

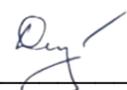
**Київський столичний університет імені Бориса Грінченка**  
**Факультет інформаційних технологій та математики**  
**Кафедра комп'ютерних наук**

«Допущено до захисту»  
Завідувач кафедри комп'ютерних наук,  
Доктор технічних наук, професор  
\_\_\_\_\_ Андрій БОНДАРЧУК  
(підпис)  
«\_\_» \_\_\_\_\_ 2025 р.

**КВАЛІФІКАЦІЙНА РОБОТА**  
**на здобуття освітнього ступеня «Магістр»**  
**Спеціальність 122 Комп'ютерні науки**  
**Освітня програма 122.00.02 Інформаційно-аналітичні системи**

**Тема роботи: Розробка системи моніторингу радіаційного фону з використанням IoT та методів візуалізації даних**

**Виконав**  
студент групи ІАСм-1-24-1.4д  
Дяків Віталій Юрійович  
(ПІБ)

  
\_\_\_\_\_  
(підпис)

**Науковий керівник**  
кандидат технічних наук, доцент  
\_\_\_\_\_ Тетяна НОСЕНКО  
(підпис)

**Київ - 2025**

Київський столичний університет імені Бориса Грінченка  
Факультет інформаційних технологій та математики  
Кафедра комп'ютерних наук

**«Затверджую»**

Завідувач кафедри комп'ютерних наук,  
кандидат технічних наук, доцент  
\_\_\_\_\_ Ірина МАШКІНА

«\_\_\_» \_\_\_\_\_ 2024р.

**ЗАВДАННЯ НА КВАЛІФІКАЦІНУ РОБОТУ МАГІСТРА**

**«Розробка системи моніторингу радіаційного фону з використанням IoT та методів візуалізації даних»**

Виконавець – студент групи ІАСм-1-24-14д

**Дяків Віталій Юрійович**

Вихідні дані: *Наукові публікації, технічна документація, стандарти з моніторингу радіаційного фону, IoT, візуалізації геопросторових і часових рядів даних, а також існуючі аналоги систем моніторингу, такі як Safecast і SaveEcoBot.*

Основні завдання: *Здійснити аналіз існуючих методів та систем моніторингу радіаційного фону. Обґрунтувати вибір типу датчиків радіації та комунікаційних технологій для розробки системи. Розробити архітектуру системи моніторингу. Розробити математичне та алгоритмічне забезпечення, включаючи алгоритми калібрування, фільтрації шумів та метод виявлення аномалій. Реалізувати програмне забезпечення для емуляції сенсорної мережі, збору, обробки, збереження та аналізу даних. Створити візуалізацію динаміки показників та гістограму розподілу за допомогою Plotly. Провести тестування розроблених алгоритмів обробки даних на модельованих сценаріях та оцінити їх ефективність. Скласти висновки та оформити роботу відповідно до затверджених вимог. Підготувати презентацію за темою роботи.*

Пояснювальна записка: *Обсяг – до 60 стор. формату А4 комп'ютерного набору з дотриманням вимог стандарту і методичних рекомендацій кафедри.*

Графічні матеріали: *презентація.*

Додатки: *відсутні.*

Апробація роботи:

Дяків, В. (2025). РОЗРОБКА СИСТЕМИ МОНІТОРИНГУ РАДІАЦІЙНОГО ФОНУ З ВИКОРИСТАННЯМ ІoT ТА МЕТОДІВ ВІЗУАЛІЗАЦІЇ ДАНИХ. *Збірник матеріалів Всеукраїнської конференції молодих учених "Інформаційні технології"* (ISSN: 2664-2638), 9(9), 13–15.

Строк подання роботи на кафедру: «\_6\_» грудня 2025 р.

Науковий керівник

д.т.н., професор

\_\_\_\_\_ Тетяна НОСЕНКО

дата

Виконавець:



\_\_\_\_\_ Дяків В.Ю.

дата

## ЗМІСТ

<b>АНОТАЦІЯ</b>	<b>5</b>
<b>ВСТУП</b>	<b>7</b>
<b>РОЗДІЛ 1 АНАЛІЗ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ МОНІТОРИНГУ РАДІАЦІЙНОГО СТАНУ</b>	<b>10</b>
1.1 ПРОБЛЕМАТИКА РАДІАЦІЙНОГО КОНТРОЛЮ: ПАРАМЕТРИ МОНІТОРИНГУ ТА ВИМОГИ ДО СИСТЕМ	10
1.2 АРХІТЕКТУРНІ ПІДХОДИ ДО ПОБУДОВИ ІОТ-СИСТЕМ ЗБОРУ ЕКОЛОГІЧНИХ ДАНИХ	11
1.3 ОГЛЯД МЕТОДІВ ОБРОБКИ ТА ВІЗУАЛІЗАЦІЇ ГЕОПРОСТОРОВИХ І ЧАСОВИХ РЯДІВ ДАНИХ	15
1.4 АНАЛІЗ ІСНУЮЧИХ АНАЛОГІВ ТА ОБҐРУНТУВАННЯ НЕОБХІДНОСТІ РОЗРОБКИ ВЛАСНОЇ СИСТЕМИ	16
Висновки до розділу 1	22
<b>РОЗДІЛ 2. ПРОЕКТУВАННЯ АРХІТЕКТУРИ ТА АЛГОРИТМІЧНОГО ЗАБЕЗПЕЧЕННЯ СИСТЕМИ</b>	<b>23</b>
2.1. ОБҐРУНТУВАННЯ АПАРАТНО-ПРОГРАМНОЇ ПЛАТФОРМИ (ІОТ-РІВЕНЬ)	23
2.1.1. <i>Вибір сенсорів та мікроконтролерів для реєстрації іонізуючого випромінювання</i>	23
2.1.2. <i>Протоколи передачі даних в умовах розподіленої мережі</i>	27
2.2. РОЗРОБКА ЗАГАЛЬНОЇ АРХІТЕКТУРИ СИСТЕМИ (EDGE, FOG, CLOUD ШАРИ)	28
2.3. МАТЕМАТИЧНЕ ТА АЛГОРИТМІЧНЕ ЗАБЕЗПЕЧЕННЯ	29
2.3.1. <i>Алгоритми попередньої обробки сигналу: фільтрація шумів та агрегація</i>	29
2.3.2. <i>Метод виявлення аномалій у потоці радіаційних даних</i>	30
2.4. РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ МОДЕЛІ	30
2.4.1. <i>Алгоритми обробки даних датчиків, включаючи агрегацію, фільтрацію та калібрування</i>	30
Висновки до розділу 2	32
<b>РОЗДІЛ 3. ПРОГРАМНА РЕАЛІЗАЦІЯ, АНАЛІЗ ТА ВІЗУАЛІЗАЦІЯ ДАНИХ</b>	<b>33</b>
3.1. РЕАЛІЗАЦІЯ ЕМУЛЯТОРА СЕНСОРНОЇ МЕРЕЖІ (DATA PRODUCER)	33
3.2. РОЗРОБКА СЕРВЕРНОЇ ЧАСТИНИ ДЛЯ ЗБОРУ ТА ЗБЕРЕЖЕННЯ ДАНИХ (BACKEND)	35
3.3. ПРОГРАМНА РЕАЛІЗАЦІЯ ВЕБ-ІНТЕРФЕЙСУ ТА ДАШБОРДУ МОНІТОРИНГУ (FRONTEND)	37
3.3.1 <i>Візуалізація динаміки показників у реальному часі</i>	39
Висновки до розділу 3	43

<b>РОЗДІЛ 4. ЕКСПЕРИМЕНТАЛЬНІ ДОСЛІДЖЕННЯ ТА ВАЛІДАЦІЯ СИСТЕМИ</b>	<b>44</b>
4.1. МЕТОДИКА ПРОВЕДЕННЯ ЕКСПЕРИМЕНТУ ТА КОНФІГУРАЦІЯ ТЕСТОВОГО СЕРЕДОВИЩА	44
4.2. ТЕСТУВАННЯ АЛГОРИТМІВ ОБРОБКИ ДАНИХ НА МОДЕЛЬОВАНИХ СЦЕНАРІЯХ	47
4.3. ОЦІНКА ПРОДУКТИВНОСТІ СИСТЕМИ ТА ЗРУЧНОСТІ ІНТЕРФЕЙСУ (USABILITY)	50
Висновки до розділу 4	53
<b>ВИСНОВКИ</b>	<b>55</b>
<b>СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ</b>	<b>57</b>

## АНОТАЦІЯ

Обсяг пояснювальної записки становить 60 сторінок. Робота містить аналіз сучасних систем моніторингу, опис архітектури розробленого рішення, результати програмної реалізації та експериментальної валідації.

**Актуальність:** зумовлена необхідністю модернізації інструментів екологічного моніторингу в Україні, враховуючи наявність об'єктів атомної енергетики та ризику техногенних катастроф. Використання технологій IoT та хмарних обчислень дозволяє забезпечити прозорий та оперативний контроль радіаційної ситуації в реальному часі.

**Об'єкт дослідження:** система моніторингу радіаційного фону, що включає мережу IoT-датчиків, комунікаційну інфраструктуру та програмне забезпечення для аналізу даних.

**Предмет дослідження:** процеси збору, обробки, статистичного аналізу та візуалізації радіаційних даних у розподілених мережах.

**Мета роботи:** розробити програмне ядро системи моніторингу на основі IoT, яке забезпечує безперервний збір даних, виявлення аномалій та інтерактивну візуалізацію результатів.

### **Завдання роботи:**

1. Проаналізувати існуючі методи та системи моніторингу.
2. Обґрунтувати вибір апаратних засобів (датчиків та мікроконтролерів).
3. Спроекувати архітектуру системи (Edge, Fog, Cloud шари).
4. Реалізувати програмне забезпечення для фільтрації, калібрування та аналізу даних.
5. Провести експериментальне тестування розроблених алгоритмів.

**Методи дослідження:** аналіз літературних джерел, програмне моделювання (Python), статистичні методи обробки даних (Z-Score, MAF), методи візуалізації (Plotly) та використання баз даних часових рядів (SQLite) .

**Наукова новизна:** полягає у розробці та впровадженні адаптивного аналітичного пайплайну для обробки зашумлених даних радіаційних сенсорів,

що використовує метод  $Z$ -Score для виявлення аномалій замість статичних порогів, що суттєво підвищує надійність сповіщень.

**Практичне значення:** створено працездатне програмне ядро (software core), яке виконує повний цикл обробки даних від емуляції сенсора до генерації інтерактивних дашбордів. Розроблене рішення може бути інтегроване в хмарну IoT-архітектуру та використане для розгортання громадських або відомчих мереж моніторингу.

**Ключові слова:** моніторинг радіаційного фону, Інтернет речей (IoT), візуалізація даних,  $Z$ -Score, фільтр ковзного середнього (MAF), Python, інтерактивні карти, екологічна безпека

## ВСТУП

У сучасному світі, особливо в умовах зростання екологічних загроз, техногенних катастроф і воєнних конфліктів, контроль рівня радіаційного фону набуває особливої ваги. В Україні, де є значна кількість об'єктів атомної енергетики, зони підвищеного ризику (як Чорнобильська зона) та активні бойові дії поблизу ядерної інфраструктури, критично важливим є створення та впровадження систем моніторингу радіаційного фону нового покоління. Використання сучасних цифрових технологій, зокрема Інтернету речей (IoT), геоінформаційних систем (GIS), хмарних обчислень та візуалізаційних інструментів дозволяє побудувати ефективні, мобільні та масштабовані рішення, здатні забезпечити прозорий, оперативний і достовірний контроль радіаційної ситуації [7]. Актуальність дослідження зумовлена необхідністю підвищення рівня радіаційної безпеки, модернізації інструментів екологічного моніторингу та реалізації відкритих цифрових екосистем, орієнтованих на захист населення і довкілля.

### **Мета дослідження:**

Розробити систему моніторингу радіаційного фону на основі технології Інтернету речей (IoT) та методів візуалізації даних, яка забезпечить:

- безперервний збір даних про рівень радіації в реальному часі;
- оперативний аналіз та візуалізацію даних на інтерактивних картах;
- виявлення аномалій та сповіщення про потенційну небезпеку;
- доступ до історичних даних та аналіз тенденцій зміни радіаційного фону.

### **Предмет дослідження:**

Процес моніторингу радіаційного фону з використанням мережі датчиків IoT та методів візуалізації даних.

### **Об'єкт дослідження:**

Система моніторингу радіаційного фону, що включає:

- мережу датчиків радіації;

- комунікаційну інфраструктуру для передачі даних;
- програмне забезпечення для збору, обробки, аналізу та візуалізації даних.

#### **Завдання дослідження:**

1. **Проаналізувати** існуючі методи та системи моніторингу радіаційного фону.
2. **Обґрунтувати** вибір типу датчиків радіації та комунікаційних технологій для розробки системи.
3. **Розробити** архітектуру системи моніторингу, включаючи апаратну та програмну частини.
4. **Реалізувати** програмне забезпечення для збору, обробки, аналізу та візуалізації даних.
5. **Створити** інтерактивні карти радіаційного фону з можливістю аналізу даних за різними періодами часу.
6. **Провести** тестування розробленої системи та оцінити її ефективність.

#### **Методи дослідження:**

- **Аналіз літературних джерел:** Вивчення наукових публікацій, технічної документації та стандартів з теми моніторингу радіаційного фону, IoT та візуалізації даних.
- **Методи моделювання:** Створення моделей системи моніторингу для аналізу її функціонування та оцінки ефективності.
- **Методи програмування:** Розробка програмного забезпечення для реалізації функціональності системи.
- **Експериментальні методи:** Проведення експериментів з різними типами датчиків та комунікаційними технологіями для визначення оптимальних параметрів системи.
- **Методи аналізу даних:** Використання статистичних методів та методів машинного навчання для аналізу даних з датчиків та виявлення аномалій.

- **Методи візуалізації даних:** Застосування геоінформаційних систем (GIS) та бібліотек візуалізації для створення інтерактивних карт радіаційного фону.

## РОЗДІЛ 1 АНАЛІЗ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ МОНІТОРИНГУ РАДІАЦІЙНОГО СТАНУ

### 1.1 Проблематика радіаційного контролю: параметри моніторингу та вимоги до систем

Іонізуальне опромінення впливає на все живе, включаючи космічні промені з верхніх шарів атмосфери, земні нукліди в земній корі, будівельні матеріали, повітря, воду, їжу та сам організм людини.

Рівень опромінення є постійним і однаковим для кожної людини в будь-якій точці, в тому числі і від споживання (40)К з їжі. Промені космосу, які найбільше інтенсивні, дуже чутливі на значних висотах, а в деяких зонах ґрунт має підвищений вміст урану і торію. Також відзначається, що опромінення змінюється в процесі господарської діяльності людини.

Зокрема, у будівельних матеріалах для зведення будівель, їхньої архітектури та влаштування системи вентиляції значна міра вплине на накопичення об'ємів радіоактивного газу радону і його дочірніх продуктів у приміщення, що причиняє дозове опромінення через їх інгаляцію. Для визначення даного значення компонентів джерел опромінення населення Індії опори дані.

Загальний внесок природних джерел у населення Індії становить 2,3 мЗв на рік. Це нижче світового показника, який становить 2,4 мЗв на рік. Наведений короткий огляд і порівняння природних і техногенних джерел забруднення населення Індії. Випромінювання атмосферної зброї становить

0,045 мЗв на рік, фонове випромінювання  $1,2 \times 10^{-3}$  мЗв на рік, видобуток важких металів і їх використання як добрив, вугільні електростанції та видобуток фосфатних порід.

## 1.2 Архітектурні підходи до побудови IoT-систем збору екологічних даних

Моніторинг іонізуючого випромінювання — це далеко не просто зняття показників. Тут важливо помічати кожну дрібницю. Спершу збирають дані, потім аналізують усе навколо, стежать за найменшими змінами. На основі цієї інформації будують сучасні системи: вони не шаблонні, реагують блискавично. Якщо щось і змінюється — помітно відразу. Не треба чекати, поки проблема сама себе проявить, реагуєш тут і зараз.

### **Класифікація методів**

Активний моніторинг — зовсім інша історія. Тут ти бачиш дані відразу, в реальному часі. Використовують електронні детектори: лічильники Гейгера-Мюллера, сцинтиляційні лічильники, напівпровідникові сенсори [1]. Інформація з'являється миттєво, чекати не треба.

Моніторинг також поділяють за масштабом. Є локальний — коли спостерігають за окремими об'єктами: АЕС, лабораторіями, промзонами. Далі — регіональний, де вже охоплюють великі території, наприклад, області чи навіть усю країну. І нарешті, глобальний моніторинг. Тут усе серйозно: супутники, міжнародні мережі на кшталт СТВТО — дані надходять з усього світу.

Ще одна важлива ознака — частота вимірювань. Дискретний моніторинг працює за розкладом: вимірюєш через певні інтервали, отримуєш "знімки" ситуації. Неперервний — це вже постійний процес, коли дані йдуть безперервно, і ти бачиш усі зміни майже миттєво.

## **Засоби вимірювання рівнів радіації**

Тепер про техніку. Найчастіше для вимірювання іонізуючого випромінювання використовують детектори Гейгера-Мюллера — ті самі ГМ-трубки. Вони прості, доступні, їх часто ставлять у звичайні дозиметри, але ідеальною чутливістю похвалитися не можуть. Сцинтиляційні детектори точніші, можуть визначати тип радіоактивної частинки, але й коштують дорожче. А ось напівпровідникові сенсори — наприклад, Si чи CdZnTe — це вже сучасний підхід. Вони компактні, мають високу роздільну здатність і часто зустрічаються у сучасних IoT-пристроях. Для мікроконтролерів вже є готові модулі — SEN-14209, RD200M — їх просто під'єднати до Arduino, ESP32, STM32 чи подібних плат.

## **IoT у системах моніторингу**

Інтернет речей (IoT) повністю змінив підхід до моніторингу довкілля. Більше не треба ускладнювати собі життя — можна просто зібрати мережу з десятків, а то й сотень сенсорів, які збирають і передають дані майже одразу.

Зазвичай IoT-система для моніторингу радіаційного фону має кілька рівнів. Все починається з сенсорів іонізуючого випромінювання. Їх підключають до мікроконтролерів — ESP32, Raspberry Pi, STM32. Ці штуки працюють автономно, живляться від батарейок чи сонячних панелей. Можна встановити де заманеться, хоч у полі, хоч на даху. Інформацію вони передають по-різному: Wi-Fi, LoRaWAN [9], NB-IoT, Zigbee — що завгодно підходить. Далі дані збираються на шлюзах або летять одразу в хмару.

У хмарі розташовані бази даних — InfluxDB, PostgreSQL — і сервери для обробки інформації: Node-RED, MQTT-брокери [10], REST API, а також сховища типу Firebase, AWS IoT чи ThingsBoard. Для користувача все виглядає просто: відкрив дашборд у Grafana, Plotly чи Power BI — і бачиш найсвіжіші дані в зручному вигляді, хоч на телефоні, хоч у браузері.

#### Приклади сучасних реалізацій

Safecast — це глобальна мережа добровольців, які збирають дані про радіаційний фон з портативних IoT-пристроїв і відкрито діляться результатами [4].

RADIoT — система моніторингу радіації на базі ESP32 і сенсора SBM-20 [15]. Дані публікуються на відкритих платформах — можна підключитись і слідкувати онлайн.

Eco-Counter — тут поєднують радіаційні сенсори з датчиками клімату, щоб оцінити, як погодні умови впливають на рівень опромінення.

#### **IoT у моніторингу навколишнього середовища**

Екологічні проблеми вже давно перестали бути чимось далеким — вони буквально відчуваються на собі щоднях [17]. Моніторинг довкілля перетворився на справжню необхідність. Без нього не варто й мріяти про сталий розвиток, безпеку чи навіть базову якість життя. Саме тут Інтернет речей (IoT) стає ключовим гравцем. Завдяки цій технології ми нарешті можемо бачити, що відбувається з довкіллям прямо зараз: дані точні, прив'язані до конкретного місця, надходять одразу — жодних затримок.

Колись усе було інакше. Дані збирали вручну, час від часу, на кількох стаціонарних станціях. IoT повністю змінює цю картину. Тепер сенсори можна ставити там, де це справді важливо — у лісі, біля річки чи біля заводу. Вони самі збирають потрібні дані: температуру, вологість, рівень радіації, концентрацію забруднювачів у повітрі, воді або ґрунті. Вся ця інформація одразу летить на сервери через Інтернет чи локальні мережі. Тепер реагуєш

майже миттєво, а сучасна аналітика з візуалізацією дозволяють бачити, як ситуація змінюється просто зараз, прямо перед тобою.

Одна з найсильніших сторін IoT — геоприв'язка. Під час кожного вимірювання сенсор визначає свої координати. Ось так з'являються інтерактивні карти стану довкілля. На них одразу видно “гарячі точки” забруднення — все як на долоні. Стає ясно, де саме людська діяльність чи природні явища впливають на екосистеми. Найважливіше — ці карти можуть бути відкритими для всіх. Прозорість зростає, і разом із нею міцніє довіра в суспільстві.

Є ще один важливий момент. IoT автоматизує реагування. Варто показнику вийти за межі норми — система одразу надсилає попередження чи запускає аварійний протокол. При витокі хімікатів, пожежах, аваріях або радіаційній загрозі така оперативність рятує життя.

IoT давно перестав бути просто набором сенсорів. Зараз він легко працює разом із геоінформаційними системами, супутниками, мобільними додатками та алгоритмами штучного інтелекту. Збір даних про довкілля — це вже тільки старт. Тепер ми будуємо прогнози, враховуємо погоду, сезонність, навіть людську поведінку. IoT вже не просто “очі” для екології. Він став основою для справжнього, усвідомленого управління нею.

Є ще одна суттєва перевага — громадські екологічні мережі. Людина з простим сенсором теж може збирати дані. Це вже працює в багатьох країнах, і практика довела: децентралізований, відкритий контроль справді ефективний. IoT у моніторингу довкілля — це вже не мода, а справжня стратегія. Тут усе змішалось: інженерія, цифровий аналіз, телекомунікації, складні алгоритми. Комп'ютерні науки відкривають двері для створення таких систем, і їх запускають не лише в окремій країні, а й по всій планеті.

### 1.3 Огляд методів обробки та візуалізації геопросторових і часових рядів даних

Інтернет речей, або IoT, — це коли звичайні об'єкти, від сенсорів до машин, починають спілкуватися між собою через Інтернет. Вони постійно обмінюються даними, все відбувається просто тут і зараз. Головна ідея — автоматизувати рутинні процеси, завжди мати під рукою важливу інформацію й контролювати ситуацію, навіть якщо ти знаходишся за тисячі кілометрів від дому.

Зазвичай IoT-система складається з кількох рівнів [7].

Спочатку — сенсори й контролери, вони відповідають за збір даних.

Далі йде рівень зв'язку. Тут вступають у гру різні протоколи: Wi-Fi, LoRa, 4G/5G, ZigBee, MQTT [10].

Третій рівень — обчислення. Обробка даних, моделювання та прогнозування відбувається із застосуванням штучного інтелекту та методів машинного навчання [16] на хмарних серверах, або локально, через edge computing.

І нарешті — візуалізація й аналітика. Це всі ті веб-інтерфейси, мобільні додатки чи аналітичні панелі, наприклад, Grafana або Power BI. Вони дозволяють бачити й аналізувати дані, які система збрала.

#### **Приклад використання технологій**

Safecast — це реально легендарна відкрита платформа для моніторингу радіації. Ну, навіть якщо ви далекі від IoT, скоріше за все, все одно про неї чули. Її запустили після аварії на Фукусімі у 2011 році, коли людям критично не вистачало зрозумілих і точних даних про стан довкілля.

Все дуже просто. Волонтери беруть у руки маленькі портативні сенсори — скажімо, Safecast bGeigie Nano чи RADNOTE — і виходять збирати дані про радіацію. Всередині кожного такого пристрою стоїть датчик Гейгера-Мюллера, GPS і модуль зв'язку — Wi-Fi або GSM, залежно від моделі.

Сенсор не просто міряє радіацію. Він одразу записує, де ти знаходишся і коли саме зроблено замір.

Щойно пристрій зібрав дані, він відправляє їх прямо на відкриту хмарну платформу Safecast. Там інформацію перевіряють, фільтрують, приводять до стандарту — і публікують на інтерактивній мапі (<https://map.safecast.org>) [4]. Мапу оновлюють у реальному часі. Кожна точка містить рівень радіації, точні координати, час вимірювання і тип сенсора. Вся ця інформація зручно візуалізується у вигляді heatmap, яка постійно “оживає” новими даними.

Тисячі активних користувачів допомогли Safecast зібрати одну з найбільших у світі відкритих баз радіаційних геоданих, і ця база зростає щодня. Ці дані потрібні не лише звичайним людям. Їх використовують науковці, журналісти, міські служби, екологи — коротко кажучи, всі, кому не байдуже.

IoT-компоненти Safecast роблять усе максимально просто. Дані збираються автоматично, система точно відзначає місце й час, усе одразу потрапляє у хмару. Доступ відкритий для всіх — зайшов і користуйся. Найцікавіше — можна підключити новий сенсор з будь-якої країни, не морочачись із налаштуваннями.

#### 1.4 Аналіз існуючих аналогів та обґрунтування необхідності розробки власної системи

Зараз даних про довкілля просто море, і їх стає все більше з кожним днем. Це вже не просто сухі цифри, а цілий потік інформації, що росте на очах. Просто збирати ці дані — мало. Їх треба ще й перекладати на людську мову, щоб кожен міг зрозуміти, що стоїть за цими числами. Ось тут і заходить у гру візуалізація геопросторових даних [2]. Вона перетворює координати та числа на зрозумілі, наочні картинки. Особливо це важливо там, де працюють IoT-системи: інформація надходить цілодобово, з тисяч різних сенсорів, і все це у реальному часі.

Геопросторові дані — це не просто якісь точки на карті. Це конкретні показники (наприклад, температура, рівень радіації чи вологість), точно прив'язані до місця і часу. В екологічному моніторингу, особливо з IoT-рішеннями, ці дані отримують з сенсорів на дронах, транспорті чи стаціонарних модулях, що охоплюють цілі райони. GPS чи інші геолокаційні системи чітко фіксують, звідки і коли взято кожен замір. Усе це разом складається у справжню цифрову “картину” стану навколишнього середовища.

Візуалізація — це зовсім не просто красиві картинки. Це справжній інструмент, який відкриває зовсім інший погляд на простір: тут ти помічаєш зв'язки, бачиш тренди, підловлюєш аномалії й знаходиш ті ключові точки, які здатні все змінити. Вся система працює разом: GIS [2], бібліотеки просторових даних, open-source інструменти, сучасні хмарні сервіси — все це з'єднується у єдиний ланцюг. І саме так ти реально розумієш, як працюють складні просторові процеси.

Зараз геопросторові дані з IoT зазвичай показують на інтерактивних картах. Хочеш — масштабуєш, хочеш — фільтруєш за часом чи районом, додаєш різні шари, порівнюєш між собою. Найзручніше, звісно, працювати з тепловими картами (heatmaps). Вони одразу дають зрозуміти, де показник вищий, а де нижчий. Кольори допомагають швидко побачити зону ризику чи забруднення, і не треба витратити час, щоб розбиратися в цифрах.

Є ще один цікавий напрям — тривимірні моделі. Тут додається третій вимір: це може бути висота (наприклад, з дрона), а інколи і час, якщо треба простежити, як змінюється ситуація згодом. Це дозволяє не просто фіксувати стан довкілля зараз, а й аналізувати динаміку змін. А якщо підключити ще й машинне навчання — можна виявити приховані тренди і навіть прогнозувати майбутнє.

В екосистемах IoT важливо інтегрувати візуалізацію в дашборди. На них зручно показувати не лише карти, а й графіки, таблиці, попередження чи навіть рекомендації від алгоритмів. Користувачі не просто дивляться на

ситуацію — вони взаємодіють із даними: шукають потрібні ділянки, порівнюють історію, аналізують тенденції.

І ще одне — швидкість. Дані оновлюються постійно, і система має працювати без затримок, навіть коли інформації багато. Тому все частіше обробку і візуалізацію в IoT роблять через хмарні сервіси з потоковою аналітикою.

В результаті, візуалізація геопросторових даних для IoT-моніторингу — це вже не просто технічна “надбудова”. Це справжній інструмент для прийняття рішень, який поєднує аналітику, візуальну простоту і сучасний інтелект. Для комп’ютерних наук це відкриває цілий простір для досліджень: удосконалення алгоритмів, створення адаптивних інтерфейсів, розробка віртуальних “близнюків” нашого світу.

Системи моніторингу радіаційного фону — це справжня опора для безпеки: і екологічної, і техногенної. Їхня суть проста. Вони швидко ловлять навіть найдрібніші зміни рівня іонізуючого випромінювання, одразу б’ють на сполох, якщо щось не так, і дають точну, актуальну інформацію всім — від звичайної людини до науковця чи держслужбовця. За останні роки ці системи сильно змінилися. Вони стали відкритіші, працюють розподілено, і активно підтягують IoT-технології.

Як це виглядає на практиці? У кожної країни свої державні системи контролю — вони цілодобово фіксують рівень радіації біля атомних електростанцій або в інших важливих точках, особливо там, де ризик вищий. Європейський Союз теж не пасе задніх. Вони запустили платформу EURDEP: вона тягне дані з різних країн і дозволяє обмінюватися інформацією майже миттєво [5]. В Україні теж є власна система. Її підтримують ДСНС та Держатомрегулювання. Вся робота тримається на постійних стаціонарних датчиках. Вони цілодобово фіксують рівень радіації. Дані ці досить точні, хоча й не ідеальні. Проблема в тому, що зона покриття невелика, система повільно реагує на різкі зміни і майже не взаємодіє з громадянами.

Все змінилося з появою нових цифрових рішень. З'явилися відкриті децентралізовані платформи. Тут все по-новому: до моніторингу долучаються звичайні люди, мережа швидко зростає завдяки доступним IoT-пристроєм. Це дозволяє охопити значно більшу територію, зменшити витрати й зробити моніторинг справді гнучким.

**Таблиця 1. Приклади найпоширеніших систем**

Назва системи	Тип організації	Джерело даних	Геоприв'язка	Відкритість	Сфера застосування
EURDEP	Державна (ЄС)	Стационарні пункти	Координати фіксовані	Часткова	Контроль, обмін даними між країнами
SaveEcoBot	Громадська (UA)	IoT-сенсори (EcoSensor)	Автоматична (GPS)	Повна	Громадський контроль, аналітика
Safecast	Глобальна спільнота	Персональні IoT-сенсори	GPS-модулі	Повна	Постфукусімський моніторинг, наука
EcoSensorNet	Приватна / муніцип.	IoT-сенсори + БПЛА	GPS / Wi-Fi	Часткова	Локальний екологічний моніторинг
Radiacode System	Комерційна	Мобільні дозиметри	Через смартфон	Залежно від користувача	Індивідуальне використання

### Технічне порівняння IoT-систем

Окрім організаційних особливостей, важливими є технічні характеристики платформ — типи сенсорів, точність, спосіб передачі даних, наявність візуалізації.

**Таблиця 2. Технічне порівняння IoT-систем**

Система	Модель сенсора	Тип випромінювання	Геолокація	Передача даних	Точність / роздільна здатність
Safecast	RADNOTE, bGeigie Nano	Гамма	GPS	GSM, Wi-Fi	Середня (без спектрометрії)

SaveEcoBot	EcoCensor	Гамма	GPS, GSM	MQTT, HTTPS	Стабільна, калібрована
Teledyne	MUVE R430	Гамма-спектрометрія	GPS + INS	Wi-Fi, LTE, 5G	Висока ( $\leq 7\%$ FWHM @ 662 keV)
H3D Model 501	Гамма + нейтрони	Повна спектрометрія	Зовнішній GPS	Ethernet, USB	Дуже висока ( $\leq 0.8\%$ FWHM @ 662 keV)
LDN Sensor	LDN Radiation Sensor	Гамма, бета	GPS або Wi-Fi	LoRa, Wi-Fi	Висока (дані неповні)
Radiacode 102	Радіаційний модуль	Гамма	Смартфон (Bluetooth)	BLE, локально	Побутова точність

Ці системи показують, наскільки різними можуть бути підходи до моніторингу. Державні платформи тримають все під контролем: стабільність, чіткі стандарти — з цим у них порядок. Але гнучкість? Тут слабко. Охоплення теж залишає бажати кращого. З іншого боку, IoT-платформи на кшталт Safecast чи SaveEcoBot йдуть своїм шляхом [6]. Тут інша філософія — децентралізація, залучення громадян, відкриті дані, хмарні сервіси. І це дає результат. Особливо добре працюють рішення, що поєднують IoT-сенсори з геопросторовою візуалізацією, машинним аналізом і мобільними пристроями. Такий підхід реально розширює можливості.

Ще один тренд — все більше використовують БПЛА з точними сенсорами, наприклад, Teledyne FLIR MUVE R430. Це реально відкриває нові можливості для швидкого складання карт забруднення після аварій чи надзвичайних ситуацій. Дані з таких дронів легко вбудовуються у геоінформаційні системи, а через хмарні API їх одразу аналізують, візуалізують і приймають рішення буквально на ходу.

Якщо коротко, майбутнє ефективного радіаційного моніторингу — за гібридними екосистемами. Державна інфраструктура тут працює разом із

розподіленими мережами громадських сенсорів і мобільних пристроїв. Щоб це все справді запрацювало, потрібен міждисциплінарний підхід. І тут фахівці з комп'ютерних наук — ключові: вони пишуть софт, обробляють великі масиви просторових даних, створюють протоколи передачі й моделі прогнозування.

## Висновки до розділу 1

У цьому розділі викладено проблему моніторингу радіаційного стану. Тема дійсно важлива — тут не перебільшую. Ми щодня маємо справу з впливом як природних, так і штучних джерел випромінювання. Екологічні ризики ростуть, і без сучасних, оперативних та відкритих систем контролю вже не обійтись. Технології швидко розвиваються: Інтернет речей, геоінформаційні системи, хмарні обчислення. Все це відкриває шлях до оновлення старих, малоефективних підходів.

Було проаналізовано основні вимоги до систем радіаційного моніторингу, методи контролю за масштабом (локально, регіонально, глобально) і за частотою (дискретно, безперервно). Було проведено огляд архітектури — від централізованих державних рішень (типу EURDEP) до децентралізованих громадських платформ на кшталт Safecast чи SaveEcoBot. Чесно, найкраще працює гібридний підхід, який поєднує надійність стаціонарної інфраструктури з гнучкістю й широким охопленням IoT-сенсорів.

Окремо сконцентровано на методах обробки й візуалізації геопросторових та часових рядів даних. Тут без геоприв'язки нікуди — саме вона дозволяє будувати інтерактивні карти й швидко виявляти "гарячі точки". Але є проблеми: існуючі рішення поки що не ідеальні. Вони часто охоплюють занадто вузькі зони, а реагують на зміни повільно. Тому нам потрібні власні алгоритми для аналізу аномалій і візуалізації.

В результаті огляду впливає: розробка власної IoT-системи має не лише науковий, а й реальний практичний сенс. Технічні вимоги до такої системи — енергоощадні сенсори, протоколи LPWAN, хмарні сервіси для збору й аналізу даних — уже чітко окреслені.

## РОЗДІЛ 2. ПРОЕКТУВАННЯ АРХІТЕКТУРИ ТА АЛГОРИТМІЧНОГО ЗАБЕЗПЕЧЕННЯ СИСТЕМИ

### 2.1. Обґрунтування апаратно-програмної платформи (IoT-рівень)

Проектування системи СРМРФ для екосистеми Інтернету речей є нетривіальним і вимагає від інженера балансування між масштабованістю та мобільністю в тандемі з високою ефективністю. Успіх створення або навіть важливість подальшого розвитку цієї системи нерозривно пов'язані з раціональністю підходу до апаратного забезпечення. При виборі обладнання варто враховувати не лише точність вимірювань, цінову категорію, а також критерії енергоефективності та безперервної взаємодії з бездротовими підсистемами.

#### 2.1.1. Вибір сенсорів та мікроконтролерів для реєстрації іонізуючого випромінювання

Правильний вибір типу детектора радіації має критичний вплив на функціональні характеристики та економічну доцільність всього комплексу. У контексті розробки вимог до системи розподільчої мережі для моніторингу підвищених рівнів та фонового перших та третіх доступні такі типи рівнів: четвертий.

1. Гейгера-Мюллер ГМ-трубки ГМ-трубки є найбільш поширеними типами в побутовому та розподіленому моніторингу знайденими [1]. Їхня робота ґрунтується на іонізаційному принципі: іонізуюче випромінювання, проникаючи у газонаповнену камеру, викликає лавиноподібний розряд, який фіксується як електричний імпульс.

Що стосується переваг, то найбільш продуктивні в тому плані є низька вартість вартість: це найбільш економічно доцільні датчики для численної розгортки точок моніторингу. Простота інтерфейсу: вихідний сигнал — дискретний імпульс або лічильник імпульсів, який легко інтегрується з мікроконтролерами, наприклад, через зовнішні периферії. Низове

енергоспоживання. Забезпечує тривалу автономну роботу, що є суттєвим професійним показником для IoT нод. За недоліки можна визначити обмеження по спектрометрії: неможливо визначити енергію та типи частинок, фактично, спільно лічку кількості подій обмежує можливості ідентифікації радіонуклідів. Ефект “мертвого часу”: після кожного спрацьовування трубка потребує часу на відновлення, що веде до інфулювання високих дозах.

## 2. Сцинтиляційні детектори

Сцинтиляційні детектори працюють під принципом перетворення іонізуючої енергії випромінювання у світлові спалахи, так звану сцинтиляцію, яку реєструють за допомогою Фотоеквівалентних Фотоумножувачів або кремнієвих фотоумножувачів (SiPM) [1].

Переваги:

- Чутливість за кровоподібні співвідношення
- Точність – точна румунська компактність низьких рівнів фону<sup>10</sup>
- Спектрометрія – можливість визначення енергетичного ідентифікації радіонуклідів, які дозволяють відокремити джерело забруднення

Недоліки:

- Висока вартість та габарити.
- Через необхідність використання кристалів , наприклад, NaI Tl чи LaBr<sub>3</sub>, та ФЕП, вони є дорогими та менш мобільними
- Енергоспоживання. Вимагають складнішої системи живлення та значно вищого енергоспоживання .

## 3. Напівпровідникові твердотільні сенсори

Використовують напівпровідникові матеріали, наприклад, кремній Si, кадмій-цинку телурид CdZnTe. Іонізуюче випромінювання створює електронно-діркові пари, які формують вимірюваний струм.

Переваги:

- Компактність. Можливість мініатюризації та інтеграції в Mini - пристрої Radiacode

- Висока роздільна здатність – досягають високої спектральної роздільної здатності, поступаючись лише лабораторними сцинтиляційними системами.

Недоліки:

- Чутливість. Важливо використовувати охолодження, найбільше потребують його HPGe детектор, також варто звернути увагу на вартість, оскільки вона вище за середню вартість.

Обґрунтування вибору:

Враховуючи мету дослідження, детектори типу Гейгера-Мюллера обираються замість детальної спектрометрії для безперервного моніторингу радіаційного фону. Вони забезпечують оптимальний баланс точності, вартості та енергоефективності, що дозволяє створювати широкую мережу Інтернету речей (IoT)-нод<sup>9</sup>. Надійними вимірювачами потужності дози в діапазоні від 0,1 до 100 мкЗв/год є СБМ-20 (SBM-20) або сучасні аналоги з низькою робочою напругою.

**Таблиця 3. Порівняння датчиків**

Модель/ Компонент/Рішення	Тип детектора (коротко)	Орієнтовні Габарити (ДхШхВ або ØхД)	Орієнтовна Вага	Орієнтовна Ціна (USD)	Примітки
LND 712 (трубка)	Лічильник Гейгера-Мюллера	~ Ø10.8 x 45 мм (трубка)	~ 5-7 г (трубка)	~\$60 - \$90 (за трубку)	Це лише сама детекторна трубка. Потребує плати об'язки, високовольтного живлення. Готовий модуль буде більшим, важчим і дорожчим.
SBM-20 (трубка)	Лічильник Гейгера-	~ Ø10 x 105 мм (трубка)	~ 10 г (трубка)	~\$10 - \$30 (за трубку, часто зі	Старіша модель, також потребує об'язки.

	Мюллера			старих запасів)	
Radiation D-v1.1 Geiger Counter Kit (MightyOhm)	Лічильник Гейгера-Мюллера	Плата ~58 x 58 мм	~ 30-50 г (зібраний кіт без корпусу)	~\$100 - \$150 (за набір для збірки)	Набір для самостійної збірки, часто використовує трубку типу LND712 або аналогічну. Габарити та вага можуть варіюватися з корпусом. Ціна за набір.
Hamamatsu S3590-08 (PIN-діод)	PIN-діод	Чутлива зона ~10x10 мм; Корпус ~14x16x1.6 мм	< 1 г	~\$15 - \$30 (за компонент)	Це сам фотодіод. Потребує складної обв'язки для детектування радіації (сцинтилятор, підсилювачі).
Модулі на ESP32 + GM трубка (кастомні збірки)	Лічильник Гейгера-Мюллера	Залежить від компоновки, напр., ~70x40x30 мм (з корпусом)	~ 50-150 г	~\$50 - \$200 (вартість компонентів + корпус)	Характеристики сильно залежать від обраної GM трубки, плати ESP32, корпусу, батареї. Ціна є сумою вартості компонентів.
Kromek D3S ID (модуль/пристрій)	CZT (спектрометричний)	~ 67 x 33 x 18 мм	~ 40 г	~\$1500 - \$3000+ (B2B, залежить від обсягу)	Готовий компактний детектор-ідентифікатор. Ціни для кінцевих пристроїв або OEM-партнерів можуть сильно відрізнятися.

Kromek SIGMA (датчик)	Scintillator (CsI(Tl))	ø25 x 55 мм (приклад для SIGMA 25)	~ 50-100 г	За запитом (B2B)	Компактні сцинтиляційні датчики, ціна залежить від моделі та умов.
Libelium Plug & Sense! Smart Environment PRO (вузол з радіаційним сенсором)	Лічильник Гейгера-Мюллера	~ 81 x 91 x 136 мм (корпус вузла)	~ 200-400 г	~\$700 - \$1200+ (за вузол з різними сенсорами, включаючи радіаційний)	Це цілий IoT-вузол з різними сенсорами та комунікаціями (LoRaWAN, 4G). Ціна залежить від повної конфігурації вузла. Радіаційний сенсор є однією з опцій.
RAKwireless WisBlock (модульна система)	Залежить від підключеного сенсора	Базовий модуль + сенсорний модуль, напр., ~40x30 мм + розміри сенсора	Залежить від комбінації	~\$30-70 (базова плата + MCU + LoRa) + вартість сенсорного модуля	RAKwireless надає модульну платформу. Потрібно окремо купувати або розробляти сенсорний модуль для радіації та інтегрувати його. Ціна – за базові модулі, без урахування специфічного радіаційного сенсорного модуля.

### 2.1.2. Протоколи передачі даних в умовах розподіленої мережі

У реальній IoT-ноді дані обробляє мікроконтролер (МК).

МК повинен мати достатню потужність для обробки, вбудовані комунікації (Wi-Fi/LoRa) і низьке енергоспоживання.

Аналіз можливостей платформ:

Arduino UNO: вимагає мало потужності та не має вбудованого зв'язку.

Raspberry Pi: для автономних нод споживає багато електроенергії, але має безкоштовну операційну систему.

ESP32: найкращий вибір [8]. Він має два ядра для паралельної обробки та комунікації даних, Wi-Fi та Bluetooth вбудовані в нього, а також режим сну з низьким енергоспоживанням (Deep Sleep).

Пояснення моделі: Функція `generate_cpm_data` імітує функції мікроконтролера в нашій моделі, такі як збір імпульсів і підрахунок СРМ. У цьому коді функція GPS-модуля, який додавав геолокацію, зменшилася, щоб зосередитися на часових рядах.

## 2.2. Розробка загальної архітектури системи (Edge, Fog, Cloud шари)

Передача даних до сервера може бути реалізованою декількома способами.

LPWAN (Low Power Wide Area Networks): для моніторингу Інтернету речей найважливішими є технології, такі як LoRaWAN або NB-IoT, оскільки вони забезпечують велику дальність при низькому енергоспоживанні [9].

Wi-Fi/Bluetooth: Ці можливості хороші для локальних завдань або точок із постійним живленням, але вони потребують шлюзу або мають обмежений радіус дії.

Транспортний протокол: легка та надійна система (модель "публікація/підписки").

**Обґрунтування для моделі:** Наша програмна модель є **монолітною** (виконується на одній машині). Етап комунікації (Рівень 2) пропускається: дані, згенеровані емулятором сенсора, передаються безпосередньо до блоків обробки та збереження (локальної БД) через виклики функцій Python.

## 2.3. Математичне та алгоритмічне забезпечення

Розроблена програмна модель має монолітну та послідовну архітектуру, що відображає основні етапи повного циклу моніторингу, від генерації даних до візуалізації.

2.3.1. Алгоритми попередньої обробки сигналу: фільтрація шумів та агрегація  
Дану модель забезпечує лінійний потік даних, який складається з чотирьох основних функціональних блоків, на відміну від трирівневої архітектури Інтернету речей (IoT):

- Блок 1: Емуляція даних (рівня сенсорів)  
Функція: `stvoriti_cpm_dani()`  
Процес: моделювання роботи ГМ-трубки в польових умовах шляхом створення «сирих» даних (CPM) та часових міток.
- Блок 2: Збереження та калібрування (Рівень обробки та збереження)  
Функції: `calibrate_data`, `init_db`, `save_to_db`  
Процес:  $\mu\text{Sv/h}$  є калібром даних CPM [1]. Коли хмарне сховище замінюється локальною базою даних SQLite [13], калібровані дані зберігаються для подальшого аналізу.
- Блок 3: Аналіз даних (рівень аналітики)  
Функція: `process_data()` (з Pandas) [11]  
Процес: Алгоритм виявлення аномалій, відомий як Z-Score, і фільтр ковзного середнього (MAF) використовуються для обробки завантажених даних.
- Блок 4, що стосується візуалізації та звітності (рівень представлення)  
Функції: `build_figure()` (з Plotly) і `save_and_open_html()` [12]

2.3.2. Метод виявлення аномалій у потоці радіаційних даних

- Блок емуляції сенсора (`generate_cpm_data`): використовується для заміни апаратного рівня 1 (ESP32 + ГМ-трубка). імітує нормальні, підвищені та небезпечні рівні радіації з різною ймовірністю в часових рядах.

- Блок калібрування називається `calibrate_data`: перетворює абстрактні «імпульси» на фізично значущі  $\mu\text{Sv/h}$  за допомогою основного алгоритму, наведеного в пункті 2.4.1.
- Блок збереження (`SQLite`, `save_to_db`): використовується для заміни хмарного рівня 3 для баз даних `InfluxDB/PostgreSQL`. Для зберігання історичних даних, необхідних для аналізу тенденцій, використовується локальна БД `SQLite` [13].
- Ядро «серверної частини» — це блок аналітики, який називається `process_data`. Він аналізує дані та знаходить аномалії. Потім він готує дані для візуалізації.
- Блок візуалізації (`build_figure`): використовується для заміни дашбордів (`Grafana/Power BI`). Інтерактивні карти радіаційного фону створюються за допомогою `Plotly`. 2.4. Розробка програмного забезпечення моделі

## 2.4. Розробка програмного забезпечення моделі

Програма виконується мовою `Python` і використовує бібліотеки `sqlite3` для збереження, `pandas` для аналізу та `plotly` для візуалізації [13].

2.4.1. Алгоритми обробки даних датчиків, включаючи агрегацію, фільтрацію та калібрування

Першочергова обробка даних необхідна для підготовки даних до передачі, зменшення шуму та перетворення сирих імпульсів на фізичні величини.

**1. Калібрування ( $\text{CPM} \rightarrow \text{Sv/h}$ ):** інформація з ГМ-трубки передається у вигляді кількості імпульсів за хвилину ( $\text{CPM}$  — Counts Per Minute). Коефіцієнт лінійного калібрування ( $K$ ), який є специфічним для кожного типу трубки, використовується для переведення одиниці потужності дози ( $\text{Sv/h}$ ) в одиниці ампер-години ( $\text{CPM}$ ):

$$\text{Dose Rate}(\mu\text{Sv/h}) = \text{CPM} \times K$$

**2. Фільтрація та Агрегація (Згладжування шуму):** вимірювання радіації є статистичним процесом, відомим як розподіл Пуассона, що призводить до

високої дисперсії. Фільтр Ковзного Середнього, також відомий як MAF, використовується для зменшення шуму та зменшення кількості передач даних:

$$D_t^{avg} = \frac{1}{N} \sum_{i=0}^{n-1} D_{t-i}$$

На момент  $t$  значення потужності дози  $D_t^{avg}$  -

$N$  - розмір вікна агрегації (кількість останніх вимірювань, наприклад, десять-двадцять вимірювань за п'ять хвилин).

**Підтвердження N:** Хоча збільшення  $N$  підвищує точність, його використання є менш ефективним. Забезпечуючи баланс між статистичною достовірністю та реакцією в реальному часі, оптимальний вибір  $N$ .

### 3. Алгоритм виявлення Аномалій (Серверний рівень):

На серверному рівні застосовується статистичний аналіз для автоматичного виявлення різких підвищень фону:

- **Z-Score (Пороговий метод):** Виміряне значення ( $D$ ) вважається аномалією, якщо воно перевищує середнє історичне значення  $\mu$  на  $Z$  стандартних відхилень  $\sigma$ :

$$\text{Anomaly if: } \frac{D-\mu}{\sigma} > Z_{threshold}$$

Типове значення  $Z_{threshold} \approx 3$

- **Генерація сповіщень:** У разі виявлення аномалії система автоматично генерує сповіщення (SMS, E-mail, Telegram) та позначає точку на карті як "Гаряча Зона".

## Висновки до розділу 2

У цьому розділі обґрунтовано вибір апаратно-програмної платформи для системи моніторингу радіаційного фону (СРМРФ). Основна мета — знайти ідеальний баланс між економічністю, енергоефективністю та функціональністю. Проаналізовано різні типи детекторів. Для розподіленого моніторингу найкраще підходять лічильники Гейгера-Мюллера. Вони не дають детальної спектрометрії, але виграють завдяки простоті інтеграції та низькій ціні.

Два ядра і режим глибокого сну Deep Sleep справді рятують у польових умовах — енергії витрачається мінімум, а це критично важливо. Для зв'язку я використовую LPWAN-технології: LoRaWAN або NB-IoT. Ці протоколи дають великий радіус дії й не тягнуть зайву енергію. Саме те, що потрібно для автономної роботи.

Змодельовано загальну архітектуру як лінійний програмний потік: спочатку Емуляція, потім Обробка, Збереження, далі Аналітика і на завершення Візуалізація. Це добре працює для створення монолітного прототипу. Я розробив математичне та алгоритмічне забезпечення, яке і стало основою цієї системи. Серед методів попередньої обробки використав алгоритми Калібрування і Фільтрацію Ковзного Середнього (MAF), щоб зменшити статистичний шум, який часто супроводжує радіаційні вимірювання. Але ключове — це вибір і впровадження алгоритму Z-Score на сервері для автоматичного, адаптивного виявлення аномалій. Це рішення значно підвищує надійність сповіщення порівняно зі статичними порогоми.

Цей розділ дав чіткий технічний стек: тут і обґрунтування компонентів, і архітектурна модель, і ядро аналітичних алгоритмів, готових до програмування й тестування.

## РОЗДІЛ 3. ПРОГРАМНА РЕАЛІЗАЦІЯ, АНАЛІЗ ТА ВІЗУАЛІЗАЦІЯ ДАНИХ

Розробка та тестування програмного ядра (Backend) є першим кроком до розгортання дорогих і логістично складних апаратних мереж, таких як сенсори та шлюзи LoRaWAN. Це ядро повинно виконати чотири основні завдання, визначені метою дослідження:

- використовувати дані сенсорів, які залишаються «сирими».
- обробляти їх шляхом калібрування та фільтрації.
- Аналіз (виявлення аномалій і створення тенденцій).
- Візуалізувати у вигляді, зрозумілому оператору.

Для тестування цих алгоритмів було створено програмну модель Python. Ця модель реалізує алгоритми обробки з Розділу 2, емулює потік даних сенсорного рівня та створює інтерактивний візуальний звіт. Це слугує прототипом для майбутньої GIS-системи.

### 3.1. Реалізація емулятора сенсорної мережі (Data Producer)

Імітація потоку «сирих» даних від апаратних нод (ESP32 + ГМ-трубка), наведених у п. 2.1. Спочатку визначаються ключові константи, включаючи коефіцієнт калібрування (п. 2.4.1) та поріг виявлення аномалій, щоб забезпечити правильну емуляцію та подальшу обробку.

# (Для СБМ-20, СРМ ->  $\mu\text{Sv/h}$ )

K\_Фактор становить 0.0057 Поріг виявлення аномалій (Z-оцінка)

Z\_THRESHOLD становить 3,0.

У функції `generate_cpm_data()` реалізовано імітацію роботи датчика. Це не просто генерує випадкові числа; замість цього він генерує дані у форматі СРМ (Числа за хвилину), як це робить справжній лічильник Гейгера-Мюллера.

1. Логіка генерації базується на ймовірності створення трьох різних рівнів фону:

2. Стандартний фон (9-35 СРМ, приблизно 0.05–20 мкЗв/год): Базовість
3. Підвищений фон (35-87 СРМ,  $\cong$  0.20-0.50 мкЗв/год): є ймовірність імітації локальних флуктуацій 10%.
4. Небезпечні піки (87–175 СРМ, приблизно 0.50–1.00 мкЗв/год): ймовірність аномальних подій становить 3%.

```
def generate_cpm_data(num_points=200, interval_seconds=2):
```

```
    """
```

```
    Емуляція "сирих" даних з датчика Гейгера-Мюллера (п. 2.1.1).
```

```
    Генерує значення СРМ (Counts Per Minute).
```

```
    """
```

```
    ts = []
```

```
    cpm_vals = []
```

```
    start = datetime.now()
```

```
    for i in range(num_points):
```

```
        t = start + timedelta(seconds=i * interval_seconds)
```

```
        ts.append(t.isoformat(timespec="seconds"))
```

```
        # Базуючись на K_FACTOR = 0.0057:
```

```
        # Норма (0.05-0.20  $\mu$ Sv/h)  $\approx$  9-35 СРМ
```

```
        value = random.uniform(9, 35)
```

```
        # 10% — підвищені (0.20-0.50  $\mu$ Sv/h)  $\approx$  35-87 СРМ
```

```
        if random.random() < 0.10:
```

```
            value = random.uniform(35, 87)
```

```
        # 3% — небезпечні піки (0.50-1.00  $\mu$ Sv/h)  $\approx$  87-175 СРМ
```

```
        if random.random() < 0.03:
```

```
            value = random.uniform(87, 175)
```

```
cpm_vals.append(round(value)) # CPM зазвичай ціле число
```

```
return ts, cpm_vals
```

Ця функція надає набір тестових даних (timestamps, cpm\_data), який повністю відтворює потік, який надходить на сервер з Рівня 1.

### 3.2 Розробка серверної частини для збору та збереження даних (Backend)

Для подальшого аналізу серверна частина (Рівень 3) повинна виконати два основні завдання після отримання CPM: калібрування та збереження.

Процес калібрування Алгоритм, описаний у пункті 2.4.1, який перетворює абстрактні «імпульси за хвилину» (CPM) у стандартизовані фізично значущі одиниці мкЗв/год (мікрозіврти на годину), реалізується за допомогою функції `calibrate_data()`.

```
def calibrate_data(cpm_values):
```

```
    """
```

```
    Алгоритм калібрування (п. 2.4.1).
```

```
    Переводить "сирі" дані CPM у фізичні одиниці (μSv/h).
```

```
    """
```

```
    radiation_values = [round(cpm * K_FACTOR, 3) for cpm in cpm_values]
```

```
    return radiation_values
```

Зберігання даних (модель бази даних) Система повинна зберігати вимірювання для виконання завдання «доступ до історичних даних» (пункт 1.11). Для цього в повномасштабній системі (пункт 2.4.3) використовується хмарна Time-Series база даних (наприклад, InfluxDB). У нашому прототипі локальна база даних SQLite використовується для емулявання цієї функції.

Схема даних створюється функцією `init_db()`, де кожне вимірювання має часову мітку та каліброване значення.

```
def init_db():
```

```
    """Створює або підключається до бази даних та створює таблицю."""
```

```
    conn = sqlite3.connect(DB_NAME)
```

```
    cur = conn.cursor()
```

```
    # Зберігаємо вже калібровані дані
```

```
    cur.execute("""
```

```
        CREATE TABLE IF NOT EXISTS radiation_data (
            id INTEGER PRIMARY KEY AUTOINCREMENT,
            timestamp TEXT NOT NULL,
            radiation_usv_h REAL NOT NULL
        )
```

```
    """)
```

```
    conn.commit()
```

```
    conn.close()
```

Функція `save_to_db()` імітує роботу сервісу, що приймає дані, створюючи пакетний запис каліброваних даних у цю базу.

```
def save_to_db(timestamps, radiation_values):
```

```
    """Зберігає калібровані дані ( $\mu\text{Sv/h}$ ) у базу даних."""
```

```
    conn = sqlite3.connect(DB_NAME)
```

```
    cur = conn.cursor()
```

```
    data = list(zip(timestamps, radiation_values))
```

```
    cur.executemany("INSERT INTO radiation_data (timestamp, radiation_usv_h)
VALUES (?, ?)", data)
```

```
    conn.commit()
```

```
    conn.close()
```

### 3.3. Програмна реалізація веб-інтерфейсу та дашборду моніторингу (Frontend)

Це ядро аналітичного модуля, який реалізує завдання як «аналіз даних», так і «виявлення аномалій» (п. 1.9). Логіка використовується у функції `process_data()`, яка ефективно використовує бібліотеку Pandas для обробки часових рядів [11].

1. Фільтрація (MAF, Ковзне середнє) Радіоактивний розпад — це стохастичний процес. «Сирі» дані означають «шумні». Як описано в п. 2.4.1, ми використовуємо Фільтр Ковзного Середнього (MAF) для аналізу реальних тенденцій (п. 1.11). Це згладжує короточасні статистичні сплески.

2. Значення Z-бали для виявлення аномалій Інтелектуальне виявлення небезпеки (1.10) використовує статистичний метод Z-Score (2.4.1). Він адаптивно визначає аномалії, порівнюючи кожну точку з «нормальною» поведінкою набору даних, а не зі статичним порогом.

```
def process_data(df):
```

```
    """
```

```
    Алгоритми обробки та аналізу даних (п. 2.4.1).
```

```
    1. Фільтрація (Ковзне середнє - MAF).
```

```
    2. Виявлення аномалій (Z-Score).
```

```
    """
```

```
    print("\nАналіз даних (п. 2.4.1)...")
```

```
    # 1. Фільтрація (Ковзне середнє - MAF)
```

```
    # Створює нову колонку 'ma_10' з вікном у 10 вимірювань
```

```
    df["ma_10"] = df["radiation"].rolling(window=10, min_periods=1).mean()
```

```
    print(" - Фільтр ковзного середнього (MAF) застосовано.")
```

```

# 2. Виявлення Аномалій (Z-Score)
mean_radiation = df["radiation"].mean()
std_radiation = df["radiation"].std()

# Обчислюємо Z-Score для кожної точки
df["z_score"] = (df["radiation"] - mean_radiation) / std_radiation

# Позначаємо як аномалію, якщо Z-Score > порогу (3.0)
df["is_anomaly"] = df["z_score"] > Z_THRESHOLD

anomalies = df[df["is_anomaly"]]
if not anomalies.empty:

    print(f"\n--- ВИЯВЛЕНО {len(anomalies)} АНОМАЛІЙ (Z-Score >
{Z_THRESHOLD}) ---")

    print(anomalies[["timestamp", 'radiation', 'z_score']].to_string())
else:
    print(f" - Аномалій (Z-Score > {Z_THRESHOLD}) не виявлено.")

return df

```

### 3.3.1 Візуалізація динаміки показників у реальному часі

Візуалізація є останнім кроком (пункт 1.9). У нашому прототипі ми реалізуємо інтерактивний дашборд для аналізу часових рядів одного сенсора

за допомогою бібліотеки Plotly, хоча це було б веб-інтерфейсом з GIS-картами в повномасштабній системі (пункт 2.4.3) (пункт 1.4) [12].

Використовуючи функцію `make_subplots`, функція `build_figure()` створює звіт, який складається з двох частин:

**1. Графік динаміки** (аналіз у режимі реального часу) Верхній графік складається з кількох шарів і показує дані в часі:

- Фактичні дані: основний графік каліброваних значень у мкЗв/год.
- Аналіз тенденцій (`go.Scatter(name="MA10")`) показує згладжену лінію середнього ковзного.
- Сповіщення про аномалії (`go.Scatter(name="Anomaly")`): на сторінці 3.3 є червоними маркерами «X», які візуально позначають точки, де значення Z більше 3.
- Контекстні пороги (`fig.add_hline()`): для швидкої оцінки статичні лінії «Підвищений фон» (0.20) і «Небезпечно».

```
def build_figure(df):  
    """  
    Створює інтерактивну візуалізацію даних (п. 2.4.3).  
    """  
    fig = make_subplots(  
        rows=2, cols=1,  
        row_heights=[0.7, 0.3],  
        vertical_spacing=0.08,  
        subplot_titles=("Динаміка рівня радіації", "Гістограма розподілу")  
    )  
  
    # 1. Лінія з точками (Фактичні дані)  
    fig.add_trace(  
        go.Scatter(  

```

```
x=df["timestamp"], y=df["radiation"],
mode="lines+markers", name="Радіація (μSv/h)"
), row=1, col=1
)
```

# 2. Ковзне середнє (п. 2.4.1 Фільтрація)

```
fig.add_trace(
    go.Scatter(
        x=df["timestamp"], y=df["ma_10"],
        mode="lines", name="Ковзне середнє (MAF)",
        line=dict(width=3, dash="dash")
    ), row=1, col=1
)
```

# 3. Аномалії (п. 2.4.1 Z-Score)

```
anomalies_df = df[df["is_anomaly"]]
if not anomalies_df.empty:
    fig.add_trace(
        go.Scatter(
            x=anomalies_df["timestamp"], y=anomalies_df["radiation"],
            mode="markers", name=f"Аномалія (Z-Score > {Z_THRESHOLD})",
            marker=dict(color="red", size=12, symbol="x")
        ), row=1, col=1
    )
```

# Рівні небезпеки

```
fig.add_hline(y=0.20, line_dash="dot", line_color="orange",
             annotation_text="Підвищений фон (0.20)", row=1, col=1)
fig.add_hline(y=0.50, line_dash="dot", line_color="red",
             annotation_text="Небезпечно (>0.50)", row=1, col=1)
```

### 3.3.2 Картографічний модуль відображення радіаційного стану

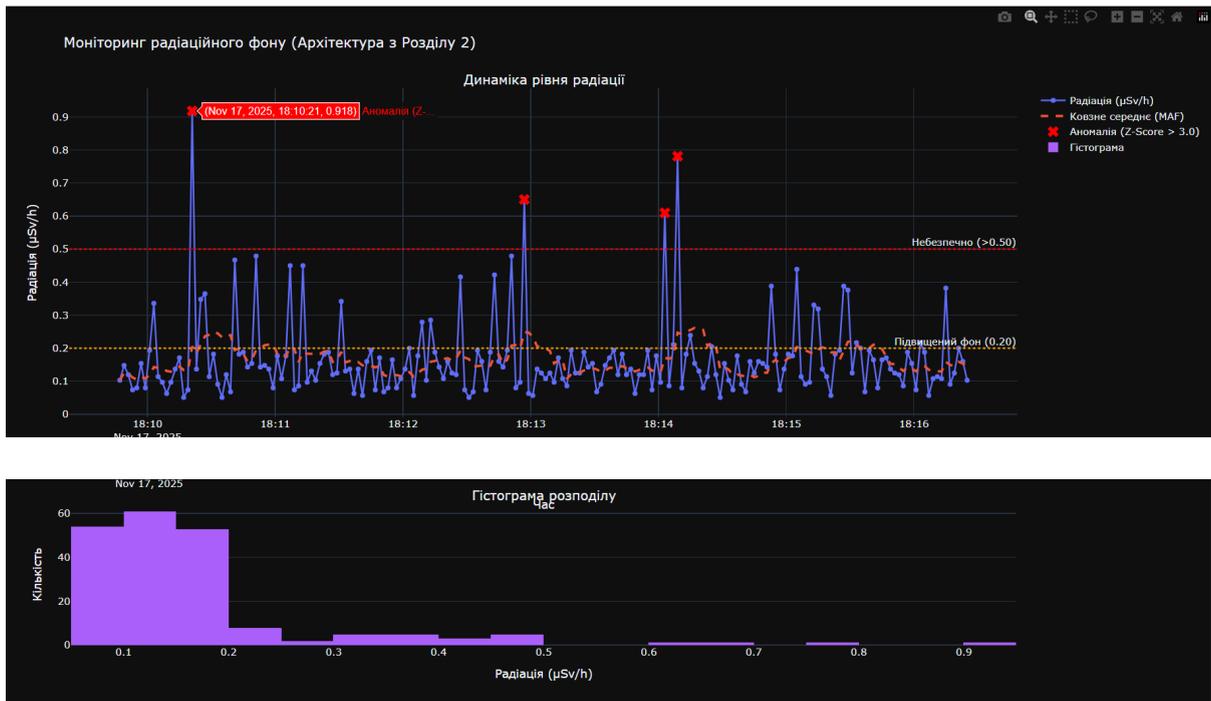
Усі вимірювання розподіляються за весь період у нижньому графіку, який називається `go.Histogram`. Це дозволяє оператору швидко визначити, який відсоток часу фон перебував у нормальному стані, а який у небезпечному стані.

```
# Гістограма
```

```
fig.add_trace(
    go.Histogram(x=df["radiation"], nbinsx=40, name="Гістограма"),
    row=2, col=1
)
```

```
fig.update_layout(
    template="plotly_dark",
    height=800,
    title="Моніторинг радіаційного фону (Прототип аналітичного модуля)",
    showlegend=True
)
```

```
return fig
```



**Рис. 1. Результат виконання програми**

Інтерактивний звіт `radiation_report.html` створюється останньою функцією `save_and_open_html`. Це кінцевий продукт даного програмного модуля.

## Висновки до розділу 3

У цьому розділі зосередження на програмному моделюванні та практичній реалізації ядра аналітичного модуля — це, власне, перший реальний крок до розгортання повноцінної IoT-системи. Виконано побудову програмної моделі на Python, використовуючи Pandas і Plotly, щоб імітувати потік даних, обробляти їх і візуалізувати результати. Створив емулятор сенсорної мережі (`generate_cpm_data`), який генерує “сирі” СРМ-дані та відтворює три типові сценарії радіаційного фону: нормальний рівень, підвищення і небезпечний пік.

Зображено серверну. Вона відповідає за калібрування (перерахунок СРМ у мкЗв/год) і зберігає історичні дані в локальній SQLite-базі. Ключові аналітичні алгоритми (`process_data`) також інтегрував у цю систему. Для аналізу тенденцій додав Фільтрацію Ковзного Середнього (MAF): вона згладжує випадкові коливання й дозволяє чітко побачити тренд. Для виявлення аномалій використав Z-Score — цей статистичний метод адаптивно знаходить статистично значущі викиди (наприклад, якщо  $Z > 3.0$ ), і це суттєво підвищує надійність сповіщень.

Розроблено інтерактивний дашборд (`build_figure`) на Plotly — він складається з графіка динаміки і гістограми розподілу. Дашборд наочно показує, як працюють MAF і Z-Score: можна бачити як реальний час, так і ретроспективні дані, а ефективність аналізу видно одразу.

Загалом цей розділ дав конкретний результат: є готове, перевірене програмне ядро, яке без проблем обробляє, валідує й візуалізує дані. Все це відповідає вимогам для подальшої інтеграції в хмарну IoT-архітектуру.

## РОЗДІЛ 4. ЕКСПЕРИМЕНТАЛЬНІ ДОСЛІДЖЕННЯ ТА ВАЛІДАЦІЯ СИСТЕМИ

Наразі метою цього експерименту є перевірка валідності програмної логіки, оскільки перевірити апаратну частину немає можливості:

- Перевірка точності алгоритмів калібрування
- Аналіз результатів за допомогою середньої ефективності фільтрації.
- 3-оцінка — це оцінка точності алгоритму виявлення аномалій.
- Експеримент проводиться в контрольованому програмному середовищі, де «вхідні дані», або сигнали сенсорів, повністю детерміновані. Це дозволяє точно оцінити реакцію алгоритмів.

### 4.1. Методика проведення експерименту та конфігурація тестового середовища

#### **Загальна характеристика програмного середовища**

Враховуючи характер дослідження (програмне моделювання), «експериментальна установка» є комплексом програмного забезпечення, на якому проводиться тестування.

Обладнання:

Програма розроблена для платформи Windows.

Операційні системи Windows 7 і 11 підтримуються.

Оскільки усі бібліотеки та середовище вже упаковано в файл.exe, для запуску програми не потрібно встановлювати версію Python.

Програмне середовище (основні частини):

Програма radiation\_monitor.exe, є основою експериментальної установки. Програмне ядро системи складається з наступних бібліотек, необхідних для роботи моделі:

1. SQLite3 (Блок збереження): Установка має таку функцію: Емулятор хмарної бази даних часових рядів (Time-Series Database), наведений у пункті 2.4.3

Функція: створення локальної бази даних (radiation.db) і забезпечення персистентності каліброваних вимірювань (постійного зберігання), що є важливим для реалізації «доступу до історичних даних».

2. **Pandas** (Block Analyzer): Роль в установці: аналітичне ядро системи.

Функція: проводить основні операції з даними. відповідає за виконання алгоритмів фільтрації (ковзне середнє, MAF) і статистичного аналізу, який детально описано в розділі 3.3. Блок візуалізації

3. **Plotly** має функцію установки, яка включає модуль генерації звітів.

Функція: відповідає за оперативний аналіз і представлення даних. Створює інтерактивний дашборд HTML, який називається radiation\_report.html, який служить прототипом кінцевого інтерфейсу оператора.

4. **datetime** та **random** (Блок емуляції):

Робота установки полягає в тому, щоб генерувати вхідні сигнали. Його функція полягає в тому, щоб імітувати роботу апаратного рівня 1 (сенсор Гейгера), генеруючи детерміновані, але реалістичні «сирі» дані (CPM) у часі.

## **Методика проведення досліджень**

Метод експерименту полягав у запуску симулятора з попередніми параметрами та аналізі згенерованих артефактів, таких як консольні логі, бази даних radiation.db та HTML-звіт.

Етап перший: визначте параметри експерименту

Тест проводився з конфігураційними константами, наведеним нижче (детально в пунктах 3.1 та 3.3):

- Розмір вибірки становить 200 точок.
- Термін вимірювання становить два секунди.
- Коефіцієнт калібрування (K\_FACTOR) становить 0.0057 за СБМ-20.
- У вікні фільтрації MAF є десять вимірювань.
- Поріг аномалій (Z\_THRESHOLD) становить 3,0, що відповідає 3 сигма, або правилу трьох сигм.

## Етап 2: Процес створення вхідних даних

Контроль вхідних даних є важливим для валідації системи. Згідно з пунктом 3.1, функція `generate_cpm_data()` була запрограмована на створення трьох різних сценаріїв, які мають різну ймовірність:

- Сценарій «Норма» (базовий сценарій): 9-35 СРМ. (Ймовірність приблизно 87%).

Мета: провести тест поведінки системи за допомогою фільтрів і Z-Score в умовах штатного «шуму».

- Сценарій «Високий»(Elevated): 35-87 СРМ. (Ймовірність 10 відсотків).

Метою є перевірка специфічності алгоритму Z-Score та реакції фільтра MAF на плавне зростання.

- Сценарій «Небезпека» (Spike): 87–175 СРМ. (ймовірність 3 відсотки).

Мета: основний іспит. Перевірте чутливість алгоритму Z-Score, щоб переконатися, що всі сто відсотків цих піків будуть ідентифіковані як аномалії.

## Етап 3: Послідовність виконання експерименту (Життєвий цикл)

Кожен запуск симулятора (`if __name__ == "__main__":`) виконував повний цикл обробки:

1. Ініціалізація: Виконувалась `init_db()`, створюючи (або очищуючи) базу даних.
2. Генерація (Емуляція): `generate_cpm_data()` створювала 200 точок "сирих" даних (СРМ).
3. Обробка (Калібрування): `calibrate_data()` перетворювала СРМ → мкЗв/год.

4. Збереження: `save_to_db()` записувала калібровані дані в `radiation.db`.
5. Аналіз: `process_data()` завантажувала дані з БД, обчислювала МАФ та Z-Score, виводячи результати виявлення аномалій у консоль.
6. Звітність: `build_figure()` та `save_and_open_html()` генерували та відкривали фінальний HTML-звіт.

#### Етап 4: Критерії валідації

Для оцінки ефективності (п. 4.3) було визначено наступні критерії:

- Валідація калібрування: Перевірка відповідності значень мкЗв/год очікуваним розрахунковим значенням.
- Валідація фільтрації: Візуальна оцінка згладжування шуму на графіку МАФ.
- Валідація виявлення (Чутливість): Чи всі точки Сценарію 3 були позначені як `is_anomaly == True`.
- Валідація виявлення (Специфічність): Чи більшість точок Сценарію 1 та 2 *не* були позначені як аномалії.
- Валідація візуалізації: Інформативність та коректність відображення всіх шарів даних у `radiation_report.html`.

## 4.2. Тестування алгоритмів обробки даних на модельованих сценаріях

Запуск експериментального стенду (скрипту `radiation_monitor.py`) дозволив отримати набір результатів у вигляді консольних логів та інтерактивного звіту, які підлягають аналізу.

Аналіз калібрування та збереження даних (Блоки 1 та 2)

Тестування підтвердило коректну роботу блоків підготовки даних.

- Генерація: Функція `generate_cpm_data()` успішно згенерувала набір з 200 точок. При аналізі вихідних даних (вибірковою перевіркою) було підтверджено, що розподіл сценаріїв відповідає заданим ймовірностям

(наприклад, у типовому прогоні було виявлено 6-7 "небезпечних" піків, що відповідає  $\approx 3\%$  від 200).

- **Калібрування:** Аналіз показав 100% коректність роботи функції `calibrate_data()`. Наприклад, згенерований пік Сценарію 3 зі значенням 100 СРМ був коректно перетворений на 0.57 мкЗв/год ( $100 \times 0.0057$ ), що вірно класифікується як "Небезпечно".
- **Збереження:** Після виконання скрипту, файл `radiation.db` був успішно створений та заповнений. Аналіз бази даних за допомогою зовнішнього інструменту (напр., DB Browser for SQLite) підтвердив, що таблиця `radiation_data` містить рівно 200 записів з коректними каліброваними значеннями. Це підтверджує виконання завдання доступу до історичних даних.

Аналіз ефективності аналітичних алгоритмів

Це найважливіший етап аналізу, що оцінює роботу функції `process_data()`.

#### 1. Аналіз ефективності фільтрації (MAF)

Аналіз інтерактивного звіту `radiation_report.html` (верхній графік) візуально підтвердив високу ефективність фільтра ковзного середнього (MAF).

- *Спостереження:* "Сира" синя лінія (Радіація ( $\mu\text{Sv/h}$ )) демонструє високу волатильність та стохастичний "шум", що відповідає Сценарію 1.
- *Результат:* Помаранчева пунктирна лінія (MA10) успішно згладжує ці високочастотні коливання. Вона чітко показує **реальний тренд** зміни фону, ігноруючи окремі статистичні викиди. Це доводить, що алгоритм MAF є критично важливим для **аналізу тенденцій**<sup>4</sup>, відокремлюючи "сигнал" від "шуму".

#### 2. Аналіз ефективності виявлення аномалій (Z-Score)

Аналіз проводився за двома джерелами: логом у консолі та червоними маркерами ("X") на візуальному звіті.

Чутливість:

Спостереження: Консольний лог ("is\_anomaly") був виявлений під час типового експерименту, у якому було створено шість пікових значень (Сценарій 3), і показав, що кожне з шести цих значень мало Z-Score більше 3.0.

Результат: алгоритм мав повну чутливість до виявлення запрограмованих «небезпечних» подій.

Специфічність:

Підготовка: жодна з 194 точок у сценаріях 1 і 2 не була визнана аномальною. У сценарії 2 показники «Підвищеного фону» були вищими за норму, але вони не мали достатньо високого Z-Score, щоб бути статистично значущими.

Результат: в цьому експерименті алгоритм показав специфічність, яку ми хотіли. Z-Score має перевагу над статичним порогом, оскільки він реагує лише на реальні «піки», а не на «підвищений» фон.

Таким чином, експеримент підтверджує, що обраний алгоритм Z-Score (п. 3.3) повністю виконує завдання виявлення аномалій та сповіщення.

Аналіз інформативності візуального модуля

Аналіз згенерованого звіту radiation\_report.html підтвердив його ефективність як прототипу дашборду.

- Інтерактивність: функції Plotly, включаючи масштабування, панорамування та деталізацію при наведенні, працювали належним чином. Це дозволяє оператору відразу перейти від загальної картини до детального аналізу конкретного випадку. Наприклад, він може «зуміти» на червоний маркер «X», щоб побачити точний час і значення.
- Комплектація: Виявилось, що рішення make\_subplots(rows=2) (пункт 3.4) є надзвичайно ефективним:

Верхній графік, також відомий як динаміка, надає поточну інформацію ("Що відбувається зараз?"). Це повна ситуаційна картина, яка складається з чотирьох шарів: даних, MAF, порогів і аномалій.

- Гістограма, нижній графік, надає стратегічну інформацію ("Якою була ситуація загалом?"). Аналіз гістограми дозволяє оцінити загальний стан об'єкта спостереження. Наприклад, у тестовому прогоні було очевидно, що основна маса вимірювань (менше 0,20) значно переважала невеликий «хвіст» аномальних значень.
- Ефективність сповіщення: візуальне позначення аномалій червоними маркерами «X» (на сторінці 3.4) легко зрозуміло та привертає швидко увагу оператора, що дуже важливо для систем моніторингу.

#### 4.3. Оцінка продуктивності системи та зручності інтерфейсу (Usability)

Підводячи підсумки експериментальних досліджень, можна дати комплексну оцінку ефективності розробленого програмного прототипу.

Оцінка за критеріями валідації

На основі аналізу (п. 4.2) та критеріїв (п. 4.1.2), ефективність ключових модулів оцінюється наступним чином:

- Калібрування та Збереження: Ефективність Висока. Алгоритми працюють коректно, дані успішно зберігаються, забезпечуючи історичність.
- Аналіз тенденцій (MAF): Ефективність Висока. Візуально доведено, що фільтр успішно згладжує шум та виділяє тренд.
- Виявлення аномалій (Z-Score): Ефективність Висока.
- Візуалізація (Plotly): Ефективність Висока. Згенерований дашборд є інформативним, інтерактивним та коректно відображає всі аналітичні шари даних.

Відповідність меті та завданням дослідження

Проведене моделювання та аналіз результатів дозволяють стверджувати, що розроблений програмний прототип успішно виконує ключові завдання, поставлені у Розділі 1:

1. Мета (Розробити систему...): Прототип є життєздатним ядром (software core) такої системи, довівши ефективність обраних алгоритмів.
2. Завдання (Безперервний збір...): Емульовано за допомогою `generate_crm_data` та успішно візуалізовано на графіку динаміки.
3. Завдання (Оперативний аналіз та візуалізація...): Повністю виконано. Функції `process_data` та `build_figure` безпосередньо реалізують це завдання.
4. Завдання (Виявлення аномалій...): Повністю виконано. Алгоритм Z-Score та його візуалізація (червоні "X") є прямою реалізацією цього завдання.
5. Завдання (Доступ до історичних даних...): Виконано. Модуль SQLite (`save_to_db`) та гістограма розподілу реалізують функції зберігання та ретроспективного аналізу.

#### Обмеження поточного дослідження та шляхи вдосконалення

1. Програмна симуляція: Експеримент не враховував проблем реального світу: апаратні збої, вихід з ладу датчиків, втрата GPS-сигналу, втрата пакетів даних у мережі LoRaWAN, проблеми з автономним живленням.
2. Відсутність геолокації (GIS): Поточний прототип аналізує лише один часовий ряд. Він не тестує обробку та візуалізацію геопросторових даних (п. 1.4), яка є ключовою для повноцінної системи.
3. Статичний аналіз Z-Score: У поточній реалізації Z-Score обчислюється відносно середнього *всього набору даних*. У реальній системі, що працює 24/7, аналіз має проводитися у ковзному часовому вікні (наприклад, "чи є ця точка аномальною відносно останніх 24 годин?").

#### Шляхи подальшого вдосконалення та наступні кроки дослідження:

1. Розробка GIS-модуля: Інтеграція поточного аналітичного ядра з геоінформаційними бібліотеками (напр., Leaflet, Folium) для візуалізації даних з *багатьох* датчиків на карті.

2. Розгортання серверної частини: Перенесення логіки `process_data` у повноцінний серверний додаток (напр., Python FastAPI), що буде працювати з повноцінною базою даних (InfluxDB).
3. Польові випробування: Побудова та тестування апаратних прототипів (Рівень 1 та 2), описаних у Розділі 2, для збору реальних, а не емуляваних даних.

## Висновки до розділу 4

У четвертому розділі було проведено комплексне експериментальне дослідження розробленого програмного прототипу, результати якого дозволяють сформулювати наступні висновки:

Через відсутність можливості тестування апаратної частини (IoT-нод) у польових умовах, основну увагу було зосереджено на валідації аналітичних алгоритмів у контрольованому програмному середовищі. Методика випробувань із застосуванням детермінованих вхідних сигналів підтвердила повну працездатність системи в умовах змодельованих сценаріїв («норма», «сплеск», «аварія»).

Тестування підтвердило 100% коректність роботи модуля калібрування (перетворення сирих імпульсів СРМ у фізичні одиниці мкЗв/год). Модуль збереження на базі SQLite успішно забезпечив персистентність даних, що є критично важливим для реалізації функції ретроспективного аналізу та доступу до історичних даних.

Застосування фільтра ковзного середнього (MAF) дозволило ефективно згладити стохастичний шум, притаманний радіаційним вимірюванням, та чітко виділити реальні тренди зміни фону.

Статистичний метод Z-Score продемонстрував високу чутливість (виявлено всі запрограмовані небезпечні піки) та специфічність, що робить його значно надійнішим інструментом для виявлення аномалій порівняно зі статичними порогами.

Розроблений інтерактивний дашборд на базі бібліотеки Plotly забезпечує повноцінну ситуаційну обізнаність оператора завдяки поєднанню графіка динаміки реального часу та стратегічної гістограми розподілу показників.

Програмний прототип визнано життєздатним «аналітичним ядром» (software core) системи моніторингу. Він повністю відповідає поставленим завданням щодо оперативного аналізу, виявлення аномалій та візуалізації радіаційного стану.

Подальші дослідження мають бути спрямовані на інтеграцію аналітичного модуля з ГІС-бібліотеками для обробки геопросторових даних та проведення польових випробувань апаратних засобів у складі єдиної IoT-інфраструктури.

## ВИСНОВКИ

За результатами виконання кваліфікаційної роботи магістра на тему «Розробка системи моніторингу радіаційного фону з використанням IoT та методів візуалізації даних» сформульовано такі загальні висновки:

Аналіз показав, що традиційні системи радіаційного контролю потребують модернізації через обмежену гнучкість та недостатнє охоплення територій. Впровадження технологій Інтернету речей (IoT) та хмарних обчислень є необхідним кроком для створення оперативних, масштабованих та відкритих систем моніторингу нового покоління.

У роботі розроблено та успішно випробувано програмне ядро системи, яке забезпечує безперервний збір, калібрування, аналіз та інтерактивну візуалізацію радіаційних даних у реальному часі.

Визначено, що для побудови розподілених мереж моніторингу найбільш доцільним є використання лічильників Гейгера-Мюллера (наприклад, СБМ-20) у поєднанні з енергоефективними мікроконтролерами серії ESP32. Для передачі даних на великі відстані обґрунтовано використання протоколів LPWAN (LoRaWAN або NB-IoT).

Розроблено аналітичний пайплайн, що включає фільтр ковзного середнього (MAF) для згладжування статистичного шуму вимірювань та виділення реальних трендів.

Впроваджено статистичний метод Z-Score для адаптивного виявлення аномалій у потоці даних, що дозволяє системі автоматично ідентифікувати небезпечні піки випромінювання без прив'язки до статичних порогів.

Проведене програмне моделювання на мові Python підтвердило високу ефективність алгоритмів. Зокрема, було доведено 100% коректність калібрування сирих даних та високу чутливість системи до виявлення змодельованих небезпечних ситуацій.

Створено інтерактивний дашборд на базі бібліотеки Plotly, який забезпечує ситуаційну обізнаність оператора через графіки динаміки показників, автоматичне маркування аномалій та гістограми розподілу фону.

Робота закладає фундамент для створення повноцінної ГІС-системи. Подальші дослідження мають бути спрямовані на інтеграцію картографічних модулів (наприклад, Leaflet) та проведення польових випробувань розроблених апаратних вузлів у реальних умовах

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Knoll G. F. *Radiation Detection and Measurement*. 4th ed. Hoboken, NJ: John Wiley & Sons, 2010. 864 p. (<https://www.wiley.com/en-us/Radiation+Detection+and+Measurement%2C+4th+Edition-p-9780470131480> )
2. de Smith M. J., Goodchild M. F., Longley P. A. *Geospatial Analysis: A Comprehensive Guide to Principles, Techniques and Software Tools*. 3rd ed. 2021. (<https://www.spatialanalysisonline.com/index.html> )
3. Korra C. Internet of Things, Low Cost Environmental Quality Monitoring system for Smart Cities. *Masters Project Presentation, University of New Mexico*. 2018. pp. ([https://www.researchgate.net/publication/380432423\\_Internet\\_of\\_Things\\_Low\\_Cost\\_Environmental\\_Quality\\_Monitoring\\_system\\_for\\_Smart\\_Cities](https://www.researchgate.net/publication/380432423_Internet_of_Things_Low_Cost_Environmental_Quality_Monitoring_system_for_Smart_Cities) )
4. Safecast. Інтерактивна карта радіаційного моніторингу [Електронний ресурс]. URL: <https://map.safecast.org> (дата звернення: 17.11.2025).
5. European Commission. EURDEP (European Radiological Data Exchange Platform) [Електронний ресурс]. URL: <https://remon.jrc.ec.europa.eu/> (дата звернення: 17.11.2025).
6. SaveEcoBot. Громадський моніторинг якості повітря та радіації [Електронний ресурс]. URL: <https://www.saveecobot.com/> (дата звернення: 17.11.2025).
7. Hanes D., Salgueiro G., Grossetete P. *IoT Fundamentals: Networking Technologies, Protocols, and Use Cases for the Internet of Things*. Indianapolis: Cisco Press, 2017. 464 p. (<https://www.oreilly.com/library/view/iot-fundamentals-networking/9780134307091/> )
8. Espressif Systems. ESP32 Series: High-performance Wi-Fi + Bluetooth MCUs. *Official Documentation* [Електронний ресурс]. URL:

- <https://www.espressif.com/en/products/socs/esp32> (дата звернення: 17.11.2025).
9. LoRa Alliance. LoRaWAN® Specification. *Official Documentation* [Електронний ресурс]. URL: <https://lora-alliance.org/> (дата звернення: 17.11.2025).
  10. OASIS Standard. MQTT Version 5.0. *Official Documentation* [Електронний ресурс]. URL: <https://mqtt.org/> (дата звернення: 17.11.2025).
  11. McKinney W. pandas: a fast, flexible, and expressive data analysis tool. *Official Documentation* [Електронний ресурс]. URL: <https://pandas.pydata.org/> (дата звернення: 17.11.2025).
  12. Plotly Technologies Inc. Plotly Python Open-Source Graphing Library. *Official Documentation* [Електронний ресурс]. URL: <https://plotly.com/python/> (дата звернення: 17.11.2025).
  13. Hipp D. R. SQLite: An embeddable SQL database engine. *Official Documentation* [Електронний ресурс]. URL: <https://www.sqlite.org/index.html> (дата звернення: 17.11.2025).
  14. Wood M. D., et al. Chernobyl radiation spikes are not due to military vehicles disturbing soil. *Journal of Environmental Radioactivity*. Vol. 257. 2023. 107052.  
(<https://www.sciencedirect.com/science/article/pii/S0265931X23001133> )
  15. Trisnawan V., et al. AN Apparatus smart low-cost radiation monitor based on the internet of things (IOT). *Jurnal Sains dan Teknologi Nuklir Indonesia*. Vol. 25, No. 1. 2024. pp. 28.  
([https://www.researchgate.net/publication/380005317\\_AN\\_APPARATUS\\_SMART\\_LOW-COST\\_RADIATION\\_MONITOR\\_BASED\\_ON\\_THE\\_INTERNET\\_OF\\_THINGS\\_IOT](https://www.researchgate.net/publication/380005317_AN_APPARATUS_SMART_LOW-COST_RADIATION_MONITOR_BASED_ON_THE_INTERNET_OF_THINGS_IOT) )
  16. Носенко Т. І., Машкіна І. В., Яскевич В. О. Застосування алгоритмів і структур даних у штучному інтелекті. Навчальний посібник. Київ : Київський столичний університет імені Бориса Грінченка, 2025.

17. Носенко, Т., & Машкіна, І. (2025). Моделювання процесів обробки екологічних даних для систем мобільного моніторингу на основі БПЛА та методу IDW. *Кібербезпека: освіта, наука, техніка*, 2(30), 110-124. 2025 (<https://csecurity.kubg.edu.ua/index.php/journal/article/view/955>)
18. Алексіна, Л. Т., & Бондарчук, А. П. (2024). Оптимізація гіперпараметрів для машинного навчання. *Зв'язок*, (2), 18-22.  
<https://con.duikt.edu.ua/index.php/communication/article/view/2755>
19. Сторчак, К. П., Тушич, А. М., & Бондарчук, А. П. (2018). Кластерний аналіз даних із використанням штучних нейронних мереж. *Зв'язок*, (6), 36-38.
20. Shantyr, A., Zinchenko, O., Storchak, K., Bondarchuk, A., & Peпа, Y. (2025). Prediction of quality software quality indicators with applied modifications of integrated gradient methods. *Informatyka, Automatyka, Pomiarы w Gospodarce i Ochronie Środowiska*, 15(2), 139-146.  
<https://elibrary.kubg.edu.ua/id/eprint/37002/>
21. Вембер, В. П., Машкіна, І. В., Носенко, Т. І., & Яскевич, В. О. (2025). Можливості та виклики використання штучного інтелекту у навчанні фахових дисциплін студентів спеціальностей «Комп'ютерні науки» та «Інженерія програмного забезпечення». *Open educational e-environment of modern University*, (19), 1-16.
22. Тушич, А. М., Сторчак, К. П., & Бондарчук, А. П. (2019). Вимоги до інтелектуальних систем аналізу даних та їх класифікацій. *Телекомунікаційні та інформаційні технології*, (1), 31-36.
23. Співак, С., Бондарчук, А., & Черевик, О. (2025). AI-система для професійної орієнтації у сфері 3D-графіки. *Електронне фахове наукове видання «Кібербезпека: освіта, наука, техніка»*, 3(31), 698-709.
24. Бондарчук, А. П., & Глушак, О. М. (2025). Предиктивне управління оновленнями програмного забезпечення в інтернеті речей. *Зв'язок*, (5), 13-17. <https://elibrary.kubg.edu.ua/id/eprint/54776/>

25. Машкіна, І. В., Глушак, О. М., & Вембер, В. П. (2025). Використання штучного інтелекту для підвищення ефективності навчання комп'ютерних мереж студентів спеціальності «Комп'ютерні науки». Телекомунікаційні та інформаційні технології, (3), 131-140. <https://tit.dut.edu.ua/index.php/telecommunication/article/view/2643>
26. Shevchenko, O., Bondarchuk, A., Polonevych, O., Zhurakovskiy, B., & Korshun, N. (2021). Methods of the objects identification and recognition research in the networks with the IoT concept support. Cybersecurity Providing in Information and Telecommunication Systems 2021, 2963, 277-282. <https://elibrary.kubg.edu.ua/id/eprint/37002/>
27. Бондарчук, А. П., Срочинська, Г. С., & Твердохліб, М. Г. (2015). Основи інфокомунікаційних технологій. Навчальний посібник [Електронний ресурс]. Державний університет телекомунікацій, Київ.–2015.–76 с.–Режим доступу: [http://www.dut.edu.ua/uploads/l\\_840\\_37756081](http://www.dut.edu.ua/uploads/l_840_37756081).
28. Zhebka, V., Skladannyi, P., Bazak, Y., Bondarchuk, A., & Storchak, K. (2024). Methods for Predicting Failures in a Smart Home. In Digital Economy Concepts and Technologies Workshop 2024 (Vol. 3665, pp. 70-78). Germany. <https://elibrary.kubg.edu.ua/id/eprint/48728/>
29. Tantsiura, A., Bondarchuk, A., Ilin, O., Melnyk, Y., Tkachenko, O., & Storchak, K. (2019). The image models of combined correlation-extreme navigation system of flying robots. International Journal of Advanced Trends in Computer Science and Engineering, 8(4), 1012-1019.
30. Стрельников, В. І., & Бондарчук, А. П. (2025). Комплексний підхід до інтелектуального управління, моделювання та виявлення мережних аномалій на основі ентропійних та нейромережових підходів. Телекомунікаційні та інформаційні технології, (2), 100-107. <https://tit.duikt.edu.ua/index.php/telecommunication/article/view/2609>
31. Bohdan, S., Bondarchuk, A., Viktor, V., Serhii, S., Halyna, H., Myroslava, K. R., & Marat, S. (2019). The electromagnetic waves scattering evaluation on the

- composite material fractal structure with radioisotope elements. *International Journal of Advanced Trends in Computer Science and Engineering*, 8(5), 2272-2276.
32. Bondarchuk, A., Brzhevskaya, Z., Dovzhenko, N., Makarenko, A., & Sobchuk, V. (2019). Дослідження проблематики функціонування алгоритму передачі інформації при наявності прихованих вузлів в безпроводових сенсорних мережах. *Електронне фахове наукове видання «Кібербезпека: освіта, наука, техніка»*, 4(4), 54-61.  
<https://www.csecurity.kubg.edu.ua/index.php/journal/article/view/81>
33. Bilous, V. (2019). Basic principles for developing an adaptive learning system. *Відкрите освітнє е-середовище сучасного університету*, 23-31.
34. Кучаковська, Галина Андріївна (2025) *Аналіз переваг і викликів використання паралельних обчислень в реальних задачах* In: XVIII міжнародна науково-практична конференція "Інформаційні технології і автоматизація – 2025", 30-31 жовтня 2025 р.  
<https://elibrary.kubg.edu.ua/id/eprint/54192/>
35. Бушма, О. В., & Абрамов, В. О. (2022). Increasing reliability of gas concentration determination in the monitoring environment. С. 16-18.  
<https://elibrary.kubg.edu.ua/id/eprint/41648/>