

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ТЕХНОЛОГІЙ ТА
ДИЗАЙНУ**

Факультет мехатроніки та комп'ютерних технологій

Кафедра комп'ютерних наук

ДИПЛОМНА БАКАЛАВРСЬКА РОБОТА

на тему

**Розробка програмного засобу для збору та обробки інформації
про параметри навколишнього середовища**

Виконав: студент групи БІТ-2-19
спеціальності 122 – Комп'ютерні науки
освітньої програми Комп'ютерні науки
Микола КРАВЧЕНКО

Науковий керівник: к.т.н., доц.
Тетяна АСТІСТОВА

Рецензент: д.т.н., проф.
Сергій КРАСНИТСЬКИЙ

Київ 2023

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ТЕХНОЛОГІЙ ТА
ДИЗАЙНУ

Факультет мехатроніки та комп'ютерних технологій
Кафедра комп'ютерних наук
Спеціальність 122 Комп'ютерні науки
Освітня програма Комп'ютерні науки

ЗАТВЕРДЖУЮ

Завідувач кафедри
комп'ютерних наук
_____ В.Ю.Щербань

«_____» _____ 2023 р.

ЗАВДАННЯ

НА ДИПЛОМНУ БАКАЛАВРСЬКУ РОБОТУ СТУДЕНТУ

Кравченко Миколі Сергійовичу

1. Тема роботи : Розробка програмного засобу для збору та обробки інформації про параметри навколишнього середовища, науковий керівник роботи Астістова Тетяна Іванівна , к.т.н., доцент затверджені наказом КНУТД від “08” листопада 2022 року №224-уч.
2. Строк подання студентом дипломної роботи 05.06. 2023
3. Вихідні дані до дипломної роботи : розробки кафедри комп'ютерних наук
4. Зміст дипломної роботи: Розділ 1. Методи інтелектуального аналізу даних; Розділ 2. Розробка архітектури програмного засобу; Розділ 3. Розробка програмного продукту.
5. Дата видачі завдання 01.02.2023 року

6. КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломної роботи	Терміни виконання етапів	Примітка про виконання
1	Вступ	30.04 2023р.	
2	Розділ 1. Методи інтелектуального аналізу даних	1.05.2023р.	
3	Розділ 2. Розробка архітектури програмного засобу	12.05.2023р.	
4	Розділ 3. Розробка програмного продукту	26.05.2023р.	
5	Висновки	27.05 2023р.	
6	Оформлення дипломної роботи	31.05 2023р.	
7	Здача дипломної роботи на кафедру для рецензування	2.06.2023р.	
8	Перевірка кваліфікаційної роботи на наявність текстових співпадінь та помилок	6.06 2023р.	
9	Подання дипломної роботи на затвердження завідувачу кафедри	10.06 2023р.	

Студент

Микола КРАВЧЕНКО

Науковий керівник
роботи

Тетяна АСТІСТОВА

Рецензент

Сергій КРАСНИТСЬКИЙ

АНОТАЦІЯ

Кравченко М.С. «Розробка програмного засобу для збору та обробки інформації про параметри навколишнього середовища»

Дипломна бакалаврська робота за спеціальністю 122 - «Комп'ютерні науки» – Київський національний університет технологій та дизайну, Київ, 2023 рік.

Темою дипломної бакалаврської роботи є розробка програмного продукту для збору, зберігання та обробки масивів даних параметрів навколишнього середовища. Збір даних відбувається з локальної портативної метеостанції через віртуальний СОМ-порт та з веб-сайту метеосервісу через АРІ. Попередня обробка даних полягає у формуванні значень вектору ознак відповідного метричного простору на базі отриманих масивів даних. Здійснюється аналіз даних з метою виявлення прихованих взаємозв'язків між параметрами та візуалізації динаміки еволюції погодних умов у визначеній місцевості.

Створення такої моделі за допомогою веб-орієнтованої системи для збору, зберігання та обробки масивів даних параметрів навколишнього середовища дасть змогу співставити її з загальною динамічною моделлю, яка надається глобальною системою метеоспостереження. В результаті можна буде виконати оцінку майбутнього стану погодних умов для генерації відповідних профілів електричної генерації електроенергії на основі відновлювальних джерел енергії.

Ключові слова : клієнт-орієнтована система, масиви даних, веб-додаток, JSON, API, JavaScript

ANNOTATION

Kravchenko M.S. "Design the program application for the data collection and information processing as to environment parameters"

Bachelor's thesis degree work specialty 122 - Computer sciences, Kyiv National University of Technologies and Design, Kyiv, 2023.

The diploma work topic is to design of a web-oriented system for collecting, storing and processing data arrays of environmental parameters. Data collection carries out from the local portable weather station by means of the virtual COM port and from the weather service website due to API. Data pre-processing consists in the formation of the feature vector value in the corresponding metric space on the basis of the obtained data arrays. Data analysis is carried out in order to identify hidden relationships between parameters and visualize the dynamics of weather condition evolution in a certain area.

The formation of such a model using a web-oriented system for collecting, storing and processing data arrays of environmental parameters will make it possible to correlate it with the general dynamic model provided by the global meteorological observation system. As a result, it will be possible to perform an assessment of the future weather state for the generation of appropriate profile in electric power generation based on renewable energy sources.

Keywords: client-oriented system, big data, web-application, JSON, API, JavaScript

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

ВЗД - Видобутку Знань з Даних

ВМД - Великі Масиви Даних

ПЗ - Програмне забезпечення

ППДАВ - Проблема План Дані Аналіз Висновки

СБ – сонячна батарея

API - Application Programming Interface

FURPS - Functionality Usability Reliability Performance Supportability

JSON - JavaScript Object Notation

KDD - Knowledge Discovery in Databases

MPA - Multi Page Application

PPDAC - Problem, Plan, Data, Analysis, Conclusion

PWA - Progressive Web Application

REST - Representational State Transfer

SATC - Software Assurance Technology Center

SOAP - Service Object Access Protocol

SPA - Single Page Application

ЗМІСТ

ВСТУП	8
РОЗДІЛ 1. МЕТОДИ ІНТЕЛЕКТУАЛЬНОГО АНАЛІЗУ ДАНИХ	10
1.1 Постановка завдання.....	10
1.2. Загальні принципи дослідження великих масивів даних.....	12
1.3. Обробка масивів даних. Основні етапи	14
1.4. Методи інтелектуального аналізу даних.....	18
РОЗДІЛ 2. РОЗРОБКА АРХІТЕКТУРИ ПРОГРАМНОГО ЗАСОБУ	20
2.1. Модель клієнт-серверної архітектура.....	20
2.2. Стандарти API.....	23
2.3. Погодні API	25
2.4. Перевірка статистичних гіпотез	27
2.5. Моделі оцінки якості веб-додатку.....	33
РОЗДІЛ 3. РОЗРОБКА ПРОГРАМНОГО ПРОДУКТУ	37
3.1. Розробка основних модулів системи	37
3.2. Алгоритм роботи програмного продукту.....	38
3.3. Модуль моніторингу параметрів.....	39
3.4. Аналіз даних.....	41
ВИСНОВКИ	47
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	49
ДОДАТКИ	51

ВСТУП

Актуальність теми. Важливим викликом для нашої держави є розгорнута Росією повномасштабна війна проти України. Ефективна державна політика, крім забезпечення військових заходів, повинна також бути направлена на реалізацію чітких та послідовних заходів, пов'язаних із забезпеченням енергетичної, екологічної та продовольчої безпеки України. Енергетична криза в Україні спричинила розвиток відновлювальної енергетики. Потужність джерел цієї енергії залежить від температури, вологості повітря, швидкості вітру, тобто від метеорологічних умов місцевості.

Наявність інформації про параметри навколишнього середовища, в яких знаходяться різні об'єкти господарської діяльності, є надзвичайно важливою в економічному, екологічному та соціальному вимірі.

Розвиток інформаційних, мережевих та мікропроцесорних технологій сформував необхідні умови для збору, зберігання та обробки даних, що забезпечують інформацію про параметри навколишнього середовища. Ця інформація може бути корисною у побудові прогнозованого профілю генерації електричної енергії. Вилучення корисної інформації з великого масиву даних є надзвичайно складним завданням з практичної точки зору, тому виникає потреба в розробці нових методів аналізу даних за допомогою комп'ютерних програмних заходів .

Мета дослідження. Мета дослідження полягає в розробці такого програмного засобу, який може здійснювати накопичування, збереження та попередню обробку масивів даних, які отримуються з засобів моніторингу параметрів навколишнього середовища, для подальшого вилучення корисної інформації з таких даних.

Завдання дослідження. На основі наявних масивів даних, які надаються портативним електронним пристроєм моніторингу параметрів навколишнього середовища та веб-порталом The Open Whether, створити програмний засіб, який накопичує, зберігає та обробляє дані з цих масивів з отриманням масивів

нових даних, в яких міститься корисна інформація для використання в прогнозуванні та навчанні нейронних мереж або в машинному навчанні.

Об'єктом дослідження є масиви даних параметрів навколишнього середовища.

Апробація отриманих результатів.

1. Kravchenko M. S. Data analysis in systems with alternative energy sources / M. S. Kravchenko, M. S. Nuzhdyak ; scientific adviser T. I. Astistova // Інноватика в освіті, науці та бізнесі: виклики та можливості : матеріали III Всеукраїнської конференції здобувачів вищої освіти і молодих учених, м. Київ, 17 листопада 2022 року. – Т. 1. – Київ : КНУТД, 2022. – С. 81-84.

2. Кравченко О.П., Кравченко М.С., Кластеризація даних параметрів навколишнього середовища // Інтелектуальні технології лінгвістичного аналізу: Тези доповідей міжнародної науково-технічної конференції. – Національний авіаційний університет. – Київ, 2022. - С.34.

3. Кравченко О.П., Кравченко М.С., Комп'ютеризована інформаційна система моніторингу параметрів оптимального функціонування агропромислового комплексу на основі маломірних дронів //

Тези доповідей міжнародної науково-технічної конференції. – Національний авіаційний університет. – Київ, 20 квітня 2023. - С

4. Астістова Т.І., Кравченко М.С., Комп'ютеризована інформаційна система для збору та обробки даних при оптимізації функціонування відновлювальних джерел енергії // II Міжнародна науково-практична Інтернет конференція молодих учених та студентів «Електромеханічні, інформаційні системи та нанотехнології». - КНУТД. – Київ, 2023. – С..

5. Астістова Т.І., Кравченко М.С., Манойлов В.Е., Розподілені системи дистанційного контролю та моніторингу навколишнього середовища // (II Міжнародна науково-практична Інтернет конференція молодих учених та студентів «Електромеханічні, інформаційні системи та нанотехнології»

.- КНУТД. – Київ, 2023. – С

РОЗДІЛ 1. МЕТОДИ ІНТЕЛЕКТУАЛЬНОГО АНАЛІЗУ ДАНИХ

1.1 Постановка завдання.

В Україні в зв'язку з наростаючою енергетичною кризою продовжується розвиток відновлювальної енергетики. Відновлювальні джерела енергії, які використовуються для виробництва електричної енергії, зокрема, на основі сонця та вітру, характеризуються невичерпним і потужним потенціалом. Проте, на відміну від традиційних джерел (на основі вуглеводнів), потужність відновлювальних джерел енергії напряму залежить від метеорологічних умов місцевості, в якій вони розташовані. Так, потужність сонячної батареї (СБ) пропорційна інтенсивності сонячного випромінювання, що падає на її поверхню та температури панелі СБ.

Останній параметр знаходиться також в залежності від температури та вологості повітря, швидкості вітрового потоку. З іншого боку, швидкість вітру та величина атмосферного тиску визначають потужність вітрогенератора. Крім того, в зв'язку із зростаючими ризиками впливу небезпечних зовнішніх чинників на екологічну безпеку навколишнього середовища (це стосується об'єктів підвищеної небезпеки, насамперед, підприємств хімічної та фармацевтичної промисловості, атомних електростанцій тощо), нагальною є потреба в забезпеченні необхідного контролю атмосфери, щоб запобігти потенційно небезпечному розвитку ситуації. І, нарешті, існує нагальна потреба належним чином оптимізувати функціонування агропромислового комплексу в аспекті вирощування та збирання життєво необхідних сільськогосподарських культур. Такі фактори навколишнього середовища, як інтенсивність сонячного випромінювання, температура та вологість повітря, швидкість вітру, атмосферний тиск, температур та вологість ґрунту є основними чинниками, що впливають на агропромислові процеси.

Завдяки мережевим, та мікропроцесорним та інформаційним, технологіям можна сформулювати умови для збору, зберігання та обробки даних інформації про параметри навколишнього середовища.

Збір даних може здійснюватися як за допомогою портативних електронних модулів моніторингу параметрів навколишнього середовища з відповідними електронними давачами (сенсорами), так і метеостанціями, розташованих у визначених локаціях. При цьому, в більшості випадків, такі метеостанції видають інформацію, яка стосується прогнозу погоди на найближчий час. З практичної точки зору оптимальне функціонування енергосистеми з відновлювальними джерелами енергії ґрунтується на досягненні максимального накладення профілів генерації та споживання електричної енергії в системі. Профіль генерації електричної потужності в такій системі характеризується, як вже згадувалося раніше, залежністю від погодних умов місцевості, в яких розташовані відновлювальні джерела електричної генерації. Прогноз погоди може містити корисну інформацію відносно чинників, які впливають на профіль генерації. Крім того, наявні параметри оточуючого середовища, які збираються портативними або локальними метеостанціями, може слугувати масивом даних, з яких можливо видобути необхідну інформацію, яка буде корисною у побудові прогнозованого профілю генерації електричної енергії, яка виробляється за допомогою відновлювальних джерел енергії. Аналогічно, вилучення прогностичної інформації для екологічних та сільськогосподарських потреб має надзвичайно важливе значення.

Вилучення інформації з даних, видобуток знань, видобуток даних, глибинний (інтелектуальний) аналіз даних (англ. *Data mining*) – це синоніми, що використовуються для визначення технології аналізу масивів даних, яка ґрунтується на виявленні в наявному масиві даних раніше невідомої, але корисної та доступної інформації; яка в подальшому використовується для знаходження певних патернів, трендів та кореляцій з метою отримання прогностичних величин бажаних параметрів.

З практичної точки зору вилучення корисної інформації з великого масиву даних складним процесом. Часто традиційні інструменти та методи аналізу даних неможливо використовувати через величезний об'єм масиву

даних. Іноді нетрадиційний характер даних унеможлиблює застосування традиційних підходів, навіть якщо набір даних відносно невеликий. В інших ситуаціях питання, на які необхідно отримати відповідь, не можуть бути вирішені за допомогою вже існуючих методів аналізу даних, і, тому, виникає потреба в розробці нові методів аналізу даних.

Отже, виникає потреба в розробці програмного засобу, що буде здійснювати накопичування, збереження та попередню обробку масивів даних параметрів навколишнього середовища для подальшого вилучення корисної інформації з таких даних для прогнозування.

1.2. Загальні принципи дослідження великих масивів даних

Виконання дослідницьких проектів, в яких використовуються великі масиви даних, вимагає підходу, який орієнтований на розв'язання задач для яких застосування специфічних статистичних інструментів є лише одним із компонентів цілісного циклу досліджень. Для розв'язання такого роду задач був запропонований цикл ППДАВ, аббревіатура якого розшифровується як *Проблема План Дані Аналіз Висновки* (англ. PPDAC - *Problem, Plan, Data, Analysis, Conclusion*) (рис. 1.1) [1].

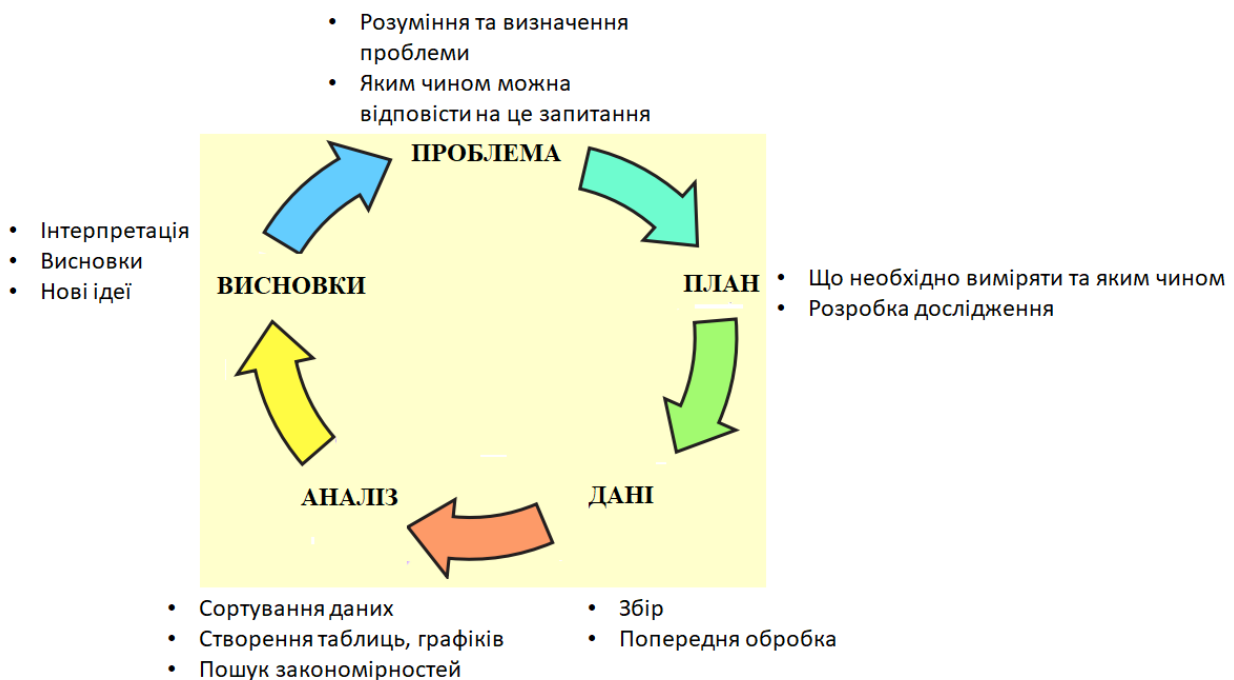


Рисунок 1.1.- Цикл ППДАВ

Структура всіх емпіричних досліджень, в яких використовуються великі масиви даних, може бути представлена моделлю, що складається з п'яти етапів. Зазначається, що така структура є корисною з двох причин:

- вона забезпечує шаблон для активного використання даних, отриманих при проведенні емпіричних досліджень;
- вона забезпечує можливість для критичного перегляду вже завершених досліджень.

Розглянемо етапи моделі:

1. На першому етапі (етап Проблема) відбувається чітке формулювання того, що потрібно отримати з дослідження. Деталізована структура дослідження та визначена термінологія, допомагають перевести контекстуальну проблему дослідження у таку форму, яка дозволяє керувати процесом дослідження в цілому та реалізовувати послідовність необхідних етапів.
2. На другому етапі (етап План) розробляється план, який стосується збору необхідних даних та зазначаються можливі методи, що можна використати для аналізу даних.
3. На третьому етапі (етап Дані) відбувається збір та попередня обробка даних.
4. На четвертому етапі (етап Аналіз) провадиться аналіз отриманих даних, здійснюється вирішення всіх необхідних питань, які були сформульовані на першому етапі.
5. На п'ятому етапі (етап Висновки) отримується необхідна інформація, що стосується результатів, що були досягнуті після виконання дослідження. Також цей етап дає можливість виявити переваги або недоліки, що стосуються попередніх етапів (*План, Дані, Аналіз*), а також можливих помилок, які могли виникнути в результаті виконання роботи.

1.3. Обробка масивів даних. Основні етапи

Сучасне суспільство часто називають інформаційним суспільством, оскільки швидкий розвиток технологій збору та зберігання даних дозволяє різноманітним організаціям отримувати та накопичувати величезні обсяги даних. Доступність таких величезних масивів даних не приносить ніякої користі, якщо вони не перетворюються в цінну інформацію. Часто складається ситуація, що споживач занурюється в надзвичайно потужний потік даних, проте не знаходить в ньому ніякої корисної інформації. Рішенням цієї проблеми є сукупність методів, які в англійській мові називаються *Data mining*, що перекладається як *Вилучення інформації з даних*, *Видобуток знань*, *Глибинний (інтелектуальний) аналіз даних*. Вони включають в себе обчислювальні процеси, які використовують математичні алгоритми для виявлення в наявному масиві даних раніше невідомої, але корисної та доступної інформації та для визначення ймовірності майбутніх подій. Ці методи також відомі під назвою *Видобуток Знань з Даних*, ВЗД (англ. *Knowledge Discovery in Databases*, KDD) [2].

Інтелектуальний аналіз даних є невід'ємною частиною загального процесу "виявлення знання з даних", в якому необроблені (первинні) дані, шляхом виконання певної послідовності етапів, перетворюються в корисну інформацію (рис. 1.2).



Рисунок 1.2- Етапи Видобутку Знань з Даних, ВЗД

Вхідні дані можуть зберігатися в різноманітних форматах, наприклад, у вигляді текстових файлів або електронних таблиць) і розміщуватися або в централізованому сховищі даних, або розподілятися між кількома сайтами. Метою етапу «Попередня обробка даних» є перетворення первинних (необроблених) вхідних даних у відповідний формат для подальшого аналізу. Операції, які при цьому виконуються, включають: 1) об'єднання масивів даних отриманих з кількох джерел; 2) очищення масивів даних від шуму та 3) вибір необхідних форматів запису даних, які є суттєвими для виконання подальшого інтелектуального аналізу даних. В загальному процесі обробки масивів даних цей етап є найбільш трудомістким та тривалим у часі.

Виконання етапу "подальша обробка даних" гарантує, що лише достовірні та корисні результати включаються в контент веб-додатку. На цьому етапі реалізується візуалізація, яка дає змогу досліджувати вихідні дані та результати, що отримуються після інтелектуального аналізу даних. Інтелектуальний аналіз даних можна визначити як набір методів, які використовуються для ефективного автоматизованого виявлення раніше невідомих, нових, корисних та зрозумілих шаблонів (патернів) у великих базах даних.

Використовуючи концепцію ППДАВ, можна в певній мірі модифікувати структуру ВЗД (рис. 1.3).

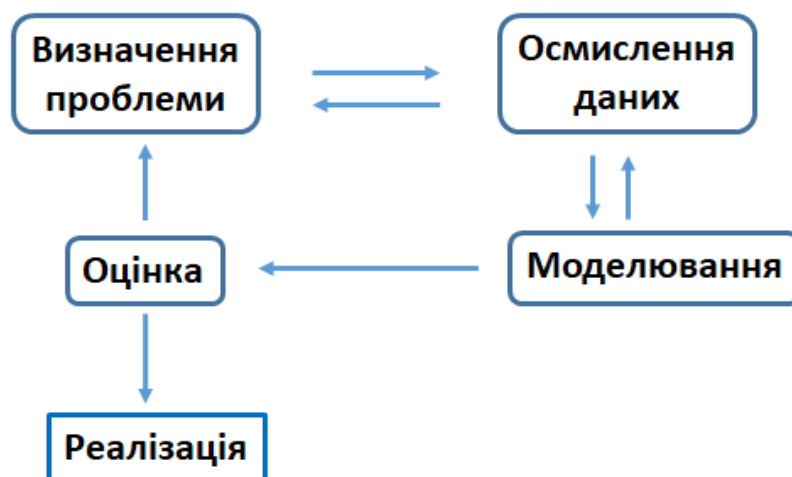


Рисунок 1.3 Модифікована структура ВЗД

Етап "Визначення проблеми"

Основна мета першого етапу процесу інтелектуального аналізу даних полягає в тому, щоб визначити вимоги та цілі такого проекту, що в свою чергу, дозволяє сформулювати його у вигляді задачі (проблеми) інтелектуального аналізу даних. Формалізація задачі надає можливість розробки плану реалізації проекту.

Етап "Осмислення даних"

На цьому етапі здійснюється збір даних з доступних джерел, виконується їх завантаження, інтеграція та ретельний огляд з метою визначення чи в змозі будуть ці дані вирішити окреслену задачу. Якщо ні, то для ефективного вирішення проблеми можна вносити додаткові дані або видаляти надлишкові дані. Крім того, на цьому етапі також ідентифікуються відсутні дані.

Етап "Попередня обробка даних"

Цей етап зазвичай займає більшу частину, в деяких випадках навіть близько 90%, часу, який витрачається на реалізацію проекту.

Етап "Моделювання"

На цьому етапі, для досягнення цілей запропонованого на першому етапі рішення, виконується реалізація обчислювальних моделей з використанням різних алгоритмів інтелектуального аналізу даних.

Етап "Оцінювання"

На цьому етапі оцінюються результати обчислювального моделювання, щоб визначити, чи задовольняє отримана модель вирішенню задачі, яка сформульована на першому етапі. Для цього виконується її тестування на основі тестових наборів даних. Якщо точність моделей на даних тестування є недостатньою, тоді повертаються до попередніх етапів, щоб точніше налаштувати ті області, які можуть бути причинами низької точності. Після досягнення задовільного рівня точності процес переходить до фази розгортання.

Етап "Реалізація"

На цьому етапі нові ідеї та цінну інформацію, що були отримані з даних, потрібно представити таким чином, щоб зацікавлені сторони могли використовувати їх у зручній для себе спосіб. Залежно від вимог проекту цей етап може бути простим (створення звіту) або складним (вимагає подальшої ітеративної обробки інтелектуального аналізу даних). Саме на цьому етапі створюються інформаційні панелі або графічні інтерфейси користувача для задоволення всіх вимог зацікавлених сторін. Інтелектуальний аналіз даних є основною операцією при вилученні інформації з масивів даних, що будуть надходити та зберігатися в контентній частині веб-додатку.

З вищенаведеного означення виводяться наступні положення:

- Інтелектуальний аналіз даних – це процес автоматизованого виявлення раніше невідомих шаблонів у великих обсягах даних.
- Цей великий обсяг даних, у більшості випадків, є історичними даними, який накопичується за певний проміжок часу і зберігається у сховищі даних.
- Інтелектуальний аналіз даних має справу з великими обсягами даних.
- Знайдені шаблони (патерни) повинні бути дійовими, новими, корисними та зрозумілими.
- Інтелектуальний аналіз даних надає можливість визначати закономірності в історичних даних, що, в свою чергу, дозволяє виконувати прогноз майбутньої поведінки об'єкту, який характеризується цими даними. При цьому, хоча інтелектуальний аналіз даних є можливим з меншими обсягами даних, проте, чим більше є даних, тим точнішим буде передбачення.

Таким чином, інтелектуальний аналіз даних є достатньо потужним інструментом для визначення взаємозв'язків та шаблонів в великому масиві даних. Більше того, окрім виявлення нових закономірностей, інтелектуальний

аналіз даних також дає змогу виявляти деякі нюанси та приховані деталі емпіричних досліджень, які неможливо зафіксувати поверхневим спостереженням. Однак, хоча цей метод і видобуває приховану інформацію з масивів даних, він не може виконати оцінку цінності видобутої інформації.

1.4. Методи інтелектуального аналізу даних

Класичні статистичні методи до сих пір знаходять своє застосування в методах інтелектуального аналізу даних. Так, *перевірка статистичної гіпотези* використовується для визначення особливостей функції розподілу генеральної сукупності величин, що знаходяться в великих масивах даних

В той же час, *кореляційно-регресійний аналіз* масивів даних дає можливість визначити функціональні та статистичні взаємозв'язки між даними в масивах даних.

Аналіз даних характеризується комплексним характером пошуку загальних особливостей досліджуваної системи. Основне питання, яке необхідно при цьому розв'язати, є питання про взаємозв'язок ознак.

Функціональний взаємозв'язок – це випадок, коли кожному значенню одного показника відповідає чітко визначене значення іншого. Статистичний взаємозв'язок – це випадок, коли одному значенню однієї величини може відповідати декілька значень іншої.

Асоціація - найбільш відомий і простий метод інтелектуального аналізу даних. В ньому здійснюється зіставлення двох або більше елементів, щоб знайти схожість, або подібність, між об'єктами, або подіями, та визначити ймовірність того, що певні об'єкти, або події, будуть знаходитися, або відбуватимуться разом. Цей метод визначає, ймовірність того, що певні події відбуватимуться разом. Якщо, наприклад, подією є покупка товарів у супермаркеті, то асоціативне правило визначає, чи придбає покупець, з високою долею імовірності, бутербродне масло в тому випадку, коли він купує хліб. Метод формулювання правил асоціацій — це метод машинного навчання,

який базується на правилах, що виявляють зв'язки між змінними (атрибутами) у великих масивах даних.

Класифікація – це операція розбиття множини об'єктів на апріорно задані групи (класи), усередині кожної з яких ці об'єкти мають приблизно однакові властивості та ознаки. Якщо відомі властивості об'єктів певного класу, і новий об'єкт відносять до цього класу, властивості, що характеризують клас, автоматично поширюються і на цей об'єкт.

До основних методів класифікації належать: дерева рішень та алгоритми покриття.

Кластеризація. Кластер – це група об'єктів, які групуються відносно подібності властивостей. Кластеризація полягає у розбитті заданої множини вибірки об'єктів на підмножини (кластери), так, щоб кожен кластер складався з схожих об'єктів, а об'єкти різних кластерів істотно відрізнялися. Кластеризацію також називають сегментацією даних, оскільки великі групи даних діляться за подібністю. Кластерний аналіз використовується в тих випадках, коли необхідно автоматично виділити деякі правила, взаємозв'язки або тенденції у сукупності даних.

Вибір методу кластеризації залежить від властивостей та характеру набору даних.

Існують наступні види кластерного аналізу:

- 1) Кластеризація розбиттям (partions method).
- 2) Ієрархічна кластеризація (hierarchical method).
- 3) Кластеризація на основі густини (density method).
- 4) Кластеризація на основі моделі розподілу (method based on distribution model).

РОЗДІЛ 2. РОЗРОБКА АРХІТЕКТУРИ ПРОГРАМНОГО ЗАСОБУ

2.1. Модель клієнт-серверної архітектури.

Веб-додаток – це програмне забезпечення, яке можна відкрити за допомогою будь-якого браузера. Ключова відмінність веб-додатків від веб-сайтів полягає у тому, що вони відображають не статичні дані в Інтернеті такі, як тексти та зображення, а динамічні програми. Клієнт-серверна_конфігурація - це концепція інформаційної мережі, в якій, з одного боку, клієнти надсилають запити, з іншого боку, сервер відповідає на надіслані запити.

Веб-додаток можна умовно розділити на:

- клієнтську частину,
- серверну частину
- інформаційне наповнення (контент)

Отримання необхідних масивів даних виконується в інформаційних системах, в яких міститься клієнт-серверну архітектуру. Це означає, що програма або сервіс складається з двох програмних частин – клієнтської та серверної.



Рисунок 2.1. Клієнт-серверна архітектура

Серверна частина веб-додатку (Back-end) – це основна частина програми, яка виконується на основі інтернет-запитів користувачів, надісланих через веб-браузери. Серверний компонент складається з двох частин: 1) логіки додатка та 2) бази даних. Логіка додатку – це головний центр керування веб-додатком, бази даних – це місце, де зберігається інформація. Серверний програмний код знаходиться на веб-сервері та відповідає на *HTTP*-запити

клієнтів, створюючи сторінку. Він, також, відповідає за зберігання різного роду даних, зокрема профілі клієнтів. Для написання коду використовують *JavaScript*, *PHP*, *Python*, тощо.

Клієнтська частина веб-додатку (*Front-end*) – це інтерактивна частина програми, що виконується в веб-браузері на різних пристроях - комп'ютері, смартфоні або планшеті користувача. Клієнтський компонент активує інтерфейс користувача веб-додатку і завантажується на пристрої у вигляді динамічних веб-сторінок. Веб-додатки запускаються на будь-яких пристроях та операційних системах, де є інтернет-браузери. Для написання клієнтського коду використовують комбінацію *JavaScript* (головна мова програмування), *CSS* (надає веб-сторінці певний вигляд), *HTML* (відповідає за структурування змісту сторінки). Вони вбудовані у більшість браузерів, тому нема потреби окремо налаштовувати пристрій. Клієнтський код взаємодіє з сервером лише через *HTTP*, й не може безпосередньо отримувати інформацію з сервера.

База даних веб-додатку – це масиви даних, в яких міститься необхідна клієнтам інформація. За запитом користувача серверна частина веб-додатку використовує ці дані для обробки та надає бажаний результат кінцевому клієнту.

Існує декілька видів архітектури додатків:

- дворівнева архітектура".
- багаторівневу архітектуру

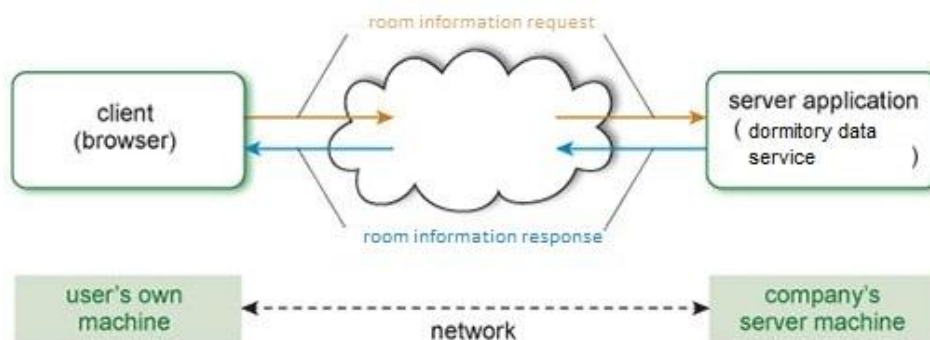


Рисунок 2.2 – Схема клієнт-серверної архітектури.

Такий додаток в комп'ютерній мережі підключає клієнта до сервера. Серверна частина забезпечує можливість підключитися будь-якої кількості клієнтів до сервера і вимагати, щоб він виконував завдання. Це "дворівнева архітектура".

Часто використовують багаторівневу архітектуру, архітектуру яка використовує проміжне програмне забезпечення для обробки запитів даних між користувачем та базою даних (рис. 2.3.)

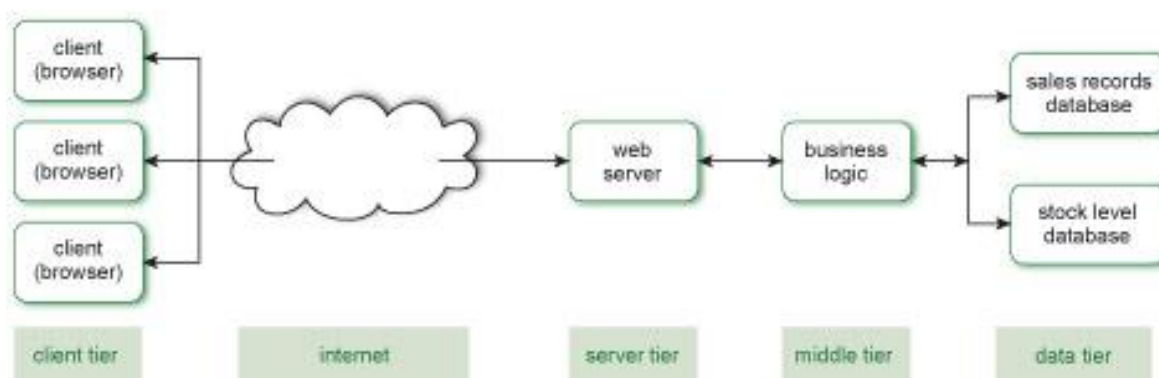


Рисунок 2.3 Схеми багаторівневої архітектури

Багаторівневу архітектуру можна називати трирівневою архітектурою (клієнт, сервер та рівні даних).

У випадку, наприклад, якщо виникла потреба у прогнозі погоди на наступний день, у веб-браузер заносять запит і він завантажує потрібні дані, використовуючи транспортний протокол HTTP. В даному випадку браузер – це клієнтська частина веб-програми, а дані, що генеруються під запит браузера – це відповідь серверної частини веб-програми.

Для досягнення цієї ж цілі можна використати програму у вигляді додатку на смартфоні, який теж є клієнтською частиною. Цей додаток запускається на Android або IOS, після чого програма звертається до сервера. Але цього разу вона запитує не сайт, а лише маленький файл – фрагмент даних у форматі JSON з інформацією про прогноз погоди.

2.2. Стандарти API,

Прикладний Інтерфейс Програмування API. Як у випадку з браузером, так і у випадку з мобільним додатком клієнтська та серверна частина спілкуються між собою за допомогою API (*Application Programming Interface*, Прикладний Інтерфейс Програмування), незважаючи на різні мови програмування, якими виконано клієнтську та серверну частину. Більше того, використовувати API можуть не тільки пара клієнт-сервер, а й, скажімо, два сервери. Наприклад, скориставшись сервісом для придбання авіаквитків (звертаючись до API цього сервісу), можна також отримати інформацію щодо прогнозу погоди в тій місцевості, до якої був придбаний квиток, тобто сервер надіслав запит на адресу іншого сервера, зв'язавшись з API сервісу метеослужби.

Сучасні API часто приймають форму веб-сервісів, які надають користувачам (як людям, так і іншим веб-сервісам) необхідну клієнтам інформацію. Зазвичай ця процедура обміну інформацією і формат передачі даних структуровані, щоб обидві сторони знали, як взаємодіяти між собою.

Найпопулярніший стандартом API є REST API, *Representational State Transfer API*, який ґрунтується на наступних принципах,

- Клієнт-серверна архітектура
- Stateless сервер
- Кешування
- Багатошарова структура
- Єдиний інтерфейс
- Код на вимогу

До стандарту REST API був протокол SOAP, *Service Object Access Protocol*, який містив правила синтаксису для оформлення запитів та відповідей, що надсилаються веб-додатками. Усі програми, які підтримували SOAP, повинні були обмінюватися даними у форматі XML через HTTP або

SMTP. Стандарт REST, на відміну від SOAP, не є протоколом, а є архітектурним стилем.

Веб-API, що відповідає підходам REST-API називають RESTFUL API [3]. RESTFUL API застосовують HTTP-запити для роботи з ресурсами – такі як GET (для отримання інформації) або POST (запит, необхідний для надсилання даних). Також застосовуються запити PUT/PATCH для оновлення або зміни та DELETE для видалення даних.

На відміну від SOAP, який використовує тільки формат XML, система RESTFUL підтримує обмін повідомленнями в різних форматах — звичайний текст, YAML, HTML, XML, JSON.

Також слід відзначити стандарт GraphQL, який дозволяє вказувати точні дані та спрощує отримання та склеювання даних з декількох джерел. Завдяки мові запитів GraphQL розробник може використовувати одне звернення API для отримання всіх необхідних даних.

Методи HTTP запитів.

При зверненні до веб-API використовуються запити HTTP, в яких містяться стандартні методи, які також називаються HTTP дієсловами:

GET — найпопулярніший тип запиту. Використовується для отримання або читання даних. Запити з використанням цього методу можуть тільки отримувати дані.

HEAD — запрошує ресурс так само, як і метод GET, але без тіла відповіді.

PUT — зазвичай використовується для оновлення ресурсу, шляхом заміни даних запиту.

POST — зазвичай використовується для створення нового ресурсу або використовується для відправки сутностей до певного ресурсу.

DELETE — видаляє дані.

OPTIONS — використовується для опису параметрів з'єднання з ресурсом.

RATCH — використовується для часткової зміни ресурсу.

2.3. Погодні API

Whether (погодні) API — це служби, які надають доступ до даних погодних умов і прогнозів погоди в реальному часі. Метеосервіс (або веб-сайт погоди) - це такий тип веб-сайту, який спеціалізується на представленні детальних звітів про погоду. Масиви даних, що знаходяться на веб-сайтах метеосервісів є історичними даними. Всі матеріали, що знаходяться на такого роду сервісі можуть набувати вигляду:

- погодинних, денних, місячних та річних звітів про погоду, яка відбулася в минулому;
- погодинних прогнозів на 7 діб вперед;
- денних узагальнених прогнозів 14 діб вперед;
- коротких прогнозів опадів на 2 години вперед з інтервалами по 15 хвилин;

Кожний онлайн метеосервіс надає послуги доступу до власного API та бази даних звітів погоди та геоположень. Ця послуга може бути безкоштовною, але, у більшості випадків, для того, щоб отримати доступ до API потрібно оформити платну підписку та отримати ліцензійний ключ доступу. Користувач має можливість ввести цей ключ до параметрів запиту та отримати повний список інформації, яка стосується погоди у різних форматах: XML; CSV; JSON. Погодні API надають доступ до даних погодних умов і прогнозів погоди в реальному часі. За допомогою таких API користувачі отримують доступ до поточних умов, погодинних та щоденних прогнозів, а також довгострокових прогнозів за допомогою лише кількох рядків коду. Це дозволяє користувачам створювати багатофункціональні програми, які надають актуальну інформацію про температуру, опади, швидкість і напрям вітру, а також про інші важливі погодні параметри.

Погодні API використовуються в у таких додатках, як, наприклад:

- онлайн-видання новин, де публікуються місцеві звіти про погоду,

- туристичні веб-сайти, які надають інформацію про поточні умови у країнах, до яких їздять туристи,
- системи домашньої автоматизації, які відстежують температурні тенденції для більш ефективного використання енергії,
- додатки для смартфонів, які допомагають користувачам планувати свої заходи на свіжому повітрі відповідно до останнього прогнозу погоди.

Більшість погодних API пропонують повний набір функцій, а також велику гнучкість під час створення інноваційних способів взаємодії з даними про погоду, зібраними постачальником послуг API. Це дозволяє розробникам використовувати ці дані для створення інноваційних веб-додатків та програм погоди для мобільних пристроїв, налаштовувати функціонування своїх програм на основі вподобань користувачів та попередніх моделей їх використання.

Щоб підключити API до проекту чи програми, потрібно мати ключ API для автентифікації запиту розробника. Створення програми в RapidAPI генерує ключ API (X-RapidAPI-Key), специфічний для цієї програми. За допомогою цього ключа надається можливість переглядати аналітичні дані на основі викликів API.

Перелік найбільш популярних Weather API:

World Weather API and Weather Forecast [4]

The Global Weather API & Mapping Platform for Business [5]

WeatherAPI.com [6]

OpenWeather [7]

Open-Meteo [8]

Meteostat Developers [9]

Weather Data & API Global Forecast & History Data [10]

Meteotrek – більше, ніж метеостанція [11]

Формати запису аналізу даних

Набір даних параметрів оточуючого середовища складається з сукупності точок, що мають визначені координати у певному розмірному просторі, де кожна точка є індивідуальною характеристикою системи у певний момент часу. При компіляції сукупності набору даних вимірюють та збирають ряд ознак, тобто даних, які описують властивості системи, яка еволюціонує у часі. Ці ознаки можуть бути

- категоріальними (визначені ознаки без певного порядку),
- порядковими (визначені ознаки, які мають внутрішній порядок),
- числовими (реальні значення параметрів).

Кожна ознака представляє один вимір простору ознак, і конкретне значення ознаки для певної точки даних розміщує точку в визначеному місці в цьому просторі. Сукупність усіх значень всіх ознак точки даних називаються вектором ознак. Чим більше ознак зібрано для набору даних, тим вищою буде розмірність результуючого вектору ознак і простору ознак. Очевидно, що зі збільшенням розмірності візуалізація всіх вимірів простору ознак стає складною для відповідної обробки, і виникає потреба у визначенні необхідних шаблонів для сприйнятливої візуалізації.

Кожен метод інтелектуального аналізу даних характеризується різними значеннями вектору ознак, які описують властивості системи, а значить і певним форматом представлення даних, який є специфічним для кожного окремого методу.

2.4. Перевірка статистичної гіпотези.

Розглядають генеральну сукупність даних параметрів навколишнього середовища та перевіряють гіпотезу про те, чи розподілена ця сукупність за певним за законом Z . Для цього використовують критерій узгодженості Пірсона χ^2 , на підставі якого перевіряється гіпотеза, яка стверджує, що вибірка спостережень параметрів навколишнього середовища має функцію розподілу Z .

Критерій узгодженості Пірсона χ^2 розраховують за формулою:

$$\chi^2 = \sum \frac{(n_i - n'_i)^2}{n'_i}$$

n_i – емпіричні частоти (кількість значень, що попадають в кожен інтервал експериментальної вибірки),

n'_i – теоретичні частоти (кількість значень, що попадають в кожен інтервал за нормальним розподілом).

Будують гістограму емпіричного розподілу величин параметрів навколишнього середовища та графік теоретичного розподілу цих величин (Рис. 2.4.).

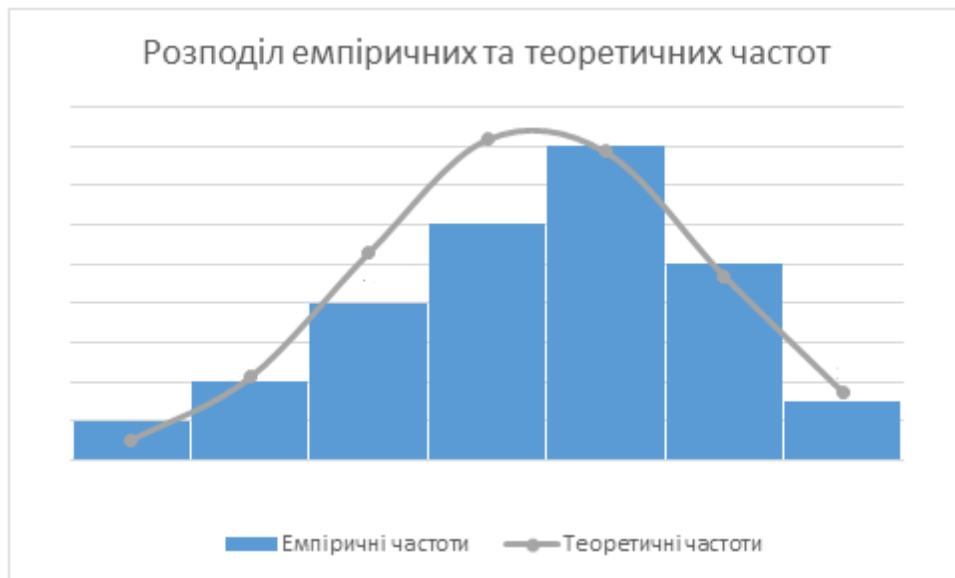


Рисунок 2.4. Графіки емпіричного та теоретичного розподілу величин

Кореляційно-регресійний аналіз масивів даних.

З масивів даних параметрів навколишнього середовища вибирають величини, що характеризують певний параметр оточуючого середовища Y . Цей параметр співставляють з іншими параметрами X_1 , X_2 та X_3 . Виконується кореляційний аналіз визначених параметрів та будуються лінії регресії, що показують співвідношення між цими параметрами (рис. 2.5).



Рисунок 2.5. Кореляції та лінії регресії параметрів навколишнього середовища Y, X1, X2 та X3.

Формулювання асоціативних правил.

Складається послідовність транзакцій та сукупність подій:

- Транзакція 1 – Події ABC
- Транзакція 2 – Події ABD
- Транзакція 3 – Події ABD
- Транзакція 4 – Події BCE
- Транзакція 5 – Події EF
- Транзакція 6 – Події E

На основі частотних характеристик подій у визначеній транзакції будуються графічні співвідношення між подіями у транзакціях (рис. 2.6)

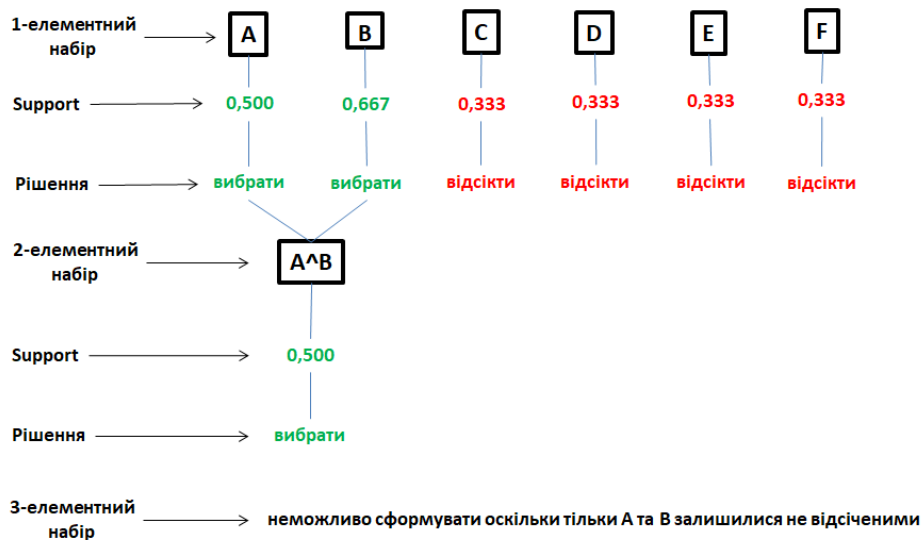


Рисунок 2.6. Співвідношення між подіями у транзакціях

На основі оцінки співвідношення між подіями у транзакціях формулюються правила асоціацій.

Класифікація та дерево рішень.

Формулюється сукупність величин, що описують зв'язки між причинами А, В, С, D та наслідком Е, дані наведені в табл. 2.1.

Таблиця 2.1. Масив даних сукупності величин

№	A	B	C	D	E
1	a1	b1	c1	d2	e2
2	a1	b1	c1	d1	e2
3	a2	b1	c1	d2	e1
4	a3	b2	c1	d2	e1
5	a3	b3	c2	d2	e1
6	a3	b3	c2	d1	e2
7	a2	b1	c2	d1	e2
8	a1	b2	c1	d2	e2
9	a1	b3	c2	d2	e1
10	a3	b2	c2	d2	e1
11	a1	b2	c2	d1	e1
12	a2	b2	c1	d1	e1
13	a2	b1	c2	d2	e1

Будується дерево рішень, яке формулює класифікацію параметрів навколишнього середовища відносно причинно-наслідкових співвідношень (рис. 2.7).

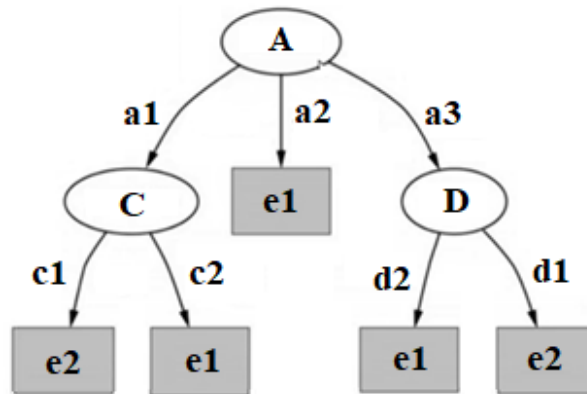


Рисунок 2.7. Дерево рішень

Кластеризація масивів даних.

Виконується кластерний аналіз параметрів навколишнього середовища з метою виявлення центрів тяжіння подій, що характеризуються визначеними ознаками. Використовується асоціативний кластерний аналіз (табл. 2.2, рис.2.8) та кластерний аналіз за методом к – середнього (табл. 2.3, рис.2.9).

Таблиця 2.2. Події та характеристичні ознаки параметрів простору А

Подія	Ознаки	
	Масив X	Масив Y
1	x1	y1
2	x2	y2
3	x3	y3
4	x4	y4
5	x5	y5
6	x6	y6

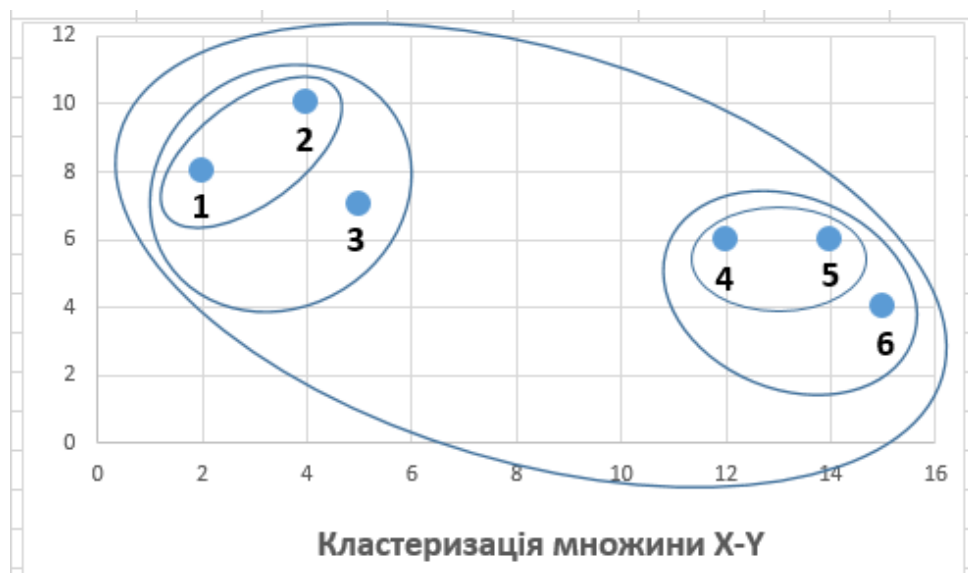


Рисунок 2.8. Асоціативний кластерний аналіз

Таблиця 2.3. Події та характеристичні ознаки параметрів простору В

Подія	Ознаки	
	Масив X	Масв Y
1	x1	y1
2	x2	y2
3	x3	y3
4	x4	y4
5	x5	y5
6	x6	y6
7	x7	y7
8	x8	y8
9	x9	y9
10	x10	y10
11	x11	y11
12	x12	y12
13	x13	y13
14	x14	y14
15	x15	y15

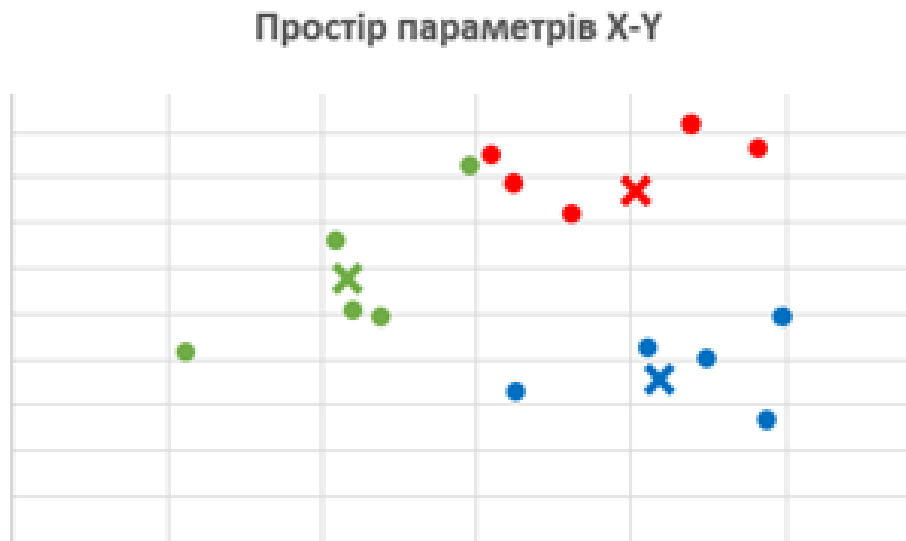


Рисунок 2.9. Кластерний аналіз за методом к-середнього

2.5. Моделі оцінки якості веб-додатку

Існуючі моделі оцінки якості програмного продукту є ієрархічними моделями. Так, модель *FURPS* характеризується наявністю двох шарів показників якості [9]. Перший шар визначає основні якісні характеристики показників, у другому шарі представлені атрибути, що пов'язані з цими характеристиками. Модель *FURPS* отримала свою назву за першими літерами основних категорій показників якості програмного забезпечення:

- *Functionality* – функціональність;
- *Usability* – практичність;
- *Reliability* – надійність;
- *Performance* – продуктивність;
- *Supportability* – експлуатаційна придатність.

Оцінки якості веб-додатку

Модель якості веб-додатку регламентується стандартом *ISO/IEC 25010:2011* та включає вісім показників: 1) функціональну придатність, 2) ефективність функціонування, 3) сумісність, 4) зручність використання

(практичність), 5) надійність, 6) захищеність, 7) супроводжуваність, 8) мобільність.

Підрозділи показників та метрики оцінки якості веб-додатку наведені в таблиці 2.4, де $X = A/B$ – значення метрики, A – виміряне значення якості, яка оцінюється; B – базове, визначене в специфікації, значення якості.

Таблиця 2.4. Показники та метрики оцінки якості веб-додатку

Назва підрозділу характеристики	Назва метрики	Вихідні дані для обчислення метрики
1) Функціональна придатність (functional suitability)		
Функціональна повнота (functional completeness)	Повнота реалізації веб-сторінок за критерієм коректності WWW Consortium	A – кількість сторінок HTML, реалізованих за критерієм WWW Consortium; B – кількість реалізованих сторінок HTML
	Повнота реалізації веб-сторінок	A – кількість реалізованих сторінок HTML; B – кількість сторінок HTML, задана у специфікації
Функціональна відповідність (functional appropriateness)	Відповідність цілям користувачів	A – кількість досягнутих цілей користувачів, пов'язаних з використанням клієнтської частини веб-додатку; B – загальна кількість цілей користувачів, пов'язаних із використанням клієнтської частини веб-додатку
	Відповідність завданням користувачів	A – кількість користувачів, які успішно виконали свої завдання за допомогою веб-програми; B – загальна кількість користувачів веб-програми
Сумісність (compatibility)		
Здатність до взаємодії (Interoperability)	Взаємодія з браузерами	A – кількість підтримуваних браузерів; B – загальна кількість різних браузерів, задана у специфікації
	Взаємодія з мобільними пристроями	A – кількість типів мобільних пристроїв, що підтримуються; B – загальна кількість типів мобільних пристроїв, задана у специфікації
	Адаптованість до роздільної здатності екрана користувача	A – кількість дозволів екрана, що підтримуються; B – загальна кількість дозволів екрану, задана у специфікації

Продовження таблиці 2.4.

Використання (usability)		
Зворотній зв'язок (feedback)	Відображення повідомлень	<i>A</i> – кількість операцій, для яких використовується область для відображення повідомлень із заданих у специфікації; <i>B</i> – загальна кількість операцій, для яких має використовуватися область для відображення повідомлень, задану у специфікації
	Відображення прогресу операцій	<i>A</i> – кількість функцій, котрим використовується відображення прогресу операцій; <i>B</i> – загальна кількість функцій, заданих у специфікації, для яких має бути реалізовано відображення прогресу операцій
	Моніторинг операцій	<i>A</i> – кількість функцій, котрим реалізовано моніторинг виконання; <i>B</i> – кількість функцій, для яких потрібний моніторинг виконання відповідно до специфікації
Простота використання (operability)	Потреба у довідці	<i>A</i> – кількість функцій, які можуть бути виконані без звернення до довідки, документації чи демонстрації; <i>B</i> – загальна кількість функцій, задана у специфікації
	Перевірка введення	<i>A</i> – кількість полів введення, для яких реалізовано перевірку вхідних даних; <i>B</i> – загальна кількість полів введення, для яких має бути реалізована перевірка вхідних даних
	Зрозумілість повідомлень	<i>A</i> – кількість повідомлень, зрозумілих користувачеві; <i>B</i> – загальна кількість реалізованих повідомлень
	Очевидність веб-сторінок	<i>A</i> – кількість веб-сторінок, призначення яких очевидне користувачеві; <i>B</i> – загальна кількість реалізованих веб-сторінок
Естетичність інтерфейсу користувача (user interface aesthetics)	Ступінь привабливості інтерфейсу користувача	<i>X</i> – ступінь привабливості інтерфейсу користувача на інтервалі $0 \leq X \leq 1$, яка визначається методом соціологічного опитування, де 1 – найкраща оцінка, 0 – найгірша

Продовження таблиці 2.4.

Супроводжуваність		
Модульність (modularity)	Реалізація відповідно до підходу MVC	1 – реалізація відповідно до підходу MVC; 0 – інакше
Повторне використання (reusability)	Використання CSS-основи (CSS framework)	1 – використання у веб-додатку CSS-основи; 0 – інакше
	Використання JavaScript-основи (JavaScript framework)	1 – використання у веб-додатку JavaScript основи; 0 – інакше
Модифікованість (modifiability)	Використання стилів у CSS-файлах	1 – вказівка стилів відображення лише в окремих CSS-файлах; 0 – інакше
	Використання JavaScript-коду в окремих файлах	1 – зберігання JavaScript коду тільки в окремих файлах; 0 – інакше
Надійність		
Завершеність (maturity)	Відповідність специфікації WWW Consortium	<i>A</i> – кількість помилок під час перевірки коректності всіх веб-сторінок за критерієм WWW Consortium; <i>B</i> – базова середня кількість помилок при перевірці коректності всіх веб-сторінок за критерієм WWW Consortium

РОЗДІЛ 3. РОЗРОБКА ПРОГРАМНОГО ЗАСОБУ

3.1. Розробка основних модулів системи

На даний час термін BigData (Великі Масиви Даних, ВМД) стає дуже популярним брендом, який активно використовують учасники, що працюють на IT ринку. Крім чистого маркетингу ВМД мають величезний вплив на всю IT індустрію. Суть у тому, що вартість зберігання та транзакцій великих обсягів інформації стрімко зменшується. Це відкриває реалізацію можливостей для всіх бажаючих. Кілька років тому вартість розробки та підтримки системи з терабайтовим обсягом даних та навантаженням тисяча транзакцій за секунду була непідйомною навіть для середньої професійної компанії. Проте тепер, стрімкий розвиток обчислювальної техніки та IT забезпечив широкий доступ до ВМД та операцій з ними.

Важливу цінність мають масиви даних, які стосуються інформації про параметри навколишнього середовища. Основним фактором, що істотним чином впливає на ці параметри є погодні умови. Саме вони формують оптимальну ефективність генеруючих засобів відновлювальної енергетики, продуктивність сільськогосподарської діяльності та екологічний профіль навколишнього середовища

Масиви даних про погодні умови отримуються за допомогою активної діяльності великої кількості державних та приватних професійних погодних станцій, яких на сьогоднішній день налічується понад 40 тисяч [23]. Більшість із цих станцій встановлено в аеропортах та великих містах світу. Важливими є також дані, що отримуються від чисельних портативних локальних метеостанцій. Ці станції характеризуються високим технічним рівнем та точністю отримуваної інформації. Коли такі метеостанції підключити до глобальної мережі моніторингу погодних умов, вони будуть вносити додаткові масиви даних, на основі яких будуть виконуватися уточнення обчислювальних моделей, які використовуються для прогнозування погодних умов на найближчі проміжки часу. Портативна локальна метеостанція здійснює моніторинг параметрів навколишнього середовища у визначеній місцевості, тому дані, які

надаються цією метеостанцією, будуть вносити уточнення в прогноз погоди для тієї місцевості, де така станція розміщена.

Таким чином, потрібно розробити програмний засіб, який дозволить здійснювати збір, зберігання, та обробку масивів даних, що знаходяться на веб-сайтах метеослужб та надходять з локальної метеостанції, щоб виявити в них інформацію, яка описує динамічну еволюцію погодних умов у визначеній місцевості. Розробка моделей, що описують еволюцію динамічних систем, є важливою науковою проблемою, яка потребує активних зусиль [24]. Тому розробка веб-додатку, який оперує з масивами даних параметрів оточуючого середовища, має наукове та практичне значення.

3.2. Алгоритм роботи програмного засобу

Програмний засіб матиме наступну конфігурацію (рис. 3.1).

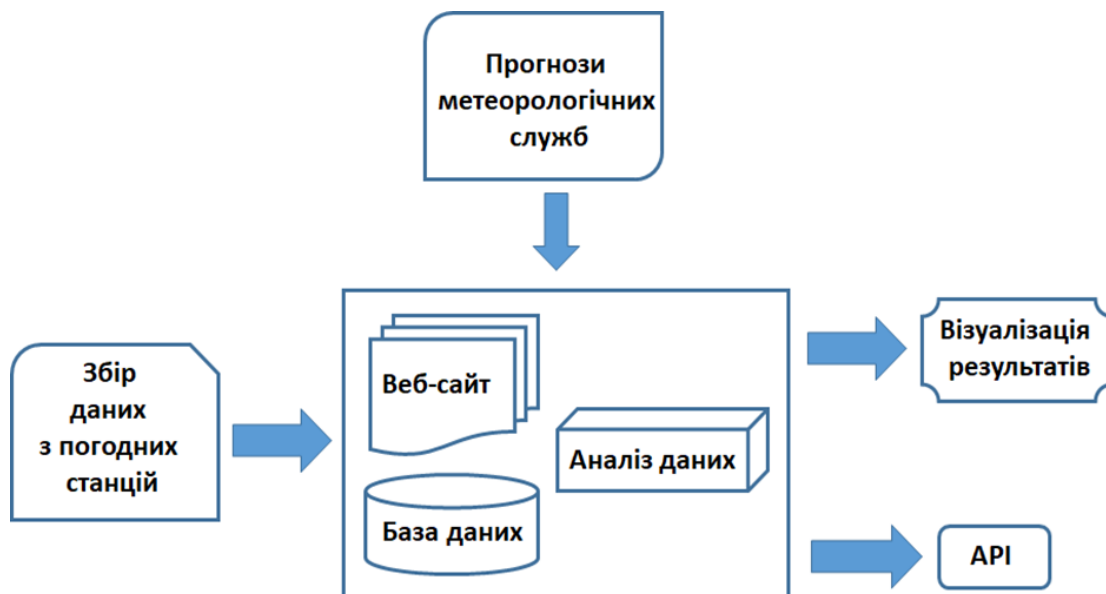


Рисунок 3.1. Загальна схема програмного засобу

Алгоритм роботи програмного продукту:

1. На вхід надходять:

- дані від професійних та аматорських погодних станцій;

- дані в яких представлені прогнози погоди від метеорологічних служб.

2. Отримані дані будуть:

- накопичуватися та зберігатися в базі даних;
- оброблятися за допомогою методів інтелектуального аналізу.

3. На вихід надходитимуть:

- візуалізовані шаблони, що характеризують кореляції, тренди та асоціації між параметрами навколишнього середовища;
- API до всіх даних про погоду, включаючи візуалізовані шаблони.

Ці масиви даних будуть накопичуватися та зберігатися в базі даних, з подальшою обробкою та аналізом. На виході будуть отримуватися результати обробки вхідних масивів даних та забезпечуватися доступ до цих результатів за допомогою API.

3.3. Модуль моніторингу параметрів

Масиви даних отримуються з модулю моніторингу параметрів навколишнього середовища та веб-сайтів метеосервісів.

Портативний електронний модуль моніторингу параметрів навколишнього середовища складається з мікроконтролера та сенсорів, які збирають необхідні дані та передають їх у пам'ять мікроконтролера (рис. 3.2)

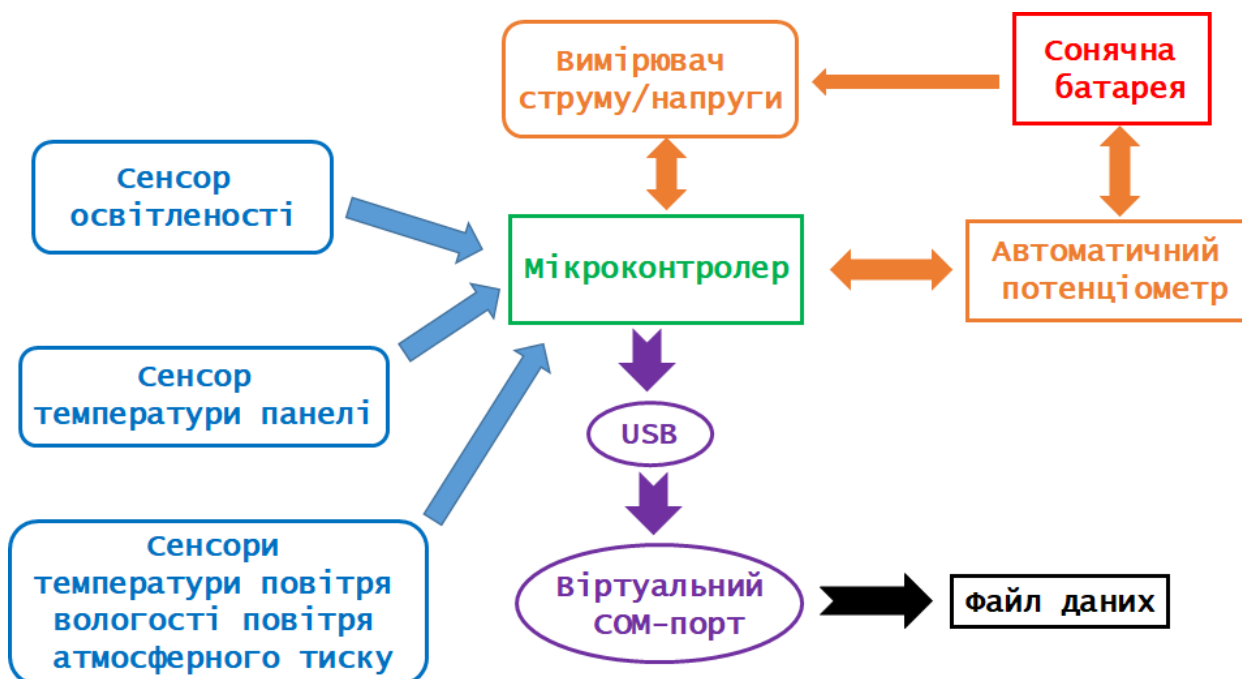


Рисунок 3.2- Схема модулю моніторингу параметрів навколишнього середовища

Перелік компонентів модулю наведений в Додатку 1, програмний код управління мікроконтролером ARDUINO в Додатку 2.

В мікроконтролері, який через USB-порт під'єднаний до комп'ютера, формується рядок даних. Цей рядок направляєтся у віртуальний СОМ-порт, через який переноситься на диск комп'ютера у вигляді набору даних (рис. 3.3).

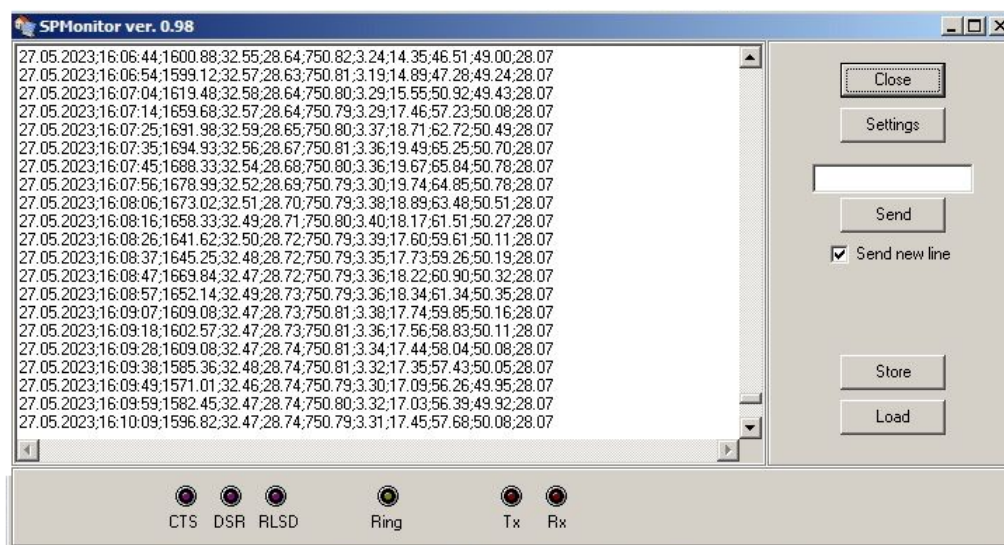


Рисунок 3.3- Віртуальний СОМ-порт

Дані з веб-сайтів метеосервісу приймаються та передаються у форматі JSON, текстового формату обміну даними для передавання структурованої інформації через мережу. Запити з відповідних сайтів надходять у вигляді, який зображено на рис. 3.4

JSON Return Object

On success a JSON object will be returned.

```
"latitude": 52.52,  
"longitude": 13.419,  
"generationtime_ms": 2.2119,  
"timezone": "Europe/Berlin",  
"timezone_abbreviation": "CEST",  
"hourly": {  
  "time": ["2022-07-01T00:00", "2022-07-01T01:00", "2022-07-01T02:00", ...],  
  "temperature_2m": [13, 12.7, 12.7, 12.5, 12.5, 12.8, 13, 12.9, 13.3, ...]  
},  
"hourly_units": {  
  "temperature_2m": "°C"  
},
```

Рисунок 3.4.-Програмний код модуля Дані у форматі JSON

3.4. Аналіз даних

Програмна частина веб-сайту «Аналіз даних» виконує попередню обробку отриманих даних, яка включає в себе визначення сукупності значень усіх наявних ознак кожної точки даних в масиві даних. Кожна ознака виражає певне значення простору ознак, яке розміщує точку даних в визначеному місці такого простору. За допомогою сукупності ознак описується динамічна еволюція погодних умов за певний проміжок часу. На підставі величин векторів простору ознак здійснюють інтелектуальний аналіз даних з наступною візуалізацією отриманих результатів.

1. Виконують попередню обробку отриманих даних, яка включає в себе:

- перевірку статистичних гіпотез та кореляційно-регресійний аналіз безпосередньо отриманих величин;
- визначення векторів ознак, які представляють собою сукупність усіх значень усіх наявних ознак точки даних. Кожна ознака представляє один вимір простору ознак, і конкретне значення ознаки для певної точки даних розміщує точку в визначеному місці в цьому просторі. Ознаки -це дані, які описують властивості системи, що еволюціонує у часі.

2. На підставі величин векторів ознак здійснюють інтелектуальний аналіз даних з наступною візуалізацією отриманих результатів. Методи інтелектуального аналізу даних включають в себе:

- Кластерний аналіз масивів даних;
- правила класифікації та дерево рішень;
- формулювання асоціативних правил.

Формат JSON дозволяє структурувати дані у вигляді об'єктів та масивів, що дуже зручно для подальшого опрацювання програмами. Формат JSON (JavaScript Object Notation) є легким, текстовим форматом обміну даними, який використовується для передачі структурованої інформації між різними програмами. Він часто використовується для передачі даних між веб-серверами та клієнтськими додатками.

Зазвичай, відповідь про погоду в JSON форматі містить багато полів з різною інформацією. Наприклад, це може бути інформація про температуру, вологість, тиск, швидкість вітру, хмарність тощо. Кожне поле має свою назву і значення, яке може бути числом, рядком, булевим значенням або навіть вкладеним об'єктом чи масивом. Запити з відповідного веб-сайту надходять у вигляді, який зображено на рис. 3.5.

```
{
  "coord": {
    "lon": -0.13,
    "lat": 51.51
  },
  "weather": [
    {
      "id": 800,
      "main": "Clear",
      "description": "clear sky",
      "icon": "01d"
    }
  ],
  "main": {
    "temp": 25.5,
    "feels_like": 27.3,
    "humidity": 60
  },
  "wind": {
    "speed": 3.1,
    "deg": 210
  },
  "clouds": {
    "all": 0
  }
}
```

Рисунок 3.5. Дані у форматі JSON через запит на веб-сайт.

У цьому прикладі ми маємо кореневий об'єкт, який містить різні поля про погоду. Наприклад, поле "coord" містить координати місцезнаходження, поле "weather" містить масив з інформацією про погоду, поле "main" містить дані про температуру та вологість, поле "wind" містить дані про швидкість вітру та його напрямок. Поле "clouds" містить інформацію про хмарність. Крім тих полів, які наведені в цьому прикладі, відповідь може містити інші поля, такі як дата та час, індекс UV, опади тощо. Все залежить від конкретного API та його можливостей. Оскільки формат JSON є широко використовуваним та має простий синтаксис, його легко опрацьовувати за допомогою різних мов програмування, включаючи JavaScript. Таким чином за допомогою JavaScript ми отримуємо значення полів з об'єкту JSON та використовуємо їх для відображення інформації на вашому веб-сайті.

Функціонал отримання погодних даних для певного міста реалізовано за допомогою коду, що створюватиме веб-сторінку із необхідним інструментарієм (рис. 3.5).

```

<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <title>Погода</title>
  </head>
  <body>
    <h1>Погода</h1>
    <p>Введіть місто:</p>
    <input type="text" id="city">
    <button onclick="getWeatherData()">Отримати погоду</button>
    <p>Температура: <span id="temperature"></span></p>
    <p>Хмарність: <span id="cloudiness"></span></p>

    <script>
      function getWeatherData() {
        const city = document.getElementById('city').value;
        const date = new Date();
        date.setDate(date.getDate() - 5);
        const timestamp = Math.floor(date.getTime() / 1000);
        const url = `https://api.openweathermap.org/data/2.5/forecast?q=${city}&units=metric&appid=YOUR_API_KEY&dt=${timestamp}`;
        fetch(url)
          .then(response => response.json())
          .then(data => {
            const temperature = data.list[0].main.temp;
            const cloudiness = data.list[0].weather[0].description;
            document.getElementById('temperature').textContent = temperature + "°C";
            document.getElementById('cloudiness').textContent = cloudiness;
          })
          .catch(error => console.error(error));
      }
    </script>
  </body>
</html>

```

Рисунок 3.6. Код на javascript для отримання погодних даних

Користувач може ввести назву міста у поле введення з ідентифікатором `id="city"`, після чого, натиснувши кнопку "Отримати погоду", запускається функція `getWeatherData()`, яка створює запит до API використовуваного нами сервісу Openweathermap для отримання даних про погоду. У функції `getWeatherData` використовується JavaScript для отримання значення введеного міста та створення URL-адреси для запиту до API OpenWeatherMap. Крім того, в коді також використовується JavaScript для отримання дати 5 днів назад та перетворення її в формат Unix-часу. Ця дата використовується в URL-адресі для отримання погодних даних за цей період. Після отримання відповіді від API, дані про погоду (температура та хмарність) витягуються з відповіді та

виводяться на веб-сторінку за допомогою методу `getElementById`. Температура виводиться в градусах Цельсія, а хмарність виводиться у вигляді текстового опису.

Запит у вікні веб-додатку оформлюється з параметрами в поточний момент часу (Рис. 3.7).

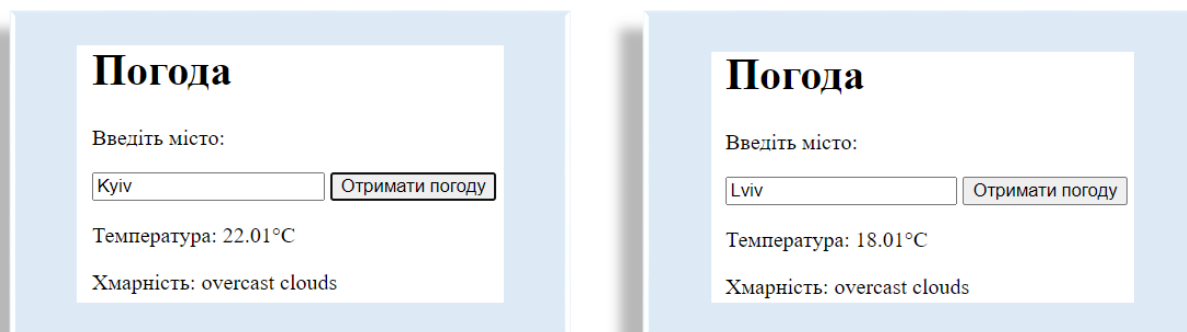


Рисунок 3.7- Вивід запити про параметри на поточний момент.

Також вивід запити можна виконати в форматі історичних даних (Рис. 3.8).

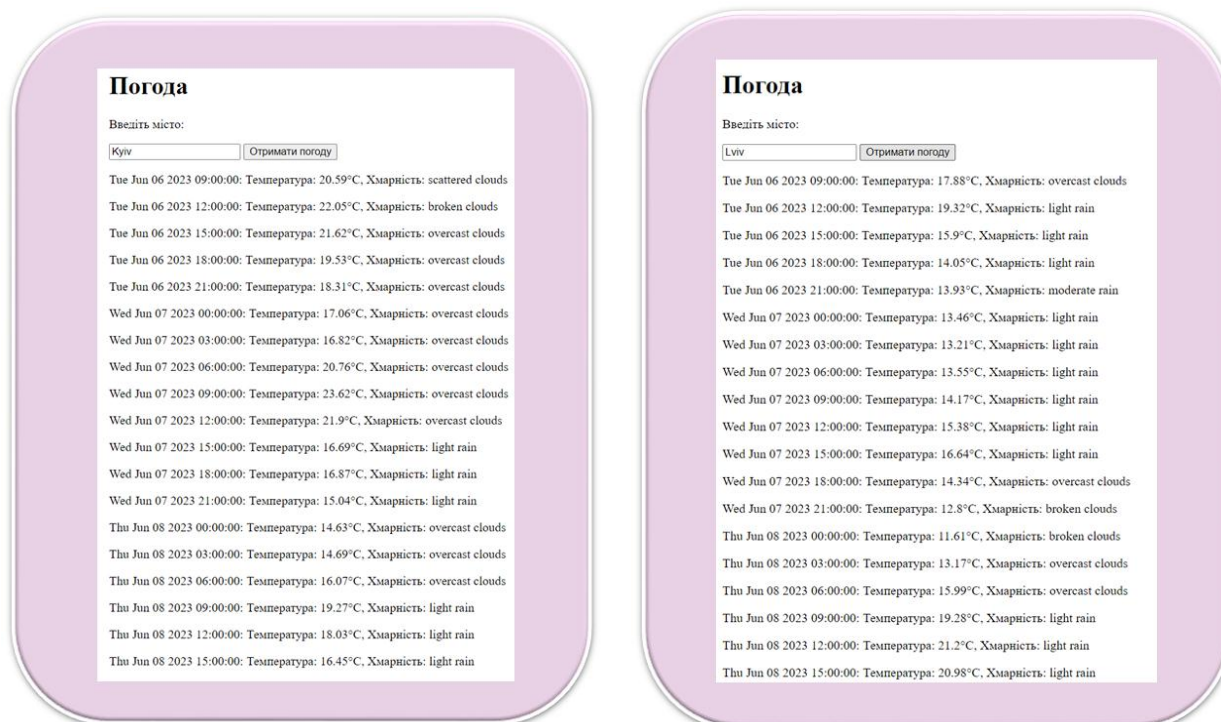


Рисунок 3.8. Вивід запити у вигляді історичних даних.

Використовуючи програмний код як основу та додатково розробляючи його функціонал, можна створити повноцінний додаток, який дозволить користувачам отримувати актуальну інформацію про погоду та прогнози погоди для обраних місць у зручному веб-інтерфейс.

ВИСНОВКИ

В дипломній бакалаврській роботі розглядаються питання, які стосуються методів збору та обробки інформації про параметри навколишнього середовища. Основним фактором, що істотним чином впливає на ці параметри, є погодні умови, які формують оптимальну ефективність генеруючих засобів відновлювальної енергетики, продуктивність сільськогосподарської діяльності та екологічний профіль навколишнього середовища.

В більшості випадків такі дані є історичними даними, в яких інформація, що описує стан якоїсь системи, накопичується за визначений проміжок часу. Використовуючи інтелектуальний аналіз можна визначати закономірності в історичних даних та передбачувати майбутній стан системи.

Розробка програмного продукту, в якому використовуються великі масиви даних, вимагає реалізацію проблеми за допомогою циклу ППДАВ – Проблема, План, Дані Аналіз, Висновки; а використання специфічних статистичних інструментів потребує компонентів інтегрального масиву.

Веб-структурований програмний продукт дозволяє здійснювати збір, зберігання та обробку масивів даних, що знаходяться на веб-сайтах метеослужб та надходять з локальної метеостанції через засоби моніторингу параметрів навколишнього середовища. Це дасть змогу виявити в них інформацію, яка описує динамічну еволюцію погодних умов у визначеній місцевості.

Використовуючи формат JSON ми отримуємо структуровані дані у вигляді об'єктів та масивів для опрацювання їх при передачі між веб-серверами та клієнтськими додатками. Це інформація про температуру, вологість, тиск, швидкість вітру, хмарність тощо. За допомогою мови JavaScript, HTML розробили фрагменти кодів для відображення інформації на веб-сайті.

Використовуючи програмний код як основу та додатково розробляючи його функціонал, можна створити повноцінний додаток, який дозволить користувачам отримувати актуальну інформацію про погоду та прогнози погоди для обраних місць у зручному веб-інтерфейсі.

Розробка моделей, що описують еволюцію динамічних систем, є важливою науковою проблемою, яка потребує активних зусиль. Тому веб-структурований програмний засіб, який оперує з масивами даних параметрів оточуючого середовища, має наукове та практичне значення.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. R.J. MacKay, R.W. Oldford /Scientific Method, Statistical Method and the Speed of Light/ Statistical Science, 2000, V.15(3), P.254-278.
2. H. Jiawei, M. Kamber, J. Pei / Data Mining: Concepts and Techniques, 2011, 3rd edition, Amsterdam: Elsevier, 795 pages.
3. Client-Server Model [Електронний ресурс] – Режим доступу до ресурсу:
4. <https://www.geeksforgeeks.org/client-server-model/>
5. The Single Page Interface Manifesto[Електронний ресурс] – Режим доступу до ресурсу: https://itsnat.sourceforge.net/php/spim/spi_manifesto_en.php
6. Single-page application vs. multiple-page application[Електронний ресурс] – Режим доступу до ресурсу: <https://medium.com/@NeotericEU/single-page-application-vs-multiple-page-application-2591588efe58>
7. Progressive Web Apps[Електронний ресурс] – Режим доступу до ресурсу:
8. <https://web.dev/progressive-web-apps/>
9. S. Purewal Learning Web App Development O'Reilly Media 303 pages
10. Seffah A., Donyaee M., Kline R.B., Padda H.K. Usability Measurement and Metrics: A consolidated model. Software Quality Journal, 2006, vol. 14, iss. 2, pp. 159–178.
11. Grady R.B., Caswell D.L. Software Metrics: Establishing a Company-Wide Program. Prentice-Hall, 1987, 275 p
12. Hyatt L.E., Rosenberg L.H. A Software Quality Model and Metrics for Identifying Project Risks and Assessing Software Quality. In: Proceedings of Product Assurance Symposium and Software Product Assurance Workshop. Noordwijk, 1996, pp. 209–212.
13. ISO/IEC 9126-1:2001. Software Engineering – Software Product Quality – Part 1: Quality Model.
14. ISO/IEC 25010:2011 (Systems and Software Engineering – Systems and Software Quality Requirements and Evaluation (SQuaRE) – System and Software Quality Models).

15. А. О. Олійник, С. О. Субботін, Інтелектуальний аналіз даних : навчальний посібник Запоріжжя : ЗНТУ, 2012. – 278 с.
16. B.Jin, S.Sahni, A.Shevat Designing Web APIs O'Reilly Media, Inc, 2018 p.
17. World Weather API and Weather Forecast [Електронний ресурс] – Режим доступу до ресурсу:<https://www.worldweatheronline.com/>
18. The Global Weather API & Mapping Platform for Business [Електронний ресурс] – Режим доступу до ресурсу:<https://www.aerisweather.com/>
19. JSON and XML Weather API and Geolocation Developer API [Електронний ресурс] – Режим доступу до ресурсу:<https://www.weatherapi.com/>
20. OpenWeather [Електронний ресурс] – Режим доступу до ресурсу: <https://openweathermap.org/>
21. Free Weather API [Електронний ресурс] – Режим доступу до ресурсу: <https://open-meteo.com/>
22. Meteostat Developers [Електронний ресурс] – Режим доступу до ресурсу: <https://dev.meteostat.net/>
23. Weather Data & API Global Forecast & History Data[Електронний ресурс] – Режим доступу до ресурсу: <https://www.visualcrossing.com/>
24. Meteotrek - більше, ніж метеостанція [Електронний ресурс] – Режим доступу до ресурсу:<https://www.meteotrek.ua/>
25. World Meteorological Organization [Електронний ресурс] – Режим доступу до ресурсу:<https://public.wmo.int/en>
26. S.Badillo, et al./An Introduction to Machine Learning/CLINICAL PHARMACOLOGY & THERAPEUTICS, 2020, V.107(4), P.871-885.
27. Міхалевський В. Ц. Розвиток інформаційного середовища для формування інтелектуальної надбудови інформаційного суспільства / В. Ц. Міхалевський, Г. І. Міхалевська // Вісник ХНУ. Технічні науки. 2020. № 1. С. 163-170.
28. Kravchenko M. S., Astistova T. I. ,Data analysis in systems with alternative energy sources / M. S. Kravchenko, M. S. Nuzhdyak ; scientific adviser T. I. Astistova // Інноватика в освіті, науці та бізнесі: виклики та можливості :

- матеріали III Всеукраїнської конференції здобувачів вищої освіти і молодих учених, м. Київ, 17 листопада 2022 року. – Т. 1. – Київ : КНУТД, 2022. С. 81-84. <https://er.knutd.edu.ua/handle/123456789/2276012>
29. Астісова Т.І., Кравченко М.С., Комп'ютеризована інформаційна система для збору та обробки даних при оптимізації функціонування відновлювальних джерел енергії // II Міжнародна науково-практична Інтернет конференція молодих учених та студентів «Електромеханічні, інформаційні системи та нанотехнології». - КНУТД. – Київ, 2023. – С. 24-25. <https://er.knutd.edu.ua/handle/123456789/23499>.
30. Астісова Т.І., Кравченко М.С., Манойлов В.Е., Розподілені системи дистанційного контролю та моніторингу навколишнього середовища // (II Міжнародна науково-практична Інтернет конференція молодих учених та студентів «Електромеханічні, інформаційні системи та нанотехнології» .- КНУТД. – Київ, 2023. – С.99 <https://er.knutd.edu.ua/handle/123456789/25654>.