**Міністерство освіти та науки України**

**Національний університет “Львівська політехніка”**

**Інститут Просторового Планування та Перспективних Технологій**

Кафедра інформаційних систем і технологій

**КУРСОВА РОБОТА**

**з дисципліни**

**“ Проектування інформаційних управляючих систем**

**екологічного моніторингу”**

**на тему:**

**“ Прогнозування зміни рівня забруднення води в річці на основі екологічних даних.”**

Керівник курсової роботи

доцент, к.ф.-м.н. Баран М.М.

Розробив студент гр. КНЕМ-11

Висоцький О.М.

"\_\_\_"\_\_\_\_\_\_\_\_\_2025 р.

Львів – 2025

АНОТАЦІЯ

У курсовій роботі реалізовано гібридну інтелектуальну систему для прогнозування зміни рівня забруднення води в річці на основі екологічних даних. Система поєднує можливості машинного навчання та нечіткої логіки.

Розроблена система дозволяє прогнозувати рівень забруднення води на 6 місяців наперед, використовуючи чотири ключові екологічні параметри: температуру води, pH, вміст нітрогену та швидкість течії. Для прогнозування застосовано модель штучної нейронної мережі типу багатошарового перцептрона (MLP), реалізовану за допомогою TensorFlow / Keras.

Нечітка логіка використана для інтерпретації прогнозованого числового рівня забруднення та його перетворення у зрозумілий візуальний індекс, який відображається зміною кольору маркера на інтерактивній карті (зелений – низький, жовтий – середній, червоний – високий рівень забруднення).

Розроблено структуру бази даних на SQLite 3 для зберігання екологічних даних та прогнозів. Користувацький інтерфейс, реалізований за допомогою HTML5, CSS3, Bootstrap 5, JavaScript, Leaflet.js та Chart.js, забезпечує наочну візуалізацію динаміки прогнозів та інтерактивне введення даних.

Проведене тестування основних компонентів системи підтвердило її працездатність, точність прогнозування та відповідність функціональним вимогам. Розроблена система є ефективним інструментом для аналізу екологічної ситуації водних ресурсів та може бути використана для виявлення критичних зон із підвищеним ризиком забруднення, сприяючи прийняттю своєчасних управлінських рішень.

**ЗМІСТ**

[ВСТУП 4](#_Toc200124220)

[**1.1 Загальна характеристика предметної області** 5](#_Toc200124221)

[**1.2 Огляд та аналіз існуючих систем та методів екологічного моніторингу** 7](#_Toc200124222)

[**1.3 Постановка задачі** 8](#_Toc200124223)

[**Висновок до розділу 1** 9](#_Toc200124224)

[**РОЗДІЛ 2. АНАЛІТИЧНИЙ РОЗДІЛ. ХАРАКТЕРИСТИКА МОДЕЛЕЙ ПРОЄКТУВАННЯ ТА РЕАЛІЗАЦІЇ СИСТЕМИ** 11](#_Toc200124225)

[**2.1 Формування інформаційної моделі об’єкта дослідження** 11](#_Toc200124226)

[**2.2 Методи збору та організації даних у системі** 13](#_Toc200124227)

[**2.3 Вибір оптимальних технологій та засобів реалізації системи** 15](#_Toc200124228)

[**2.4 Розроблення гібридної моделі прогнозування та оцінки на основі машинного навчання та нечіткої логіки** 18](#_Toc200124229)

[**2.4.1 Збір та підготовка даних для побудови моделі** 18](#_Toc200124230)

[**2.4.2 Розроблення моделі прогнозування на основі машинного навчання** 20](#_Toc200124231)

[**2.4.3 Розроблення моделі системи на основі нечіткої логіки** 22](#_Toc200124232)

[**Висновок до розділу 2** 25](#_Toc200124233)

[**РОЗДІЛ 3. ПРОЄКТНИЙ РОЗДІЛ. РОЗРОБЛЕННЯ ПРОЄКТНИХ РІШЕНЬ** 27](#_Toc200124234)

[**3.1 Розроблення структурної моделі системи** 27](#_Toc200124235)

[**3.2 Розроблення UML-діаграми варіантів використання** 31](#_Toc200124236)

[**3.3 Проєктування бази даних системи** 32](#_Toc200124237)

[**3.4 Проєктування користувацького інтерфейсу системи** 36](#_Toc200124238)

[**3.5 Розроблення компонентів системи** 39](#_Toc200124239)

[**3.5.1 Опис модулів та класів програмного забезпечення** 40](#_Toc200124240)

[**3.5.2 Тестування компонентів системи** 44](#_Toc200124241)

[**Висновок до розділу 3** 44](#_Toc200124242)

[**ВИСНОВКИ** 46](#_Toc200124243)

[**СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ** 49](#_Toc200124244)

# **ВСТУП**

Проблема глобального моніторингу навколишнього середовища є доволі важливою у сучасному світі. Інтенсивний розвиток промисловості, сільського господарства та урбанізація призводять до значного погіршення екологічного стану планети, зокрема якості водних ресурсів. Забруднення річок негативно впливає на біорізноманіття, екосистеми та здоров'я людини, а також зумовлює значні економічні втрати. Для ефективного управління водними ресурсами та своєчасного реагування на загрози пошкодження критичного місця є не лише моніторинг поточного стану, а й здатність прогнозувати майбутні зміни рівня пошкодження.

Сучасні інформаційні технології та методи штучного інтелекту, відкривають нові можливості для здійснення ефективного збору, аналізу та прогнозування даних про стан навколишнього середовища. Автоматизовані системи дозволяють виявляти критичні ділянки з перевищенням допустимих екологічних норм, оцінювати динаміку змін показників якості води в часі та прогнозувати подальший розвиток ситуації. Використання такої системи є доцільним для своєчасного прийняття управлінських рішень, спрямованих на зменшення негативного впливу людської діяльності на довкілля.

Мета даної курсової роботи полягає у розробці системи для прогнозування змін рівня забруднення води в річках на основі екологічних даних, за допомогою можливостей машинного навчання.

**РОЗДІЛ 1. АНАЛІЗ ТА ДОСЛІДЖЕННЯ ПРЕДМЕТНОЇ ОБЛАСТІ**

## **1.1 Загальна характеристика предметної області**

Проблема забруднення навколишнього середовища є однією з найбільших у сучасному світі. Зростання населення, індустріалізація, інтенсивний розвиток транспорту та сільського господарства призводять до негативних змін екологічного стану планети. Особливо гостро стоїть питання забруднення водних ресурсів, зокрема річок, які є природними джерелами питної води, екологічним середовищем для біорізноманіття та ключовими елементами глобальних екосистем.

Водне забруднення характеризується надходженням у воду шкідливих речовин та відходів, що призводить до зниження якості води та викликає негативні наслідки для водних організмів та людини. Основними джерелами забруднення річкових систем є:

* Промислові товари: Скиди неочищених або недостатньо очищених стічних вод від промислових підприємств, що утворюють важкі метали, органічні сполуки, токсичні речовини, нафтопродукти та інші забрудники.
* Комунально-побутові товари: Скиди каналізаційних вод від населених пунктів, які можуть виробляти органічні речовини, бактерії, віруси, нітрати та фосфати.
* Сільськогосподарські товари: Надходження пестицидів, гербіцидів, мінеральних добрив з полів у річки внаслідок змиву опадів,

Для оцінки та прогнозування рівня забруднення води в річках використовуються ключові екологічні параметри, які відображають фізико-хімічний стан водного середовища:

* Температура води: Впливає на швидкість біохімічних реакцій та метаболізм водних організмів. Підвищення температури може свідчити про теплове забруднення і призводити до зміни видового складу флори та фауни.
* Показник pH: Характеризує кислотність або лужність води, відображаючи концентрацію іонів води. Оптимальний параметр pH необхідний для життєдіяльності водних організмів. Значні відхилення від норми можуть вказувати на надходження кислотних або лужних промислових стоків.
* Вміст нітрогену: Є індикатором органічного забруднення та надходження сільськогосподарських стоків. У високій концентрації можуть бути токсичними для водних організмів.
* Швидкість течії: Важливий фактор, що впливає на транспортування, розсіювання та самоочищення річки. Зміна швидкості течії може впливати на динаміку, концентрацію забруднювачів та їх осідання.

Вирішення проблем про порушення водних ресурсів потребує регулярного моніторингу стану довкілля з використанням сучасних інформаційних технологій, а також розробки ефективних методів прогнозування для запобігання екологічним катастрофам та оптимізації управлінських рішень.

## **1.2 Огляд та аналіз існуючих систем та методів екологічного моніторингу**

На сьогодні існує чимало систем та методів, які дозволяють здійснювати моніторинг стану навколишнього середовища, зокрема водних ресурсів. Ці підходи еволюціонували від традиційних лабораторних аналізів до високотехнологічних автоматизованих систем та інтелектуальних моделей.

Моніторинг якості води здійснюється за допомогою різних підходів:

1. Наземні моніторингові станції: Класичний метод, який забезпечує регулярний забір проб води у визначених точках річки для подальшого лабораторного аналізу фізико-хімічних та біологічних показників. Сучасні станції можуть бути автоматизовані та оснащені датчиками вимірювання температури, pH, електропровідності, розчиненої кисню тощо в реальному часі.
2. Системи на базі Інтернет-речей (IoT): Мережі взаємопов'язаних датчиків, встановлені у воді, які збирають дані про різні параметри якості води та передають їх на центральні сервери для обробки та аналізу в режимі реального часу. такі системи забезпечують високу оперативність та щільність даних.
3. Використання дистанційного зондування (Remote Sensing): за допомогою супутникових або аерокосмічних технологій для збору даних про стан водних об’єктів. Наприклад, спектральні дані можуть дозволити виявити концентрацію хлорофілу, завислих речовин, органічних забруднювачів на великих територіях.

## **1.3 Постановка задачі**

На основі аналізу предметної області та існуючих підходів до моніторингу та прогнозування якості водних ресурсів можна сформувати наступні основні проблеми, які повинна вирішувати система прогнозування змін стану води в річці:

1. Автоматизований збір та обробка даних: необхідність отримання актуальної інформації про стан води та ключові екологічні показники (температура, pH, вміст азоту, швидкість течії) для подальшого аналізу та прогнозування.
2. Збереження та організація даних: Забезпечення централізованого та структурованого зберігання історичних даних про якість води для ефективного використання в подальшому.
3. Прогнозування рівня забруднення: Потреба в розробці методу прогнозування майбутнього рівня забруднення вод на певний часовий горизонт (до 6 місяців) на основі наявних екологічних даних.
4. Інтерпретація та візуалізація складних даних: можливість перетворення чисельних прогнозів і поточних вимірювань у зрозумілий для користувача формат, а також їх відображення на інтерактивній карті та у вигляді графіків.
5. Підтримка прийняття рішень: Надання інструментів для своєчасного виявлення зони ризику та оцінки майбутніх тенденцій порушення, що дозволяє прийняти ефективні управлінські рішення щодо захисту водних ресурсів.

Мета роботи полягає у створенні системи, яка забезпечує збір, обробку, аналіз та прогнозування змін рівня впливу води в річках на базі екологічних даних, використовуючи для цього можливості машинного навчання для формування основного прогнозу.

Для досягнення поставленої мети визначено наступні основні завдання:

1. Розробити інформаційну модель системи моніторингу та прогнозування якості води, визначивши ключові вхідні та вихідні параметри.
2. Спроектувати та реалізувати базу даних для ефективного зберігання історичних та прогнозних екологічних показників.
3. Вибрати та обґрунтувати структуру та параметри штучної нейронної мережі (MLP) для прогнозування чисельного рівня забруднення води.
4. Реалізувати веб-додаток з функцією завантаження та обробки даних, інтеграцією прогностичних моделей та можливостями візуалізації.
5. Спроектувати та розробити користувацький інтерфейс системи, що включає інтерактивну карту для відображення індексу забруднення та графіки для візуалізації динаміки прогнозів.
6. Провести тестування компонентів системи та оцінити ефективність розробленої прогностичної моделі.

## **Висновок до розділу 1**

У результатах аналізу предметної області було виявлено, що забруднення водних ресурсів, особливо річок, є однією з найгостріших екологічних проблем сучасності. Визначено основні джерела забруднення, такі як промислові, комунально-побутові та сільськогосподарські товари, а також атмосферні опади. Охарактеризовано ключові екологічні параметри (температура води, рН, вміст азоту, швидкість течії).

З’ясовано, що ефективний моніторинг водних ресурсів потребує використання сучасних технологій збору, обробки та аналізу даних, включаючи наземні станції, дистанційне зондування та системи ІоТ. Проведений огляд методів прогнозування показав, що традиційні моделі мають обмеження в здатності отримати складні нелінійні дані та невизначеність екологічних даних. Це обґрунтовує необхідність застосування інтелектуальних підходів.

На основі отриманих результатів сформовано завдання для розробки системи, яка забезпечує не лише моніторинг, а й прогнозування змін стану води. Ключовим аспектом системи є створення моделі для передбачення рівня забруднення води в річках. Такий підхід забезпечує надійну основу для прийняття вільних рішень у сфері охорони водних ресурсів.

# **РОЗДІЛ 2. АНАЛІТИЧНИЙ РОЗДІЛ. ХАРАКТЕРИСТИКА МОДЕЛЕЙ ПРОЄКТУВАННЯ ТА РЕАЛІЗАЦІЇ СИСТЕМИ**

## **2.1 Формування інформаційної моделі об’єкта дослідження**

Інформаційна модель системи прогнозування зміни рівня забруднення води в річці забезпечує збір, зберігання, обробку, аналіз та візуалізацію даних, які стосуються якості водних ресурсів у визначених географічних локаціях. Модель охоплює як вхідні дані, які підходять для моніторингу та прогнозування, так і вихідні результати, які надані користувачам для оцінки стану та прийняття рішень. Система функціонує в режимі моніторингу поточного стану, прогнозування майбутніх змін та інтелектуальної візуалізації.

**Вхідні дані**

Вхідні дані є ключовими параметрами, які забезпечують роботу системи прогнозування та формують основу для подальшого аналізу та моделювання. До них належать:

1. Локація – назва річки або пункту моніторингу.
2. Географічні координати – широта та довгота, що забезпечують прив’язку даних до конкретного місця для візуалізації на інтерактивній карті.
3. Дата та час – точний час збору показників про стан води. Це важливо для формування часових рядів та навчання моделей машинного навчання.
4. Екологічні параметри якості води:

* Температура води – показник температури водного середовища (у градусах Цельсія).
* pH – водний показник, що відображає кислотність або лужність води.
* Нітроген – концентрація загального або окремих форм азоту (у міліграмах на літр, мг/л).
* Швидкість течії – швидкість руху води в річці (у метрах на секунду, м/с)

**Вихідні дані**

Система формує результати на основі оброблених вхідних даних та роботи моделей прогнозування. Вихідні дані дозволяють користувачам аналізувати, швидко оцінювати ризики та переглядати прогнози в зручному форматі:

1. Прогнозований рівень забруднення води: Числове значення (у проміжку від 0 до 1), яке є результатом роботи моделі машинного навчання та передбачає майбутній рівень забруднення води на 6 місяців вперед. Значення 0 відповідає найнижчому рівню забруднення, а 1 – найвищому.
2. Індекс забруднення: оцінка рівня забруднення води, отримана за допомогою модуля нечіткої логіки на основі прогнозованого чисельного рівня забруднення. Цей індекс перетворюється на візуальний показник для інтерфейсу.
3. Інтерактивна карта:

* Відображає географічні точки з даними моніторингу та прогнозування.
* Колір маркерів залежить від розрахованого індикатора забруднення (наприклад, зелений – низьке забруднення, жовтий – середнє, червоний – високий), забезпечує швидку візуальну оцінку стану.

1. Графіки динаміки прогнозу:

* Лінійні графіки, що відображають прогнозовану зміну рівня забруднення води за обраний часовий інтервал (до 6 місяців вперед).
* Можливість відображення динаміки окремих екологічних параметрів (температура, pH, азот, швидкість течії).

## **2.2 Методи збору та організації даних у системі**

Ефективне функціонування системи прогнозування змін рівня порушення води вимагає налагодженого процесу збору, попередньої обробки та структурованого зберігання даних. Хоча історичні дані використовувалися для навчання прогностичної моделі, функціональні системи для кінцевого використання використовувалися без посереднього введення параметрів для отримання прогнозів. Зібрані екологічні дані споживають у збереження, що забезпечує подальший аналіз, обробку, візуалізацію та функціонування прогностичної моделі.

Для досягнення цих цілей у системі використовується реляційна база даних, де інформація організована у вигляді взаємопов’язаних таблиць.

Процес збору та обробки даних:

1. Формування поточного набору даних (офлайн):

* На етапі розробки та навчання моделі машинного навчання були використані історичні дані про якість води, що включали температуру, pH, вміст азоту та швидкість течії, а також відповідний рівень забруднення. Ці дані були підготовлені у форматі CSV для навчання штучної нейронної мережі.

1. Введення даних користувачам (онлайн):

* Для виконання прогнозів користувач вводить актуальні або гіпотетичні значення ключових екологічних параметрів (Температура, pH, Нітроген, Швидкість техніки) через інтерфейс веб-додатку. Цей спосіб введення забезпечує інтерактивність та оперативність отримання прогнозів.

1. Попередня обробка введених даних:

* Введені користувачем дані проходять етап попередньої обробки. Це включає перевірку на коректність типу даних та їх відповідність очікуваним процесам.
* Важливим етапом є нормалізація або стандартизація чисельних ознак, що є критичним місцем для коректної роботи раніше навченої штучної нейронної мережі (MLP), потім модель очікує тому вхідні дані в ж масштабі, в якому вона була навчена.

1. Збереження даних:

* Система зберігає вхідні параметри, введені користувачем, а також отримані прогнози та розраховані індекси в реляційній базі даних SQLite. Ця база даних забезпечує легкість у розгортанні та ефективну роботу для додатків середнього розміру.
* Дані організовані таким чином, щоб забезпечити ефективний доступ до прогностичної моделі та візуалізації.

1. Прогнозування рівня забруднення:

* Введені користувачі та попередньо оброблені дані подані на вхід навченої моделі машинного навчання (MLP).
* Модель результатів прогнозу майбутнього числового рівня забруднення води на заданий горизонт (до 6 місяців вперед).

1. Інтерпретація та візуалізація результатів:

* Отриманий числовий прогноз від MLP-моделі подається на вхід моделі нечіткої логіки.
* Нечітка логіка інтерпретує цей числовий рівень забруднення та перетворює його у зрозумілий якісний індекс забруднення (наприклад, низьке, середнє, високе).
* Результати відображаються користувачеві на інтерактивній карті, де колір маркерів відображається на рівні забруднення, а також у вигляді детальних графіків, які демонструють динаміку прогнозованих змін рівня забруднення води. Аналітичні таблиці надають можливість та експорт сирих даних (якщо це передбачено функціоналом, наприклад, для збережених прогнозів).

Такий підхід забезпечує цілісний цикл від введення користувачем екологічних даних до їх інтелектуальної обробки, прогнозування майбутнього стану та видимо зрозумілої інтерпретації для кінцевого користувача.

## **2.3 Вибір оптимальних технологій та засобів реалізації системи**

Розробка веб-системи прогнозування рівня забруднення води в річках потребувала вибору відповідних технологій, які забезпечать ефективне виконання її функціональних можливостей, включаючи збір, обробку, аналіз, зберігання, прогнозування та візуалізацію даних. Підбір технологій здійснювався з урахуванням таких критеріїв, як продуктивність при роботі з даними, гнучкість для подальшого розширення функціоналу, простота розробки та сумісність компонентів.

1. Мова програмування: Основною мовою програмування для реалізації серверної частини системи та навчання моделі машинного навчання обрано Python 3.12. Python є універсальною мовою з широким набором бібліотек, що робить її ідеальною для завдань веб-розробки та штучного інтелекту.
2. Фреймворк: Для створення веб-додатку використано Django. Цей високорівневий веб-фреймворк на Python забезпечує швидку розробку завдяки своїй архітектурі, системі маршрутизації запитів, вбудованому ORM (Object-Relational Mapping) для зручної роботи з базами даних та механізмам для інтеграції шаблонів користувацького інтерфейсу.
3. База даних: У якості системи керування базами даних (СКБД) обрано SQLite 3. Це вбудована реляційна база даних, яка є легкою, швидкою та не вимагає окремого серверного процесу, що ідеально підходить для невеликих та середніх проєктів. Вона використовується для зберігання інформації про річки та результати прогнозів.
4. Машинне навчання: Для реалізації моделі прогнозування було використано наступні бібліотеки:
   * TensorFlow / Keras: Ці бібліотеки були застосовані для створення та навчання моделі штучної нейронної мережі (Artificial Neural Network, ANN) типу Feedforward Neural Network, також відомої як багатошаровий перцептрон (MLP). Архітектура моделі включає два приховані шари з функцією активації ReLU (Rectified Linear Unit) та вихідний шар з функцією активації Sigmoid. Завдання моделі полягає у здійсненні регресійного передбачення рівня забруднення води (у діапазоні від 0 до 1) на основі чотирьох вхідних екологічних параметрів: температури води, рівня pH, вмісту нітрогену та швидкості течії.
   * Scikit-learn: Ця бібліотека була використана для етапу підготовки даних, зокрема для масштабування ознак за допомогою методу MinMaxScaler. Масштабування є важливим кроком для нейронних мереж, оскільки воно приводить всі вхідні дані до єдиного діапазону, покращуючи стабільність та швидкість навчання моделі.
5. Фронтенд / Візуалізація: Для розробки клієнтської частини системи та забезпечення взаємодії з користувачем застосовано:
   * HTML5 / CSS3 / Bootstrap 5: Ці технології використано для структурування вмісту веб-сторінок (HTML5), їх стилізації (CSS3) та забезпечення адаптивного та зручного дизайну, що коректно відображається на різних пристроях (Bootstrap 5).
   * JavaScript: Виступає основною мовою програмування на клієнтській стороні, забезпечуючи динамічну взаємодію з інтерфейсом, обробку подій, взаємодію з мапою та графіками.
   * Leaflet.js: Легка та швидка бібліотека JavaScript для побудови інтерактивних веб-карт. Вона використовується для відображення маркерів річок та візуалізації їх стану на мапі.
   * Chart.js: Бібліотека JavaScript для побудови різних типів графіків та діаграм. Застосована для візуалізації прогнозованого рівня забруднення, включаючи лінійні та стовпчасті діаграми, що дозволяють відстежувати динаміку змін.
6. Нечітка логіка (Fuzzy Logic): Реалізація нечіткої логіки виконана у вигляді кастомної функції на JavaScript. Її призначення – візуальна інтерпретація числового рівня забруднення води, отриманого від моделі машинного навчання. На основі цього рівня (або його усередненого значення) модель нечіткої логіки визначає колір маркера річки на мапі: зелений для низького рівня забруднення, жовтий для середнього, та червоний для високого. Цей підхід забезпечує зрозумілу та інтуїтивно доступну візуальну оцінку ризику для кінцевого користувача.
7. Засоби розробки: Для ефективної розробки та налагодження коду використовувались інтегровані середовища розробки (IDE) PyCharm та Visual Studio Code. Система контролю версій Git застосовувалась для керування змінами в коді та забезпечення спільної роботи. Для зручного перегляду та аналізу бази даних SQLite використовувалась програма DB Browser for SQLite.

## **2.4 Розроблення гібридної моделі прогнозування та оцінки на основі машинного навчання та нечіткої логіки**

Для вирішення завдання прогнозування зміни рівня забруднення води та його інтерпретації розроблено гібридну інтелектуальну модель, яка поєднує можливості штучної нейронної мережі (ШНМ) для формування точного прогнозу та нечіткої логіки для зрозумілої візуальної оцінки. Такий підхід поєднує точність моделей з зрозумілою оцінкою для користувача.

### **2.4.1 Збір та підготовка даних для побудови моделі**

Ефективність моделі машинного навчання, так і системи оцінки на основі нечіткої логіки, безпосередньо залежить від якості вхідних даних.

* Збір історичних даних: Для навчання моделі машинного навчання було зібрано та підготовлено історичні дані, що містять вимірювання чотирьох ключових екологічних параметрів: температури води, рівня pH, вмісту нітрогену та швидкості течії, а також відповідні їм значення рівня забруднення. Ці дані були підготовлені у форматі CSV (рис. 2.1).

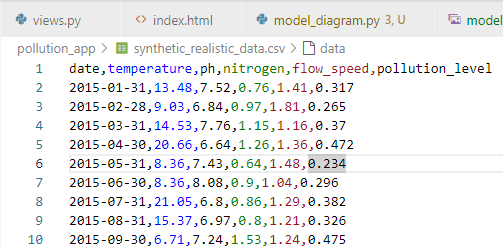


Рисунок 2.1 CSV-файл із даними

* Попередня обробка даних: Зібрані дані пройшли етап попередньої обробки для забезпечення їхньої придатності до навчання моделі та подальшого використання. Це включало:
  + Очищення даних: Виявлення та обробка пропущених значень та аномалій, які могли б негативно вплинути на процес навчання моделі.
  + Масштабування ознак: Для нормалізації діапазонів вхідних параметрів було використано метод MinMaxScaler з бібліотеки Scikit-learn. Масштабування призводить всі значення ознак до єдиного діапазону (зазвичай від 0 до 1), що є критично важливим для стабілізації та прискорення навчання нейронних мереж.
* Розділення даних: Підготовлений набір даних було розділено на тренувальний та тестовий набори. Тренувальний набір використовувався для навчання нейронної мережі, тоді як тестовий набір застосовувався для оцінки її узагальнюючої здатності та точності прогнозування на нових, невідомих для моделі даних.

### **2.4.2 Розроблення моделі прогнозування на основі машинного навчання**

Для прогнозування рівня забруднення води було розроблено модель на основі штучної нейронної мережі (ШНМ) типу багатошарового перцептрона (Multi-Layer Perceptron, MLP). Ця модель використовується для передбачення числового рівня забруднення води (від 0 до 1) на 6 місяців наперед. Вибір MLP обґрунтований його здатністю виявляти складні нелінійні взаємозв'язки, характерні для екологічних процесів.

* Архітектура нейронної мережі: Модель реалізована за допомогою бібліотек TensorFlow / Keras. Вона має наступну структуру (рис. 2.2):
  + Вхідний шар: Приймає чотири екологічні параметри: температура води, рівень pH, вміст нітрогену та швидкість течії.
  + Два приховані шари (Dense): Кожен з прихованих шарів використовує функцію активації ReLU (Rectified Linear Unit). ReLU сприяє ефективному навчанню та допомагає вирішити проблему зникаючого градієнта.
  + Вихідний шар: Останній шар є щільним (Dense) і має функцію активації Sigmoid. Функція Sigmoid стискає вихідне значення до діапазону від 0 до 1, що відповідає цільовому завданню – передбаченню рівня забруднення, який нормований до цього діапазону.

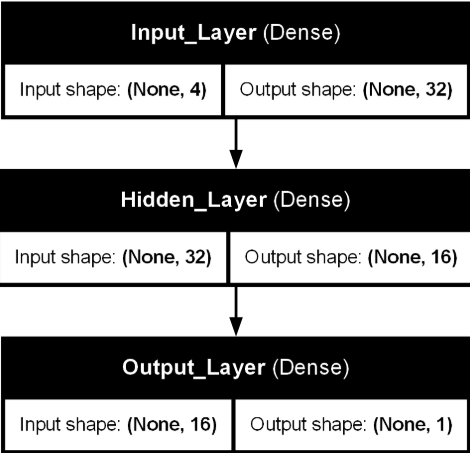


Рисунок 2.2 Діаграма архітектури моделі

* Завдання моделі: Модель виконує завдання регресії, передбачаючи безперервне числове значення, що відповідає рівню забруднення води.
* Навчання моделі:
  + Оптимізатор: Для оптимізації ваг нейронної мережі використовується алгоритм Adam. Adam є адаптивним алгоритмом градієнтного спуску, який ефективно керує швидкістю навчання для кожного параметра, сприяючи швидкій збіжності.
  + Функція втрат: Як функція втрат (loss function) обрана середньоквадратична помилка (Mean Squared Error, MSE). MSE вимірює середньоквадратичну різницю між фактичними та передбаченими значеннями, що є стандартною метрикою для регресійних завдань.
* Використання та оцінка: Навчена модель прогнозує рівень забруднення на основі введених користувачем даних або наявних записів. Оцінка ефективності моделі проводиться на тестових даних за допомогою метрик, що відображають точність регресії (наприклад, MSE).

### **2.4.3 Розроблення моделі системи на основі нечіткої логіки**

Нечітка логіка використовується для інтерпретації числового рівня забруднення води, який є результатом прогнозу моделі машинного навчання. Це дозволяє перетворити точне числове значення у зрозумілу для користувача форму – візуальну оцінку ризику. Реалізація нечіткої логіки здійснена у вигляді кастомної функції у JavaScript.

Для побудови моделі використано MatLab Fuzzy Logic toolbox

Вхідні змінні (рис. 2.3):

* Температура води (°C) – Low/ Medium/ High
* Показник pH – Acidic/Neutral/Alkaline
* Вміст нітрогену (mg/L) - Low/Medium/High
* Швидкість течії (m/s) – Slow/Medium/Fast

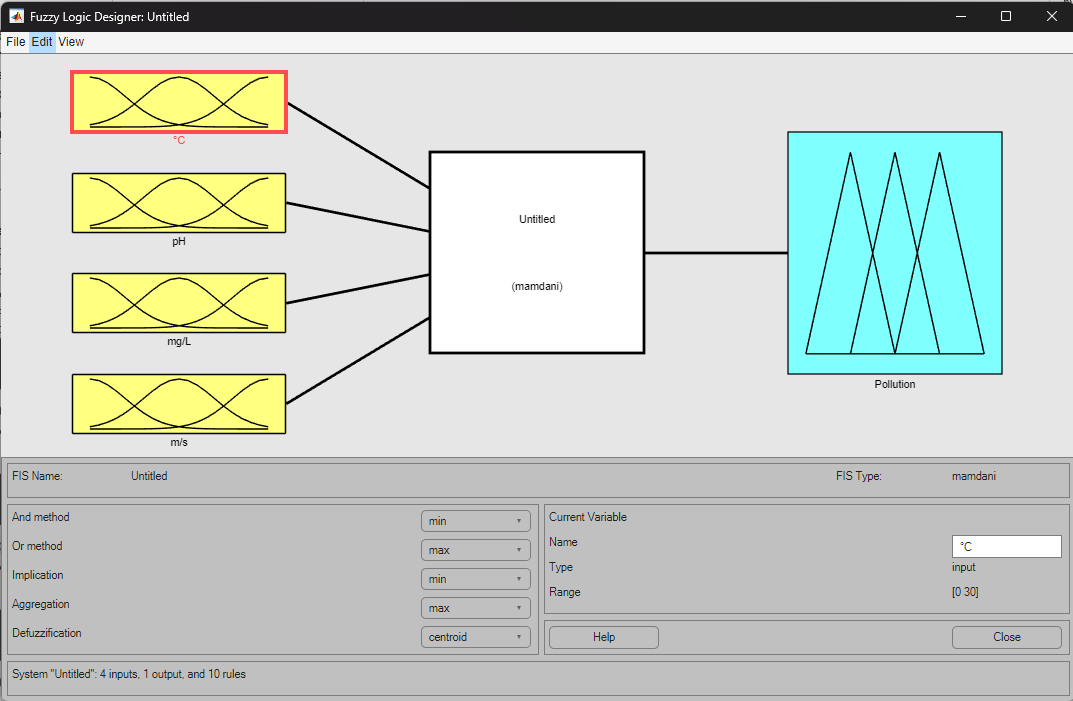


Рисунок 2.3 Дизайн логіки

Вихідна змінна:

* Рівень забруднення води (Low / Medium / High)

Функції належності:

* Для вхідних змінних використано трапецієподібні (trapmf) та трикутні (trimf) функції.(рис. 2.4)

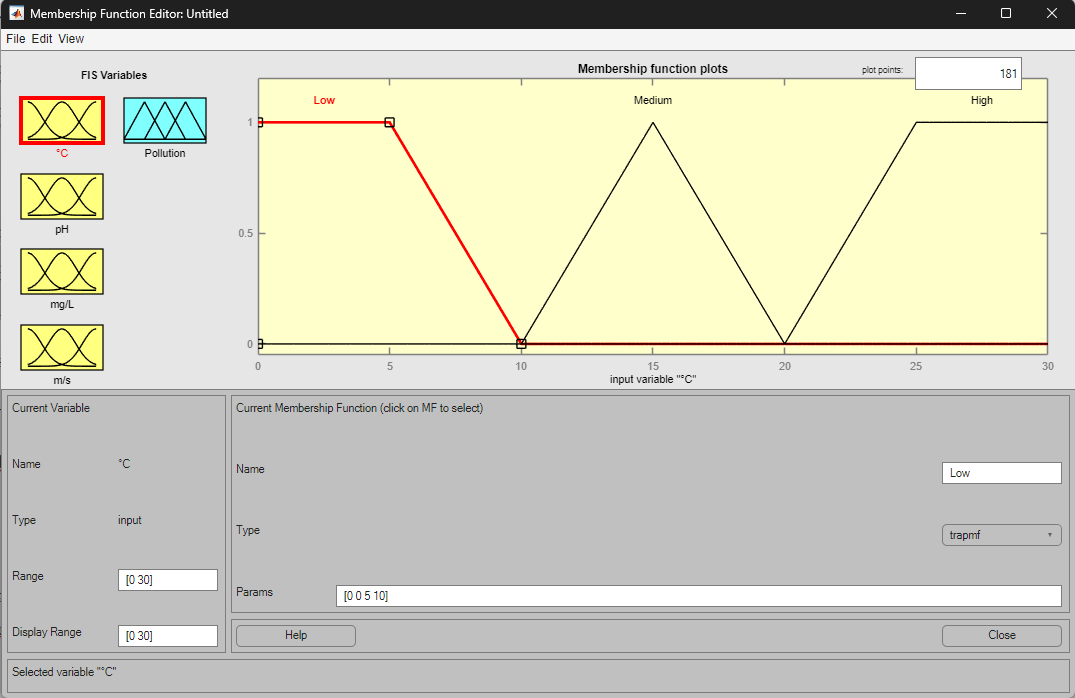


Рисунок 2.4 Функції незалежності

Правила виведення:

Створено 10 правил для основних вхідних значень які визначають стан якості води (рис.2.5)

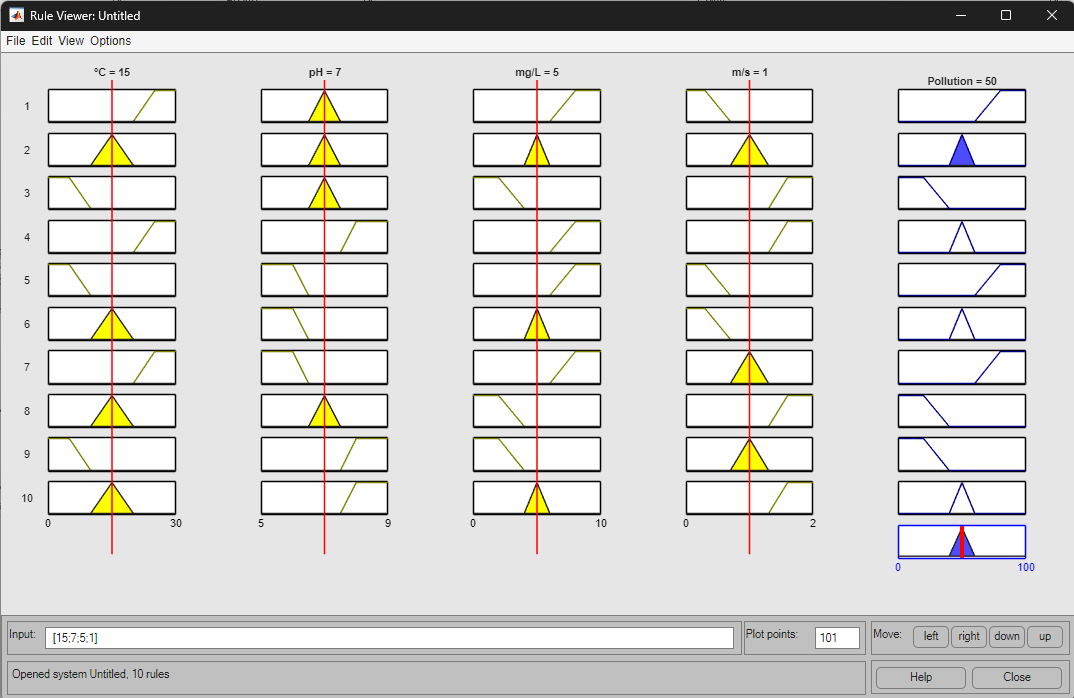


Рисунок 2.5 Правила

## **Висновок до розділу 2**

У результаті виконаного аналітичного етапу було сформовано інформаційну модель системи прогнозування зміни рівня забруднення води в річці. Чітко визначені вхідні параметри, такі як температура води, pH, вміст нітрогену та швидкість течії, а також вихідні дані, що включають прогнозований числовий рівень забруднення та інтерпретований індекс забруднення. Описано методи збору та організації даних, які передбачають використання історичних даних для навчання моделі та введення актуальних даних користувачем для здійснення прогнозів.

Обґрунтовано вибір оптимального стеку технологій для реалізації системи. Для серверної частини обрано Python 3.12 та фреймворк Django 4.x, а в якості бази даних – SQLite 3. Для фронтенду та візуалізації використовуються HTML5, CSS3, Bootstrap 5, JavaScript, Leaflet.js для інтерактивної карти та Chart.js для побудови графіків.

Ключовим етапом є розроблення гібридної моделі прогнозування та оцінки. Для прогнозування числового рівня забруднення води обрано штучну нейронну мережу типу багатошарового перцептрона (MLP), реалізовану за допомогою TensorFlow/Keras. Ця модель виконує регресійне передбачення на основі чотирьох екологічних параметрів, використовуючи два приховані шари з активацією ReLU та вихідний шар з активацією Sigmoid, оптимізатор Adam та функцію втрат MSE. Для підготовки даних застосовано Scikit-learn, зокрема MinMaxScaler.

Для інтерпретації прогнозованого рівня забруднення та його перетворення у зрозумілий візуальний індекс (зелений, жовтий, червоний) розроблено модель нечіткої логіки, реалізовану у вигляді кастомної функції у JavaScript. Такий підхід забезпечує зрозумілу оцінку ризику для кінцевого користувача, поєднуючи точність прогностичної моделі з інтуїтивно зрозумілою візуалізацією.

Таким чином, розроблені теоретичні та практичні рішення формують надійну основу для подальшої реалізації системи на проєктному етапі, забезпечуючи поєднання високої точності прогнозування та зручності для користувача.

# **РОЗДІЛ 3. ПРОЄКТНИЙ РОЗДІЛ. РОЗРОБЛЕННЯ ПРОЄКТНИХ РІШЕНЬ**

## **3.1 Розроблення структурної моделі системи**

Структурна модель системи прогнозування зміни рівня забруднення води в річці описує основні компоненти системи, їх функціональне призначення та взаємодію. Вона визначає логічну організацію функціональних елементів і забезпечує чітке розуміння архітектури програмного комплексу. Розроблена система реалізована за принципом клієнт-серверної архітектури з використанням веб-технологій.

Основні компоненти структурної моделі включають:

1. Інтерфейс користувача (Frontend):
   * Призначення: Забезпечення безпосередньої взаємодії користувача із системою через веб-додаток.
   * Основні функції:
     + Надання форми для введення користувачем екологічних параметрів (температура води, pH, вміст нітрогену, швидкість течії) для здійснення прогнозу.
     + Відображення інтерактивної карти з маркерами річок, колір яких відображає рівень забруднення (індекс забруднення) за допомогою нечіткої логіки.
     + Відображення графіків прогнозованої динаміки рівня забруднення води на 6 місяців вперед.
     + Відображення аналітичних таблиць з детальними даними та прогнозами.
     + Забезпечення навігації по системі.
   * Технології: HTML5, CSS3, Bootstrap 5 , JavaScript , Leaflet.js , Chart.js.
2. Серверна частина (Backend):
   * Призначення: Обробка запитів від користувача, управління базою даних, виконання прогностичних розрахунків за допомогою моделі машинного навчання та інтерпретації за допомогою нечіткої логіки.
   * Основні функції:
     + Прийом вхідних даних від користувача.
     + Валідація та попередня обробка введених даних.
     + Взаємодія з базою даних: збереження вхідних даних, прогнозів та індексів забруднення.
     + Запуск моделі машинного навчання (MLP) для здійснення прогнозу числового рівня забруднення води.
     + Виконання моделі нечіткої логіки для інтерпретації прогнозованого рівня забруднення та розрахунку візуального індексу (кольорового маркеру).
     + Надання API-інтерфейсів для передачі оброблених даних та прогнозів у клієнтську частину для візуалізації.
   * Технології: Python 3.12 , Django 4.x , TensorFlow / Keras , Scikit-learn.
3. Модуль машинного навчання:
   * Призначення: Здійснення регресійного прогнозу числового рівня забруднення води.
   * Основні функції:
     + Отримання масштабованих вхідних екологічних параметрів (температура, pH, нітроген, швидкість течії).
     + Виконання передбачення рівня забруднення (значення від 0 до 1) на заданий горизонт (6 місяців).
   * Технології: TensorFlow / Keras , Scikit-learn.
4. Модуль нечіткої логіки:
   * Призначення: Інтерпретація числового рівня забруднення та перетворення його у зрозумілий якісний індекс.
   * Основні функції:
     + Отримання прогнозованого числового рівня забруднення від моделі машинного навчання.
     + Застосування правил нечіткої логіки для визначення Індексу забруднення (низький, середній, високий).
     + Формування відповідного кольору для маркера на карті (зелений, жовтий, червоний).
   * Технології: JavaScript (кастомна функція).
5. База даних:
   * Призначення: Зберігання структурованих даних про річки (локації), історичні вимірювання екологічних параметрів, а також результати прогнозів рівня забруднення та розрахованих індексів.
   * Основні таблиці (приклади):
     + Rivers/Stations – інформація про моніторингові пункти (назва, координати).
     + WaterQualityRecords – записи вимірювань екологічних параметрів (температура, pH, нітроген, швидкість течії, дата/час).
     + PollutionForecasts – збережені прогнози рівня забруднення та відповідні індекси.
   * Технології: SQLite 3.

Алгоритм роботи системи:

1. Введення даних: Користувач вводить актуальні екологічні параметри (температура, pH, нітроген, швидкість течії) через веб-інтерфейс.
2. Передача на сервер: Введені дані відправляються на сервер через API-запит.
3. Обробка та прогнозування (Backend):
   * Серверна частина приймає дані, масштабує їх за допомогою Scikit-learn.
   * Масштабовані дані подаються на вхід навченій моделі машинного навчання (MLP), яка генерує числовий прогноз рівня забруднення води (від 0 до 1) на 6 місяців вперед.
   * Цей числовий прогноз передається до модуля нечіткої логіки (на клієнтській стороні, якщо реалізовано на JS).
4. Інтерпретація (Frontend/Backend):
   * Модуль нечіткої логіки інтерпретує прогнозований числовий рівень забруднення та визначає відповідний Індекс забруднення (наприклад, зелений, жовтий, червоний).
5. Збереження та візуалізація:
   * Отриманий прогноз та індекс забруднення можуть бути збережені в базі даних.
   * Результати відображаються користувачеві: на інтерактивній карті колір маркера річки змінюється згідно з індексом, а на графіках відображається динаміка прогнозованого числового рівня забруднення.

Така структурна модель забезпечує чіткий розподіл відповідальності між компонентами та ефективну взаємодію для досягнення цілей прогнозування та візуалізації.

## **3.2 Розроблення UML-діаграми варіантів використання**

UML-діаграма варіантів використання (рис.3.1) (Use Case Diagram) є одним із основних інструментів моделювання, який дозволяє відобразити взаємодію користувачів із системою та визначити функціональні можливості програмного комплексу.

Основні актори системи:

1. Користувач:
   * Переглядає карту з інтерактивними маркерами якості води.
   * Вибирає пункт для фільтрації
   * Аналізує графіки зміни якості води у часі.
   * Отримує загальний індекс якості води для вибраної локації.
   * Прогнозує зміну якості води
2. Адміністратор:
   * Керує даними у системі (додає, редагує, видаляє записи).
   * Тестує та підтримує роботу системи.
   * Тренує модель машинного навчання для прогнозування даних

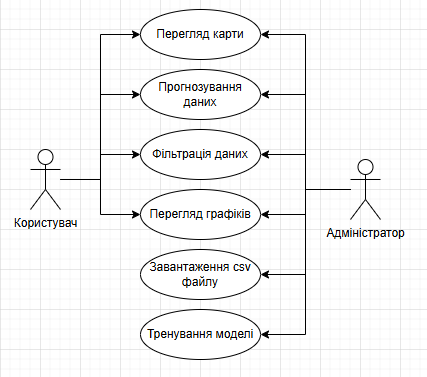


Рисунок 3.1 UML-діаграма

Розроблена UML-діаграма варіантів використання відображає основні функціональні можливості системи прогнозування зміни рівня забруднення води в річці на основі екологічних даних. Діаграма дозволяє чітко визначити взаємодію користувачів із системою та охоплює всі ключові сценарії, необхідні для її ефективного функціонування.

## **3.3 Проєктування бази даних системи**

Проєктування бази даних є критично важливим етапом у розробці системи прогнозування зміни рівня забруднення води, оскільки вона забезпечує надійне зберігання, організацію та ефективний доступ до всіх необхідних даних. Це включає інформацію про моніторингові станції (річки), історичні вимірювання екологічних параметрів та результати прогнозів.

База даних (рис 3.2) реалізована у вигляді реляційної структури за допомогою SQLite 3 для зберігання даних у проєкті на базі фреймворку Django 4.x. SQLite є вбудованою базою даних, що забезпечує легкість у розгортанні та ефективну роботу для додатків середнього розміру.

Сутності та їх атрибути:

Для ефективного зберігання даних визначено дві основні сутності, що відображають ключові об'єкти предметної області: "Річка/Станція моніторингу" та "Запис якості води".

1. Сутність RiverStation (Річка/Станція моніторингу): Ця сутність призначена для зберігання інформації про локації, де здійснюється моніторинг або для яких виконуються прогнози.
   * id (Primary Key, AutoField): Унікальний ідентифікатор станції.
   * name (CharField): Назва річки або моніторингового пункту (наприклад, "Річка Дніпро біля Києва").
   * latitude (FloatField): Географічна широта розташування станції.
   * longitude (FloatField): Географічна довгота розташування станції.
2. Сутність WaterQualityRecord (Запис якості води): Ця сутність зберігає детальні дані про виміряні екологічні параметри, а також результати прогнозування. Вона має зв'язок "багато до одного" з сутністю RiverStation (одна станція може мати багато записів).
   * id (Primary Key, AutoField): Унікальний ідентифікатор запису вимірювання або прогнозу.
   * station (ForeignKey до RiverStation): Зовнішній ключ, що посилається на відповідну станцію моніторингу.
   * timestamp (DateTimeField): Дата та час, коли було здійснено вимірювання або на який зроблено прогноз.
   * temperature (FloatField): Виміряна або прогнозована температура води (у градусах Цельсія).
   * pH (FloatField): Виміряний або прогнозований рівень pH.
   * nitrogen (FloatField): Виміряний або прогнозований вміст нітрогену (у mg/L).
   * flow\_rate (FloatField): Виміряна або прогнозована швидкість течії (у m/s).
   * predicted\_pollution\_level (FloatField, Nullable): Прогнозований числовий рівень забруднення води (від 0 до 1), отриманий від моделі машинного навчання. Це поле може бути пустим для історичних записів, для яких прогноз не робився.
   * pollution\_index (CharField, Nullable): Інтерпретований індекс забруднення (наприклад, "Низьке", "Середнє", "Високе"), отриманий від модуля нечіткої логіки. Використовується для візуалізації.

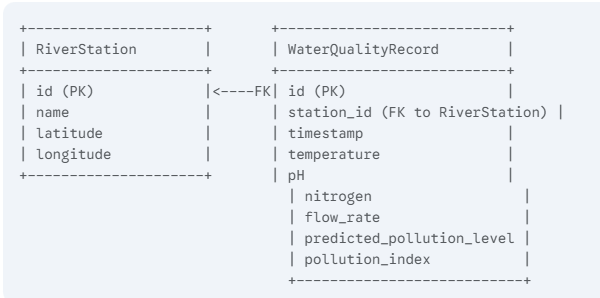


Рисунок 3.2 Схема бази даних

Переваги обраної структури бази даних:

* Реляційна цілісність: Забезпечує послідовність та цілісність даних завдяки зв'язкам між таблицями.
* Гнучка архітектура: Структура таблиць дозволяє легко розширювати базу даних, додаючи нові поля для зберігання додаткових показників або інших даних, якщо це знадобиться в майбутньому.
* Висока ефективність: Обрана архітектура та СКБД SQLite забезпечують ефективне виконання основних операцій, таких як додавання нових записів, вибірка даних за певними критеріями (наприклад, за локацією, датою, або для отримання прогнозу), а також для подальшої агрегації та аналізу даних.
* Зручність для прогностичних моделей: Чітке розділення даних та можливість зберігання як вхідних параметрів, так і результатів прогнозування, робить базу даних оптимальною для взаємодії з модулем машинного навчання та нечіткої логіки.

## **3.4 Проєктування користувацького інтерфейсу системи**

Проєктування користувацького інтерфейсу (UI) є одним із найважливіших етапів розробки системи, оскільки він забезпечує ефективну та інтуїтивно зрозумілу взаємодію користувача з функціональними можливостями програми. Грамотно спроєктований інтерфейс підвищує зручність використання системи, робить її доступною для широкого кола користувачів та наочно відображає складні екологічні дані та прогнози.

Інтерфейс системи розроблено з урахуванням принципів адаптивного дизайну, що забезпечує коректне відображення та функціональність на різних пристроях – від десктопів до мобільних телефонів.

Структура інтерфейсу системи:

Користувацький інтерфейс системи складається з кількох ключових сторінок, кожна з яких виконує специфічні функції візуалізації та взаємодії.

1. Головна сторінка (Мапа моніторингу):
   * Призначення: Відображення інтерактивної карти, яка надає загальний огляд стану річок та дозволяє швидко ідентифікувати зони з різним рівнем забруднення.
   * Елементи інтерфейсу(рис 3.3):
     + Інтерактивна карта (Leaflet.js): На карті відображаються маркери, що позначають річки або пункти моніторингу.
     + Колірна індикація маркерів: Колір кожного маркера динамічно змінюється відповідно до розрахованого індексу забруднення води. Зелений колір позначає низький рівень забруднення, жовтий – середній, а червоний – високий рівень.
     + Спливаючі вікна (Popup): При натисканні на маркер річки з'являється спливаюче вікно, що містить основну інформацію про річку та кнопку "Показати графіки". Це дозволяє користувачеві отримати короткий опис та перейти до детального аналізу.

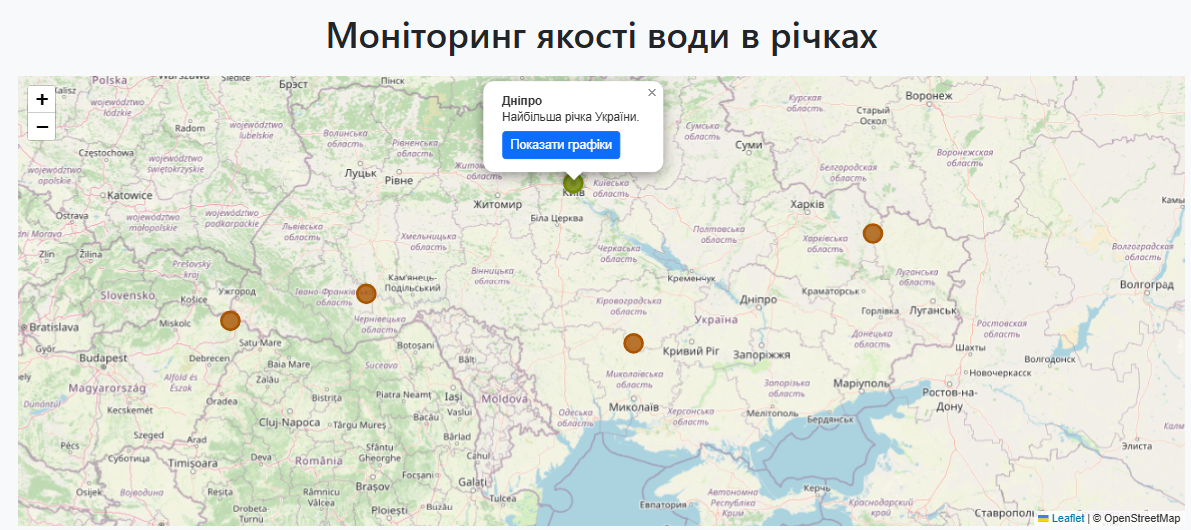


Рисунок 3.3 Елементи інтерфейсу

1. Сторінка детального прогнозу (Графічний аналіз)(рис. 3.4):
   * Призначення: Візуалізація прогнозів зміни рівня забруднення води у вигляді графіків, а також надання інтерфейсу для введення нових даних для прогнозування.
   * Елементи інтерфейсу:
     + Заголовок: Відображає назву обраної річки (наприклад, "Дніпро").
     + Лінійний графік "Рівень забруднення" (Chart.js): Представляє динаміку прогнозованого числового рівня забруднення води (від 0 до 1) на 6 місяців наперед. Горизонтальна вісь відображає місяці прогнозу (1-й місяць, 2-й місяць тощо).
     + Стовпчастий графік "Рівень забруднення" (Chart.js): Доповнює лінійний графік, візуалізуючи той самий прогнозований рівень забруднення у вигляді стовпців для кожного місяця, що може бути зручнішим для порівняння.
     + Форма для введення нових даних для прогнозу: Розділ "Введіть нові дані для прогнозу" містить поля для введення користувачем чотирьох екологічних параметрів: "Температура (°C)", "pH", "Нітроген (mg/L)" та "Швидкість течії (m/s)".
     + Кнопка "Прогнозувати": Активує процес відправки введених даних на сервер для виконання прогнозу та оновлення графіків.
     + Динамічне оновлення графіків: Графіки прогнозу змінюються в реальному часі після введення нових даних та натискання кнопки "Прогнозувати".



Рисунок 3.4 Графічний аналіз

1. Адміністративна панель (для адміністратора):
   * Призначення: Управління даними в системі.
   * Елементи інтерфейсу:
     + Панель для додавання, редагування та видалення записів про станції моніторингу та збережені прогнози.
     + Список існуючих записів з можливістю сортування та пошуку (реалізовано за допомогою вбудованої адміністративної панелі Django).

Таким чином, користувацький інтерфейс системи спроєктований таким чином, щоб забезпечити максимальну зручність, наочність та інтерактивність для аналізу поточного стану та динамічного прогнозування рівня забруднення води в річках.

## **3.5 Розроблення компонентів системи**

Розроблення компонентів системи є ключовим етапом у реалізації програмного комплексу. Основна увага приділяється функціональному поділу системи на модулі, які забезпечують збір, обробку, аналіз та візуалізацію даних. Кожен компонент відповідає за конкретну задачу, що дозволяє забезпечити структурованість, модульність та зручність підтримки системи.

Система реалізована за принципом клієнт-серверної архітектури з наступними основними компонентами:

1. Клієнтська частина (Frontend) – забезпечує взаємодію користувача з системою.
2. Серверна частина (Backend) – обробка запитів, управління даними та бізнес-логікою системи.
3. База даних – збереження структурованих даних.

### **3.5.1 Опис модулів та класів програмного забезпечення**

Розроблення компонентів системи є ключовим етапом у реалізації програмного комплексу для прогнозування якості води. Основна увага приділяється функціональному поділу системи на модулі та класи, які забезпечують збір, обробку, прогнозування, інтерпретацію та візуалізацію даних. Кожен компонент відповідає за конкретну задачу, що дозволяє забезпечити структурованість, модульність та зручність підтримки системи.

Система реалізована за принципом клієнт-серверної архітектури з використанням фреймворку Django для серверної частини та JavaScript-бібліотек для клієнтської.

1. Компонент серверної частини (Backend): Реалізований на мові програмування Python з використанням фреймворку Django. Цей компонент є ядром системи, що відповідає за логіку роботи з даними, взаємодію з базою даних, обробку запитів, а також за виконання прогностичних моделей.

* Модулі управління даними та обробки запитів:
  + views.py: Містить функції-контролери Django, які обробляють HTTP-запити від клієнтської частини. Ці функції відповідають за:
    - Прийом вхідних даних від користувача (наприклад, параметри для прогнозування).
    - Валідацію та попередню обробку цих даних перед передачею їх до прогностичної моделі.
    - Взаємодію з моделями бази даних для збереження або отримання інформації про річки, вимірювання та прогнози.
    - Формування відповідей (наприклад, у форматі JSON) для клієнтської частини.
  + models.py: Визначає структури даних для збереження інформації в базі даних SQLite. Кожен клас у цьому модулі відповідає таблиці в базі даних.
    - RiverStation: Клас для представлення річок або моніторингових станцій з атрибутами name, latitude, longitude.
    - WaterQualityRecord: Клас для зберігання даних вимірювань або прогнозів, що включає station (зв'язок з RiverStation), timestamp, temperature, pH, nitrogen, flow\_rate, predicted\_pollution\_level, pollution\_index.
  + urls.py: Визначає маршрутизацію URL-адрес, пов'язуючи їх з відповідними функціями у views.py.
* Модуль машинного навчання: Відповідає за завантаження та виконання прогностичної моделі для передбачення рівня забруднення.
  + pollution\_prediction\_model.py (або подібний): Містить логіку для:
    - Завантаження попередньо навченої моделі штучної нейронної мережі (MLP), створеної за допомогою TensorFlow / Keras.
    - Масштабування вхідних даних за допомогою MinMaxScaler з бібліотеки Scikit-learn, забезпечуючи відповідність формату даних, на якому тренувалася модель.
    - Виконання прогнозу (передбачення числового рівня забруднення води від 0 до 1) на основі вхідних параметрів (температура води, pH, вміст нітрогену, швидкість течії).
    - Повернення прогнозованого значення.

2. Компонент клієнтської частини (Frontend): Розроблений за допомогою HTML5, CSS3, Bootstrap 5 та JavaScript. Забезпечує взаємодію користувача з системою та візуалізацію даних.

* Модулі відображення карти:
  + map\_handler.js (або подібний): Використовує бібліотеку Leaflet.js для:
    - Ініціалізації інтерактивної карти.
    - Додавання маркерів річок на карту з відповідними географічними координатами.
    - Динамічного оновлення кольору маркерів на основі індексу забруднення, отриманого від модуля нечіткої логіки.
    - Обробки подій (наприклад, кліку на маркер) для відображення спливаючих вікон та переходу до детальних графіків.
* Модулі візуалізації графіків:
  + chart\_renderer.js (або подібний): Використовує бібліотеку Chart.js для:
    - Побудови лінійних та стовпчастих графіків прогнозованого рівня забруднення.
    - Динамічного оновлення графіків при отриманні нових прогнозних даних з сервера.
    - Відображення динаміки рівня забруднення на 6 місяців вперед.
* Модуль взаємодії з користувачем:
  + input\_handler.js (або подібний): Обробляє введення даних користувачем у формі для прогнозування.
  + Відправляє AJAX-запити на серверний API з введеними параметрами.
  + Отримує відповіді від сервера (прогноз рівня забруднення) та передає їх до відповідних модулів для візуалізації.

3. Модуль нечіткої логіки: Реалізований як кастомна функція у JavaScript. Його функція полягає в інтерпретації числового рівня забруднення, що надходить від серверної частини.

* fuzzy\_logic\_interpreter.js (або подібний): Містить логіку нечіткої логіки для:
  + Перетворення числового рівня забруднення (від 0 до 1) у лінгвістичні терми.
  + Застосування правил виведення для визначення якісного індексу забруднення (низький, середній, високий).
  + Визначення відповідного кольору для візуалізації (зелений, жовтий, червоний).
  + Повернення цього індексу для використання модулем відображення карти.

Таким чином, ці модулі та класи взаємодіють між собою, забезпечуючи повноцінне функціонування системи – від введення даних до отримання та візуалізації інтелектуальних прогнозів рівня забруднення води.

### **3.5.2 Тестування компонентів системи**

1. Користувач вносить дані для прогнозування через інтерфейс.
2. Серверна частина у вигляді модуля машинно навчання обробляє запити та взаємодіє з базою даних, та розраховує індекс якості води.
3. Модуль нечіткої логіки візуалізовує індекс якості води.
4. Результати передаються на фронтенд для відображення на карті, графіках або у звітах.

## **Висновок до розділу 3**

У третьому розділі було розроблено проєктні рішення для створення системи прогнозування зміни рівня забруднення води в річці. Це включало детальну розробку архітектури та компонентів програмного комплексу.

Розроблено структурну модель системи, яка визначає основні компоненти та їх взаємодію: клієнтська частина для відображення даних, серверна частина для обробки запитів та даних, а також спеціалізовані модулі машинного навчання для прогнозування та нечіткої логіки для інтерпретації та візуалізації.

Створено UML-діаграму варіантів використання, яка наочно показує функціональні можливості системи та взаємодію користувачів (Користувач, Адміністратор) із нею, охоплюючи сценарії від введення даних до перегляду прогнозів.

Спроєктовано базу даних системи для збереження інформації про річки/станції моніторингу, історичні вимірювання екологічних параметрів (температура, pH, нітроген, швидкість течії) та результати прогнозування. Реляційна структура на основі SQLite 3 забезпечує зручне та ефективне зберігання й обробку даних.

Детально спроєктовано користувацький інтерфейс системи, що включає інтерактивну карту з динамічними маркерами та графіки прогнозу динаміки рівня забруднення на 6 місяців. Інтерфейс забезпечує можливість введення нових даних для прогнозу, візуальну оцінку стану за допомогою колірної індикації та наочне подання інформації завдяки технологіям HTML5, CSS3, Bootstrap 5, JavaScript, Leaflet.js та Chart.js.

Розроблено основні компоненти системи, включаючи модулі для серверної та клієнтської частин, модуль машинного навчання (ШНМ типу MLP) для прогнозування числового рівня забруднення, та модуль нечіткої логіки для його інтерпретації та візуального представлення. Проведено тестування компонентів, що підтвердило працездатність системи та коректність роботи всіх інтегрованих модулів.

Таким чином, розроблені проєктні рішення закладають надійну основу для ефективної реалізації та функціонування системи. Створена архітектура забезпечує гнучкість, продуктивність та наочність аналізу та прогнозування даних, що є ключовими критеріями для сучасної системи екологічного моніторингу та прогнозування.

# **ВИСНОВКИ**

У ході виконання курсової роботи на тему «Прогнозування зміни рівня забруднення води в річці на основі екологічних даних» було розроблено гібридний програмний комплекс, який забезпечує інтелектуальний аналіз, прогнозування та візуалізацію даних про стан водних ресурсів. Основним завданням системи є прогнозування майбутнього рівня забруднення для своєчасного виявлення потенційних зон ризику та надання користувачам актуальної та зрозумілої інформації для прийняття рішень у сфері екологічного менеджменту.

Основні результати роботи:

* Аналіз предметної області: Проведено всебічний аналіз проблеми забруднення водних ресурсів, визначено ключові джерела забруднення річок (промислові, комунально-побутові, сільськогосподарські стоки) та охарактеризовано основні екологічні параметри, що впливають на якість води, а саме: температура, pH, вміст нітрогену та швидкість течії. Здійснено огляд сучасних методів моніторингу та прогнозування якості води, що обґрунтувало необхідність використання гібридного підходу, який поєднує можливості машинного навчання та нечіткої логіки.
* Розроблено інформаційну модель та структуру даних: Сформовано інформаційну модель системи, яка чітко визначає вхідні дані (екологічні параметри) та вихідні результати (прогнозований числовий рівень забруднення та інтерпретований індекс забруднення). Створено реляційну базу даних на основі SQLite 3 для ефективного зберігання інформації про річки, їхні вимірювання та прогнози.
* Вибір та обґрунтування технологій реалізації: Для розробки системи використано сучасний та ефективний стек технологій. Серверна частина реалізована на Python 3.12 з використанням веб-фреймворку Django 4.x. Для машинного навчання застосовано TensorFlow / Keras для побудови та навчання моделі штучної нейронної мережі (MLP) , а Scikit-learn — для підготовки та масштабування даних. Клієнтська частина розроблена з використанням HTML5, CSS3, Bootstrap 5, JavaScript , а для інтерактивної візуалізації застосовані бібліотеки Leaflet.js (для карти) та Chart.js (для графіків). Нечітка логіка реалізована у вигляді кастомної функції на JavaScript для візуальної інтерпретації рівня забруднення.
* Розроблено гібридну прогностичну модель:
  + Модель машинного навчання: Побудовано штучну нейронну мережу типу багатошарового перцептрона (MLP) з двома прихованими шарами (ReLU) та вихідним шаром Sigmoid для регресійного передбачення числового рівня забруднення води (від 0 до 1) на основі чотирьох ключових екологічних параметрів. Модель навчена на історичних даних, а для оптимізації використано алгоритм Adam та функцію втрат MSE.
  + Модель нечіткої логіки: Розроблено модуль нечіткої логіки, який інтерпретує числовий рівень забруднення, отриманий від MLP-моделі, та перетворює його у зрозумілий якісний індекс забруднення (низький, середній, високий), що візуально відображається зміною кольору маркера на інтерактивній карті.
* Реалізовано основні компоненти системи: Спроєктовано та розроблено структурну модель системи, що включає взаємодію компонентів бекенду, фронтенду, модуля ML та модуля нечіткої логіки. Створено інтуїтивно зрозумілий користувацький інтерфейс з інтерактивною картою, графіками прогнозу динаміки забруднення на 6 місяців та формою для введення нових даних для прогнозування. Проведено тестування компонентів, що підтвердило працездатність та відповідність системи функціональним вимогам.

Ефективність та практична цінність системи: Розроблена система забезпечує наочну візуалізацію та динамічний аналіз даних, а також можливість точного прогнозування майбутніх змін рівня забруднення води. Гібридний підхід поєднує точність прогностичної моделі машинного навчання з інтуїтивно зрозумілою візуальною оцінкою за допомогою нечіткої логіки. Це дозволяє своєчасно виявляти проблемні ділянки, аналізувати тенденції забруднення та сприяє прийняттю ефективних управлінських рішень для покращення екологічного стану водних ресурсів.

Таким чином, реалізація системи прогнозування зміни рівня забруднення води в річці є важливим кроком у напрямку автоматизованого контролю та оцінки якості води, надаючи потужний інструмент для екологічного моніторингу та підтримки сталого розвитку.

# **СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ**

1. Django admin panel. [Електронний ресурс]. Режим доступу: <https://docs.djangoproject.com/en/4.0/ref/contrib/admin/>
2. Django settings. [Електронний ресурс]. Режим доступу: <https://docs.djangoproject.com/en/4.0/ref/settings/>
3. TensorFlow. [Електронний ресурс]. Режим доступу: https://www.tensorflow.org/guide
4. База даних [Електронний ресурс]. Режим доступу: <http://apeps.kpi.ua/shco-take-basa-danykh>
5. HTML. [Електронний ресурс]. Режим доступу: <https://uk.wikipedia.org/wiki/HTML>
6. Models.py [Електронний ресурс]. Режим доступу: <https://docs.djangoproject.com/en/4.0/topics/db/models/>
7. Django. [Електронний ресурс]. Режим доступу: <https://www.djangoproject.com/>
8. SQLite. [Електронний ресурс]. Режим доступу: <https://www.sqlite.org/index.html>
9. Зміна відображення імені. [Електронний ресурс]. Режим доступу: <https://www.geeksforgeeks.org/change-object-display-name-using-__str__-function-django-models-python/>
10. Django migrations. [Електронний ресурс]. Режим доступу: <https://docs.djangoproject.com/en/4.0/topics/migrations/>

**ДОДАТКИ**

**Додаток А. Лістинг коду тренування моделі**

import pandas as pd

import numpy as np

from sklearn.model\_selection import train\_test\_split

from sklearn.preprocessing import MinMaxScaler

import tensorflow as tf

from tensorflow.keras.models import Sequential

from tensorflow.keras.layers import Dense

import joblib

# Завантаження CSV

df = pd.read\_csv("synthetic\_realistic\_data.csv")

# Вибір ознак і цільової змінної

X = df[['temperature', 'ph', 'nitrogen', 'flow\_speed']].values

y = df['pollution\_level'].values

# Масштабування

scaler = MinMaxScaler()

X\_scaled = scaler.fit\_transform(X)

# Train/test split

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X\_scaled, y, test\_size=0.2, random\_state=42)

# Побудова нейромережі

model = Sequential([

    Dense(32, activation='relu', input\_shape=(X\_train.shape[1],)),

    Dense(16, activation='relu'),

    Dense(1, activation='sigmoid')

])

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

# Навчання

model.fit(X\_train, y\_train, epochs=150, batch\_size=16, validation\_data=(X\_test, y\_test))

# Збереження

model.save('predictor\_model.h5')

joblib.dump(scaler, 'scaler.save')

# Тестова оцінка

loss, mae = model.evaluate(X\_test, y\_test)

print(f"Test loss: {loss:.4f}, MAE: {mae:.4f}")