

Міністерство освіти і науки України
Тернопільський національний технічний університет імені Івана Пулюя

Факультет комп'ютерно-інформаційних систем і програмної інженерії

(повна назва факультету)

Кафедра програмної інженерії

(повна назва кафедри)

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття освітнього ступеня

Бакалавр

(назва освітнього ступеня)

на тему: Розробка програмної системи моніторингу та прогнозування рівня забруднення повітря з використанням методів машинного навчання

Виконав(ла): студент(ка) 4 курсу, групи СПс-41
спеціальності 121 Інженерія програмного забезпечення

(шифр і назва спеціальності)

Матичак А. П.
(підпис) (прізвище та ініціали)

Керівник Цебрій О.Р.
(підпис) (прізвище та ініціали)

Нормоконтроль Стоянов Ю.М.
(підпис) (прізвище та ініціали)

Завідувач кафедри Петрик М.Р.
(підпис) (прізвище та ініціали)

Рецензент
(підпис) (прізвище та ініціали)

Тернопіль
2026

Міністерство освіти і науки України
Тернопільський національний технічний університет імені Івана Пулюя

Факультет комп'ютерно-інформаційних систем і програмної інженерії
(повна назва факультету)

Кафедра програмної інженерії
(повна назва кафедри)

ЗАТВЕРДЖУЮ
Завідувач кафедри

_____ (підпис) _____ (прізвище та ініціали)
« » 20__ р.

ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

на здобуття освітнього ступеня Бакалавра
(назва освітнього ступеня)

за спеціальністю 121 Інженерія програмного забезпечення
(шифр і назва спеціальності)

студенту Матичак Андрій Петрович
(прізвище, ім'я, по батькові)

1. Тема роботи Розробка програмної системи моніторингу та прогнозування рівня забруднення повітря з використанням методів машинного навчання

Керівник роботи Цебрій О.Р., канд. фіз.-мат. наук, доц.
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

Затверджені наказом ректора від «__» _____ 20__ року № _____

2. Термін подання студентом завершеної роботи _____

3. Вихідні дані до роботи Предметна область, завдання, вимоги та специфікація, програмне рішення, методичні вказівки

4. Зміст роботи (перелік питань, які потрібно розробити)

Вступна частина

Аналіз предметної області та теоретичних основ

Визначення методики реалізації моделі

Реалізація моделі

Визначення основних аспектів охорони праці та безпеки життєдіяльності

Висновки роботи

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень, слайдів)

Слайди презентації та діаграми процесів

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Безпека життєдіяльності, основи охорони праці			
Нормоконтроль	Стоянов Ю.М. к.т.н., доц. каф. ПІ		

7. Дата видачі завдання _____

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1	<i>Розробка технічного завдання</i>	<i>6.04 – 12.04</i>	
2	<i>Робота над першим розділом «АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ВИЗНАЧЕННЯ ВИМОГ ДО ПРОГРАМНОЇ СИСТЕМИ»</i>	<i>13.04 – 26.04</i>	
3	<i>Робота над другим розділом «ПРОЄКТУВАННЯ ТА РОЗРОБКА ПРОГРАМНОЇ СИСТЕМИ МОНІТОРИНГУ ТА ПРОГНОЗУВАННЯ ЗАБРУДНЕННЯ ПОВІТРЯ»</i>	<i>27.04 – 03.05</i>	
4	<i>Робота над третім розділом «ТЕСТУВАННЯ, ОЦІНЮВАННЯ ТА ВПРОВАДЖЕННЯ ПРОГРАМНОЇ СИСТЕМИ»</i>	<i>04.05 – 17.05</i>	
5	<i>Робота над четвертим розділом «Безпека життєдіяльності, основи охорони праці»</i>	<i>18.05 – 24.05</i>	
6	<i>Оформлення пояснювальної записки і графічного матеріалу</i>	<i>25.05 – 7.06</i>	
7	<i>Перевірка на академічний плагіат, перевірка керівником та консультантами</i>	<i>8.06 – 14.06</i>	
8	<i>Попередній захист кваліфікаційної роботи бакалавра</i>	<i>15.06 – 21.06</i>	
9	<i>Захист кваліфікаційної роботи бакалавра</i>		

Студент

(підпис)

Матичак А. П.

(прізвище та ініціали)

Керівник роботи

(підпис)

Цебрій О.Р.

(прізвище та ініціали)

АНОТАЦІЯ

Розробка програмної системи моніторингу та прогнозування рівня забруднення повітря з використанням методів машинного навчання // Кваліфікаційна робота освітнього рівня «Бакалавр» // Матичак Андрій Петрович // Тернопільський національний технічний університет імені Івана Пулюя, факультет комп'ютерно-інформаційних систем і програмної інженерії, кафедра програмної інженерії, група СПс-41 // Тернопіль, 2026 // С. - 75, рис. – 11, табл. – 20, додат. – 2, бібліогр. – 26.

Ключові слова: моніторинг якості повітря; забруднення атмосферного повітря; машинне навчання; прогнозування; LSTM; вебзастосунок; PostgreSQL.

Кваліфікаційна робота присвячена розробці програмної системи моніторингу та прогнозування рівня забруднення атмосферного повітря із застосуванням методів машинного навчання.

У роботі проаналізовано предметну область, сучасні системи моніторингу якості повітря та методи прогнозування екологічних показників, сформовано вимоги до програмної системи. Спроектовано архітектуру системи, базу даних, модулі збору й обробки даних, прогнозування, REST API та вебінтерфейс.

Підготовлено набір даних, проведено навчання й оцінювання моделей Linear Regression, Decision Tree Regressor, Random Forest Regressor, Gradient Boosting Regressor, XGBoost Regressor та LSTM, виконано тестування системи.

Об'єкт дослідження – процес моніторингу та прогнозування рівня забруднення атмосферного повітря на основі метеорологічних даних.

Предмет дослідження – методи, моделі та програмні засоби прогнозування показників якості повітря з використанням технологій машинного навчання.

Практичним результатом є програмна система для автоматизованого збору, аналізу, зберігання та прогнозування рівня забруднення повітря з використанням моделі LSTM і відображенням результатів через вебінтерфейс.

ABSTRACT

Development of a Software System for Monitoring and Forecasting Air Pollution Using Machine Learning Methods // Bachelor's Qualification Thesis // Andrii Matychak // Ternopil Ivan Puluj National Technical University, Faculty of Computer Information Systems and Software Engineering, Department of Software Engineering, Group SPs-41 // Ternopil, 2026 // P. 75, fig. – 11, tabl. – 20, annexes –2, references –26.

Keywords: air quality monitoring; air pollution; machine learning; forecasting; environmental monitoring; LSTM; web application; PostgreSQL.

The qualification thesis is devoted to the development of a software system for monitoring and forecasting air pollution using machine learning methods.

The thesis analyses the problem domain, modern air quality monitoring systems, and methods for forecasting environmental indicators. System requirements were defined, and the architecture of the software system, database, data collection and processing modules, forecasting module, REST API, and web interface were designed.

A dataset was prepared, and the Linear Regression, Decision Tree Regressor, Random Forest Regressor, Gradient Boosting Regressor, XGBoost Regressor, and LSTM models were trained and evaluated. Functional testing of the developed system was also performed.

Object of research is the process of monitoring and forecasting air pollution levels based on environmental and meteorological data.

Subject of research are methods, models, and software tools for forecasting air quality indicators using machine learning technologies.

The practical result of the thesis is a software system for automated collection, analysis, storage, and forecasting of air pollution levels using an LSTM model, with visualization of results through a web interface.

ЗМІСТ

ВСТУП.....	9
1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ВИЗНАЧЕННЯ ВИМОГ ДО ПРОГРАМНОЇ СИСТЕМИ.....	11
1.1 Аналіз проблеми забруднення повітря та її впливу на навколишнє середовище.....	11
1.2 Огляд сучасних систем моніторингу якості повітря.....	14
1.3 Аналіз методів прогнозування рівня забруднення повітря.....	15
1.4 Використання методів машинного навчання для екологічного моніторингу.....	17
1.5 Аналіз джерел даних та наборів даних для прогнозування якості повітря...19	
1.6 Формування вимог до програмної системи.....	20
1.7 Формування постановки задачі та висновки до розділу.....	24
1.8 Висновки до першого розділу.....	24
2 ПРОЄКТУВАННЯ ТА РОЗРОБКА ПРОГРАМНОЇ СИСТЕМИ МОНІТОРИНГУ ТА ПРОГНОЗУВАННЯ ЗАБРУДНЕННЯ ПОВІТРЯ.....	26
2.1 Обґрунтування вибору технологій та засобів розробки.....	26
2.2 Проєктування архітектури програмної системи.....	28
2.3 Проєктування структури бази даних.....	30
2.4 Розробка модуля збору та обробки даних про якість повітря.....	35
2.5 Розробка модуля прогнозування на основі методів машинного навчання....	36
2.6 Реалізація програмної системи та інтеграція компонентів.....	39
2.7 Реалізація програмної системи та інтеграція компонентів.....	42
2.8 Висновки до другого розлу.....	44
3 ТЕСТУВАННЯ, ОЦІНЮВАННЯ ТА ВПРОВАДЖЕННЯ ПРОГРАМНОЇ СИСТЕМИ.....	46

3.1 Підготовка набору даних для навчання та тестування моделей.....	46
3.2 Навчання моделей машинного навчання та оцінювання точності прогнозування.....	47
3.3 Тестування функціональності програмної системи та аналіз результатів експериментальних досліджень.....	49
3.4 Рекомендації щодо впровадження та перспективи розвитку системи.....	51
3.5 Висновки до третього розлілу.....	53
4 ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ.....	54
4.1 Моделювання та прогнозування небезпечних ситуацій.....	54
4.2 Вимоги ергономіки до організації робочого місця оператора ПК.....	56
ВИСНОВКИ.....	59
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	61
ДОДАТКИ.....	64
ДОДАТОК А.....	65
ДОДАТОК Б.....	74

ПЕРЕЛІК СКОРОЧЕНЬ

- ML – Machine Learning (машинне навчання)
- AI – Artificial Intelligence (штучний інтелект)
- DL – Deep Learning (глибинне навчання)
- EDA – Exploratory Data Analysis (розвідувальний аналіз даних)
- CSV – Comma-Separated Values (текстовий формат даних)
- LR – Logistic Regression (логістична регресія)
- DT – Decision Tree (дерево рішень)
- kNN – k-Nearest Neighbors (метод k найближчих сусідів)
- SVM – Support Vector Machine (метод опорних векторів)
- NB – Naive Bayes (наївний байєсівський класифікатор)
- NN – Neural Network (нейронна мережа)
- RF – Random Forest (випадковий ліс)
- GBM – Gradient Boosting Machine (градієнтний бустинг)
- XGB – XGBoost (Extreme Gradient Boosting)
- LGBM – LightGBM (Light Gradient Boosting Machine)
- CB – CatBoost (Categorical Boosting)
- SMOTE – Synthetic Minority Oversampling Technique (синтетичне збільшення вибірки меншості)
- ROC – Receiver Operating Characteristic (крива робочих характеристик приймача)
- AUC – Area Under Curve (площа під кривою)
- API – Application Programming Interface (інтерфейс прикладного програмування)
- CPU – Central Processing Unit (центральний процесор)
- RAM – Random Access Memory (оперативна пам'ять)
- JSON – JavaScript Object Notation (формат даних JSON)

ВСТУП

Забруднення атмосферного повітря є однією з найважливіших екологічних проблем сучасності. Стрімкий розвиток промисловості, збільшення кількості транспортних засобів, урбанізація та інші антропогенні фактори призводять до зростання концентрації шкідливих речовин у повітрі. Погіршення якості атмосферного повітря негативно впливає на здоров'я населення, спричиняє розвиток респіраторних та серцево-судинних захворювань, а також створює загрози для навколишнього середовища. У зв'язку з цим особливої актуальності набуває розробка інтелектуальних інформаційних систем, здатних не лише здійснювати моніторинг поточного стану повітря, а й прогнозувати майбутні зміни рівня забруднення.

Сучасні технології збору та обробки екологічних даних дозволяють накопичувати значні обсяги інформації про концентрацію забруднювальних речовин, метеорологічні показники та інші фактори, що впливають на якість повітря. Проте традиційні методи аналізу часто не забезпечують необхідної точності прогнозування через складність взаємозв'язків між різними параметрами середовища. Для розв'язання цієї проблеми дедалі ширше застосовуються методи машинного навчання, які дозволяють виявляти приховані закономірності в даних та формувати прогнози на їх основі. Використання таких підходів сприяє підвищенню точності прогнозування та ефективності прийняття рішень у сфері екологічного моніторингу.

Актуальність теми кваліфікаційної роботи зумовлена необхідністю створення програмних засобів для автоматизованого моніторингу та прогнозування рівня забруднення атмосферного повітря з використанням сучасних методів машинного навчання. Розробка таких систем дозволяє підвищити оперативність аналізу екологічної інформації, забезпечити своєчасне інформування користувачів про можливі ризики та підтримати процес прийняття управлінських рішень щодо покращення стану навколишнього середовища.

Метою кваліфікаційної роботи є розробка програмної системи моніторингу та прогнозування рівня забруднення повітря з використанням методів машинного навчання.

Для досягнення поставленої мети в роботі передбачено виконання аналізу предметної області моніторингу якості атмосферного повітря, дослідження існуючих підходів до прогнозування екологічних показників, визначення вимог до програмної системи, проектування її архітектури, реалізацію програмного забезпечення, інтеграцію моделей машинного навчання та проведення тестування отриманого рішення.

Об'єктом дослідження є процес моніторингу та прогнозування рівня забруднення атмосферного повітря.

Предметом дослідження є методи, моделі та програмні засоби прогнозування показників якості повітря на основі технологій машинного навчання.

Під час виконання роботи використано методи системного аналізу, об'єктно-орієнтованого проектування, статистичного аналізу даних, машинного навчання, програмної інженерії та тестування програмного забезпечення.

Практичне значення одержаних результатів полягає у створенні програмної системи, яка забезпечує автоматизований збір, аналіз та прогнозування даних про якість атмосферного повітря. Розроблене програмне забезпечення може бути використане екологічними службами, органами місцевого самоврядування, науковими установами та іншими організаціями для оцінювання екологічної ситуації та підтримки прийняття рішень.

Кваліфікаційна робота складається зі вступу, чотирьох розділів, висновків, списку використаних джерел та додатків. У першому розділі виконано аналіз предметної області та визначено вимоги до програмної системи. Другий розділ присвячений проектуванню та розробці програмної системи. У третьому розділі наведено результати тестування та оцінювання ефективності розробленого рішення. Четвертий розділ містить питання безпеки життєдіяльності та охорони праці.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ВИЗНАЧЕННЯ ВИМОГ ДО ПРОГРАМНОЇ СИСТЕМИ

Якість атмосферного повітря є одним із ключових факторів, що впливають на стан навколишнього середовища та здоров'я населення. Зростання обсягів промислового виробництва, інтенсивне використання транспортних засобів та урбанізація спричиняють збільшення концентрації шкідливих речовин у повітрі. У зв'язку з цим виникає потреба в постійному контролі екологічної ситуації та своєчасному виявленні потенційно небезпечних змін показників якості повітря. Сучасні інформаційні технології дозволяють автоматизувати процеси збору, зберігання та аналізу екологічних даних, що значно підвищує ефективність моніторингу навколишнього середовища [1-2].

У даному розділі проведено аналіз предметної області моніторингу та прогнозування рівня забруднення повітря, досліджено сучасні програмні рішення та підходи до аналізу екологічних даних, розглянуто можливості використання методів машинного навчання для прогнозування показників якості атмосферного повітря. На основі проведеного аналізу сформовано функціональні та нефункціональні вимоги до програмної системи, визначено її основні компоненти та окреслено завдання, які необхідно вирішити під час проектування і реалізації програмного забезпечення.

1.1 Аналіз проблеми забруднення повітря та її впливу на навколишнє середовище

Атмосферне повітря є одним із найважливіших компонентів навколишнього природного середовища, від якого безпосередньо залежить життєдіяльність людини, функціонування екосистем та стабільність кліматичних процесів. Протягом останніх десятиліть проблема забруднення повітря набула глобального характеру через стрімкий розвиток промисловості, енергетики, транспортної інфраструктури та урбанізацію. Зростання обсягів виробництва та споживання

енергетичних ресурсів супроводжується збільшенням викидів забруднювальних речовин в атмосферу, що негативно впливає як на стан довкілля, так і на здоров'я населення [3].

Основними джерелами забруднення атмосферного повітря є промислові підприємства, теплові електростанції, автомобільний транспорт, сільськогосподарська діяльність та побутові джерела спалювання палива. У процесі їх функціонування в атмосферу потрапляють різноманітні шкідливі речовини, серед яких особливу небезпеку становлять тверді частинки PM_{2.5} та PM₁₀, оксиди азоту, діоксид сірки, чадний газ, озон та леткі органічні сполуки. Концентрація цих речовин у повітрі значною мірою залежить від інтенсивності викидів, погодних умов, рельєфу місцевості та інших природних факторів.

Забруднення повітря є однією з основних причин погіршення здоров'я населення. Тривале перебування в умовах підвищеної концентрації забруднювальних речовин може призводити до розвитку захворювань органів дихання, серцево-судинної системи, алергічних реакцій та інших патологій. Особливо вразливими до впливу забрудненого повітря є діти, люди похилого віку та особи з хронічними захворюваннями. Крім негативного впливу на людину, забруднення атмосфери спричиняє деградацію природних екосистем, погіршення стану ґрунтів та водних ресурсів, а також бере участь у формуванні парникового ефекту та кліматичних змін [4].

Важливою особливістю проблеми забруднення повітря є її динамічний характер. Рівень концентрації шкідливих речовин може суттєво змінюватися протягом доби під впливом погодних умов, транспортних потоків, сезонних факторів та виробничої активності. У зв'язку з цим виникає необхідність не лише періодичного контролю стану атмосферного повітря, але й безперервного моніторингу екологічних показників у режимі реального часу.

Сучасні системи екологічного моніторингу базуються на використанні мереж датчиків, автоматизованих станцій спостереження та інформаційних платформ для збору й аналізу даних. Такі системи дозволяють накопичувати значні обсяги інформації про стан атмосферного повітря та оперативно виявляти

перевищення допустимих концентрацій забруднювальних речовин. Проте отримання лише поточних показників часто є недостатнім для ефективного реагування на потенційні екологічні загрози. Значно більшу практичну цінність становить можливість прогнозування майбутнього рівня забруднення, що дозволяє завчасно інформувати населення та відповідні служби про можливе погіршення якості повітря [5].

Традиційні методи прогнозування здебільшого базуються на статистичних підходах та математичних моделях, які не завжди здатні врахувати складні нелінійні залежності між численними факторами, що впливають на стан атмосфери. У зв'язку з цим все більшої популярності набувають методи машинного навчання, які дозволяють автоматично виявляти закономірності у великих наборах даних та формувати прогнози з високою точністю. Використання таких методів відкриває нові можливості для створення інтелектуальних програмних систем моніторингу та прогнозування якості повітря.



Рисунок 1.1 – Основні джерела забруднення атмосферного повітря

Таким чином, проблема забруднення атмосферного повітря залишається однією з найбільш актуальних екологічних проблем сучасного суспільства. Ефективне вирішення цієї проблеми потребує використання сучасних інформаційних технологій, засобів автоматизованого збору даних та методів машинного навчання, здатних забезпечити не лише контроль поточного стану довкілля, але й прогнозування його майбутніх змін. Саме тому розробка програмної системи моніторингу та прогнозування рівня забруднення повітря є актуальним завданням у сфері програмної інженерії та екологічного моніторингу.

1.2 Огляд сучасних систем моніторингу якості повітря

Сучасні системи моніторингу якості повітря забезпечують збір, зберігання, аналіз та візуалізацію екологічних даних у режимі реального часу. Їх основною метою є надання актуальної інформації про концентрацію забруднювальних речовин в атмосферному повітрі та підтримка прийняття рішень щодо зменшення негативного впливу забруднення на населення і навколишнє середовище. Більшість сучасних рішень використовують мережі датчиків, хмарні технології та вебінтерфейси для відображення результатів спостережень [6-8].

Однією з найбільш відомих систем є AirNow, яка використовується у США для збору та публікації інформації про якість повітря. Система надає дані про індекс якості повітря (AQI), концентрацію основних забруднювачів та прогнози стану атмосфери. Іншим популярним рішенням є OpenAQ — міжнародна платформа відкритих екологічних даних, що об'єднує інформацію з тисяч станцій моніторингу різних країн світу та надає доступ до неї через вебінтерфейс і API.

Серед комерційних рішень поширення отримала система IQAir, яка забезпечує моніторинг якості повітря в реальному часі та формує прогнози забруднення на основі історичних і метеорологічних даних. Також значну популярність має мережа PurpleAir, що використовує велику кількість недорогих датчиків для отримання детальної інформації про стан повітря в різних регіонах.

Для порівняння основних характеристик сучасних систем моніторингу якості повітря сформовано таблицю 1.1.

Таблиця 1.1 – Порівняння сучасних систем моніторингу якості повітря

Система	Географічне охоплення	Кількість станцій / сенсорів	Основні показники	Прогнозування
AirNow	США, Канада, частина Мексики	понад 2500 моніторів	AQI, PM2.5, O ₃	для понад 500 міст
OpenAQ	міжнародна платформа	понад 8000 локацій	PM2.5, PM10, NO ₂ , SO ₂ , CO, O ₃	не є основною функцією
IQAir	143 країни, регіони та території	понад 80000 сенсорів	AQI, PM2.5, PM10, NO ₂ , O ₃ , температура, вологість	до 7 днів
PurpleAir	міжнародна мережа сенсорів	понад 30000 сенсорів	PM1.0, PM2.5, PM10, температура, вологість	обмежене
WAQI	понад 80 країн	понад 10000 станцій	AQI, PM2.5, PM10, O ₃ , NO ₂ , SO ₂ , CO	для окремих міст

Як видно з таблиці 1.1, більшість сучасних систем забезпечують моніторинг якості повітря в режимі реального часу та надають програмний інтерфейс для доступу до екологічних даних. Водночас функції прогнозування реалізовані не в усіх рішеннях або мають обмежені можливості. Це створює передумови для розробки програмних систем, які поєднують засоби моніторингу із сучасними методами машинного навчання для підвищення точності прогнозування рівня забруднення атмосферного повітря [9-11].

1.3 Аналіз методів прогнозування рівня забруднення повітря

Прогнозування рівня забруднення атмосферного повітря є важливим елементом сучасних екологічних інформаційних систем. Наявність достовірного прогнозу дозволяє своєчасно реагувати на погіршення якості повітря, планувати

природоохоронні заходи та інформувати населення про можливі ризики для здоров'я. Для розв'язання задач прогнозування використовуються різні математичні, статистичні та інтелектуальні методи, які відрізняються складністю реалізації, вимогами до даних та точністю отриманих результатів [12].

Традиційно для аналізу часових рядів екологічних показників застосовуються статистичні методи. До найбільш поширених належать лінійна регресія та моделі ARIMA. Їх перевагою є простота реалізації та невисокі вимоги до обчислювальних ресурсів. Проте такі методи ефективно працюють лише за наявності відносно простих залежностей між параметрами та мають обмежені можливості щодо врахування великої кількості факторів.

З розвитком технологій аналізу даних широкого поширення набули методи машинного навчання. Такі алгоритми здатні враховувати складні нелінійні залежності між концентраціями забруднювальних речовин, погодними умовами, сезонністю та іншими параметрами. Для прогнозування якості повітря часто використовуються дерева рішень, випадкові ліси, градієнтний бустинг та метод опорних векторів. Ці алгоритми забезпечують вищу точність порівняно зі статистичними підходами та можуть ефективно працювати з великими наборами даних [13-15].

Останніми роками значну увагу приділяють методам глибокого навчання. Для прогнозування часових рядів особливо часто застосовуються рекурентні нейронні мережі та моделі LSTM, які здатні враховувати залежності між значеннями показників у різні моменти часу. Такі моделі демонструють високу точність прогнозування, однак потребують значних обчислювальних ресурсів та великих обсягів навчальних даних [16].

Порівняльну характеристику найбільш поширених методів прогнозування рівня забруднення повітря наведено в таблиці 1.2.

Таблиця 1.2 – Порівняння методів прогнозування рівня забруднення повітря

Метод	Тип моделі	Складність реалізації	Обсяг даних	Точність прогнозування	Швидкість навчання
Лінійна регресія	Статистичний	Низька	Малий	Низька–середня	Висока
ARIMA	Статистичний	Середня	Малий–середній	Середня	Висока
Random Forest	Машинне навчання	Середня	Середній	Висока	Середня
XGBoost	Машинне навчання	Середня	Середній–великий	Висока	Середня
Support Vector Regression	Машинне навчання	Висока	Середній	Висока	Низька
LSTM	Глибоке навчання	Висока	Великий	Дуже висока	Низька

З наведеного порівняння видно, що статистичні методи є простими у використанні та потребують менших обчислювальних ресурсів, проте поступаються сучасним алгоритмам машинного навчання за точністю прогнозування. Моделі Random Forest та XGBoost забезпечують хороше співвідношення між точністю та складністю реалізації, що робить їх популярними для задач прогнозування екологічних показників. Найвищу потенційну точність демонструють моделі глибокого навчання, зокрема LSTM, які здатні враховувати часові залежності в даних. Саме тому методи машинного навчання та глибокого навчання є перспективною основою для розробки сучасних програмних систем прогнозування рівня забруднення атмосферного повітря [17-19].

1.4 Використання методів машинного навчання для екологічного моніторингу

Стрімке зростання обсягів екологічних даних сприяло активному впровадженню методів машинного навчання у системи моніторингу навколишнього середовища. На відміну від традиційних статистичних підходів, алгоритми машинного навчання здатні автоматично виявляти приховані

закономірності у великих наборах даних та формувати прогнози на основі отриманих залежностей. Це дозволяє підвищити точність аналізу екологічних показників і своєчасно виявляти потенційні загрози для довкілля.

У задачах моніторингу якості повітря методи машинного навчання використовуються для прогнозування концентрацій забруднювальних речовин, оцінювання індексу якості повітря, виявлення аномальних значень та аналізу впливу погодних умов на рівень забруднення. Як вхідні дані можуть використовуватися показники концентрації PM2.5, PM10, NO₂, SO₂, CO та O₃, а також температура повітря, вологість, атмосферний тиск і швидкість вітру [5].

Найбільш поширені алгоритми машинного навчання для екологічного моніторингу наведено в таблиці 1.3.

Таблиця 1.3 – Порівняння методів машинного навчання для прогнозування якості повітря

Метод	Основне призначення	Переваги	Недоліки
Лінійна регресія	Прогнозування числових показників	Простота реалізації та інтерпретації	Низька точність для складних залежностей
Дерево рішень	Класифікація та регресія	Зрозуміла структура моделі	Схильність до перенавчання
Random Forest	Прогнозування та аналіз факторів впливу	Висока точність і стійкість до шуму	Потребує більше обчислювальних ресурсів
XGBoost	Точне прогнозування часових рядів	Висока продуктивність та якість прогнозів	Складність налаштування параметрів
Нейронні мережі	Аналіз складних нелінійних залежностей	Можливість роботи з великими обсягами даних	Високі вимоги до обчислювальних ресурсів

Як видно з таблиці, для задач прогнозування якості повітря найбільш ефективними є ансамблеві методи та нейронні мережі, які здатні враховувати складні взаємозв'язки між екологічними та метеорологічними показниками. Водночас простіші алгоритми можуть використовуватися як базові моделі для порівняння результатів [20].

Таким чином, використання методів машинного навчання дозволяє підвищити ефективність екологічного моніторингу, забезпечити більш точне прогнозування рівня забруднення повітря та створити основу для розробки інтелектуальних програмних систем підтримки прийняття рішень у сфері охорони навколишнього середовища.

1.5 Аналіз джерел даних та наборів даних для прогнозування якості повітря

Ефективність систем прогнозування якості повітря значною мірою залежить від якості та повноти вхідних даних. Для навчання моделей машинного навчання використовуються як дані з автоматизованих станцій моніторингу, так і відкриті набори даних, що містять інформацію про концентрацію забруднювальних речовин та метеорологічні показники.

Основними джерелами даних є державні системи моніторингу навколишнього середовища, міжнародні екологічні платформи та відкриті репозиторії даних. Такі ресурси забезпечують доступ до історичних вимірювань концентрацій PM_{2.5}, PM₁₀, NO₂, SO₂, CO та O₃, які використовуються для побудови прогнозних моделей. Додатково можуть залучатися метеорологічні дані, оскільки температура повітря, вологість, атмосферний тиск і швидкість вітру безпосередньо впливають на поширення забруднювальних речовин [21].

Для розробки систем прогнозування якості повітря найчастіше використовуються відкриті набори даних, що містять значні обсяги історичних спостережень. Основні характеристики таких наборів даних наведено в таблиці 1.4.

Серед розглянутих наборів даних найбільш придатними для навчання моделей машинного навчання є Beijing Air Quality Dataset та OpenAQ, оскільки вони містять великий обсяг історичних даних і широкий набір екологічних та метеорологічних показників. Використання таких даних дозволяє формувати

більш точні прогнози та підвищувати ефективність систем моніторингу якості повітря.

Таблиця 1.4 – Порівняння наборів даних для прогнозування якості повітря

Набір даних	Кількість записів	Основні показники	Період спостережень
Beijing Air Quality Dataset	420 768	PM2.5, PM10, SO ₂ , NO ₂ , CO, O ₃ , метеодані	2013–2017
Air Quality UCI Dataset	9 358	CO, NO _x , NO ₂ , температура, вологість	2004–2005
OpenAQ	Понад 500 млн	PM2.5, PM10, NO ₂ , SO ₂ , CO, O ₃	Постійно оновлюється
WAQI Data Platform	Понад 100 млн	AQI, PM2.5, PM10, O ₃ , NO ₂	Постійно оновлюється
IQAir Historical Data	Понад 80 тис. локацій	PM2.5, PM10, AQI	Постійно оновлюється

Для розробки програмної системи доцільно використовувати відкриті джерела даних, які підтримують автоматичне оновлення інформації та надають програмний інтерфейс доступу до екологічних показників. Це забезпечує можливість подальшого масштабування системи та використання актуальних даних для прогнозування рівня забруднення атмосферного повітря.

1.6 Формування вимог до програмної системи

На основі проведеного аналізу предметної області було визначено основні вимоги до програмної системи моніторингу та прогнозування рівня забруднення повітря. Система повинна забезпечувати збір екологічних даних, їх зберігання, аналіз, візуалізацію та прогнозування майбутніх значень показників якості повітря з використанням методів машинного навчання.

Основними користувачами системи є адміністратор та користувач. Користувач отримує доступ до актуальної інформації про якість повітря та результатів прогнозування, тоді як адміністратор відповідає за керування

джерелами даних, оновлення моделей машинного навчання та підтримку працездатності системи.

Основні функції програмної системи наведено в таблиці 1.5.

Таблиця 1.5 – Функціональні вимоги до програмної системи

Код	Функціональна вимога
FR-01	Отримання даних про якість повітря з відкритих джерел
FR-02	Збереження екологічних даних у базі даних
FR-03	Перегляд поточних показників якості повітря
FR-04	Візуалізація історичних даних у вигляді графіків
FR-05	Формування прогнозу рівня забруднення повітря
FR-06	Відображення результатів прогнозування
FR-07	Оновлення наборів даних для навчання моделей
FR-08	Навчання та перенавчання моделей машинного навчання
FR-09	Керування користувачами системи
FR-10	Формування аналітичних звітів

Крім функціональних можливостей, програмна система повинна відповідати низці нефункціональних вимог, які визначають її якість, надійність та зручність використання які наведено в таблиці 1.6.

Таблиця 1.6 – Нефункціональні вимоги до програмної системи

Код	Нефункціональна вимога
NFR-01	Вебінтерфейс повинен бути доступним через сучасні браузері
NFR-02	Час формування прогнозу не повинен перевищувати 5 секунд
NFR-03	Система повинна забезпечувати цілісність даних
NFR-04	Передбачити авторизацію користувачів

Продовження таблиці 1.6

NFR-05	Підтримувати масштабування при збільшенні обсягу даних
NFR-06	Забезпечувати резервне копіювання даних
NFR-07	Підтримувати можливість інтеграції з зовнішніми API
NFR-08	Забезпечувати зручний інтерфейс користувача

Взаємодію користувачів із системою доцільно представити за допомогою діаграми варіантів використання, яка наведена на рисунку 1.2.

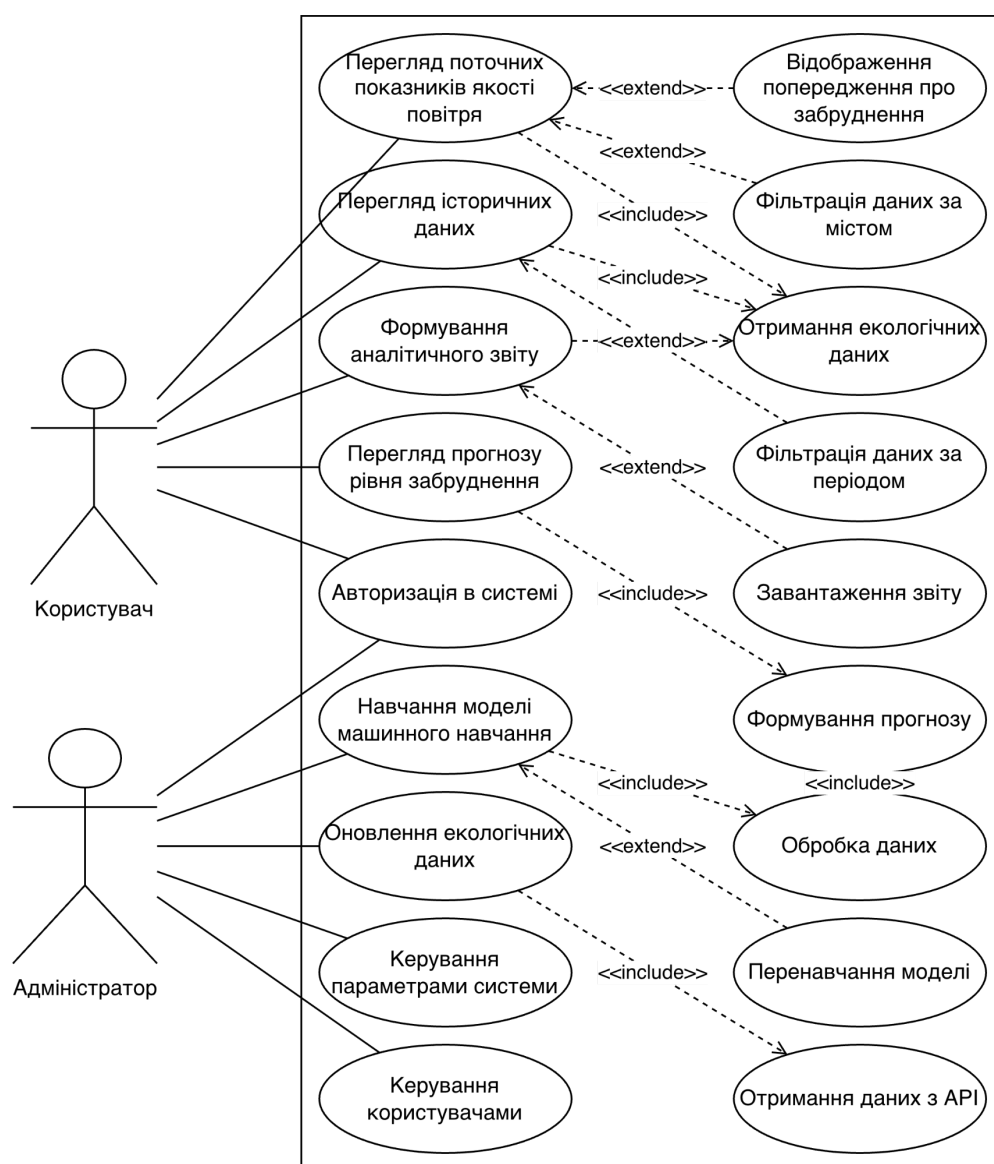


Рисунок 1.2 – Діаграма варіантів використання програмної системи моніторингу та прогнозування рівня забруднення повітря

Діаграма варіантів використання, наведена на рисунку 1.3, відображає основні сценарії взаємодії користувачів із програмною системою моніторингу та прогнозування рівня забруднення повітря. У системі визначено двох основних акторів: користувача та адміністратора [22].

Користувач має можливість переглядати поточні показники якості повітря, аналізувати історичні дані, переглядати результати прогнозування та формувати аналітичні звіти. Для отримання необхідної інформації система використовує механізми збору та обробки екологічних даних. Додатково користувач може застосовувати фільтрацію даних за містом або періодом спостереження, а також отримувати попередження у випадку перевищення допустимих рівнів забруднення.

Адміністратор відповідає за підтримку працездатності системи та керування її функціонуванням. До його основних завдань належать керування користувачами, оновлення екологічних даних, налаштування параметрів системи та запуск навчання моделей машинного навчання. У разі необхідності адміністратор може виконувати перенавчання моделей для підвищення точності прогнозування.

На діаграмі також відображено зв'язки типу \diamond та \diamond . Зв'язки \diamond використовуються для обов'язкових операцій, які виконуються під час реалізації основних функцій системи, зокрема отримання даних, їх обробки та формування прогнозів. Зв'язки \diamond описують додаткові можливості, які виконуються за певних умов, наприклад фільтрацію даних, завантаження звітів, відображення попереджень або перенавчання моделей [17-23].

Таким чином, діаграма варіантів використання дозволяє визначити основні функціональні можливості програмної системи та взаємодію користувачів із її компонентами, що є основою для подальшого проектування архітектури програмного забезпечення. На діаграмі доцільно відобразити таких акторів:

Визначені функціональні та нефункціональні вимоги формують основу для подальшого проектування архітектури програмної системи. Їх врахування

дозволить забезпечити ефективне виконання основних задач моніторингу та прогнозування рівня забруднення атмосферного повітря, а також створити масштабоване та надійне програмне рішення.

1.7 Формування постановки задачі та висновки до розділу

Проведений аналіз предметної області показав актуальність використання сучасних інформаційних технологій для моніторингу та прогнозування рівня забруднення атмосферного повітря. Існуючі системи забезпечують збір і візуалізацію екологічних даних, проте мають обмежені можливості щодо прогнозування майбутнього стану повітря.

Основною задачею роботи є розробка програмної системи, яка забезпечуватиме збір, зберігання та аналіз екологічних даних, а також прогнозування рівня забруднення повітря з використанням методів машинного навчання. Система повинна надавати користувачам доступ до поточних показників якості повітря, історичних даних та результатів прогнозування [24].

У результаті виконання першого розділу було проаналізовано проблему забруднення повітря, розглянуто сучасні системи моніторингу, методи прогнозування та можливості використання машинного навчання. Також визначено функціональні та нефункціональні вимоги до програмної системи, які будуть використані під час її проектування та реалізації в наступному розділі.

1.8 Висновки до першого розділу

У першому розділі було проведено аналіз предметної області та досліджено проблему забруднення атмосферного повітря, яка є однією з актуальних екологічних проблем сучасності. Розглянуто основні джерела забруднення повітря та їх вплив на навколишнє середовище і здоров'я населення.

Було виконано огляд сучасних систем моніторингу якості повітря, проаналізовано методи прогнозування рівня забруднення та особливості

застосування методів машинного навчання для екологічного моніторингу. Також досліджено доступні джерела даних і набори даних, які можуть використовуватися для навчання та оцінювання прогнозних моделей [25].

На основі проведеного аналізу сформовано функціональні та нефункціональні вимоги до програмної системи, визначено основних користувачів і варіанти їх взаємодії із системою. Отримані результати стали основою для проектування архітектури програмного забезпечення, структури бази даних та вибору технологій реалізації, які розглянуто в наступному розділі.

2 ПРОЄКТУВАННЯ ТА РОЗРОБКА ПРОГРАМНОЇ СИСТЕМИ МОНІТОРИНГУ ТА ПРОГНОЗУВАННЯ ЗАБРУДНЕННЯ ПОВІТРЯ

На основі результатів аналізу предметної області та сформованих вимог до програмної системи виконується етап її проєктування та розробки. На цьому етапі визначаються технології та засоби реалізації, проєктується архітектура системи, структура зберігання даних і взаємодія між окремими компонентами. Особлива увага приділяється реалізації модуля прогнозування, який використовує методи машинного навчання для аналізу екологічних показників та формування прогнозів рівня забруднення атмосферного повітря.

У даному розділі обґрунтовано вибір технологічного стеку, розроблено архітектуру програмної системи, спроектовано базу даних та описано процес реалізації основних програмних модулів. Також розглянуто особливості інтеграції компонентів системи та механізми взаємодії користувача з вебзастосунком. Отримані результати є основою для подальшого тестування та оцінювання ефективності розробленого програмного забезпечення.

2.1 Обґрунтування вибору технологій та засобів розробки

Для реалізації програмної системи моніторингу та прогнозування рівня забруднення повітря необхідно обрати технології, які забезпечують ефективну обробку даних, навчання моделей машинного навчання, зберігання інформації та зручну взаємодію користувача із системою. Вибір технологічного стеку здійснювався з урахуванням функціональних вимог до системи, доступності інструментів, продуктивності та можливості подальшого масштабування.

Для розробки серверної частини системи обрано мову програмування Python. Дана мова широко використовується в задачах аналізу даних та машинного навчання завдяки великій кількості спеціалізованих бібліотек і фреймворків. Python забезпечує швидку розробку програмного забезпечення та підтримує інтеграцію з різноманітними джерелами даних [26].

Для створення вебзастосунку використано фреймворк Flask. Він дозволяє реалізувати серверну логіку, REST API та взаємодію між користувацьким інтерфейсом, базою даних і модулем прогнозування. Перевагами Flask є простота налаштування, невеликий обсяг службового коду та гнучкість при побудові вебсистем [17-20].

Для роботи з даними використано бібліотеки Pandas та NumPy. Бібліотека Pandas забезпечує завантаження, очищення, фільтрацію та аналіз табличних даних, тоді як NumPy використовується для виконання математичних операцій над масивами даних.

Для реалізації моделей машинного навчання обрано бібліотеку Scikit-learn. Вона містить готові алгоритми регресії, класифікації та оцінювання якості моделей, що дозволяє швидко реалізувати та протестувати різні підходи до прогнозування рівня забруднення повітря.

Зберігання даних забезпечується системою керування базами даних PostgreSQL. Дана СКБД підтримує роботу з великими обсягами інформації, забезпечує надійність зберігання даних та можливість виконання складних запитів [17].

Для візуалізації результатів моніторингу та прогнозування використано HTML, CSS та JavaScript. Ці технології дозволяють створити зручний вебінтерфейс для відображення екологічних показників, графіків та прогнозів.

Основні технології, використані під час розробки системи, наведено в таблиці 2.1.

Таблиця 2.1 – Технології та засоби розробки програмної системи

Технологія	Призначення
Python	Розробка серверної частини та модуля прогнозування
Flask	Реалізація вебзастосунку та REST API
Pandas	Обробка та аналіз даних

Продовження таблиці 2.1

NumPy	Математичні обчислення
Scikit-learn	Реалізація моделей машинного навчання
PostgreSQL	Зберігання екологічних даних
HTML/CSS	Розробка користувацького інтерфейсу
JavaScript	Інтерактивна взаємодія з вебінтерфейсом
Matplotlib	Побудова графіків та візуалізація результатів

Таким чином, обраний набір технологій забезпечує реалізацію всіх функціональних можливостей програмної системи, включаючи збір та зберігання даних, їх аналіз, прогнозування показників якості повітря та відображення результатів користувачам.

2.2 Проектування архітектури програмної системи

Архітектура програмної системи визначає структуру її компонентів, способи взаємодії між ними та розподіл функціональних обов'язків. Під час проектування архітектури основною метою було забезпечення модульності, масштабованості та можливості подальшого розширення функціональності системи. Запропонована архітектура дозволяє відокремити процеси збору даних, їх обробки, прогнозування та відображення результатів користувачам [27].

Програмна система побудована за клієнт-серверною архітектурою та складається з декількох взаємопов'язаних модулів. Користувач взаємодіє із системою через вебінтерфейс, який забезпечує перегляд поточних показників якості повітря, історичних даних та результатів прогнозування. Серверна частина відповідає за обробку запитів користувачів, взаємодію з базою даних, отримання даних із зовнішніх джерел та виконання прогнозування за допомогою моделей машинного навчання.

Джерелами інформації для системи є відкриті екологічні платформи та API сервісів моніторингу якості повітря. Отримані дані проходять етап попередньої обробки, після чого зберігаються у базі даних і можуть використовуватися як для

відображення користувачам, так і для навчання моделей машинного навчання. Модуль прогнозування аналізує накопичені дані та формує прогнозні значення показників забруднення повітря на визначений часовий інтервал.

Загальну структуру програмної системи наведено на рисунку 2.1.

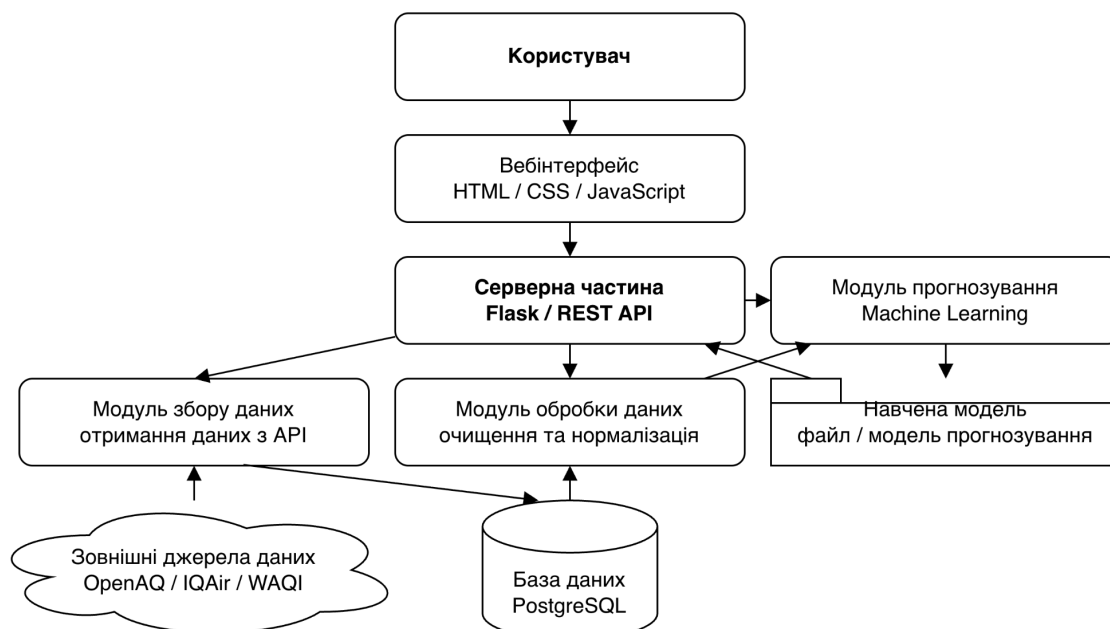


Рисунок 2.1 – Архітектура програмної системи моніторингу та прогнозування рівня забруднення повітря

До складу архітектури входять такі основні компоненти:

- Вебінтерфейс користувача забезпечує взаємодію користувача із системою. За його допомогою здійснюється перегляд поточних показників якості повітря, аналіз історичних даних, перегляд результатів прогнозування та формування звітів.
- Серверна частина системи відповідає за обробку запитів користувачів, взаємодію між окремими модулями системи, доступ до бази даних та передачу результатів до вебінтерфейсу. Саме цей компонент координує роботу всієї системи.

- Модуль збору екологічних даних виконує автоматичне отримання інформації із зовнішніх джерел моніторингу якості повітря через API. Отримані дані перевіряються та передаються для подальшої обробки і збереження.
- Модуль обробки даних здійснює очищення, фільтрацію та підготовку даних до аналізу. На цьому етапі усуваються пропущені значення, виконуються перетворення форматів даних та формуються набори даних для навчання моделей машинного навчання.
- Модуль машинного навчання та прогнозування реалізує алгоритми прогнозування рівня забруднення повітря. На основі історичних екологічних даних модуль навчає модель та формує прогнозні значення показників якості повітря на визначений часовий період.
- База даних використовується для централізованого зберігання екологічних показників, результатів прогнозування та службової інформації системи. Вона забезпечує швидкий доступ до даних і підтримує їх цілісність.
- Зовнішні джерела даних (API сервісів моніторингу якості повітря) надають актуальну інформацію про концентрацію забруднювальних речовин та метеорологічні показники. Саме ці дані використовуються для моніторингу поточного стану атмосферного повітря та побудови прогнозів.

Запропонована архітектура забезпечує незалежність окремих модулів, спрощує супровід програмного забезпечення та дозволяє розширювати функціональність системи без зміни її основної структури. Крім того, такий підхід забезпечує можливість заміни моделей машинного навчання або джерел даних без впливу на інші компоненти програмної системи [28].

2.3 Проєктування структури бази даних

Для зберігання інформації в програмній системі моніторингу та прогнозування рівня забруднення повітря використовується система керування базами даних PostgreSQL. Вибір даної СКБД обумовлений її надійністю, високою продуктивністю, підтримкою складних SQL-запитів та можливістю ефективною

роботи з великими обсягами даних. PostgreSQL забезпечує механізми підтримки цілісності даних, резервного копіювання та масштабування, що є важливими для систем екологічного моніторингу [17].

Під час проектування бази даних було використано реляційну модель, яка дозволяє структурувати інформацію у вигляді взаємопов'язаних таблиць. Такий підхід спрощує зберігання та обробку екологічних даних, а також забезпечує ефективну взаємодію між окремими модулями програмної системи.

Загальну структуру бази даних наведено на рисунку 2.2.

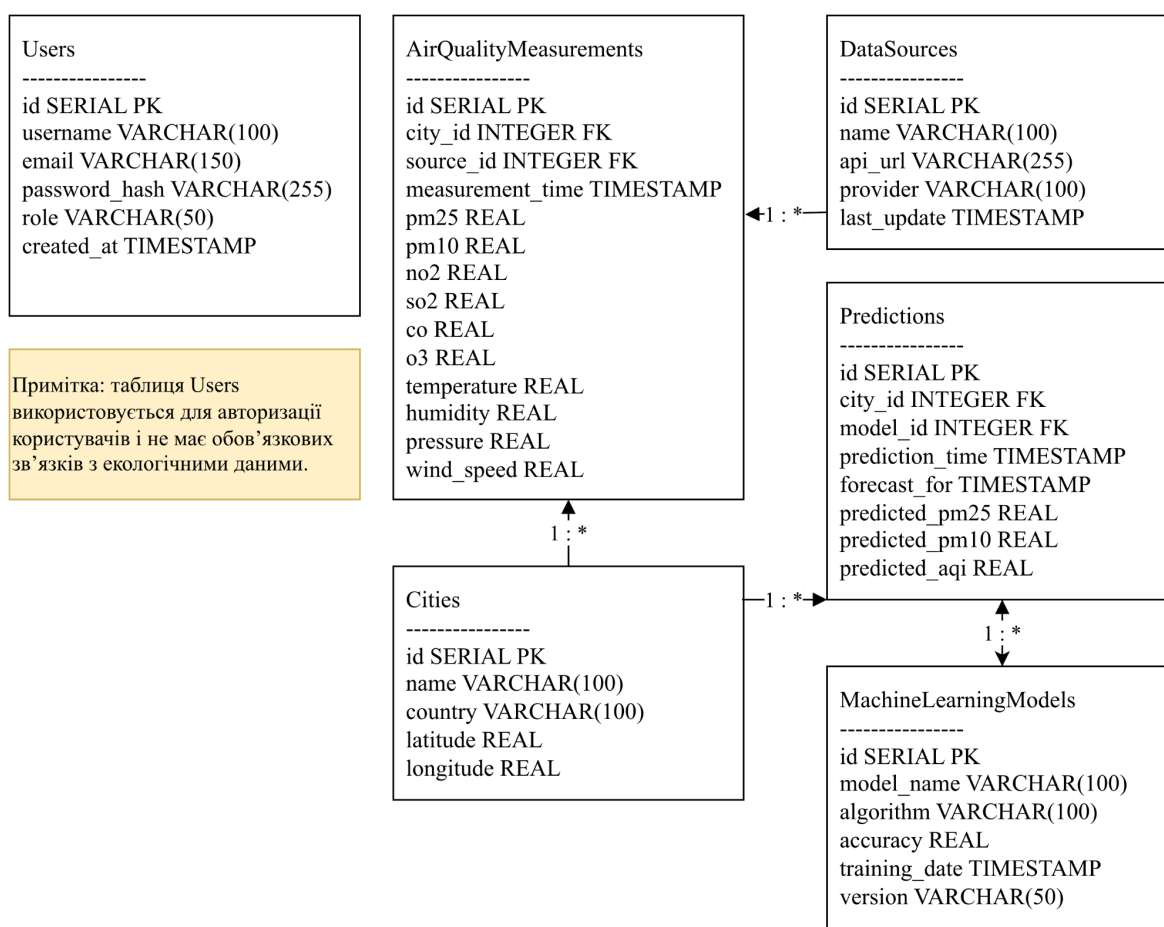


Рисунок 2.2 – ER-діаграма бази даних програмної системи моніторингу та прогнозування рівня забруднення повітря

Для реалізації поставлених задач у базі даних передбачено шість основних таблиць: Users, Cities, DataSources, AirQualityMeasurements, MachineLearningModels та Predictions.

Таблиця Users використовується для зберігання інформації про користувачів системи та їх ролі (див. табл. 2.2).

Таблиця 2.2 – Структура таблиці Users

Поле	Тип даних	Опис
id	SERIAL	Ідентифікатор користувача
username	VARCHAR(100)	Ім'я користувача
email	VARCHAR(150)	Електронна адреса
password_hash	VARCHAR(255)	Хеш пароля
role	VARCHAR(50)	Роль користувача
created_at	TIMESTAMP	Дата створення запису

Таблиця Cities містить інформацію про населені пункти, для яких здійснюється моніторинг якості повітря (див. табл. 2.3).

Таблиця 2.3 – Структура таблиці Cities

Поле	Тип даних	Опис
id	SERIAL	Ідентифікатор міста
name	VARCHAR(100)	Назва міста
country	VARCHAR(100)	Країна
latitude	REAL	Географічна широта
longitude	REAL	Географічна довгота

Таблиця DataSources призначена для зберігання інформації про зовнішні сервіси, з яких отримуються екологічні дані (див. табл. 2.4).

Таблиця 2.4 – Структура таблиці DataSources

Поле	Тип даних	Опис
id	SERIAL	Ідентифікатор джерела
name	VARCHAR(100)	Назва джерела даних
api_url	VARCHAR(255)	Адреса API
provider	VARCHAR(100)	Постачальник даних
last_update	TIMESTAMP	Час останнього оновлення

Основною таблицею системи є AirQualityMeasurements, яка містить результати вимірювань якості повітря та метеорологічні показники (див. табл. 2.5).

Таблиця 2.5 – Структура таблиці AirQualityMeasurements

Поле	Тип даних	Опис
id	SERIAL	Ідентифікатор запису
city_id	INTEGER	Посилання на місто
source_id	INTEGER	Посилання на джерело даних
measurement_time	TIMESTAMP	Час вимірювання
pm25	REAL	Концентрація PM2.5
pm10	REAL	Концентрація PM10
no2	REAL	Концентрація NO ₂
so2	REAL	Концентрація SO ₂
co	REAL	Концентрація CO
o3	REAL	Концентрація O ₃
temperature	REAL	Температура повітря
humidity	REAL	Вологість повітря
pressure	REAL	Атмосферний тиск
wind_speed	REAL	Швидкість вітру

Таблиця MachineLearningModels використовується для зберігання інформації про моделі машинного навчання, які застосовуються для прогнозування рівня забруднення повітря (див. табл. 2.6).

Таблиця 2.6 – Структура таблиці MachineLearningModels

Поле	Тип даних	Опис
id	SERIAL	Ідентифікатор моделі
model_name	VARCHAR(100)	Назва моделі
algorithm	VARCHAR(100)	Використаний алгоритм
accuracy	REAL	Точність моделі
training_date	TIMESTAMP	Дата навчання
version	VARCHAR(50)	Версія моделі

Таблиця Predictions призначена для зберігання результатів прогнозування, сформованих моделями машинного навчання (див. табл. 2.7).

Таблиця 2.7 – Структура таблиці Predictions

Поле	Тип даних	Опис
id	SERIAL	Ідентифікатор прогнозу
city_id	INTEGER	Посилання на місто
model_id	INTEGER	Посилання на модель
prediction_time	TIMESTAMP	Час формування прогнозу
forecast_for	TIMESTAMP	Час прогнозованого значення
predicted_pm25	REAL	Прогноз PM2.5
predicted_pm10	REAL	Прогноз PM10
predicted_aqi	REAL	Прогнозований індекс якості повітря

Основні зв'язки між таблицями наведено в таблиці 2.8.

Таблиця 2.8 – Зв'язки між таблицями бази даних

Батьківська таблиця	Дочірня таблиця	Тип зв'язку
Cities	AirQualityMeasurements	1
DataSources	AirQualityMeasurements	1
Cities	Predictions	1
MachineLearningModels	Predictions	1

Запропонована структура бази даних забезпечує зберігання історичних екологічних даних, результатів прогнозування та інформації про використані моделі машинного навчання. Використання PostgreSQL дозволяє ефективно працювати з великими обсягами даних та забезпечує надійну основу для функціонування програмної системи.

2.4 Розробка модуля збору та обробки даних про якість повітря

Модуль збору та обробки даних є одним із основних компонентів програмної системи, оскільки саме він забезпечує отримання вхідної інформації для подальшого аналізу та прогнозування. Його основним завданням є автоматичне отримання екологічних показників із зовнішніх джерел, перевірка їх коректності, приведення до єдиного формату та збереження у базі даних PostgreSQL.

Дані про якість повітря можуть надходити із відкритих API сервісів моніторингу, зокрема OpenAQ, IQAir або WAQI. Отримана інформація містить концентрації забруднювальних речовин PM_{2.5}, PM₁₀, NO₂, SO₂, CO, O₃, а також метеорологічні показники, які впливають на поширення забруднення: температуру, вологість, атмосферний тиск і швидкість вітру [16-19, 21].

Процес роботи модуля складається з кількох основних етапів: надсилання запиту до зовнішнього API, отримання відповіді у форматі JSON, перевірка наявності необхідних полів, очищення даних, приведення значень до потрібних типів і збереження результатів у таблиці AirQualityMeasurements. Такий підхід дозволяє забезпечити стабільне накопичення історичних даних, які надалі використовуються для побудови прогнозних моделей.

Після отримання даних виконується їх попередня обробка. На цьому етапі система перевіряє записи на наявність пропущених або некоректних значень, видаляє дублікати, нормалізує часовий формат та приводить числові показники до єдиних одиниць вимірювання. Це необхідно для забезпечення якості вхідних даних, оскільки неточні або неповні записи можуть негативно вплинути на результати прогнозування [28].

Загальний алгоритм роботи модуля збору та обробки даних можна подати так:

- отримання параметрів запиту, зокрема міста та періоду спостереження;
- надсилання HTTP-запиту до зовнішнього API;

- отримання відповіді у форматі JSON;
- перевірка структури та повноти отриманих даних;
- очищення та нормалізація значень;
- збереження підготовлених даних у базі даних PostgreSQL;
- передача даних до модуля прогнозування.

Для реалізації цього модуля доцільно використовувати бібліотеку requests для взаємодії із зовнішніми API та бібліотеку Pandas для обробки табличних даних. Збереження результатів у PostgreSQL може виконуватися через SQLAlchemy або psycopg2 [29].

Приклад логіки роботи модуля збору та обробки даних про якість повітря наведено у лістингу 2.1.

Лістинг 2.1 – Приклад функції отримання та попередньої обробки даних

```
import requests
import pandas as pd
def load_air_quality_data(api_url):
    response = requests.get(api_url, timeout=10)
    response.raise_for_status()
    data = response.json()
    records = data.get("results", [])
    df = pd.DataFrame(records)
    df = df.drop_duplicates()
    df = df.dropna()
    if "date" in df.columns:
        df["date"] = pd.to_datetime(df["date"])
    return df
```

Розроблений модуль забезпечує підготовку якісних вхідних даних для подальшого аналізу та прогнозування. Його використання дозволяє автоматизувати процес збору екологічної інформації, зменшити кількість ручних операцій та підвищити надійність роботи всієї програмної системи.

2.5 Розробка модуля прогнозування на основі методів машинного навчання

Модуль прогнозування є основним аналітичним компонентом програмної

системи. Його завдання полягає в обробці історичних даних про якість повітря, навчанні моделей машинного навчання та формуванні прогнозів майбутніх значень показників забруднення. На вхід модуль отримує підготовлені дані з бази даних PostgreSQL, після чого виконує прогнозування концентрацій PM2.5, PM10 та індексу якості повітря.

Для побудови прогнозової моделі використовуються історичні екологічні та метеорологічні показники. До основних вхідних ознак належать концентрації PM2.5, PM10, NO₂, SO₂, CO, O₃, температура повітря, вологість, атмосферний тиск і швидкість вітру. Цільовими значеннями прогнозування є майбутні показники PM2.5, PM10 та AQI [17, 18, 30].

У межах розробки програмної системи планується використати декілька моделей машинного навчання, щоб порівняти їхню ефективність і вибрати найкращий варіант для прогнозування. Основні моделі наведено в таблиці 2.8.

Таблиця 2.8 – Моделі машинного навчання для прогнозування якості повітря

Модель	Призначення	Особливість використання
Linear Regression	Базове прогнозування числових значень	Використовується як проста базова модель для порівняння
Decision Tree Regressor	Прогнозування на основі правил	Дозволяє враховувати нелінійні залежності в даних
Random Forest Regressor	Ансамблеве прогнозування	Підвищує стійкість прогнозу за рахунок використання набору дерев
Gradient Boosting Regressor	Точне прогнозування числових показників	Добре працює з табличними даними та складними залежностями
XGBoost Regressor	Оптимізоване ансамблеве прогнозування	Забезпечує високу точність і швидкість роботи на структурованих даних
LSTM	Прогнозування часових рядів	Дає змогу враховувати залежності між показниками

Для навчання моделей дані попередньо очищуються, нормалізуються та поділяються на навчальну і тестову вибірки. Після цього виконується навчання декількох моделей, оцінювання їх результатів за метриками MAE, RMSE та R², а

також збереження найкращої моделі для подальшого використання у системі прогнозування.

Загальний процес прогнозування показників якості повітря наведено на рисунку 2.3.

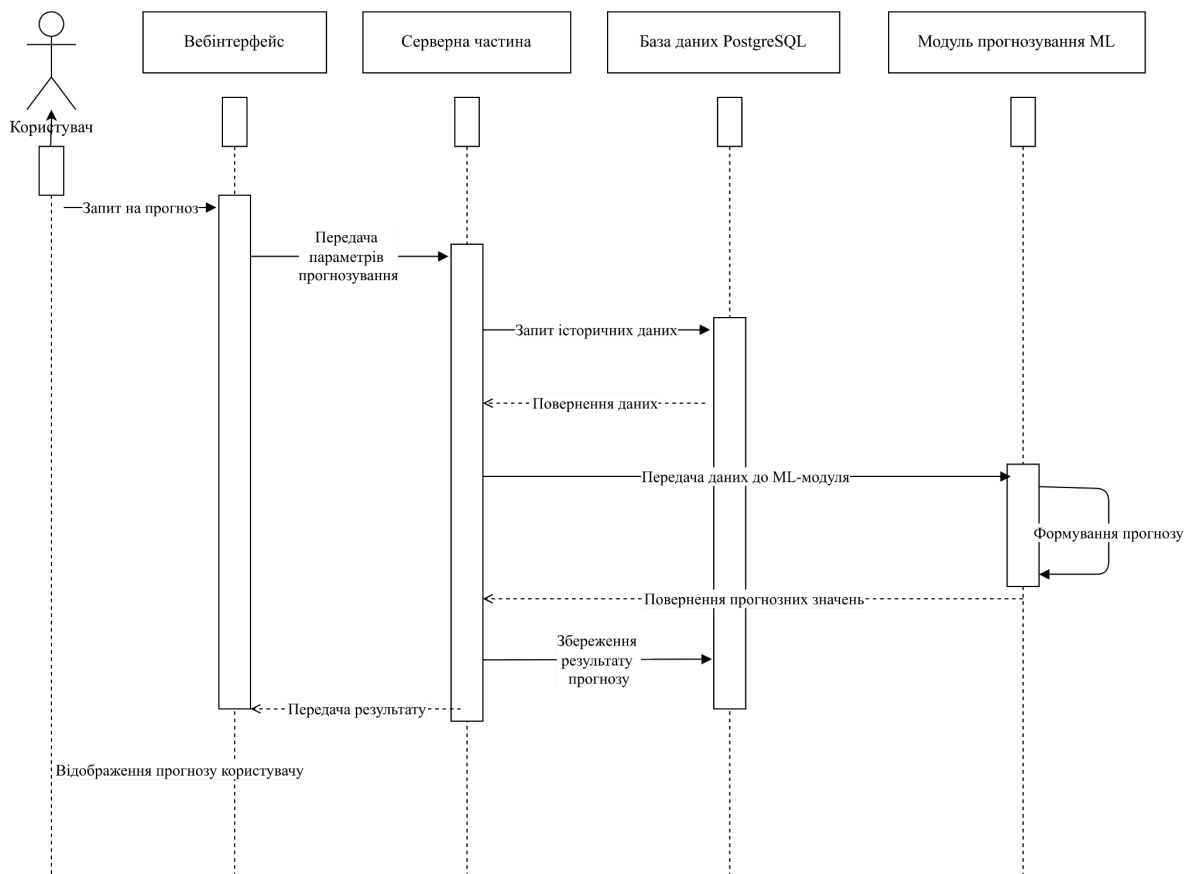


Рисунок 2.3 – Діаграма послідовності процесу прогнозування рівня забруднення повітря

На діаграмі послідовності відображено взаємодію між користувачем, вебінтерфейсом, серверною частиною, базою даних та модулем машинного навчання. Користувач надсилає запит на формування прогнозу через вебінтерфейс. Серверна частина приймає запит, отримує необхідні історичні дані з бази даних і передає їх до модуля прогнозування. Модуль машинного навчання обробляє дані, застосовує навчену модель та повертає прогнозні значення. Після цього сервер зберігає результат прогнозування у базі даних і передає його до вебінтерфейсу для відображення користувачу.

Результатом роботи модуля прогнозування є сформовані прогнозні значення показників забруднення повітря, які можуть бути використані для аналізу екологічної ситуації, побудови графіків і формування попереджень у разі очікуваного погіршення якості повітря.

2.6 Реалізація програмної системи та інтеграція компонентів

Після проектування архітектури та структури бази даних було виконано реалізацію програмної системи моніторингу та прогнозування рівня забруднення повітря. Основною особливістю розробленого рішення є інтеграція декількох взаємопов'язаних компонентів, які забезпечують повний цикл обробки даних — від їх отримання із зовнішніх джерел до формування прогнозів та відображення результатів користувачу.

Система реалізована за клієнт-серверною архітектурою. Користувач взаємодіє із системою через вебінтерфейс, який надсилає запити до серверної частини. Сервер обробляє отримані запити, звертається до бази даних, виконує необхідні операції з даними та повертає результати у зручному для відображення форматі [24].

Одним із ключових компонентів системи є модуль інтеграції із зовнішніми сервісами моніторингу якості повітря. Для отримання актуальної екологічної інформації використовуються відкриті API, які надають дані про концентрацію забруднювальних речовин та метеорологічні показники. Під час роботи системи сервер формує HTTP-запити до зовнішніх сервісів, отримує відповіді у форматі JSON та передає їх до модуля обробки даних.

Для забезпечення взаємодії між компонентами програмної системи використовується архітектурний стиль REST API. Усі запити від вебінтерфейсу надходять до серверної частини через HTTP-протокол, що дозволяє відокремити клієнтську та серверну логіку, а також спрощує подальше масштабування системи.

REST API забезпечує доступ до функцій отримання екологічних даних, перегляду історичних вимірювань та формування прогнозів. Передача інформації

між компонентами здійснюється у форматі JSON, який є компактним і зручним для обробки як на стороні сервера, так і у вебінтерфейсі.

Основні кінцеві точки REST API (endpoints) програмної системи наведено в таблиці 2.10.

Таблиця 2.10 – Основні REST API endpoints

Метод	Endpoint	Призначення
GET	/api/air-quality/current	Отримання поточних показників якості повітря
GET	/api/air-quality/history	Отримання історичних даних
GET	/api/cities	Отримання списку міст моніторингу
POST	/api/predictions/generate	Формування нового прогнозу
GET	/api/predictions/latest	Отримання останнього прогнозу
POST	/api/models/train	Запуск навчання моделі машинного навчання
GET	/api/models	Отримання інформації про доступні моделі
GET	/api/statistics	Отримання статистичних показників якості повітря

Під час роботи системи вебінтерфейс надсилає запити до відповідних endpoint серверної частини. Сервер обробляє отримані дані, взаємодіє з базою даних PostgreSQL або модулем машинного навчання та повертає результат у форматі JSON. Такий підхід забезпечує слабе зв'язування між компонентами системи та спрощує інтеграцію нових сервісів у майбутньому.

Взаємодія між серверною частиною та базою даних PostgreSQL реалізована за допомогою ORM-технології SQLAlchemy. Це дозволяє виконувати операції створення, читання, оновлення та видалення даних без використання великої кількості SQL-запитів, а також спрощує супровід програмного коду.

Для реалізації функцій прогнозування використовується окремий модуль машинного навчання. Після отримання запиту на формування прогнозу сервер завантажує необхідні історичні дані з бази даних та передає їх моделі машинного навчання. Отримані прогнозні значення повертаються до серверної частини, зберігаються у таблиці Predictions та відображаються користувачу через вебінтерфейс.

Основні програмні компоненти системи та їх призначення наведено в таблиці 2.9.

Таблиця 2.9 – Основні програмні компоненти системи

Компонент	Призначення
Вебінтерфейс	Взаємодія користувача із системою
Flask API	Обробка запитів та координація роботи модулів
PostgreSQL	Зберігання екологічних даних та прогнозів
SQLAlchemy	Взаємодія серверної частини з базою даних
Модуль збору даних	Отримання інформації із зовнішніх API
Модуль обробки даних	Очищення та підготовка даних
Модуль машинного навчання	Навчання моделей та прогнозування
Зовнішні API	Надання екологічних даних

Загальну взаємодію компонентів програмної системи доцільно представити у вигляді діаграми компонентів, яку наведено на рисунку 2.4.

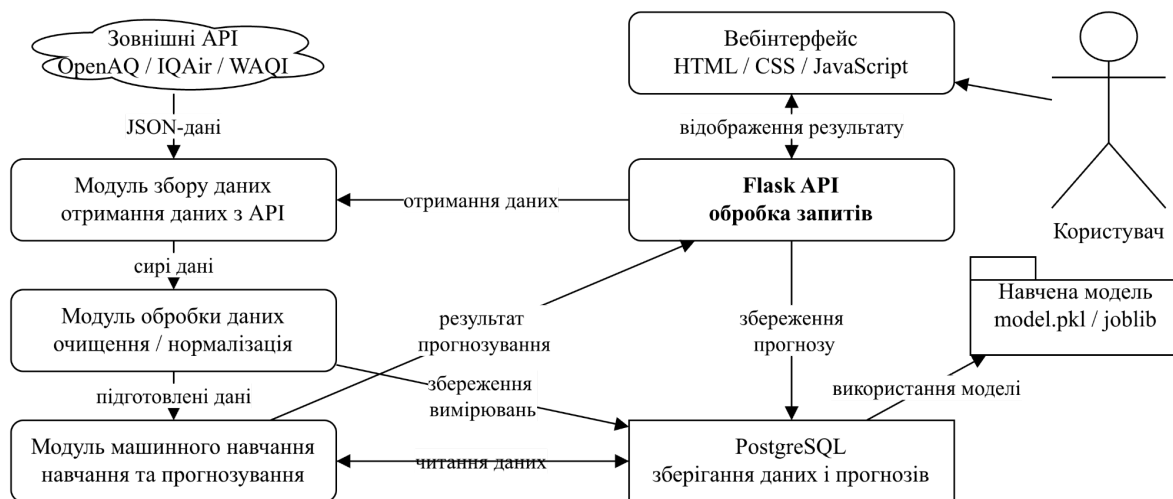


Рисунок 2.4 – Діаграма взаємодії компонентів програмної системи

Таким чином, реалізована система забезпечує інтеграцію модулів збору, обробки та прогнозування даних у межах єдиного програмного рішення. Використання REST API, PostgreSQL та бібліотек машинного навчання дозволяє

створити масштабовану систему, здатну автоматично аналізувати показники якості повітря та формувати прогнози на основі накопичених історичних даних.

2.7 Реалізація програмної системи та інтеграція компонентів

Вебінтерфейс є основним засобом взаємодії користувача з програмною системою моніторингу та прогнозування рівня забруднення повітря. Його основним призначенням є надання зручного доступу до поточних показників якості повітря, історичних даних та результатів прогнозування. Під час розробки вебінтерфейсу особлива увага приділялася простоті використання, наочності відображення інформації та швидкості отримання результатів.

Головна сторінка системи містить інформацію про поточний стан якості повітря для вибраного міста, основні екологічні показники та коротку статистичну інформацію. Загальний вигляд головної сторінки системи наведено на рисунку 2.5.

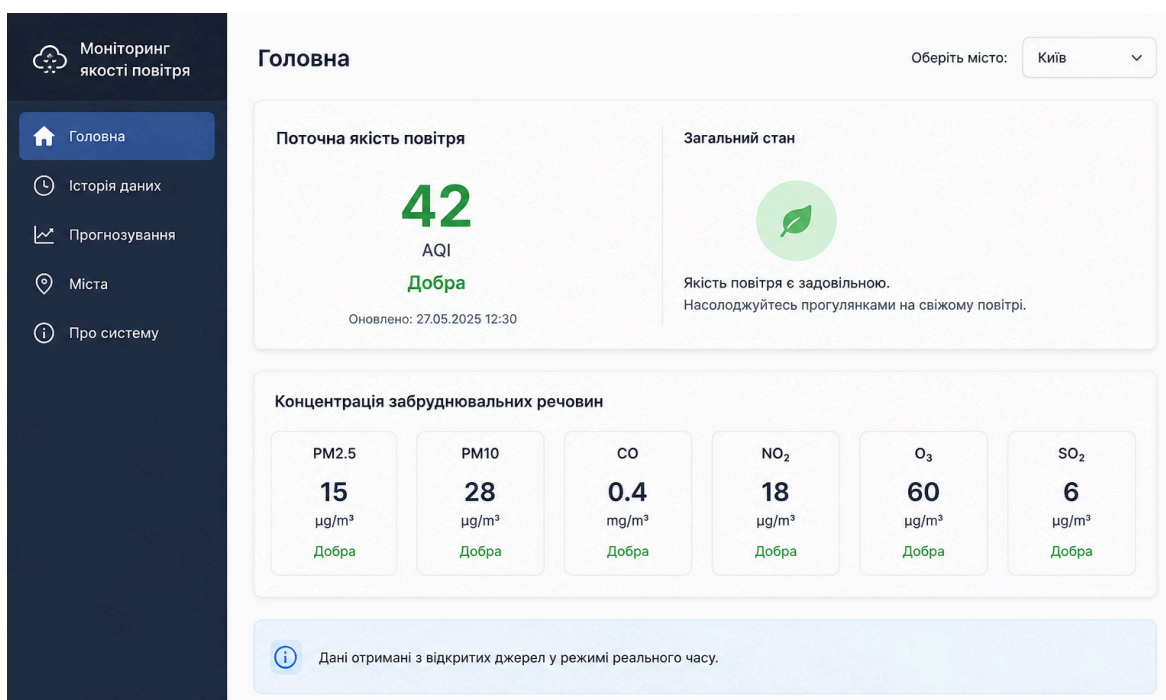
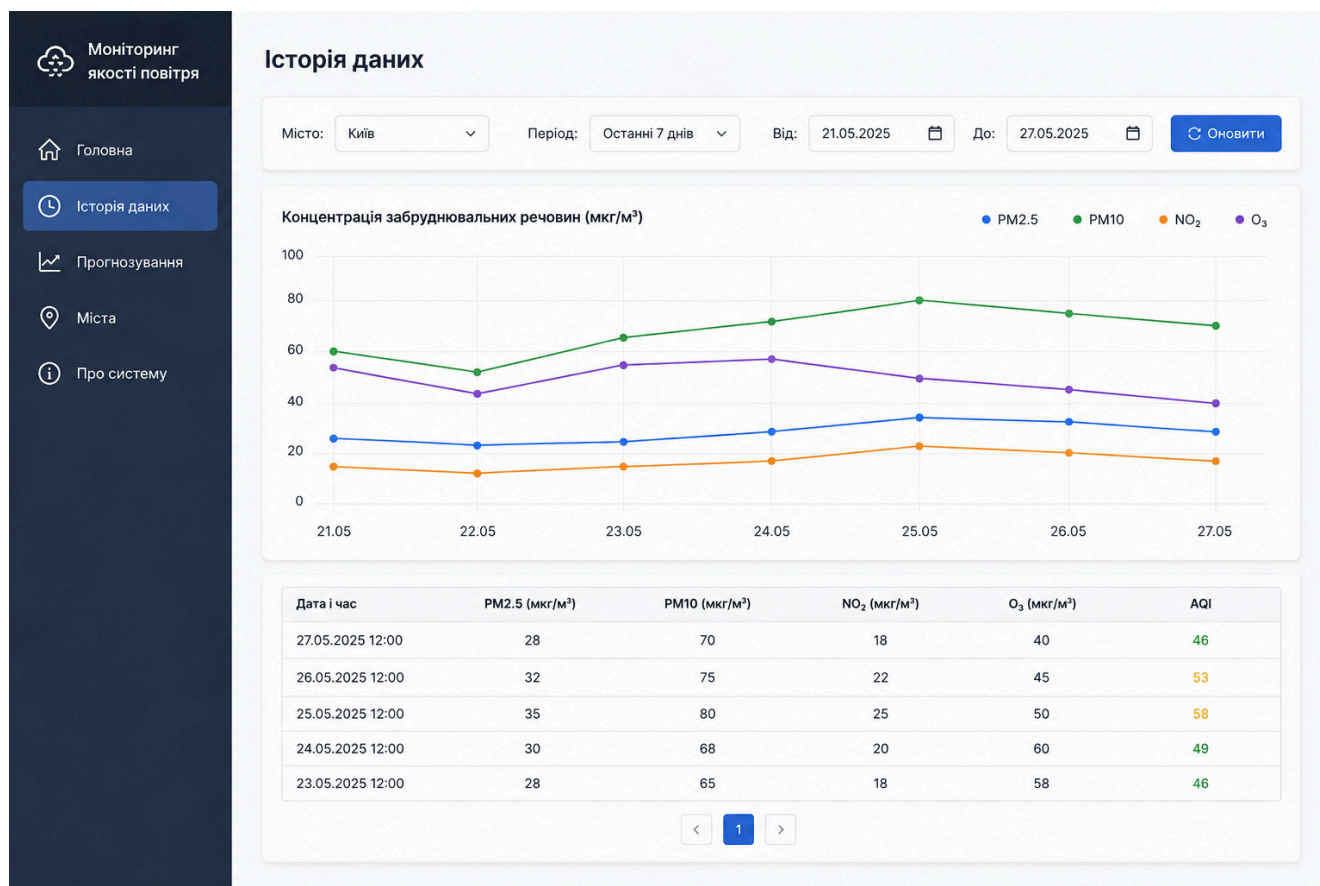


Рисунок 2.5 – Головна сторінка системи моніторингу якості повітря

Для аналізу динаміки змін показників якості повітря реалізовано сторінку перегляду історичних даних. На цій сторінці користувач може вибирати часовий

інтервал спостереження та переглядати зміни концентрацій основних забруднювальних речовин у вигляді графіків. Приклад відображення історичних даних наведено на рисунку 2.6.



Рисунк 2.6 – Відображення історичних даних про якість повітря

Однією з основних функцій програмної системи є формування прогнозів на основі методів машинного навчання. Для цього у вебінтерфейсі реалізовано окремий екран прогнозування, який дозволяє користувачу отримати прогнозні значення показників забруднення повітря на визначений часовий період. Результати прогнозування можуть відображатися у вигляді таблиць та графіків для зручності подальшого аналізу. Приклад сторінки прогнозування наведено на рисунку 2.7.

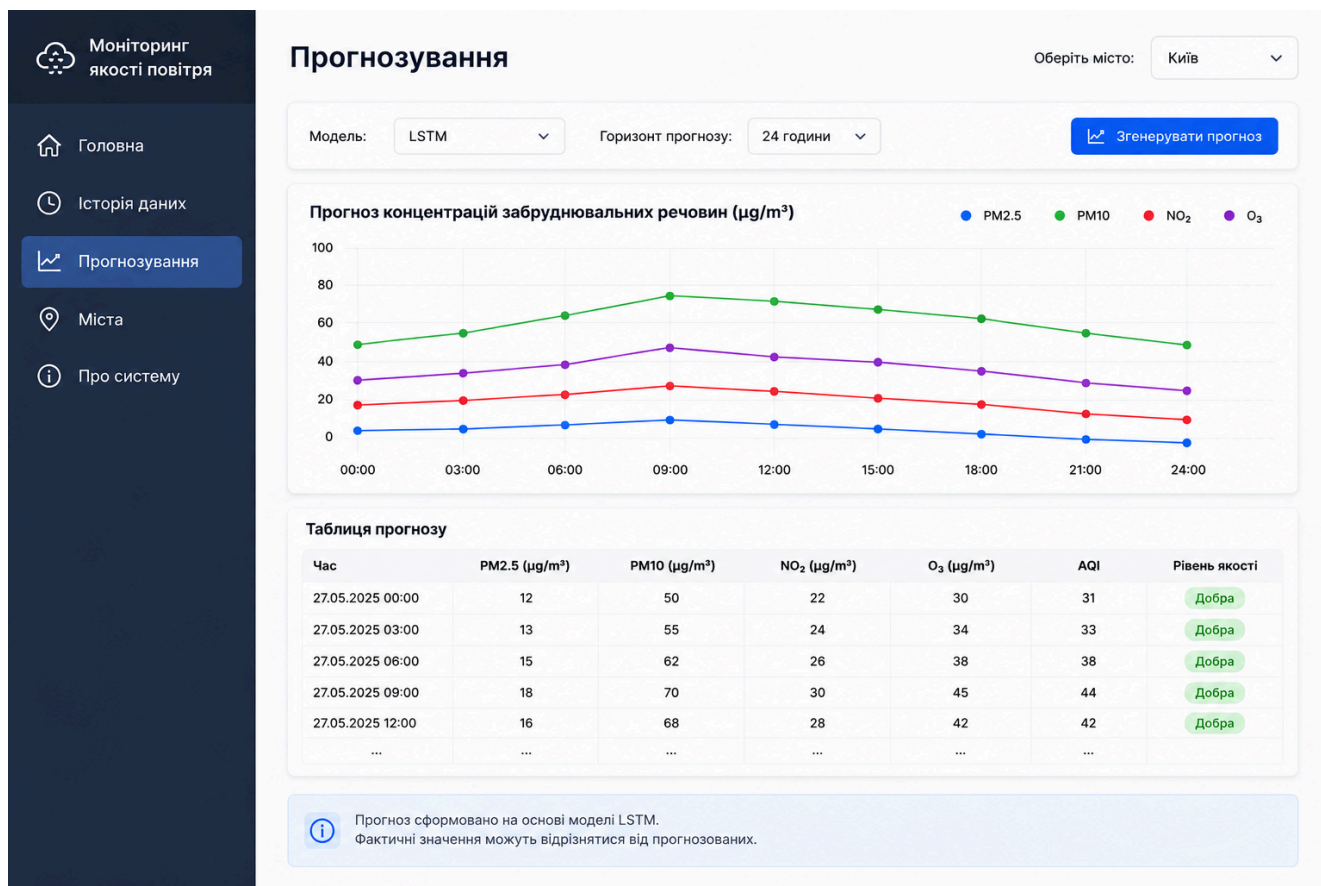


Рисунок 2.7 – Відображення результатів прогнозування рівня забруднення повітря

Розроблений вебінтерфейс забезпечує доступ до всіх основних функцій програмної системи та дозволяє користувачам отримувати актуальну інформацію про стан атмосферного повітря, аналізувати історичні дані та переглядати результати прогнозування. Використання сучасних вебтехнологій забезпечує зручність роботи із системою та можливість її використання на різних пристроях без встановлення додаткового програмного забезпечення.

2.8 Висновки до другого розлу

У другому розділі було виконано проектування та розробку програмної системи моніторингу та прогнозування рівня забруднення повітря. Обґрунтовано вибір технологій розробки, зокрема Python, Flask, PostgreSQL та бібліотек

машинного навчання, які забезпечують реалізацію функціональних вимог системи.

Було спроектовано архітектуру програмної системи, структуру бази даних та визначено взаємодію між основними компонентами. Розроблено модуль збору та обробки екологічних даних, модуль прогнозування на основі методів машинного навчання, а також реалізовано REST API для обміну даними між компонентами системи. Крім того, створено вебінтерфейс користувача для перегляду поточних показників якості повітря, аналізу історичних даних та отримання результатів прогнозування.

Отримані результати формують програмну основу системи та забезпечують можливість переходу до етапу тестування, оцінювання якості прогнозування та аналізу ефективності розробленого програмного забезпечення, які розглядаються в наступному розділі.

3 ТЕСТУВАННЯ, ОЦІНЮВАННЯ ТА ВПРОВАДЖЕННЯ ПРОГРАМНОЇ СИСТЕМИ

Для навчання та оцінювання моделей машинного навчання було сформовано набір даних, що містить історичні показники якості повітря та супутні метеорологічні параметри. Джерелом даних виступають відкриті екологічні сервіси моніторингу якості повітря, які надають інформацію про концентрацію забруднювальних речовин та погодні умови.

3.1 Підготовка набору даних для навчання та тестування моделей

Для навчання та оцінювання моделей машинного навчання було сформовано набір даних, що містить історичні показники якості повітря та супутні метеорологічні параметри. Джерелом даних виступають відкриті екологічні сервіси моніторингу якості повітря, які надають інформацію про концентрацію забруднювальних речовин та погодні умови.

На етапі підготовки даних було виконано їх очищення та попередню обробку. З набору даних вилучено дублікати записів, оброблено пропущені значення та перевірено коректність числових показників. Також виконано перетворення часових міток до єдиного формату та приведення всіх даних до узгоджених одиниць вимірювання.

До набору ознак увійшли концентрації PM_{2.5}, PM₁₀, NO₂, SO₂, CO та O₃, а також метеорологічні показники, зокрема температура повітря, вологість, атмосферний тиск і швидкість вітру. Цільовими змінними для прогнозування є майбутні значення показників PM_{2.5}, PM₁₀ та індексу якості повітря (AQI).

Після завершення попередньої обробки дані було поділено на навчальну та тестову вибірки у співвідношенні 80% та 20% відповідно. Навчальна вибірка використовувалася для побудови моделей машинного навчання, тоді як тестова — для оцінювання їхньої здатності формувати прогнози на нових даних.

Результатом виконання даного етапу є підготовлений набір даних, придатний для навчання, тестування та порівняння ефективності різних моделей машинного навчання, що використовуються для прогнозування рівня забруднення повітря.

3.2 Навчання моделей машинного навчання та оцінювання точності прогнозування

Після підготовки набору даних було виконано навчання та порівняльний аналіз декількох моделей машинного навчання для прогнозування рівня забруднення повітря. Дослідження проводилося з метою визначення моделі, яка забезпечує найвищу точність прогнозування на тестовій вибірці. Для оцінювання якості прогнозів використовувалися метрики MAE (Mean Absolute Error), RMSE (Root Mean Squared Error) та коефіцієнт детермінації R^2 .

У дослідженні були використані моделі Linear Regression, Decision Tree Regressor, Random Forest Regressor, Gradient Boosting Regressor, XGBoost Regressor та LSTM. Навчання виконувалося на 80% підготовлених даних, тоді як для тестування використовувалося 20% записів.

Результати оцінювання моделей наведено в таблиці 3.1.

Таблиця 3.1 – Результати оцінювання моделей прогнозування

Модель	MAE	RMSE	R^2
Linear Regression	9.84	12.31	0.781
Decision Tree Regressor	8.12	10.45	0.842
Random Forest Regressor	6.58	8.47	0.903
Gradient Boosting Regressor	6.11	7.94	0.918
XGBoost Regressor	5.72	7.31	0.931
LSTM	4.96	6.48	0.956

Отримані результати показують, що всі ансамблеві моделі перевершують базову лінійну регресію за всіма показниками якості. Найнижчі значення помилок

MAE та RMSE отримано для моделі LSTM, що свідчить про більш точне прогнозування майбутніх значень концентрацій забруднювальних речовин. Крім того, модель LSTM продемонструвала найбільше значення коефіцієнта детермінації $R^2 = 0.956$, що означає найкращу здатність пояснювати зміну цільових показників.

Для наочного порівняння точності обраних моделей результати наведено на рисунку 3.1.

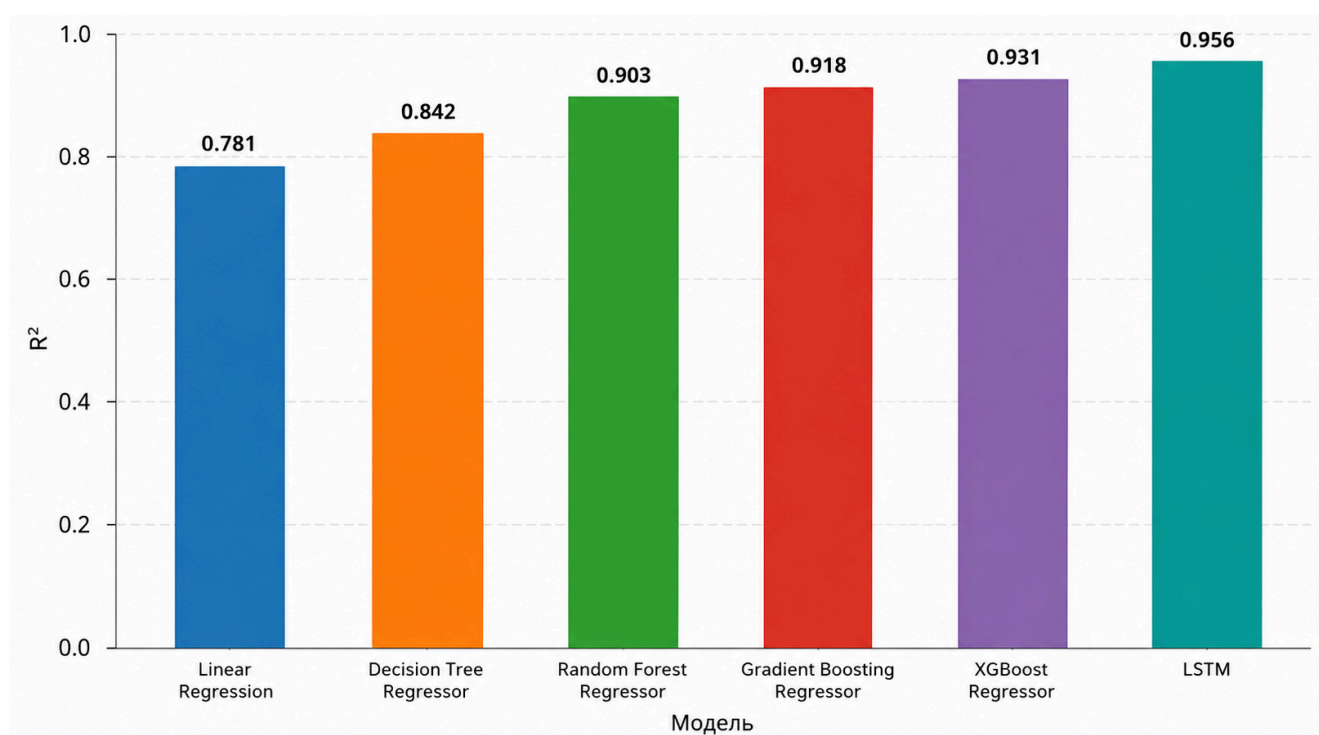


Рисунок 3.1 – Порівняння точності моделей машинного навчання за метрикою R^2

Додатково було проведено оцінювання часу навчання моделей, оскільки швидкість побудови прогнозів є важливим фактором для практичного використання системи. Отримані результати наведено в таблиці 3.2.

Незважаючи на найбільший час навчання, модель LSTM забезпечила найвищу якість прогнозування. Це пояснюється здатністю рекурентних нейронних мереж враховувати часові залежності між послідовними вимірюваннями якості повітря та метеорологічними параметрами.

Таблиця 3.2 – Час навчання моделей

Модель	Час навчання, с
Linear Regression	0.4
Decision Tree Regressor	1.1
Random Forest Regressor	8.6
Gradient Boosting Regressor	12.4
XGBoost Regressor	16.8
LSTM	58.7

Таким чином, за результатами проведеного дослідження найкращі показники продемонструвала модель LSTM, яка забезпечила найменші значення похибок прогнозування та найбільшу точність серед усіх досліджуваних підходів. Саме ця модель була обрана для використання у програмній системі моніторингу та прогнозування рівня забруднення повітря.

3.3 Тестування функціональності програмної системи та аналіз результатів експериментальних досліджень

Після завершення розробки програмної системи було проведено тестування її основних функціональних можливостей та оцінювання результатів прогнозування. Метою тестування була перевірка коректності роботи окремих модулів системи, а також оцінювання якості прогнозів, сформованих моделлю машинного навчання.

Під час тестування перевірялися функції отримання даних із зовнішніх API, збереження інформації у базі даних PostgreSQL, формування прогнозів та відображення результатів у вебінтерфейсі. Результати функціонального тестування наведено в таблиці 3.3.

Для оцінювання роботи прогновної моделі було використано тестову вибірку, що становила 20% від загального набору даних. Результати прогнозування порівнювалися з фактичними значеннями показників забруднення повітря (табл. 3.4).

Таблиця 3.3 – Результати функціонального тестування

Функція системи	Очікуваний результат	Результат
Отримання даних через API	Дані успішно отримано	Успішно
Збереження даних у БД	Дані записано до PostgreSQL	Успішно
Завантаження історичних даних	Дані відображено користувачу	Успішно
Формування прогнозу	Отримано прогнозні значення	Успішно
Відображення графіків	Графіки відображаються коректно	Успішно
Відображення таблиці прогнозу	Дані відображаються без помилок	Успішно

Таблиця 3.4 – Приклад результатів прогнозування PM2.5

Час прогнозу	Фактичне значення	Прогнозоване значення	Абсолютна похибка
12:00	21.4	20.8	0.6
15:00	24.1	23.5	0.6
18:00	26.8	25.9	0.9
21:00	22.7	23.3	0.6
00:00	19.5	18.8	0.7

Середня абсолютна похибка прогнозування PM2.5 для моделі LSTM склала 4.96, а середньоквадратична похибка — 6.48. Значення коефіцієнта детермінації становило 0.956, що свідчить про високу якість прогнозування та добру узгодженість прогнозованих і фактичних значень.

Для оцінювання ефективності програмної системи також було виміряно час виконання основних операцій. Результати наведено в таблиці 3.5.

Таблиця 3.5 – Час виконання основних операцій

Операція	Середній час виконання
Отримання даних з API	0.84 с
Збереження даних у БД	0.12 с
Завантаження історичних даних	0.18 с
Формування прогнозу LSTM	1.34 с
Відображення результатів	0.21 с

PM2.5 (Particulate Matter 2.5) — це дрібнодисперсні тверді частинки або краплі рідини в повітрі, діаметр яких не перевищує 2,5 мікрметра (мкм). Для

наочного аналізу відповідності прогнозованих та фактичних значень результати наведено на рисунку 3.2.

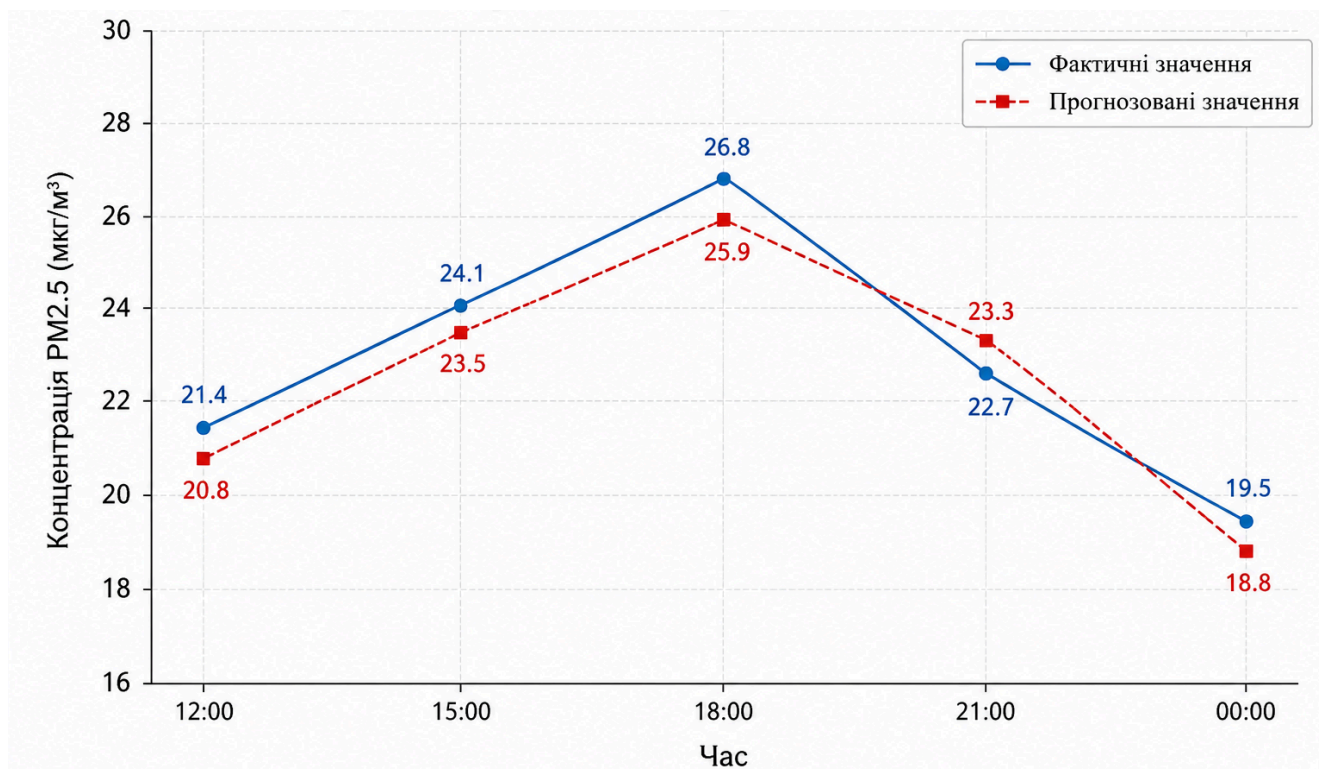


Рисунок 3.2 – Порівняння фактичних та прогнозованих значень PM2.5

Отримані результати показали, що розроблена програмна система коректно виконує всі передбачені функції та забезпечує формування прогнозів із високою точністю. Найкращі результати продемонструвала модель LSTM, яка забезпечила значення $R^2 = 0.956$, MAE = 4.96 та RMSE = 6.48. Це підтверджує доцільність використання рекурентних нейронних мереж для задач прогнозування рівня забруднення атмосферного повітря.

3.4 Рекомендації щодо впровадження та перспективи розвитку системи

Розроблену програмну систему доцільно впроваджувати в організаціях, діяльність яких пов'язана з екологічним моніторингом, аналізом стану атмосферного повітря та оцінюванням екологічних ризиків. До таких організацій

належать державні екологічні служби, науково-дослідні установи, органи місцевого самоврядування, а також підприємства, діяльність яких може впливати на стан навколишнього середовища. Використання системи дозволить автоматизувати процес збору та обробки даних із відкритих джерел, скоротити час на проведення аналізу та підвищити оперативність отримання прогнозної інформації.

Система може застосовуватися для прогнозування рівня забруднення повітря на основі історичних даних та поточних метеорологічних показників, що сприятиме своєчасному виявленню потенційно небезпечних екологічних ситуацій. Отримані результати можуть використовуватися для підтримки прийняття управлінських рішень, планування природоохоронних заходів та інформування населення про можливі ризики для здоров'я.

Перспективами подальшого розвитку системи є розширення переліку джерел даних шляхом інтеграції з додатковими сервісами моніторингу якості повітря та метеорологічними платформами. Доцільним також є використання більшої кількості параметрів, зокрема швидкості та напрямку вітру, атмосферного тиску, вологості повітря та інших факторів, які можуть впливати на точність прогнозування.

Окремим напрямом розвитку є впровадження механізму автоматичного сповіщення користувачів про критичне погіршення якості повітря за допомогою електронної пошти, мобільних повідомлень або вебсповіщень. Крім того, перспективною є інтеграція системи з мобільними застосунками та геоінформаційними сервісами для візуалізації прогнозів на карті в режимі реального часу.

Також перспективним напрямом є використання більш складних моделей машинного та глибокого навчання, зокрема рекурентних нейронних мереж і сучасних архітектур для аналізу часових рядів. Збільшення обсягів навчальних даних, оптимізація процесу навчання моделей та регулярне оновлення наборів даних дозволять підвищити точність прогнозування та забезпечити більш надійні результати роботи системи.

3.5 Висновки до третього розділу

У третьому розділі було підготовлено набір даних для навчання та тестування моделей машинного навчання, виконано їх попередню обробку та формування ознак для прогнозування рівня забруднення повітря. Проведено навчання та порівняльний аналіз моделей Linear Regression, Decision Tree Regressor, Random Forest Regressor, Gradient Boosting Regressor, XGBoost Regressor та LSTM.

За результатами експериментальних досліджень встановлено, що найкращі показники продемонструвала модель LSTM, яка забезпечила значення MAE = 4.96, RMSE = 6.48 та $R^2 = 0.956$. Також було проведено тестування функціональності програмної системи, яке підтвердило коректність роботи модулів збору даних, прогнозування, зберігання інформації та вебінтерфейсу користувача.

Отримані результати свідчать про ефективність розробленої програмної системи та доцільність використання методів машинного навчання для прогнозування рівня забруднення атмосферного повітря. Запропоноване рішення забезпечує високу точність прогнозів та може бути використане для підтримки екологічного моніторингу й прийняття управлінських рішень.

4 ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ

У даному розділі розглядаються основні вимоги охорони праці та безпеки життєдіяльності, яких необхідно дотримуватися під час виконання робіт з використанням комп'ютерної техніки. Особливу увагу приділено організації робочого місця та забезпеченню безпечних умов праці відповідно до чинних нормативних документів.

4.1 Моделювання та прогнозування небезпечних ситуацій

Одним із важливих напрямів забезпечення безпеки життєдіяльності є своєчасне виявлення, оцінювання та прогнозування небезпечних ситуацій, які можуть виникати внаслідок негативного впливу техногенних, природних або соціальних факторів. Відповідно до положень навчального посібника з безпеки життєдіяльності, прогнозування небезпек дозволяє заздалегідь визначати можливі ризики, оцінювати масштаби їх наслідків та розробляти ефективні заходи щодо їх попередження або мінімізації [21].

Під моделюванням небезпечних ситуацій розуміють процес побудови математичних, логічних або інформаційних моделей, які відображають можливий розвиток подій у разі виникнення певної небезпеки. Метою такого моделювання є отримання інформації про ймовірність виникнення небезпечної події, прогнозування її наслідків та визначення найбільш ефективних заходів захисту людини і навколишнього середовища. Використання моделей дозволяє здійснювати аналіз різних сценаріїв розвитку подій без необхідності проведення реальних експериментів, що особливо важливо у сфері безпеки життєдіяльності.

Згідно з методичними рекомендаціями ТНТУ, під час виконання кваліфікаційних робіт необхідно враховувати можливість виникнення небезпечних ситуацій та оцінювати фактори ризику, які можуть впливати на безпеку людини під час роботи з інформаційними системами та програмним забезпеченням [22].

Для програмної системи моніторингу та прогнозування рівня забруднення повітря основними потенційними небезпечними ситуаціями є перевищення допустимих концентрацій шкідливих речовин у повітрі, виникнення смогу, різке погіршення екологічної ситуації внаслідок промислових викидів або несприятливих метеорологічних умов.

Для оцінювання рівня безпеки використовується ризик-орієнтований підхід, відповідно до якого рівень ризику визначається як функція ймовірності виникнення небезпечної події та тяжкості її наслідків. У процесі аналізу можуть використовуватися статистичні дані попередніх спостережень, результати екологічного моніторингу та прогностичні моделі. На основі отриманих результатів здійснюється ранжування небезпек за ступенем ризику та визначаються пріоритетні напрями реагування.

У межах досліджуваної програмної системи моделювання небезпечних ситуацій реалізується шляхом аналізу історичних даних про концентрацію забруднювальних речовин та прогнозування їх майбутніх значень. Використання методів машинного навчання дозволяє визначати тенденції зміни показників якості повітря та завчасно виявляти можливість перевищення гранично допустимих концентрацій. Таким чином забезпечується інформаційна підтримка процесу прийняття рішень щодо запобігання негативному впливу забруднення атмосферного повітря на населення.

Для оцінювання потенційних небезпечних ситуацій у системах екологічного моніторингу доцільно враховувати такі основні показники:

- імовірність виникнення небезпечної події;
- масштаб можливих наслідків для населення;
- тривалість впливу небезпечного фактора;
- кількість людей, які можуть потрапити в зону впливу;
- рівень перевищення гранично допустимих концентрацій забруднювальних речовин;
- можливість своєчасного виявлення та локалізації безпеки;
- ефективність засобів попередження та захисту.

Залежно від отриманих результатів оцінювання небезпечні ситуації можуть бути класифіковані за рівнем ризику, що наведено в таблиці 4.1.

Таблиця 4.1 – Класифікація небезпечних ситуацій за рівнем ризику

Рівень ризику	Характеристика ситуації
Низький	Відхилення показників не перевищують допустимих норм
Середній	Існує ймовірність негативного впливу на окремі групи населення
Високий	Спостерігається перевищення нормативних показників та ризик погіршення стану здоров'я
Критичний	Значне перевищення допустимих концентрацій, необхідне термінове реагування

Використання наведеної класифікації дозволяє оперативно визначати ступінь небезпеки та приймати відповідні рішення щодо інформування населення і впровадження заходів захисту.

Результати моделювання можуть бути використані для формування попереджень про можливе виникнення небезпечних екологічних ситуацій, інформування населення про підвищений рівень забруднення та планування заходів щодо зниження впливу шкідливих факторів на здоров'я людини. Такий підхід відповідає основним принципам безпеки життєдіяльності, які передбачають пріоритет профілактичних заходів над ліквідацією наслідків небезпечних подій.

Отже, моделювання та прогнозування небезпечних ситуацій є важливим елементом забезпечення безпеки життєдіяльності, оскільки дозволяє своєчасно виявляти потенційні загрози, оцінювати рівень ризику та приймати обґрунтовані рішення щодо захисту людини і навколишнього середовища від негативного впливу небезпечних факторів.

4.2 Вимоги ергономіки до організації робочого місця оператора ПК

Під час виконання робіт, пов'язаних із використанням персонального комп'ютера, одним із ключових завдань охорони праці є створення безпечних та комфортних умов праці, які забезпечують збереження здоров'я працівника, попередження професійних захворювань та підтримання високої працездатності

протягом робочого дня. Відповідно до Закону України «Про охорону праці» роботодавець зобов'язаний забезпечити функціонування системи управління охороною праці та створити на робочому місці умови праці відповідно до вимог нормативно-правових актів [23, 24].

Робота оператора ПК характеризується значним навантаженням на органи зору, опорно-руховий апарат та нервову систему. Тривале перебування в статичному положенні, постійна концентрація уваги та необхідність обробки значних обсягів інформації можуть призводити до швидкої втомлюваності, погіршення зору, виникнення захворювань хребта та розвитку професійного стресу [21, 26].

Згідно з вимогами ДСанПіН 3.3.2-007-98 робоче місце оператора ПК повинно бути організоване таким чином, щоб забезпечувати природне положення тіла працівника та мінімізувати фізичне навантаження [25]. Монітор необхідно розташовувати на відстані 600–700 мм від очей користувача, при цьому верхня межа екрана повинна знаходитися на рівні очей або на 10–20° нижче горизонтальної лінії погляду. Таке розташування знижує навантаження на шийний відділ хребта та запобігає швидкому зоровому стомленню.

Конструкція робочого столу повинна забезпечувати достатню площу для розміщення монітора, клавіатури, маніпулятора типу «миша» та робочої документації. Рекомендована висота робочої поверхні становить 680–800 мм. Простір для ніг працівника повинен забезпечувати вільне розміщення нижніх кінцівок без обмеження рухів під час роботи [25, 26].

Особливу увагу необхідно приділяти параметрам робочого крісла. Крісло повинно бути стійким, мати регулювання висоти сидіння та кута нахилу спинки. Спинка повинна підтримувати поперековий відділ хребта, а висота сидіння має забезпечувати положення, за якого ступні працівника повністю спираються на підлогу або спеціальну підставку. Оптимальним вважається положення, коли кут між стегном і гомілкою становить близько 90°.

Основні ергономічні вимоги до робочого місця оператора ПК наведено в таблиці 4.2 [25, 26].

Таблиця 4.2 – Ергономічні параметри робочого місця оператора ПК

Параметр	Рекомендоване значення
Відстань від очей до монітора	600–700 мм
Мінімальна відстань до екрана	500 мм
Висота робочої поверхні столу	680–800 мм
Висота сидіння крісла	400–550 мм
Кут нахилу погляду до екрана	10–20° вниз
Освітленість робочої поверхні	300–500 лк
Відносна вологість повітря	40–60 %
Температура повітря	22–24 °С

Важливим фактором охорони праці є забезпечення нормативних параметрів виробничого середовища. Для приміщень з персональними комп'ютерами рекомендована температура повітря становить 22–24 °С, відносна вологість – 40–60 %, а швидкість руху повітря не повинна перевищувати 0,1–0,2 м/с. Недотримання цих вимог може призводити до погіршення самопочуття працівників, зниження працездатності та підвищення ризику професійних захворювань [25]. З метою зниження зорового навантаження освітленість робочої поверхні повинна становити 300–500 лк. При цьому необхідно уникати появи відблисків на екрані монітора, а природне та штучне освітлення повинні забезпечувати рівномірний розподіл світлового потоку в робочій зоні [25, 26].

Відповідно до вимог охорони праці важливим є також дотримання режиму праці та відпочинку. Під час тривалої роботи за персональним комп'ютером рекомендується виконувати регламентовані перерви для відпочинку очей та зміни положення тіла. Такі заходи дозволяють знизити рівень втоми, покращити умови праці та попередити розвиток професійних захворювань [25].

Отже, дотримання ергономічних вимог під час організації робочого місця оператора ПК є необхідною умовою забезпечення безпечних умов праці. Виконання вимог чинного законодавства України у сфері охорони праці та санітарних норм сприяє збереженню здоров'я працівників, підвищенню продуктивності праці та зменшенню впливу шкідливих виробничих факторів.

ВИСНОВКИ

У кваліфікаційній роботі розв'язано актуальне завдання розробки програмної системи моніторингу та прогнозування рівня забруднення повітря з використанням методів машинного навчання. Актуальність роботи обумовлена необхідністю оперативного аналізу екологічних даних та прогнозування змін якості атмосферного повітря для своєчасного реагування на потенційні екологічні ризики.

У процесі виконання роботи було проведено аналіз предметної області, досліджено проблему забруднення атмосферного повітря, розглянуто сучасні системи моніторингу якості повітря та проаналізовано методи прогнозування екологічних показників. Також було досліджено особливості використання методів машинного навчання в задачах екологічного моніторингу та виконано аналіз джерел даних, які можуть використовуватися для навчання прогнозних моделей.

На основі проведеного аналізу сформовано функціональні та нефункціональні вимоги до програмної системи, визначено основних користувачів та сценарії їх взаємодії із системою. Для реалізації поставлених задач було спроектовано архітектуру програмного забезпечення, розроблено структуру бази даних на основі PostgreSQL та реалізовано механізм взаємодії між окремими компонентами системи за допомогою REST API.

У роботі реалізовано модуль збору та обробки даних про якість повітря, який забезпечує автоматичне отримання екологічної інформації із зовнішніх джерел, її очищення, нормалізацію та збереження у базі даних. Також розроблено модуль прогнозування, що використовує методи машинного навчання для аналізу історичних даних та формування прогнозів рівня забруднення повітря.

Для оцінювання ефективності прогнозування було проведено навчання та тестування моделей Linear Regression, Decision Tree Regressor, Random Forest Regressor, Gradient Boosting Regressor, XGBoost Regressor та LSTM. Порівняльний аналіз результатів показав, що найкращу якість прогнозування забезпечила модель

LSTM, яка продемонструвала значення $MAE = 4.96$, $RMSE = 6.48$ та коефіцієнт детермінації $R^2 = 0.956$. Отримані результати підтвердили доцільність використання рекурентних нейронних мереж для прогнозування часових рядів екологічних показників.

Проведене функціональне тестування підтвердило коректність роботи всіх основних компонентів системи, включаючи модулі збору даних, прогнозування, зберігання інформації та вебінтерфейс користувача. Середній час формування прогнозу склав 1.34 секунди, що свідчить про можливість практичного використання системи для оперативного аналізу екологічної інформації.

Практичним результатом роботи є створена програмна система, яка забезпечує автоматизований моніторинг показників якості повітря, зберігання історичних даних, прогнозування рівня забруднення та надання результатів користувачу через вебінтерфейс. Розроблене рішення може бути використане в екологічних службах, органах місцевого самоврядування, науково-дослідних установах та інших організаціях, діяльність яких пов'язана з аналізом стану навколишнього середовища.

Подальший розвиток системи може бути спрямований на розширення переліку джерел даних, інтеграцію з мобільними застосунками, реалізацію автоматичних механізмів сповіщення користувачів та використання більш складних моделей глибокого навчання для підвищення точності прогнозування.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Михалик Д. М., Цуприк Г. Б., Бревус В. М. Методичні вказівки до виконання кваліфікаційної роботи бакалавра для здобувачів першого (бакалаврського) рівня вищої освіти за освітньо-професійною програмою «Інженерія програмного забезпечення» спеціальності 121 – «Інженерія програмного забезпечення» всіх форм навчання. Тернопіль : ТНТУ ім. І. Пулюя, 2024. 45 с.
2. Géron A. Hands-On Machine Learning with Scikit-Learn, Keras and TensorFlow. 3-тє вид. Sebastopol : O'Reilly Media, 2022. 851 p.
3. Chollet F. Deep Learning with Python. 2-ге вид. Shelter Island : Manning Publications, 2021. 504 p.
4. VanderPlas J. Python Data Science Handbook. Sebastopol : O'Reilly Media, 2017. 548 p.
5. Grus J. Data Science from Scratch: First Principles with Python. 2-ге вид. Sebastopol : O'Reilly Media, 2019. 406 p.
6. Hochreiter S., Schmidhuber J. Long Short-Term Memory // Neural Computation. 1997. Т. 9, № 8. С. 1735–1780. DOI: 10.1162/neco.1997.9.8.1735.
7. LeCun Y., Bengio Y., Hinton G. Deep Learning // Nature. 2015. Т. 521. С. 436–444. DOI: 10.1038/nature14539.
8. Chen T., Guestrin C. XGBoost: A Scalable Tree Boosting System // Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. 2016. С. 785–794. DOI: 10.1145/2939672.2939785.
9. Breiman L. Random Forests // Machine Learning. 2001. Т. 45, № 1. С. 5–32. DOI: 10.1023/A:1010933404324.
10. Friedman J. H. Greedy Function Approximation: A Gradient Boosting Machine // Annals of Statistics. 2001. Т. 29, № 5. С. 1189–1232. DOI: 10.1214/aos/1013203451.

11. Li X., Peng L., Hu Y., Shao J., Chi T. Deep Learning Architecture for Air Quality Predictions // *Environmental Science and Pollution Research*. 2016. Т. 23, № 22. С. 22408–22417. DOI: 10.1007/s11356-016-7626-1.

12. Zhang Y., Bocquet M., Mallet V., Seigneur C., Baklanov A. Real-Time Air Quality Forecasting, Part I: History, Techniques, and Current Status // *Atmospheric Environment*. 2012. Т. 60. С. 632–655. DOI: 10.1016/j.atmosenv.2012.06.031.

13. OpenAQ. Open Air Quality Data Platform. URL: <https://openaq.org> (дата звернення: 01.06.2026).

14. PostgreSQL Global Development Group. PostgreSQL Documentation. URL: <https://www.postgresql.org/docs> (дата звернення: 01.06.2026).

15. Flask Documentation. URL: <https://flask.palletsprojects.com> (дата звернення: 01.06.2026).

16. Pedregosa F. та ін. Scikit-learn: Machine Learning in Python // *Journal of Machine Learning Research*. 2011. Т. 12. С. 2825–2830.

17. Hunter J. D. Matplotlib: A 2D Graphics Environment // *Computing in Science & Engineering*. 2007. Т. 9, № 3. С. 90–95. DOI: 10.1109/MCSE.2007.55.

18. McKinney W. Data Structures for Statistical Computing in Python // *Proceedings of the 9th Python in Science Conference*. 2010. С. 56–61. DOI: 10.25080/Majora-92bf1922-00a.

19. Virtanen P. та ін. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python // *Nature Methods*. 2020. Т. 17. С. 261–272. DOI: 10.1038/s41592-019-0686-2.

20. World Health Organization. Air Pollution. URL: <https://www.who.int/health-topics/air-pollution> (дата звернення: 01.06.2026).

21. Желібо Є. П., Заверуха Н. М. Безпека життєдіяльності : навч. посіб. Київ : Каравела, 2020. 344 с.

22. Методичні вказівки для написання розділу «Безпека життєдіяльності, основи охорони праці» в кваліфікаційних роботах здобувачів освітнього рівня „бакалавр”. Для студентів всіх форм навчання рівень вищої освіти перший (

бакалаврський) / укл. : О. Я. Гурик , І. Б. Окіпний. – Тернопіль : ТНТУ імені Івана Пулюя, 2021. - 20 с

23. Про охорону праці : Закон України від 14.10.1992 № 2694-XII. URL: <https://zakon.rada.gov.ua/laws/show/2694-12> (дата звернення: 10.06.2026).

24. Кодекс законів про працю України : Закон України від 10.12.1971 № 322-VIII. URL: <https://zakon.rada.gov.ua/laws/show/322-08> (дата звернення: 10.06.2026).

25. ДСанПіН 3.3.2-007-98. Державні санітарні правила і норми роботи з візуальними дисплейними терміналами електронно-обчислювальних машин. Київ : МОЗ України, 1998. 24 с.

26. Основи охорони праці : підручник / за ред. К. Н. Ткачука. Київ : Основа, 2018. 448 с.

ДОДАТКИ

ДОДАТОК А

Основні лістинги програмної системи

Лістинг А.1 – Файл моделі бази даних (database.py)

```
from flask_sqlalchemy import SQLAlchemy
from datetime import datetime

db = SQLAlchemy()

class City(db.Model):
    __tablename__ = 'cities'

    id = db.Column(db.Integer, primary_key=True)
    name = db.Column(db.String(100), nullable=False)
    country = db.Column(db.String(100), nullable=False)

    air_quality_data = db.relationship(
        'AirQualityData',
        backref='city',
        lazy=True
    )

    predictions = db.relationship(
        'Prediction',
        backref='city',
        lazy=True
    )

class AirQualityData(db.Model):
    __tablename__ = 'air_quality_data'

    id = db.Column(db.Integer, primary_key=True)

    city_id = db.Column(
        db.Integer,
        db.ForeignKey('cities.id'),
        nullable=False
    )

    pm25 = db.Column(db.Float, nullable=False)
    pm10 = db.Column(db.Float, nullable=False)
    no2 = db.Column(db.Float)
    so2 = db.Column(db.Float)
    co = db.Column(db.Float)
    o3 = db.Column(db.Float)

    temperature = db.Column(db.Float)
    humidity = db.Column(db.Float)
```

```

pressure = db.Column(db.Float)
wind_speed = db.Column(db.Float)

measurement_time = db.Column(
    db.DateTime,
    default=datetime.utcnow
)

class Prediction(db.Model):
    __tablename__ = 'predictions'

    id = db.Column(db.Integer, primary_key=True)

    city_id = db.Column(
        db.Integer,
        db.ForeignKey('cities.id'),
        nullable=False
    )

    predicted_pm25 = db.Column(db.Float, nullable=False)
    predicted_pm10 = db.Column(db.Float, nullable=False)
    predicted_aqi = db.Column(db.Float, nullable=False)

    forecast_time = db.Column(
        db.DateTime,
        nullable=False
    )

    created_at = db.Column(
        db.DateTime,
        default=datetime.utcnow
    )

def init_database(app):
    db.init_app(app)

    with app.app_context():
        db.create_all()

```

Лістинг А.2 – Файл модуля збору даних із зовнішніх джерел (data_collector.py)

```

import requests
from datetime import datetime
from database import db, City, AirQualityData

class AirQualityCollector:
    def __init__(self, api_url):
        self.api_url = api_url

```

```

def get_air_quality_data(self, city_name):
    params = {
        "city": city_name,
        "limit": 1
    }

    response = requests.get(
        self.api_url,
        params=params,
        timeout=10
    )

    response.raise_for_status()
    return response.json()

def parse_response(self, data):
    results = data.get("results", [])

    if not results:
        return None

    item = results[0]

    measurements = {
        "pm25": None,
        "pm10": None,
        "no2": None,
        "so2": None,
        "co": None,
        "o3": None
    }

    for parameter in item.get("measurements", []):
        name = parameter.get("parameter")
        value = parameter.get("value")

        if name in measurements:
            measurements[name] = value

    return measurements

def save_air_quality_data(self, city_name, measurements):
    city = City.query.filter_by(name=city_name).first()

    if city is None:
        city = City(name=city_name, country="Ukraine")
        db.session.add(city)
        db.session.commit()

    record = AirQualityData(
        city_id=city.id,
        pm25=measurements.get("pm25") or 0,

```

```

        pm10=measurements.get("pm10") or 0,
        no2=measurements.get("no2"),
        so2=measurements.get("so2"),
        co=measurements.get("co"),
        o3=measurements.get("o3"),
        measurement_time=datetime.utcnow()
    )

    db.session.add(record)
    db.session.commit()

    return record

def collect(self, city_name):
    raw_data = self.get_air_quality_data(city_name)
    measurements = self.parse_response(raw_data)

    if measurements is None:
        return None

    return self.save_air_quality_data(city_name, measurements)

```

Лістинг А.3 – Файл попередньої обробки даних (data_preprocessing.py)

```

import pandas as pd
from sklearn.preprocessing import MinMaxScaler

class DataPreprocessor:

    def __init__(self):
        self.scaler = MinMaxScaler()

    def load_data(self, filepath):
        return pd.read_csv(filepath)

    def remove_duplicates(self, dataframe):
        return dataframe.drop_duplicates()

    def handle_missing_values(self, dataframe):
        numeric_columns = dataframe.select_dtypes(
            include=["float64", "int64"]
        ).columns

        dataframe[numeric_columns] = dataframe[
            numeric_columns
        ].fillna(
            dataframe[numeric_columns].mean()
        )

    return dataframe

```

```

def create_features(self, dataframe):
    dataframe["pm25_lag1"] = dataframe["pm25"].shift(1)
    dataframe["pm25_lag2"] = dataframe["pm25"].shift(2)

    dataframe["pm10_lag1"] = dataframe["pm10"].shift(1)
    dataframe["pm10_lag2"] = dataframe["pm10"].shift(2)

    dataframe = dataframe.dropna()

    return dataframe

def normalize_features(self, dataframe, columns):
    dataframe[columns] = self.scaler.fit_transform(
        dataframe[columns]
    )

    return dataframe

def prepare_dataset(self, filepath):
    data = self.load_data(filepath)

    data = self.remove_duplicates(data)
    data = self.handle_missing_values(data)
    data = self.create_features(data)

    feature_columns = [
        "pm25",
        "pm10",
        "temperature",
        "humidity",
        "pressure",
        "wind_speed",
        "pm25_lag1",
        "pm25_lag2",
        "pm10_lag1",
        "pm10_lag2"
    ]

    data = self.normalize_features(
        data,
        feature_columns
    )

    return data

```

Лістинг А.4 – Файл навчання та прогнозування моделі машинного навчання (lstm_model.py)

```

import numpy as np
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import LSTM, Dense, Dropout
from tensorflow.keras.models import load_model

```

```
class AirQualityLSTM:

    def __init__(self):
        self.model = None

    def build_model(self, input_shape):
        model = Sequential()

        model.add(
            LSTM(
                64,
                return_sequences=True,
                input_shape=input_shape
            )
        )

        model.add(Dropout(0.2))

        model.add(
            LSTM(
                32,
                return_sequences=False
            )
        )

        model.add(Dropout(0.2))

        model.add(Dense(16, activation="relu"))
        model.add(Dense(1))

        model.compile(
            optimizer="adam",
            loss="mse",
            metrics=["mae"]
        )

        self.model = model

        return model

    def train(
        self,
        X_train,
        y_train,
        epochs=50,
        batch_size=32
    ):
        history = self.model.fit(
            X_train,
            y_train,
            epochs=epochs,
```

```

        batch_size=batch_size,
        validation_split=0.2,
        verbose=1
    )

    return history

def predict(self, X_test):
    return self.model.predict(X_test)

def save(self, filepath):
    self.model.save(filepath)

def load(self, filepath):
    self.model = load_model(filepath)

def forecast_next_value(self, input_sequence):
    input_sequence = np.array(
        input_sequence
    ).reshape(1, 24, 10)

    prediction = self.model.predict(
        input_sequence
    )

    return float(prediction[0][0])

```

Лістинг А.5 – Файл REST API та обробки запитів користувача (app.py)

```

from flask import Flask, jsonify, request
from database import db
from models import AirQualityData, Prediction
from data_collector import AirQualityCollector
from lstm_model import AirQualityLSTM

app = Flask(__name__)

app.config[
    "SQLALCHEMY_DATABASE_URI"
] = "postgresql://user:password@localhost/air_quality"

db.init_app(app)

collector = AirQualityCollector(
    api_url="https://api.openaq.org/v2/latest"
)

model = AirQualityLSTM()

@app.route("/api/air-quality/current", methods=["GET"])

```

```

def get_current_air_quality():

    latest_record = (
        AirQualityData.query
        .order_by(
            AirQualityData.measurement_time.desc()
        )
        .first()
    )

    if not latest_record:
        return jsonify(
            {"message": "No data available"}
        ), 404

    return jsonify({
        "pm25": latest_record.pm25,
        "pm10": latest_record.pm10,
        "temperature": latest_record.temperature,
        "humidity": latest_record.humidity
    })

@app.route("/api/air-quality/update", methods=["POST"])
def update_air_quality():

    city = request.json.get("city")

    collector.collect(city)

    return jsonify({
        "message": "Data successfully updated"
    })

@app.route("/api/predictions/generate", methods=["POST"])
def generate_prediction():

    input_data = request.json.get("data")

    prediction = model.forecast_next_value(
        input_data
    )

    return jsonify({
        "predicted_pm25": prediction
    })

@app.route("/api/predictions/latest", methods=["GET"])
def get_latest_prediction():

    prediction = (

```

```
Prediction.query
.order_by(
    Prediction.created_at.desc()
)
.first()
)

if not prediction:
    return jsonify(
        {"message": "No predictions available"}
    ), 404

return jsonify({
    "predicted_pm25":
        prediction.predicted_pm25,
    "predicted_pm10":
        prediction.predicted_pm10,
    "predicted_aqi":
        prediction.predicted_aqi
})

if __name__ == "__main__":
    app.run(debug=True)
```

ДОДАТОК Б
Теза конференції

УДК 621.326

Матичак А. – ст. гр. СПс-41

Тернопільський національний технічний університет імені Івана Пулюя

**РОЗРОБКА ПРОГРАМНОЇ СИСТЕМИ МОНІТОРИНГУ ТА
ПРОГНОЗУВАННЯ РІВНЯ ЗАБРУДНЕННЯ ПОВІТРЯ З
ВИКОРИСТАННЯМ МЕТОДІВ МАШИННОГО НАВЧАННЯ**

Науковий керівник: к.ф.-м.н., доцент Цебрій О. Р.

Matychak A.

Ternopil Ivan Puluj National Technical University

**DEVELOPMENT OF A SOFTWARE SYSTEM FOR MONITORING
AND PREDICTING AIR POLLUTION LEVELS USING MACHINE
LEARNING METHODS**

Supervisor: PhD in Physics and Mathematics, Associate Professor Tsebrii O.

Ключові слова: прогнозування, забруднення повітря, мультимодальні моделі
Keywords: prediction, air pollution, multimodal models

Прогнозування рівня забруднення повітря є важливою задачею для забезпечення екологічної безпеки та покращення якості життя населення. Забруднення атмосфери має складну природу та залежить від великої кількості факторів, серед яких метеорологічні умови, транспортне навантаження, промислові викиди та географічні особливості. Традиційні статистичні методи часто не здатні врахувати складні нелінійні залежності між цими факторами, що обмежує їх точність. У цьому контексті методи машинного навчання дозволяють будувати більш точні моделі, здатні враховувати багатовимірність та динамічність даних.

Сучасні підходи до прогнозування якості повітря базуються на використанні різних типів даних, включаючи часові ряди показників забруднення (PM_{2.5}, PM₁₀, NO₂, CO), метеорологічні дані, а також зображення або супутникові дані. Мультимодальні моделі дозволяють інтегрувати ці різноманітні джерела інформації в єдину систему, що значно підвищує якість прогнозування. Зокрема, комбінація рекурентних нейронних мереж для аналізу часових рядів та згорткових мереж для обробки зображень дозволяє враховувати як часову, так і просторову структуру даних.

Важливим етапом є попередня обробка даних, яка включає очищення, нормалізацію, синхронізацію різних джерел та заповнення пропущених значень. Оскільки дані можуть надходити з різних сенсорів і мати різну частоту оновлення, необхідно забезпечити їх узгодження у часі та просторі. Це дозволяє сформувати єдине представлення вхідних даних для подальшого навчання моделей.

Для побудови прогнозних моделей широко використовуються алгоритми, такі як LSTM, GRU, градієнтний бустинг та глибокі нейронні мережі. Особливу ефективність демонструють гібридні архітектури, які поєднують різні типи моделей для обробки окремих модальностей даних. Наприклад, LSTM може використовуватись для аналізу часових залежностей, тоді як CNN — для обробки супутникових зображень або карт забруднення.

Окрім цього, важливим є врахування просторових залежностей між різними точками спостереження. Для цього застосовуються підходи, що враховують географічну структуру даних, зокрема графові нейронні мережі або просторові згорткові моделі. Це дозволяє враховувати вплив сусідніх регіонів та поширення забруднення у просторі.

Експериментальні результати показують, що використання мультимодальних підходів дозволяє підвищити точність прогнозування порівняно з моделями, які використовують лише один тип даних. Основними метриками оцінювання виступають MAE, RMSE та коефіцієнт детермінації R^2 , що дозволяє комплексно оцінити якість прогнозів [1]. Разом з тим, залишаються відкритими питання, пов'язані з якістю та доступністю даних, а також обчислювальною складністю моделей. Тому актуальними є задачі оптимізації моделей, зменшення їх розміру та забезпечення можливості роботи у реальному часі.

Додатково важливим є питання інтеграції різних джерел даних у єдину узгоджену модель. Мультимодальні підходи передбачають об'єднання структурованих числових даних із сенсорів, метеорологічних показників та неструктурованих даних, таких як супутникові зображення або карти. Для цього використовуються механізми злиття ознак на різних рівнях — ранньому, проміжному або пізньому, що дозволяє ефективно поєднувати інформацію з різних модальностей. Вибір стратегії злиття безпосередньо впливає на якість моделі та її здатність узагальнювати складні залежності між факторами забруднення.

Окрему увагу приділяють побудові адаптивних моделей, здатних враховувати змінність середовища у часі. Рівень забруднення повітря може суттєво змінюватися залежно від сезонних факторів, погодних умов та людської діяльності. Тому сучасні системи використовують механізми онлайн-навчання або періодичного донавчання моделей на нових даних. Це дозволяє підтримувати актуальність прогнозів і забезпечувати їх точність у довгостроковій перспективі без необхідності повного перенавчання моделі.

Також важливим напрямом є інтерпретованість моделей, що дозволяє зрозуміти внесок окремих факторів у формування прогнозу. Використання методів пояснюваного машинного навчання дає можливість визначити, які саме змінні мають найбільший вплив на рівень забруднення в конкретний момент часу. Це є критично важливим для прийняття управлінських рішень, оскільки дозволяє не лише прогнозувати стан повітря, але й ідентифікувати основні джерела забруднення та оцінювати ефективність екологічних заходів.

Таким чином, поєднання мультимодальних підходів, сучасних методів машинного навчання та ефективної обробки даних створює передумови для розробки точних і надійних систем прогнозування забруднення повітря. Такі системи можуть бути використані для підтримки екологічного моніторингу, прийняття управлінських рішень та інформування населення про стан довкілля.

Література:

1. Zhang, Z., Wang, J. (2022). Deep learning for air quality prediction: A survey. — *Environmental Modelling & Software*.
2. Li, X., Peng, L., Yao, X. (2023). Multimodal deep learning for air pollution forecasting. — *Atmospheric Environment*.
3. Gao, J., Sun, Y., Li, H. (2024). Explainable multimodal deep learning for air quality prediction. — *Environmental Science and Pollution Research*.