL з розширенням PostGIS. Це поєднання дало змогу виконувати просторові запити, маніпулювати просторовими даними та виконувати складний просторовий аналіз за допомогою SQL. Клієнтська частина була реалізована за допомогою Angular фреймворку через його структуру та наявність підтримки типізації вищеписаних компонентів з-під коробки. Сама клієнтська частина розроблена як односторінковий застосунок. Швидкість оновлення та розширення системи з точки зору клієнтського застосунку забезпечується модульністю та компонентно-орієнтовним підходом.

Запропонована комбінація моделі CALPUFF та IDW алгоритму. може допомогти уточнити модельний прогноз та надати більш повну картину розподілу забруднень у повітрі, що є важливим для оцінки ефектів забруднення на здоров'я та навколишнє середовище. Отриманий результат було растровано для швидкості відображення та була зроблена маска для отримання шлейфу, шляхом фільтрації нульових значень. Порівнявши результати прогнозу погоди, що є важливою складовою для оцінки якості повітря, виявлено перевагу Weather Research and Forecasting (WRF) моделі у порівнянні зі стандартними авторегресійними моделями. Цьому сприяло врахування більшої кількості факторів.

В умовах зростаючого рівня промислового виробництва та автотранспорту, питання якості повітря стає насущним для розвинених та розвиваючихся країн. Впровадження ефективних інформаційних систем моніторингу дозволить оперативно реагувати на негативні зміни та забезпечить створення умов для здорового та безпечного середовища для мешканців.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Aeroqual [Електронний ресурс] – Режим доступу до ресурсу: <https://www.aeroqual.com/product/air-monitoring-software>.
2. Viasala [Електронний ресурс] – Режим доступу до ресурсу: <https://www.vaisala.com/en/products/software/bl-view>.
3. Breeze-software [Електронний ресурс] – Режим доступу до ресурсу: <https://www.breeze-software.com/software/AERMOD>.
4. WHO Air Quality Guidelines [Електронний ресурс] – Режим доступу до ресурсу: https://www.c40knowledgehub.org/s/article/WHO-Air-Quality-Guidelines?language=en_US.
5. Молодець Б.В, Тарасов В.П., Булана Т.М., Байбуз О.Г. Аналіз існуючих технологій моніторингу повітря. Регіональний міжвузівський збірник «Системні Технології». 2021. Том 3. С. 67-78 DOI: <https://www.doi.org/10.34185/1562-9945-3-134-2021-08>
6. Guidance document on achievement determination Canadian ambient air quality standards for fine particulate matter and ozone [Електронний ресурс] – Режим доступу до ресурсу: https://www.ccme.ca/files/Resources/air/aqms/pn_1_483_gdad_eng.pdf
7. Державні санітарні правила охорони атмосферного повітря населених місць (від забруднення хімічними та біологічними речовинами) [Електронний ресурс]. – 2000. – Режим доступу до ресурсу: <https://zakon.rada.gov.ua/rada/show/v0201282-97/ed20000223>.
8. Air quality standards [Електронний ресурс]. – 2019. – Режим доступу до ресурсу: <https://www.eea.europa.eu/themes/air/air-quality-concentrations/air-quality-standards>.
9. European Air Quality Index [Електронний ресурс] – 2021. – Режим доступу до ресурсу: <https://airindex.eea.europa.eu/>.
10. 40 CFR Part 50 – National primary and secondary ambient air quality standards [Електронний ресурс] – 2023. – Режим доступу до ресурсу: <https://www.law.cornell.edu/cfr/text/40/part-50>.

11. 40 CFR Appendix E to Part 58 - Probe and Monitoring Path Siting Criteria for Ambient Air Quality [Электронный ресурс] – 2023. – Режим доступа до ресурсу: [https://www.law.cornell.edu/cfr/text/40/appendix-D to part 58](https://www.law.cornell.edu/cfr/text/40/appendix-D%20to%20part%2058).
12. Ambient Air Monitoring Protocol For PM_{2.5} and Ozone Canada-wide Standards for Particulate Matter and Ozone [Электронный ресурс] – Режим доступа до ресурсу: [https://www.ccme.ca/files/Resources/air/pm_ozone/pm_oz_cws_monitoring_protocol pn1456 e.pdf](https://www.ccme.ca/files/Resources/air/pm_ozone/pm_oz_cws_monitoring_protocol_pn1456_e.pdf).
13. Han S., Kundhikanjana W., Towashiraporn P., Stratoulas D. Interpolation-Based Fusion of Sentinel-5P, SRTM, and Regulatory-Grade Ground Stations Data for Producing Spatially Continuous Maps of PM_{2.5} Concentrations Nationwide over Thailand. Atmosphere. 2022. 13. 161. DOI: <https://doi.org/10.3390/atmos13020161> .
14. Gitahi J., Hahn M. High-resolution urban air quality monitoring using Sentinel satellite images and low-cost ground-based sensor networks. E3S Web of Conferences. 2022. Vol. 171, 02002. DOI: <https://doi.org/10.1051/e3sconf/202017102002> .
15. Chen L., Wang J., Wang H., Jin T. Urban Air Quality Assessment by Fusing Spatial and Temporal Data from Multiple Study Sources Using Refined Estimation Methods. ISPRS Int. J. Geo-Inf. 2022, 11, 330. DOI: <https://doi.org/10.3390/ijgi11060330> .
16. Olevskyi V, Hnatushenko Volodymyr, Korotenko G.M., Olevska Yu., Obydenyi Y. Application of two-dimensional Padé-type approximations for image processing. Radio Electronics, Computer Science, Control, 2023, № 1, P. 99-106. DOI: <https://doi.org/10.15588/1607-3274-2023-1-10> .
17. Usama M.. Urban Air Quality Measurements: A Survey. Preprints.org; 2022. DOI: <https://doi.org/10.20944/preprints202204.0232.v1> .
18. Singh T., Sharma N., Satakshi, Kumar M. Analysis and forecasting of air quality index based on satellite data. Inhal Toxicol. Vol. 35(1-2). 2023. P. 24-39. DOI: <https://doi.org/10.1080/08958378.2022.2164388>

19. Ahmed A., Ali A., Mahboob M., Humaira F. Comparison between Local and Global Methods to Develop AQI in Representing the Spatial Pattern of Air Quality of Dhaka City. The Dhaka University Journal of Earth and Environmental Sciences. 2023. P. 131-149. DOI: <https://doi.org/10.3329/dujees.v11i1.63716>.
20. Serant O., Yarema N., Согор А.Р., Геба М.С. Створення екологічних карт Львівщини за даними моніторингу атмосферного повітря. Young Scientist. 2018. №5. С. 23-27.
21. Afghan F., Habib H., Akhunzada N. et al. Customization of GIS for spatial and temporal analyses of Air Quality Index trends in Kabul city. Model. Earth Syst. Environ. 2022. Vol. 8. P. 5097–5106. DOI: <https://doi.org/10.1007/s40808-022-01396-5>.
22. Zhang H., Nie Y., Deng Q., Liu Y., Lyu Q., Qiyuan, Zhang B. Spatio-Temporal Changes in Air Quality of the Urban Area of Chongqing from 2015 to 2021 Based on a Missing-Data-Filled Dataset. Atmosphere. 2022. Vol. 13. Article 1473. DOI: <https://doi.org/10.3390/atmos13091473>.
23. Shihab A. Assessment of Air Quality through Multiple Air Quality Index Models: A Comparative Study. Journal of Ecological Engineering. 2023. Vol. 24. P. 110-116. DOI: <https://doi.org/10.12911/22998993/159398>.
24. Liu R., Pang L., Yang Y., Yidian G., Yuxing G., Bei G., Feng L., Li W. Air Quality—Meteorology Correlation Modeling Using Random Forest and Neural Network. Sustainability. 2023. Vol. 15. Article 4531. DOI: <https://doi.org/10.3390/su15054531>.
25. Fassò A., Rodeschini J., Fusta Moro A. Agrimonia: a dataset on livestock, meteorology and air quality in the Lombardy region, Italy. Sci Data. 2023. Vol. 10. Article 143. <https://doi.org/10.1038/s41597-023-02034-0>.
26. Abu El-Magd S., Soliman G., Morsy M. Environmental hazard assessment and monitoring for air pollution using machine learning and remote sensing. Int. J. Environ. Sci. Technol. 2023. Vol. 20. P. 6103–6116. DOI: <https://doi.org/10.1007/s13762-022-04367-6>.

27. Rowley A., Karakuş O. Predicting air quality via multimodal AI and satellite imagery. *Remote Sensing of Environment*. 2023. Vol. 293. Article 113609. DOI: <https://doi.org/10.1016/j.rse.2023.113609>.
28. Ridzuan N., Ujang U., Azri S., Choon T. L. Visualising urban air quality using AERMOD, CALPUFF and CFD models: a critical review. *Int. Arch. Photogramm. Remote Sens. Spatial Inf. Sci.* 2020. Vol. XLIV-4/W3-2020. P. 355–363. DOI: <https://doi.org/10.5194/isprs-archives-XLIV-4-W3-2020-355-2020>.
29. Boadh R., Kumar Rajoria Y., Ramachandra Prasad V. Air Pollution Dispersion Using Coupled AERMOD-WRF Modeling System and Generation of Gridded Emission Inventory of NOX over Nagpur 2023. DOI: <http://dx.doi.org/10.5772/intechopen.108230>..
30. Fadhil N., Jaf Saba. Air Desperation Quality by Using AERMOD Software Program. *Malaysian Journal of Computer Science*. 2022. Vol. 7. P. 2022-2023. DOI: <https://doi.org/10.5281/gsjcs.2022.7098203> .
31. Shubbar R.M., et al. Modeling Air Dispersion of Pollutants Emitted from the Daura Oil Refinery, Baghdad- Iraq using the CALPUFF Modeling System. *Journal of environmental informatics letters*, 2019, P. 28-39. doi: <https://doi.org/10.3808/jeil.201900014> .
32. Satellite Observations Part II. Track Pollution from Space [Электронный ресурс] – Режим доступа до ресурсу: <https://terra.nasa.gov/citizen-science/air-quality/part-ii-track-pollution-from-space>.
33. Advanced Satellite Tracks Air Pollution in Extraordinary Detail [Электронный ресурс] – Режим доступа до ресурсу: <https://eos.org/articles/advanced-satellite-tracks-air-pollution-in-extraordinary-detail>.
34. World`s Air Pollution: Real-time Air Quality Index [Электронный ресурс] – Режим доступа до ресурсу: <https://waqi.info/>
35. National oceanic and atmospheric administration [Электронный ресурс] – Режим доступа до ресурсу: <https://www.ncei.noaa.gov/products/weather-climate-models/global-forecast>;

36. Моніторинг радіаційної обстановки / Моніторинг [Електронний ресурс] – Режим доступу до ресурсу: <https://gcsk.gov.ua/radionuklidnij-monitoring/>.
37. Barry R.G., Chorley R.J. Atmosphere, Weather and Climate. 1st ed. Routledge, 2009. DOI: <https://doi.org/10.4324/9780203871027>.
38. Meriwether J.W., Gardner C.S. A review of the mesosphere inversion layer phenomenon. J. Geophys. Res. 2000. Vol. 105, Issue D10. P. 12405–12416. DOI: <https://doi.org/10.1029/2000JD900163>.
39. Lente G., Ösz K. Barometric formulas: various derivations and comparisons to environmentally relevant observations. ChemTexts. 2020. Vol. 6. P. 13. DOI: <https://doi.org/10.1007/s40828-020-0111-6>.
40. Zanneti P. Air Pollution Modeling. Theories, Computational Methods and Available Software. New York: Springer New York, 2013,. 444, ISBN 978-1-4757-4467-5. DOI: <https://doi.org/10.1007/978-1-4757-4465-1>.
41. Ars S., Broquet G., Kwok C.Y. et al. Statistical atmospheric inversion of local gas emissions by coupling the tracer release technique and local-scale transport modelling: a test case with controlled methane emissions. Atmos. Meas. Tech. 2017. Vol. 10. P. 5017–5037. DOI: <https://doi.org/10.5194/AMT-10-5017-2017>.
42. Korsakissok I., Mallet V. Comparative Study of Gaussian Dispersion Formulas within the Polyphemus Platform: Evaluation with Prairie Grass and Kincaid Experiments. J. Appl. Meteor. Climatol. 2009. Vol. 48. P. 2459–2473. DOI: <https://doi.org/10.1175/2009JAMC2160.1>.
43. Capelli L., Sironi S., Del Rosso R., Guillot J-M. Measuring odours in the environment vs. dispersion modelling: A review. Atmospheric Environment. 2013. Vol. 79. P. 731-743. DOI: <https://doi.org/10.1016/j.atmosenv.2013.07.029>.
44. Visscher A.D. Gaussian dispersion modeling. В книзі "Air Dispersion Modeling", ред. A.D. Visscher. 2013. Chapter 6. P. 141-200. <https://doi.org/10.1002/9781118723098.ch6>.

45. Giaioti D., Oshurok D., Skrynyk O. The Chernobyl nuclear accident ^{137}Cs cumulative depositions simulated by means of the CALMET/CALPUFF modelling system. Atmospheric Pollution Research. 2018. Vol. 9, Issue 3. P. 502-512. DOI: <https://doi.org/10.1016/j.apr.2017.11.007>.
46. Zhou H., Gao B., Deng F. Dust diffusion in large-scale urban construction combining WRF and CALPUFF model—take Xiamen as an example. Journal of Environmental Engineering and Landscape Management. 2023. Vol. 31, Issue 4. P. 288–306. DOI: <https://doi.org/10.3846/jeelm.2023.20044>.
47. Молодець Б.В, Байбуз О.Г. Використання методу обернено зважених відстаней з моделлю CALPUFF для інтерполяції концентрацій забруднюючих речовин у повітрі. Збірник наукових праць «Актуальні проблеми автоматизації та інформаційних технологій», м. Дніпро, 2024, С. 156–163. DOI: <https://www.doi.org/10.15421/432315>.
48. Inverse Distance Weighting (IDW) Interpolation [Електронний ресурс] – Режим доступу до ресурсу – <https://gisgeography.com/inverse-distance-weighting-idw-interpolation/>.
49. Kowsalyadevi, Balaji N.V. Inception to sensor and IoT technology. International Journal of Information Technology, Research and Applications. 2023. Vol. 2. P. 17-23. DOI: <https://doi.org/10.59461/ijitra.v2i4.61>.
50. Sajid, Rafat, Islam Sirajul, Rakib Abul, Kaur Amandeep. Interpretation on the Google Cloud Platform and Its Wide Cloud Services. International Journal of Security and Privacy in Pervasive Computing. 2022. Vol. 14. P. 1-7. DOI: <https://doi.org/10.4018/IJSPPC.313586>.
51. Wiechork K., Charão A. Investigating the Performance of Moodle Database Queries in Cloud Environments. 2020. P. 269-275. DOI: <https://doi.org/10.5220/0009792202690275>.
52. Khan S.M. Top 20 Microsoft Azure Cloud Services Introduction of top 20 Microsoft Azure Cloud Services. 2023. DOI: <https://doi.org/10.13140/RG.2.2.29583.66728>.

53. Luma-Osmani S., Idrizi F., Ademi S., Fetai R. Above the Clouds: A Brief Overview of Microsoft Azure Environments and Applications. Natural Science. 2018. Vol. 3. P. 79-88.
54. Choi W., Taeseok C., Heo S. A Comparative Study of Automated Machine Learning Platforms for Exercise Anthropometry-Based Typology Analysis: Performance Evaluation of AWS SageMaker, GCP VertexAI, and MS Azure. Bioengineering. 2023. Vol 10(8). 891. DOI: <https://doi.org/10.3390/bioengineering10080891> .
55. Choudhary A. & Verma P. & Rai P.. (2021). A walkthrough of Amazon Elastic Compute Cloud (Amazon EC2): A Review. International Journal for Research in Applied Science and Engineering Technology. Vol. 9. P. 93-97. DOI: <https://doi.org/10.22214/ijraset.2021.38764> .
56. What is Amazon Elastic Block Store? [Электронный ресурс] – Режим доступа до ресурсу – <https://docs.aws.amazon.com/ebs/latest/userguide/what-is-ebs.html> .
57. Amazon Simple Storage Service Documentation [Электронный ресурс] – Режим доступа до ресурсу – <https://docs.aws.amazon.com/s3/>
58. Cai, S., Gallina, B., Nyström, D. et al. Data aggregation processes: a survey, a taxonomy, and design guidelines. Computing. 2019. Vol.101, P. 1397–1429 DOI: <https://doi.org/10.1007/s00607-018-0679-5> .
59. Khder M. Web Scraping or Web Crawling: State of Art, Techniques, Approaches and Application. International Journal of Advances in Soft Computing and its Applications. 2021. Vol. 13. P. 145-168. DOI: <https://doi.org/10.15849/IJASCA.211128.11> .
60. Lotfi C., Srinivasan S., Ertz M., Latrous I. Web Scraping Techniques and Applications: A Literature Review. SCRS Conference Proceedings on Intelligent Systems. 2022. P. 381-392. DOI: <https://doi.org/10.52458/978-93-91842-08-6-38> .
61. Kurose J., Ross K. Computer Networking: A Top-Down Approach, Global Edition. Pearson Education, 2018. p. 867. ISBN 978-1-292-15360-5.

62. Douglas E. Internetworking with TCP/IP Volume One. 6th ed. O'Reilly Media, Inc., 2013. p. 744 ISBN 978-0-1374-6419-7.
63. RFC 768: User Datagram Protocol [Електронний ресурс] – Режим доступу до ресурсу – <https://www.rfc-editor.org/rfc/rfc768>
64. Hunt C. TCP/IP Network Administration. 3rd Edition. O'Reilly Media, Inc., 2002. p. 725 ISBN 978-0-596-00297-8.
65. Fall K., Stevens R. TCP/IP illustrated. 2nd ed. Addison-Wesley, Німеччина: Addison-Wesley, 2012. p. 1017. ISBN 978-0-321-33631-6
66. REST - Semantic Web Standards (w3.org) [Електронний ресурс] – Режим доступу до ресурсу – <https://www.w3.org/2001/sw/wiki/REST>
67. RFC 2616: Hypertext Transfer Protocol -- HTTP/1.1 [Електронний ресурс] – Режим доступу до ресурсу – <https://www.rfc-editor.org/rfc/rfc2616>
68. Gourley D., Totty B., Sayer M., Aggarwal A., Reddy S. HTTP: The Definitive Guide. Definitive Guides. O'Reilly Media, 2002. p. 658. ISBN 978-1-4493-7958-2.
69. Молодець Б.В., Болдирев Д.О., Децик В.О., Булана Т.М., Грабовець Д.М. Розробка модульного навчального додатку з використання гнучкої архітектури docker контейнерів. Міжнародна науково-технічна конференція Інформаційні технології в металургії та машинобудуванні, м. Дніпро, 2022. С. 298-300 doi: <https://doi.org/10.34185/1991-7848.itmm.2022.01.059>.
70. Moaut A. Using Docker: Developing and Deploying Software with Containers. O'Reilly Media, Inc. 2015. p. 354 ISBN 978-1-4919-1591-2.
71. Richer J., Sanso A. OAuth 2 in Action. Manning. 2017. p. 360. ISBN 978-1-63835-228-0.
72. RFC 6749: The OAuth 2.0 Authorization Framework [Електронний ресурс] – Режим доступу до ресурсу – <https://www.rfc-editor.org/rfc/rfc6749>
73. What is a REST API? | IBM [Електронний ресурс] – Режим доступу до ресурсу – <https://www.ibm.com/topics/rest-apis>

74. Masse, M.: REST API Design Rulebook: Designing Consistent RESTful Web Service Interfaces. O'Reilly Media, Inc. 2011. p. 116. ISBN: 9781449319915
75. Rodríguez, C., Báez, M., Daniel, F., Casati, F., Trabucco, J.C., Canali, L., Percannella, G. REST APIs: A Large-Scale Analysis of Compliance with Principles and Best Practices. International Conference on Web Engineering. 2016 P. 21-39 DOI: https://doi.org/10.1007/978-3-319-38791-8_2 .
76. Maleshkova M., Pedrinaci C., Domingue J. Investigating Web APIs on the World Wide Web. Eighth IEEE European Conference on Web Services. 2010, P. 107-114, DOI: <https://doi.org/10.1109/ECOWS.2010.9> .
77. L. Gao, C. Zhang and L. Sun, RESTful Web of Things API in Sharing Sensor Data. International Conference on Internet Technology and Applications. 2011. P. 1-4, DOI: <https://doi.org/10.1109/ITAP.2011.6006157> .
78. Petrillo, F., Merle, P., Moha, N., & Guéhéneuc, Y. Are REST APIs for Cloud Computing Well-Designed? An Exploratory Study. International Conference on Service Oriented Computing. 2016. P. 157-170 DOI: https://doi.org/10.1007/978-3-319-46295-0_10 .
79. GraphQL[Электронный ресурс] – Режим доступа до ресурсу <https://spec.graphql.org/October2021/>
80. Mikula, M., Dzieńkowski M. Comparison of REST and GraphQL web technology performance. Journal of Computer Sciences Institute. Vol. 16. 2020 P. 309-316. DOI: <https://doi.org/10.35784/jcsi.2077> .
81. Lawi, A., Panggabean, B., Yoshida T. Evaluating GraphQL and REST API Services Performance in a Massive and Intensive Accessible Information System. Computers. Vol.10. 138. 2021 DOI: <https://doi.org/10.3390/computers10110138> .
82. Lakhai V., Bachynskyy R. Investigation of Serverless Architecture. Advances in Cyber-Physical Systems. Vol. 6. 2021 P. 134-139. DOI: <https://doi.org/10.23939/acps2021.02.134> .

83. Kumari A., Sahoo B. Serverless Architecture for Healthcare Management Systems. Handbook of Research on Mathematical Modeling for Smart Healthcare Systems. 2022. P. 203-227 DOI: <https://doi.org/10.4018/978-1-6684-4580-8.ch011> .
84. dApp authentication with Amazon Cognito and Web3 proxy with Amazon API Gateway [Електронний ресурс] – Режим доступу до ресурсу: <https://aws.amazon.com/blogs/architecture/dapp-authentication-with-amazon-cognito-and-web3-proxy-with-amazon-api-gateway/> .
85. Молодець Б.В, Булана Т.М. Аналіз існуючих архітектур для розробки системи оцінки якості повітря. Регіональний міжвузівський збірник «Системні Технології», м. Дніпро, 2024, Випуск 3(152). С. 128–139 doi: <https://www.doi.org/10.34185/1562-9945-3-152-2024-13>.
86. Dragoni, N., Giallorenzo, S., Lluch-Lafuente, A., Mazzara, M., Montesi, F., Mustafin, R., Safina, L. Microservices: Yesterday, Today, and Tomorrow. Present and Ulterior Software Engineering. 2016 P. 195-216 DOI: https://doi.org/10.1007/978-3-319-67425-4_12 .
87. Villamizar M., Garcés O., Ochoa L., Castro H., Salamanca L., Verano M. M., Casallas R., Gil S., Valencia C., Zambrano A., Lang M. Infrastructure Cost Comparison of Running Web Applications in the Cloud Using AWS Lambda and Monolithic and Microservice Architectures. IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGrid). 2016. P.179-182. DOI: <https://doi.org/10.1109/CCGrid.2016.37>.
88. Simple microservices architecture on AWS - Implementing Microservices on AWS (amazon.com) [Електронний ресурс] – Режим доступу до ресурсу: <https://docs.aws.amazon.com/whitepapers/latest/microservices-on-aws/simple-microservices-architecture-on-aws.html>.
89. Abgaz Y., Mccarren A., Elger P., Solan D., Lapuz N., Bivol M., Jackson G., Yilmaz M., Buckley J., Clarke, P. Decomposition of Monolith Applications Into Microservices Architectures: A Systematic Review. IEEE

Transactions on Software Engineering. 2023 P. 1-32. DOI: <https://doi.org/10.1109/TSE.2023.3287297> .

90. What is SOA (Service-Oriented Architecture)? [Электронный ресурс] – Режим доступа до ресурсу: <https://aws.amazon.com/what-is/service-oriented-architecture/>

91. Lewis G., Smith D., Chapin N., Kontogiannis K. (2010). Proceedings of the Fourth International Workshop on a Research Agenda for Maintenance and Evolution of Service- Oriented Systems. Carnegie Mellon University. 2010. ISBN: 0-9786956-8-2

92. GDAL Grid Tutorial, 2023 [Электронный ресурс] – Режим доступа до ресурсу: https://gdal.org/tutorials/gdal_grid_tut.html#interpolation-of-the-scattered-data.

93. Hinestroza-Ramirez J., Rengifo-Castro J., Quintero O., Yarce Botero A., Rendon-Perez A. Non-Parametric and Robust Sensitivity Analysis of the Weather Research and Forecast (WRF) Model in the Tropical Andes Region. Atmosphere. 2023 14. 686. DOI: <https://doi.org/10.3390/atmos14040686> .

94. Molodets, B., Hnatushenko Volodymyr, Boldyriev D., Bulana, T. Information System of Air Quality Assessment Using Data Interpolation from Ground Stations. CEUR Workshop Modern Machine Learning Technologies and Data Science Workshop (MoMLeT&DS 2023), Lviv 2023. Vol. 3426. P. 233–245. URL: <https://ceur-ws.org/Vol-3426/>.

95. Antonenko S.V., Molodets B.V., Bulana T.M. Modeling of forecasting the danger of emergencies, System technologies. 2019. Vol. 120. (1) P.44-49.

96. Jiménez P., Dudhia J., Thompson G., Lee J., Brummet T. Improving the cloud initialization in WRF-Solar with enhanced short-range forecasting functionality: The MAD-WRF model. Solar Energy. 2022. Vol. 239. P. 221-233. DOI: <https://doi.org/10.1016/j.solener.2022.04.055> .

97. Bullington-McGuire R. Docker for Developers: Develop and run your application with Docker containers using DevOps tools for continuous delivery. 2020

98. Hnatushenko Volodymyr, Hnatushenko Vik., Dorosh N., Solodka N., Liashenko O. Non-relational approach to developing knowledge bases of expert system prototype. Naukovyi Visnyk Natsionalnoho Hirnychoho Universytetu. 2022. Vol. 2. P.112-117. DOI: <https://10.33271/nvngu/2022-2/112>.
99. Docker & Kubernetes 3: minikube Django with Redis and Celery - 2020 [Електронний ресурс] – Режим доступу до ресурсу: https://bogotobogo.com/DevOps/Docker/Docker_Kubernetes_Minikube_3_Django_with_Redis_Celery.php
100. Molodets, B., Hnatushenko Volodymyr, Boldyriev D. , Bulana T. Information System of Air Quality Assessment Based of Ground Stations and Meteorological Data Monitoring. CEUR Workshop Intelligent Information Technologies & Systems of Information Security (IntelITSIS 2023). 2023.Vol. 3373, P. 206–216. URL: <https://ceur-ws.org/Vol-3373/>.
101. Browser Market Share Worldwide | Statcounter Global Stats [Електронний ресурс] – Режим доступу до ресурсу: <https://gs.statcounter.com/browser-market-share/all/worldwide/2023>
102. Browser Version Market Share Worldwide | Statcounter Global Stats [Електронний ресурс] – Режим доступу до ресурсу: <https://gs.statcounter.com/browser-version-market-share/all/worldwide/2023>
103. Angular - Browser support [Електронний ресурс] – Режим доступу до ресурсу: <https://angular.io/guide/browser-support>
104. Kaluža M., Vukelic B., Comparison of front-end frameworks for web applications development. Zbornik Veleučilišta u Rijeci. 2018 Vol. 6. P.261-282. DOI: <https://doi.org/10.31784/zvr.6.1.19> .

Список праць здобувача за темою дисертації

Статті що індексуються в наукометричній базі Scopus:

1. Molodets, B., Hnatushenko Volodymyr, Boldyriev D., Bulana, T. Information System of Air Quality Assessment Using Data Interpolation from Ground Stations. CEUR Workshop Modern Machine Learning Technologies and Data Science Workshop (MoML&T&DS 2023), Lviv 2023. Vol. 3426. P. 233–245. Режим доступу до ресурсу: <https://ceur-ws.org/Vol-3426/>.

2. Molodets, B., Hnatushenko Volodymyr, Boldyriev D. , Bulana T. Information System of Air Quality Assessment Based of Ground Stations and Meteorological Data Monitoring. CEUR Workshop Intelligent Information Technologies & Systems of Information Security (IntelITSIS 2023). 2023.Vol. 3373, P. 206–216. Режим доступу до ресурсу: <https://ceur-ws.org/Vol-3373/>.

Статті у наукових фахових виданнях України:

3. Молодець Б.В, Тарасов В.П., Булана Т.М., Байбуз О.Г. Аналіз існуючих технологій моніторингу повітря. Регіональний міжвузівський збірник «Системні Технології», м. Дніпро, 2021, Випуск 3(134). С. 67–78. doi: <https://www.doi.org/10.34185/1562-9945-3-134-2021-08>. Режим доступу до ресурсу: <https://journals.nmetau.edu.ua/index.php/st/article/view/748>.

4. Молодець Б.В, Байбуз О.Г. Використання методу обернено зважених відстаней з моделлю CALPUFF для інтерполяції концентрацій забруднюючих речовин у повітрі. Збірник наукових праць «Актуальні проблеми автоматизації та інформаційних технологій», м. Дніпро, 2024, С. 156–163. doi: <https://www.doi.org/10.15421/432315>. Режим доступу до ресурсу: <https://actualproblems.dp.ua/index.php/APAIT/article/view/240>.

5. Молодець Б.В, Булана Т.М. Аналіз існуючих архітектур для розробки системи оцінки якості повітря. Регіональний міжвузівський збірник «Системні Технології», м. Дніпро, 2024, Випуск 3(152). С.128–139 doi:

<https://www.doi.org/10.34185/1562-9945-3-152-2024-13>. Режим доступу до ресурсу: <https://journals.nmetau.edu.ua/index.php/st/article/view/1702>

Наукові праці, які засвідчують апробацію матеріалів дисертації:

6. Молодець Б.В. Створення системи агрегування даних за допомогою Celery. VII Всеукраїнську науково-практичну конференцію «Перспективні напрямки сучасної електроніки, інформаційних і комп'ютерних систем». м. Дніпро, 2022. С.48-49. Режим доступу до ресурсу: <http://meics.dnure.dp.ua/files/MEICS-2022.pdf>

7. Молодець Б.В., Болдирев Д.О., Децик В.О., Булана Т.М., Грабовець Д.М. Розробка модульного навчального додатку з використання гнучкої архітектури docker контейнерів. Міжнародна науково-технічна конференція Інформаційні технології в металургії та машинобудуванні, м. Дніпро, 2022. С. 298-300 doi: <https://doi.org/10.34185/1991-7848.itmm.2022.01.059> Режим доступу до ресурсу: <https://journals.nmetau.edu.ua/index.php/itmm/article/view/1175>

8. Молодець Б.В, Болдирев Д.О, Булана Т.М. Аналітичне рішення задач перенесення забруднень у двовимірному просторі. Перспективні напрямки сучасної електроніки, інформаційних і комп'ютерних систем (MEICS-2021), м. Дніпро, 2021. С. 21-22. Режим доступу до ресурсу: <http://meics.dnure.dp.ua/files/MEICS-2021.pdf>

9. Болдирев Д.О., Молодець Б.В., Булана Т.М. Розробка програмного додатку для вирішення задачі перенесення забруднень у двовимірному просторі. XIX міжнародна науково-практична конференція Математичне та програмне забезпечення інтелектуальних систем (МПЗІС-2021), м. Дніпро, 2021. С. 21-22. Режим доступу до ресурсу: <http://mpzis.dnu.dp.ua/wp-content/uploads/2021/12/mpzis-2021.pdf>