

Харківський національний університет радіоелектроніки

Факультет _____ комп'ютерної інженерії та управління _____

Кафедра _____ електронних обчислювальних машин _____

Рівень вищої освіти _____ перший (бакалаврський) _____

Спеціальність _____ 123 «Комп'ютерна інженерія» _____
(код і повна назва)

Тип програми _____ освітньо-професійна _____
(освітньо-професійна або освітньо-наукова)

Освітня програма _____ Комп'ютерна інженерія _____
(повна назва)

ЗАТВЕРДЖУЮ:

Зав. кафедри _____
(підпис)

“ _____ ” _____ 20__ р.

ЗАВДАННЯ

НА КВАЛІФІКАЦІЙНУ РОБОТУ

здобувачеві _____ Готрі Герману Івановичу _____
(прізвище, ім'я, по батькові)

1. Тема роботи _____ Інформаційна система обробки даних з безпроводної мережі _____

затверджена наказом по університету від “ 26 ” травня 2025 р. № 425 Ст

2. Термін подання здобувачем роботи до екзаменаційної комісії _____ 17 червня 2025 р.

3. Вхідні дані до роботи _____
інформаційна система

Google Colab

обробка даних

Matlab

Безпроводна мережа

4. Перелік питань, що потрібно опрацювати у роботі _____

Аналіз предметної області

Проектування інформаційної системи обробки даних із безпроводної мережі

Програмна реалізація системи в середовищах Matlab та Google Colab

Тестування та аналіз результатів

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій 12 слайдів

6. Консультанти розділів роботи (заповнюється за наявності консультантів згідно з наказом, зазначеним у п.1)

Найменування розділу	Консультант (посада, прізвище, ім'я, по батькові)	Позначка консультанта про виконання розділу	
		підпис	дата

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Строк / терміни виконання етапів роботи	Примітка
1	Отримання завдання та аналіз літератури	09.06.2025–11.06.2025	
2	Огляд існуючих рішень та алгоритмів	12.06.2025–15.06.2025	
3	Вибір датасетів та архітектури системи	16.06.2025–26.06.2025	
4	Вибір програмних засобів	27.06.2025–28.06.2025	
5	Програмна реалізація	29.06.2025–02.07.2025	
6	Аналіз отриманих результатів	03.07.2025–10.07.2025	
7	Оформлення записки	11.07.2025–14.07.2025	

Дата видачі завдання “ 09 ” червня 2025 р.

Здобувач


(підпис)

Керівник роботи


(підпис)

ст. викл. Владислав ДЯЧЕНКО

(посада, власне ім'я, прізвище)

РЕФЕРАТ

Пояснювальна записка кваліфікаційної роботи: 63 с., 17 рис., 2 дод., 12 джерел.

ІНФОРМАЦІЙНА СИСТЕМА, СЕНСОРНІ ДАНІ, БЕЗПРОВІДНА МЕРЕЖА, MATLAB, GOOGLE COLAB, ОБРОБКА СИГНАЛІВ, ЗГЛАДЖУВАННЯ, ВИЯВЛЕННЯ АНОМАЛІЙ, СПЕКТРАЛЬНИЙ АНАЛІЗ, ТЕМПЕРАТУРНІ ДАНІ, ТИСК, ВОЛОГІСТЬ, КОРЕЛЯЦІЙНИЙ АНАЛІЗ, АНАЛІЗ ЧАСОВИХ РЯДІВ, FFT, ІОТ, ОБРОБКА ДАНИХ.

Метою кваліфікаційної роботи є створення функціональної інформаційної системи, яка забезпечує обробку та візуалізацію даних, отриманих з безпроводної мережі, з використанням програмних засобів Matlab і Google Colab.

У ході виконання кваліфікаційної роботи розроблено та реалізовано інформаційну систему обробки даних з безпроводної сенсорної мережі, яка функціонує на базі двох комплементарних середовищ: Matlab та Google Colab. Архітектура системи побудована з урахуванням функціонального розмежування: Matlab відповідає за моделювання даних, первинну обробку та базову візуалізацію, тоді як Google Colab реалізує розширену інтелектуальну аналітику, включаючи кореляційний аналіз, виявлення аномалій, спектральну декомпозицію та трендову оцінку. Отримані результати підтверджують працездатність розробленої інформаційної системи, її здатність адаптуватися до різних джерел даних, виявляти ключові закономірності та забезпечувати основу для подальшого розширення з метою прогнозування, автоматичного моніторингу або інтеграції в складніші IoT-інфраструктури. Розділення обробки між Matlab та Colab дозволило поєднати інженерну точність із гнучкістю машинного аналізу.

ABSTRACT

Bachelor's thesis: 63 pages, 17 figures, 2 appendices, 12 sources.

INFORMATION SYSTEM, SENSOR DATA, WIRELESS NETWORK, MATLAB, GOOGLE COLAB, SIGNAL PROCESSING, SMOOTHING, ANOMALY DETECTION, SPECTRAL ANALYSIS, TEMPERATURE DATA, PRESSURE, HUMIDITY, CORRELATION ANALYSIS, TIME SERIES ANALYSIS, FFT, IOT, DATA PROCESSING.

The major goal of this thesis is to develop a functional information system that enables the processing and visualization of data obtained from a wireless network using MATLAB and Google Colab tools.

In order to an information system for processing data from a wireless sensor network was designed and developed. The system operates on the basis of two complementary environments: MATLAB and Google Colab. The system architecture is structured with functional separation: MATLAB is responsible for data modeling, initial processing, and basic visualization, while Google Colab provides extended intelligent analytics, including correlation analysis, anomaly detection, spectral decomposition, and trend evaluation. The results obtained confirm the operability of the developed information system, its adaptability to various data sources, its ability to identify key patterns, and its suitability as a foundation for further expansion toward forecasting, automated monitoring, or integration into more complex IoT infrastructures. The division of processing between MATLAB and Colab made it possible to combine engineering precision with the flexibility of machine-based analysis.

ЗМІСТ

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ	8
ВСТУП	9
1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ	11
1.1 Особливості сучасних безпроводних мереж	11
1.2 Протоколи передачі даних у безпроводних мережах.....	12
1.3 Сучасні інформаційні системи обробки даних	15
1.4 Огляд методів і засобів обробки даних, отриманих з безпроводних мереж	17
1.5 Огляд існуючих програмних засобів.....	19
2 ПРОЕКТУВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ ОБРОБКИ ДАНИХ ІЗ БЕЗПРОВІДНОЇ МЕРЕЖІ	23
2.1 Вимоги до інформаційної системи.....	23
2.2 Структурна схема інформаційної системи	25
2.3 Вибір та обґрунтування апаратних і програмних засобів.....	27
3 ПРОГРАМНА РЕАЛІЗАЦІЯ СИСТЕМИ В СЕРЕДОВИЩАХ MATLAB ТА GOOGLE COLAB.....	29
3.1 Архітектура інформаційної системи обробки даних з безпроводної мережі	30
3.2 Особливості середовищ Matlab та Colab для розробки інформаційних систем	32
3.3 Структура програмного забезпечення в Matlab	34
3.4 Реалізація обробки даних та візуалізація результатів у Matlab.....	36
3.5 Розробка та реалізація системи обробки даних в середовищі Google Colab.....	38
4 ТЕСТУВАННЯ ТА АНАЛІЗ РЕЗУЛЬТАТІВ	41
4.1 Методика тестування інформаційної системи	41
4.2 Аналіз отриманих результатів в Matlab	42

4.3 Аналіз отриманих результатів в Google Colab.....	45
ВИСНОВКИ.....	49
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ	51
ДОДАТОК А Графічний матеріал кваліфікаційної роботи.....	53
ДОДАТОК Б Програмний код.....	60
Б.1 Лістинг файлу Gotra.m	60
Б.2 Лістинг коду в Google Colab.....	62

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ

API – програмний інтерфейс прикладного програмування

CNN – згортоква нейронна мережа

Colab – Google Colaboratory

CSV – формат текстового файлу зі значеннями, розділеними комами

CPU – центральний процесор

FFT – швидке перетворення Фур'є

GUI – графічний інтерфейс користувача

IoT – Інтернет речей

JSON – формат обміну даними JavaScript Object Notation

LSTM – довготривала короткочасна пам'ять

MAT – формат файлів Matlab, що містить змінні середовища

ML – машинне навчання

PCA – аналіз головних компонент

RAM – оперативна пам'ять

RMS – середньоквадратичне значення

SNR – співвідношення сигнал/шум

WSN – безпроводна сенсорна мережа

ВСТУП

У сучасному світі розвиток безпроводних технологій набув особливого значення, охоплюючи найрізноманітніші сфери діяльності: від промислових процесів до побутових застосунків. У зв'язку з поширенням пристроїв Інтернету речей (IoT), сенсорних мереж і мобільних систем, суттєво зросла потреба в ефективному зборі, передачі, обробці та аналізі великого обсягу даних у реальному часі. В умовах зростаючої складності інформаційних потоків особливу актуальність набувають інформаційні системи, здатні працювати з безпроводними мережами, оптимізуючи процеси обробки та представлення отриманої інформації.

У цій кваліфікаційній роботі основну увагу зосереджено на розробці інформаційної системи для обробки даних, які надходять із безпроводної мережі. Основна ідея полягає в поєднанні інструментальних засобів для збору, фільтрації, аналізу та візуалізації даних у зручній і гнучкій формі. Практична реалізація системи здійснюється за допомогою програмних середовищ Matlab і Google Colab. Ці середовища надають широкі можливості для роботи з числовими даними, машинного навчання, обробки сигналів і побудови графіків, а також дозволяють адаптувати систему до різних обчислювальних платформ.

Метою роботи є створення функціональної інформаційної системи, яка забезпечує обробку та візуалізацію даних, отриманих з безпроводної мережі, з використанням програмних засобів Matlab і Google Colab.

В процесі реалізації ставляться наступні завдання:

- проаналізувати принципи організації безпроводних мереж;
- визначити оптимальні методи обробки даних;
- розробити алгоритми їх представлення;
- перевірити працездатність створеної системи у тестовому середовищі.

Особливу увагу приділено аналізу продуктивності системи та можливостям її подальшого розширення.

Варто зазначити, робота за такою актуальною тематикою може мати не лише теоретичну, але й практичну цінність, оскільки являє собою інтеграцію сучасних технологій для вирішення задач, що постають при роботі з безпроводними інформаційними потоками.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Особливості сучасних безпроводних мереж

Сучасні безпроводні мережі є невід'ємною складовою інформаційної інфраструктури, яка забезпечує мобільність, гнучкість та масштабованість процесів передачі даних. З огляду на інтенсивний розвиток цифрових технологій, зростання обсягів даних і підвищення вимог до швидкості обміну інформацією, безпроводні мережі набули особливої актуальності у сфері телекомунікацій, автоматизації, промислових систем керування, екологічного моніторингу та побутових рішень. Ці мережі дозволяють організовувати зв'язок між пристроями без фізичних кабельних з'єднань, що істотно знижує витрати на інфраструктуру, спрощує розгортання систем і створює можливості для мобільного використання.

У технологічному аспекті безпроводні мережі базуються на використанні радіочастотного спектру, інфрачервоного випромінювання або лазерного зв'язку для передавання даних. Основою їхньої архітектури є вузли – пристрої, здатні приймати, передавати та іноді обробляти інформацію. Залежно від специфіки мережі, вузли можуть виступати як сенсори, контролери, ретранслятори або кінцеві точки. Різноманіття протоколів, зокрема IEEE 802.11, Zigbee, Bluetooth, LoRaWAN та інших, дозволяє адаптувати мережі до конкретних умов використання, таких як відстань передачі, енергоспоживання, пропускна здатність чи кількість підключених пристроїв.

Суттєвою рисою сучасних безпроводних мереж є динамічність їхньої топології, яка змінюється відповідно до фізичного розташування вузлів, змін умов навколишнього середовища або енергетичних обмежень. Це зумовлює потребу в адаптивних алгоритмах маршрутизації, які здатні забезпечити ефективну доставку пакетів у непередбачуваних умовах. Крім того, мережі

повинні гарантувати надійність передачі даних, стійкість до втрат, збоїв і втручання, що вимагає застосування методів корекції помилок, шифрування, автентифікації та механізмів захисту від атак.

Особливу увагу в розвитку безпроводних мереж приділяють питанням енергозбереження, оскільки значна частина вузлів працює від автономних джерел живлення. У зв'язку з цим розробляються спеціалізовані енергетично ефективні протоколи комунікації, методи зменшення інтенсивності передачі даних, зменшення часу активної роботи пристроїв та оптимізації обчислювального навантаження.

Не менш важливим аспектом є забезпечення масштабованості мереж, що дозволяє ефективно обслуговувати сотні й тисячі пристроїв без погіршення якості зв'язку чи зниження продуктивності. Така властивість особливо важлива для великих сенсорних мереж, які використовуються у «розумних» містах, аграрному моніторингу або системах громадської безпеки. Водночас, зростання кількості підключених пристроїв вимагає оптимізації протоколів доступу до середовища передачі та ефективного управління частотними ресурсами.

Загалом, сучасні безпроводні мережі є складними технічними системами, що поєднують апаратні, програмні та математичні компоненти, які взаємодіють у реальному часі. Їхнє вивчення, аналіз і подальший розвиток потребують міждисциплінарного підходу, який охоплює комп'ютерні науки, електроніку, телекомунікації та прикладну математику. Саме тому створення ефективної інформаційної системи обробки даних, здатної працювати з такими мережами, є не лише технічним завданням, а й актуальним викликом у контексті цифровізації сучасного суспільства.

1.2 Протоколи передачі даних у безпроводних мережах

Безпроводні мережі функціонують на основі спеціалізованих протоколів, які регламентують порядок, формат і спосіб передавання

інформації між пристроями. Ці протоколи (рисунок 1.1) забезпечують організацію взаємодії на різних рівнях моделі OSI, включаючи фізичний, каналний, мережевий і транспортний рівні, з метою забезпечення надійної, ефективної та безпечної передачі даних у середовищі з потенційними втратами, перешкодами та обмеженням енергетичних ресурсів.

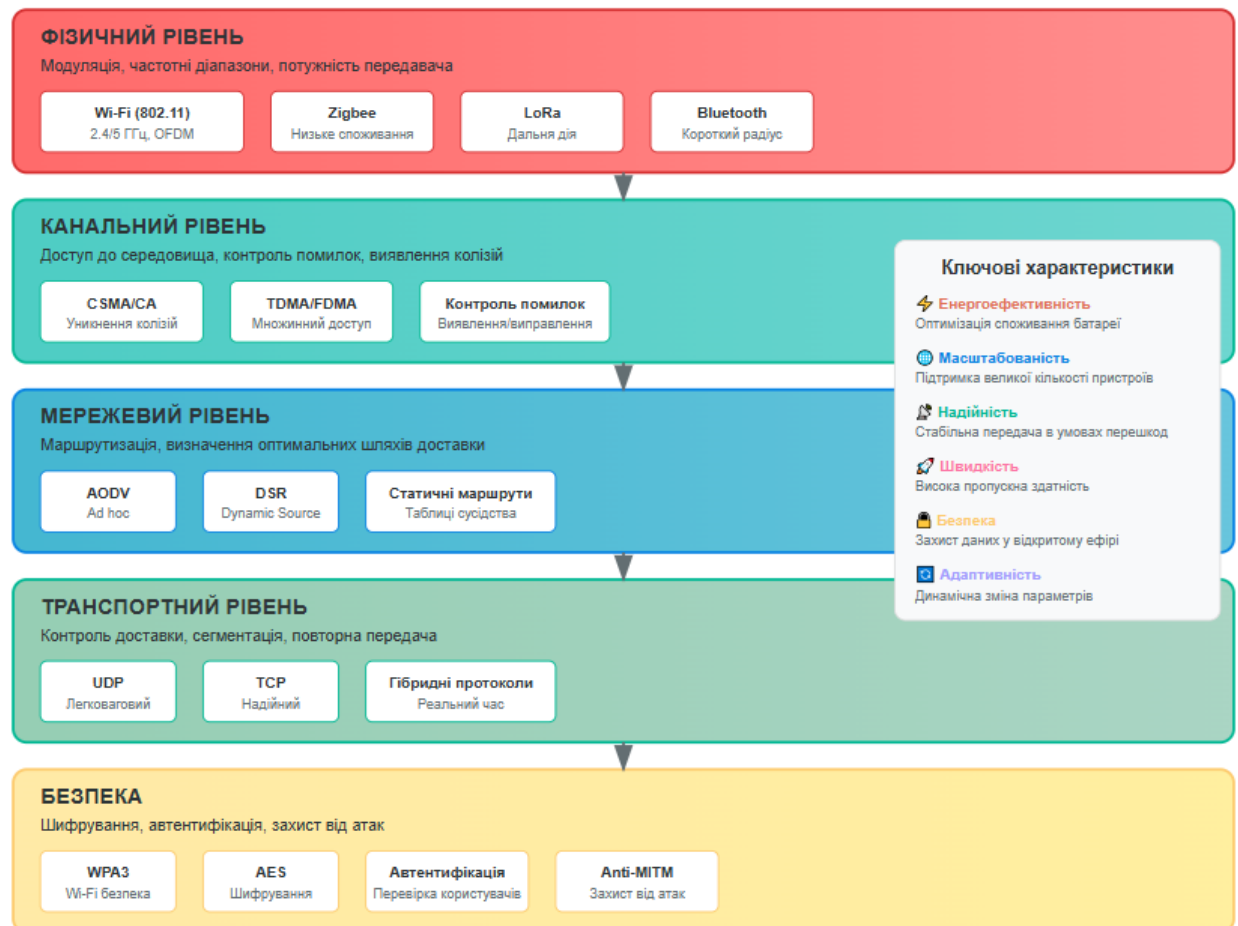


Рисунок 1.1 – Протоколи передачі даних у безпроводних мережах

Функціонування будь-якої безпроводної мережі починається на фізичному рівні, де визначаються параметри модуляції, частотні діапазони, потужність передавача та типи використовуваних сигналів. Наприклад, стандарт IEEE 802.11 передбачає використання радіочастотних смуг 2.4 ГГц і 5 ГГц з ортогональним частотним мультиплексуванням, що дозволяє забезпечити високу швидкість передачі в умовах багатошляхового розповсюдження сигналів. На цьому ж рівні працюють протоколи Zigbee та

LoRa, які відрізняються низьким енергоспоживанням і дальністю дії, що робить їх придатними для розгортання сенсорних мереж у розподілених або важкодоступних локаціях.

Канальний рівень, який відповідає за організацію доступу до середовища передачі та контроль помилок, має вирішальне значення для ефективного функціонування безпроводної мережі. У багатьох протоколах, зокрема Wi-Fi та Bluetooth, застосовуються методи множинного доступу з розділенням у часі або частоті, а також протоколи виявлення колізій, які мінімізують ймовірність одночасної передачі даних кількома пристроями. Крім того, на цьому рівні можуть використовуватись коди виявлення та виправлення помилок, що забезпечують цілісність переданої інформації навіть за наявності шумів і спотворень сигналу.

На мережевому рівні відбувається маршрутизація даних, визначення оптимальних шляхів для їх доставки від джерела до одержувача. Це особливо актуально для динамічних або мобільних мереж, де топологія може змінюватися з часом. Протоколи, такі як AODV та DSR дозволяють адаптивно будувати маршрути в реальному часі, з урахуванням поточного стану мережі. Для статичних або слабо змінних систем використовуються простіші механізми з попередньо визначеними маршрутами або алгоритмами маршрутизації на основі таблиць сусідства.

Транспортний рівень відповідає за контроль доставки даних, сегментацію великих пакетів, повторну передачу в разі втрат і забезпечення порядку доставки. У безпроводних мережах часто виникає необхідність у легковагих протоколах з низьким оверхедом, таких як UDP, що особливо важливо для сенсорних систем з обмеженими ресурсами. Проте для критично важливих даних або систем реального часу можуть застосовуватись модифіковані варіанти TCP або гібридні протоколи, які забезпечують баланс між надійністю та затримками передачі.

Суттєвою складовою сучасних протоколів є реалізація заходів безпеки: шифрування, автентифікація користувачів, захист від атак типу «людина

посередині» та спуфінгу. Впровадження криптографічних механізмів у безпроводних протоколах, таких як WPA3 для Wi-Fi або AES-шифрування для Zigbee, дозволяє гарантувати конфіденційність і цілісність даних навіть у відкритому радіоефірі.

Таким чином, протоколи передачі даних у безпроводних мережах є багаторівневими системами, що забезпечують усі аспекти функціонування зв'язку від фізичної передачі сигналів до логічної організації потоків інформації. Їх вибір залежить від конкретних вимог системи: швидкості, енергоефективності, надійності, масштабованості та безпеки. Саме через їхню гнучкість і здатність до адаптації ці протоколи є основою сучасних інформаційних систем, зокрема таких, які спрямовані на обробку даних у розподіленому середовищі безпроводних мереж.

1.3 Сучасні інформаційні системи обробки даних

Інформаційні системи обробки даних у сучасних умовах відіграють ключову роль у забезпеченні ефективного функціонування як окремих технологічних процесів, так і цілих організаційних структур. Їхнє призначення полягає в автоматизованому зборі, зберіганні, обробці, аналізі та візуалізації великих обсягів даних з різних джерел, що дозволяє приймати обґрунтовані управлінські рішення в реальному або наближеному до реального часі. З огляду на розширення сфер використання цифрових технологій, особливо у галузях з розподіленою інфраструктурою таких як: енергетика, аграрний сектор, транспорт і логістика; сучасні інформаційні системи повинні відповідати високим вимогам до масштабованості, адаптивності та стійкості до збоїв.

Технічно інформаційна система є багаторівневою архітектурною конструкцією, яка включає апаратні засоби, програмне забезпечення та бази даних, об'єднані через мережу для досягнення єдиної мети – перетворення сирих даних на корисну інформацію. Ці системи функціонують на основі

алгоритмів аналізу даних, що можуть бути як класичними (статистичними), так і сучасними – з використанням методів машинного навчання, нейронних мереж або обробки природної мови. В умовах зростання обсягів даних (Big Data) виникає потреба у використанні хмарних обчислень, розподіленої обробки, паралельних обчислень і контейнеризації, що дозволяє підвищити продуктивність і надійність системи.

Суттєвим чинником у побудові інформаційної системи є можливість інтеграції з сенсорними або телеметричними пристроями, особливо в контексті безпроводних мереж. Дані, які надходять із таких джерел, часто є неповними, зашумленими або мають часову прив'язку, що потребує додаткової попередньої обробки, зокрема фільтрації, нормалізації, інтерполяції або синхронізації. Саме тому у сучасних інформаційних системах впроваджуються модулі попередньої обробки сигналів, які дозволяють підвищити якість інформаційного потоку ще до його аналізу.

Окрему увагу слід приділити інтерфейсам взаємодії користувача з інформаційною системою. Зрозумілий і інтуїтивний графічний інтерфейс сприяє ефективному використанню системи навіть непрофесійними користувачами, зменшує кількість помилок при введенні даних та покращує загальний досвід взаємодії. Крім того, система має забезпечувати можливість інтерактивного перегляду, побудови графіків, статистичних звітів і отримання попереджень у разі виявлення аномалій у потоці даних.

У загальному випадку сучасна інформаційна система має бути гнучкою та масштабованою, з можливістю адаптації до нових форматів даних, підключення нових сенсорів, зміни джерел живлення або структури мережі. Така система повинна також враховувати вимоги до безпеки, доступності, збереження конфіденційності, резервного копіювання та журналювання подій, що є критично важливими у будь-якому середовищі, де обробляються дані в реальному часі.

Таким чином, сучасні інформаційні системи обробки даних є складними і динамічними структурами, що функціонують на стику

апаратного і програмного забезпечення, з урахуванням специфіки джерел даних і потреб кінцевого користувача. Їх розвиток нерозривно пов'язаний із прогресом у сфері телекомунікацій, обчислювальних технологій і аналітики, що формує підґрунтя для впровадження інтелектуальних рішень у різних галузях сучасного цифрового суспільства.

1.4 Огляд методів і засобів обробки даних, отриманих з безпроводних мереж

Обробка даних, отриманих з безпроводних мереж (рисунок 1.2), є однією з ключових задач сучасної прикладної інформатики, адже від правильності, швидкості та точності інтерпретації таких даних залежить ефективність прийняття рішень у широкому спектрі застосувань: від технічного моніторингу до автоматичного регулювання та прогнозування. Ці дані, як правило, мають часову структуру, можуть бути просторово розподіленими, містити шуми, втрати або дублікати, що обумовлює необхідність у застосуванні комплексного підходу до їхньої обробки.



Рисунок 1.2 – Обробка даних з безпроводних мереж

Однією з перших стадій є попередня обробка, що включає фільтрацію, виявлення та корекцію пропущених або аномальних значень, згладжування та нормалізацію. У випадку роботи із сенсорними мережами, що функціонують у нестабільних умовах, надзвичайно важливо здійснити зменшення рівня шуму, який може бути зумовлений як фізичними впливами (температура, вологість, радіоперешкоди), так і внутрішніми помилками вимірювальних пристроїв. Для цього часто застосовуються фільтри низьких частот, медіанні фільтри або методи згортки, зокрема фільтр Калмана.

Після підготовки даних відбувається їх аналітична обробка. Основу складають алгоритми статистичного аналізу, регресійного моделювання, кластеризації та відображення основних закономірностей. Для аналізу часових рядів використовуються методи автокореляції, ковзного середнього, перетворення Фур'є, а також підходи до виявлення змін у тренді або аномалій. При необхідності визначення взаємозв'язків між параметрами в потоці даних можуть застосовуватися кореляційний аналіз, головні компоненти або методи незалежних компонент.

Сучасні інформаційні системи дедалі частіше інтегрують елементи машинного навчання, що дозволяє проводити класифікацію, прогнозування або виявлення складних залежностей. У контексті безпроводних мереж особливу цінність мають алгоритми, здатні працювати в реальному часі з обмеженими обчислювальними ресурсами. Сюди належать спрощені варіанти дерев рішень, нейронних мереж, методів підтримки векторів або класифікація на основі k -ближчих сусідів. У системах, де можливе накопичення значних масивів даних, можуть бути застосовані методи глибокого навчання, зокрема LSTM-мережі для обробки послідовностей.

Для візуалізації результатів обробки використовуються інтерактивні графіки, гістограми, теплові карти, топологічні схеми або часові діаграми. Це не лише полегшує сприйняття інформації, але й дозволяє оперативно виявити потенційні проблеми в роботі мережі, наприклад, нестабільні вузли, часті втрати зв'язку або аномальні зміни параметрів середовища.

З технічної точки зору, обробка даних може здійснюватись як на централізованому сервері, так і в децентралізованому режимі: безпосередньо на вузлах мережі або проміжних хабах. Застосування хмарних технологій дозволяє масштабувати системи обробки та забезпечити доступ до обчислювальних ресурсів, зберігання даних та аналітичних сервісів незалежно від фізичного розміщення користувача. Водночас зростає значення технологій краєвих обчислень (edge computing), коли частина аналітичної обробки переноситься ближче до джерела даних, що знижує затримки та зменшує навантаження на канал зв'язку.

Засоби реалізації процесу обробки охоплюють широкий спектр інструментів, серед яких особливе місце посідають Matlab і Python (зокрема в середовищі Google Colab). Matlab має розвинуту бібліотеку функцій для обробки сигналів, статистики та побудови моделей, тоді як Colab надає доступ до гнучких інструментів Python з великою кількістю відкритих бібліотек (NumPy, Pandas, Scikit-learn, TensorFlow, Matplotlib), а також можливість інтеграції з хмарними сервісами для зберігання та обробки даних.

Отже, ефективна обробка даних з безпроводних мереж вимагає використання складної системи методів та інструментів, які дозволяють не лише підготувати дані, а й трансформувати їх у придатну до аналізу форму, забезпечити якісну візуалізацію та прийняття оперативних рішень. Саме така цілісна методологія стає основою для побудови інформаційних систем нового покоління, орієнтованих на високий ступінь автоматизації та інтелектуалізації процесів.

1.5 Огляд існуючих програмних засобів

Огляд існуючих програмних засобів, призначених для обробки даних із безпроводних мереж, демонструє значний діапазон інструментів – від спеціалізованих аналітичних утиліт до комплексних платформ для моделювання та аналізу. Зокрема, для моніторингу та аналізу мережевого

трафіку широко використовуються пакетні аналізатори. Wireshark, що відрізняється відкритим кодом і багатофункціональною графічною оболонкою, дозволяє захоплювати пакети в моніторному режимі, фільтрувати й декодувати структуру протоколів на багатьох рівнях мережевої моделі. Аналогічно функціонує Kismet – утиліта для пасивного виявлення SSID, клієнтів мереж і збору широкого спектра інформації про 802.11 інфраструктуру.

ПАКЕТНІ АНАЛІЗАТОРИ		
Параметр	Wireshark	Kismet
Призначення	Захоплення та аналіз пакетів	Пасивне виявлення мереж
Платформа	Windows, Linux, macOS	Linux, macOS, FreeBSD
Інтерфейс	Графічний (Qt)	Web UI (localhost:2501)
Ліцензія	Open Source	Open Source
Складність	Середня	Середня
Переваги	Потужна фільтрація, широка підтримка протоколів	Пасивний режим, GPS, wardriving
Недоліки	Великий обсяг даних, потребує досвіду	Вимагає monitor mode, складне налаштування
Типові користувачі	Мережеві адміністратори, інженери безпеки	Пентестери, дослідники безпеки

Рисунок 1.3 – Пакетні аналізатори

Іншими відомими представниками є CommView та Aircrack-ng, що характеризуються орієнтацією на детальний аналіз безпроводного трафіку. CommView пропонує можливості декодування кадрів, аналізу рівнів сигналу та статистики по вузлах, у той час як Aircrack-ng забезпечує інструменти для аналізу безпеки, включаючи функції перехоплення і криптоаналізу мереж із ключами WEP і WPA.

Ще одним напрямом є платформи класу GNU Radio – програмні середовища для реального або емуляційного оброблення радіосигналів. GNU Radio дозволяє створювати блокові схеми SDR (Software Defined Radio), що

застосовуються як у наукових дослідженнях, так і в реальних проектах обробки радіо- та мережевого трафіку. Для тих, кому важливий аналіз на рівні цілісного середовища, доступні системні рішення того ж MATLAB. MathWorks пропонує ряд інструментів (Wireless Communications Toolbox, 5G Toolbox, Simulink), що дозволяють моделювати, симулювати і тестувати безпроводні системи різного рівня складності – від 5G до адаптивних алгоритмів обробки сигналу.

SDR ТА МОДЕЛЮВАННЯ		
Параметр	GNU Radio	MATLAB Wireless Toolbox
Призначення	SDR блокові схеми	5G/LTE моделювання
Платформа	Linux, Windows, macOS	Windows, Linux, macOS
Інтерфейс	GNU Radio Companion	MATLAB IDE + Apps
Ліцензія	Open Source	Комерційна
Складність	Висока	Середня
Переваги	Гнучкість, наукова підтримка	Професійний, готові моделі
Недоліки	Крута крива навчання	Дорогий, потребує ліцензії
Типові користувачі	Інженери-дослідники, радіоаматори	Інженери 5G/LTE, корпорації

Рисунок 1.4 – SDR та моделювання

Для аналітики й обробки даних у середовищі Python (що актуально при роботі через Google Colab) застосовують відкриті бібліотеки на кшталт Scapy, яка дозволяє обробляти трафік у реальному часі, Pandas для структурування даних і Scikit-learn або TensorFlow для аналітичного та прогнозного машинного навчання. Інтеграція MATLAB і Python за допомогою MATLAB Engine API забезпечує взаємодію між обчислювальними ядрами двох середовищ, що дозволяє, наприклад, генерувати дані MATLAB для подальшого аналізу в Python .

Окремо варто згадати комерційні системи моніторингу мереж, такі як SolarWinds NPM, PRTG, ManageEngine OpManager та Nagios. Ці рішення відрізняються масштабованістю, гнучкою системою сповіщень,

багаторівневими дашбордами і можливістю інтегрувати Моніторинг безпроводних аспектів мережі, роблячи їх популярними в корпоративному середовищі.

КОМЕРЦІЙНІ СИСТЕМИ МОНІТОРИНГУ			
Параметр	PRTG	SolarWinds NPM	Nagios
Призначення	Універсальний моніторинг	Корпоративний моніторинг	Системний моніторинг
Платформа	Windows	Windows Server	Linux, Unix
Інтерфейс	Web UI	Web UI	Web UI + Config
Ліцензія	Freemium (100 сенсорів)	Комерційна	Open Source
Складність	Легка	Середня	Висока
Переваги	Простота, безкоштовна версія	Корпоративний рівень, масштабованість	Безкоштовний, гнучкість
Недоліки	Обмеження версії, тільки Windows	Дуже дорогий	Складна конфігурація
Типові користувачі	SMB, IT відділи	Великі корпорації, MSP	Linux адміністратори, DevOps

Рисунок 1.5 – Аналіз комерційних систем моніторингу

Таким чином, на ринку присутні інструменти різного рівня: від вузькопрофільних відкритих програм для роботи з пакетним трафіком до потужних комерційних і наукових платформ для моделювання, аналізу та прогнозування поведінки безпроводних мереж. Географія суто наукових інструментів (MATLAB, GNU Radio) прекрасно поєднується з практичними утилітами (Wireshark, Kismet, Aircrack-ng), що створює багатоступеневу екосистему для вирішення задач, подібних до тих, що визначені у вашому дослідженні.

2 ПРОЕКТУВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ ОБРОБКИ ДАНИХ ІЗ БЕЗПРОВІДНОЇ МЕРЕЖІ

2.1 Вимоги до інформаційної системи

Побудова ефективної інформаційної системи обробки даних з безпроводної мережі вимагає чіткого формулювання вимог, які визначають її функціональну придатність, надійність, адаптивність і технічну реалізацію. Ці вимоги формуються з урахуванням особливостей джерел даних, умов експлуатації, типів користувачів і цільових задач аналізу, які вирішує система.

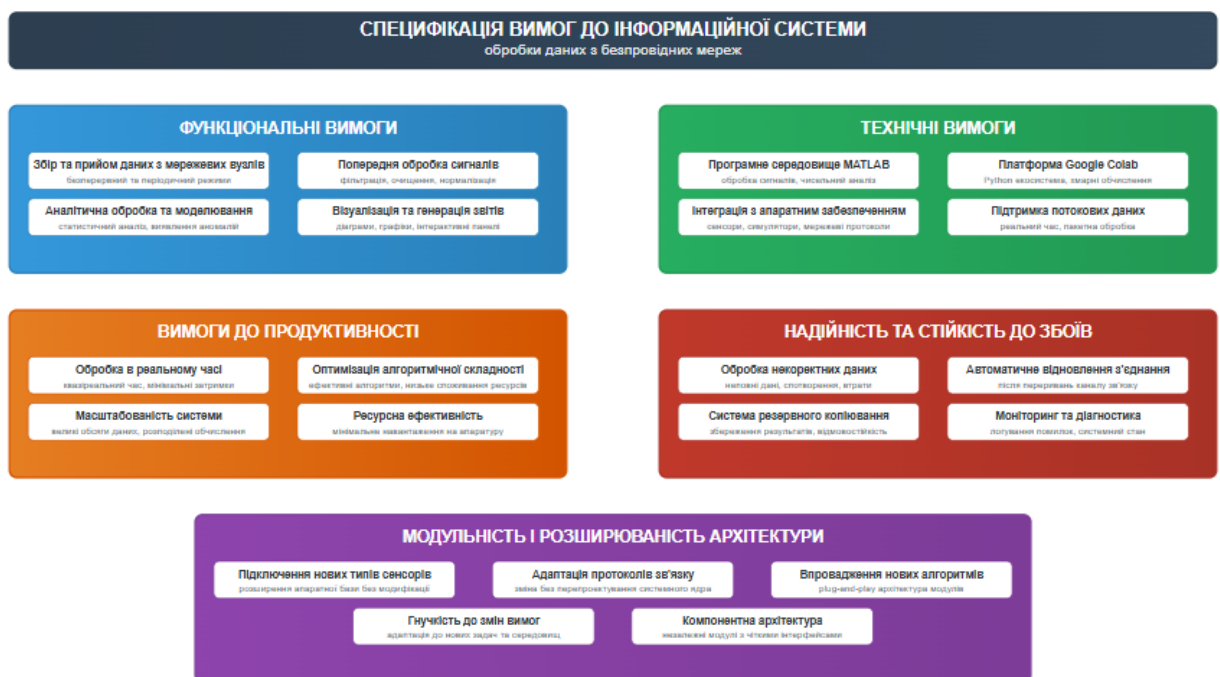


Рисунок 2.1 – Специфікації вимог до інформаційної системи обробки даних безпроводної мережі

Функціонально система повинна забезпечувати безперервний або періодичний прийом даних з одного чи декількох вузлів безпроводної мережі, виконувати попередню обробку з метою очищення, нормалізації та фільтрації

сигналу, здійснювати аналітичну обробку, яка включає базову статистику, виявлення аномалій, трендів та, за потреби, класифікацію або прогнозування. Отримані результати повинні бути представлені у візуальній формі, зручною для кінцевого користувача: діаграми, графіки часових рядів, сигнальні сповіщення або інтерфейсні звіти.

З технічного боку система повинна бути реалізована у програмному середовищі, яке дозволяє працювати з числовими даними, графічно представляти результати та інтегруватися з безпроводними пристроями або симуляторами. У цьому проєкті такими середовищами обрано Matlab і Google Colab, які забезпечують як розширені засоби обробки сигналів та чисельного аналізу, так і доступ до бібліотек для роботи з потоковими або пакетними даними, у тому числі засобами Python.

Продуктивність системи також виступає критичним параметром. З огляду на характер вхідних даних (великі обсяги, нестабільність трафіку, часові обмеження), обробка має бути виконана в реальному або квазіреальному часі, з мінімальними затримками, без суттєвого навантаження на апаратне середовище. У зв'язку з цим важливим є застосування оптимізованих алгоритмів, зменшення обчислювальної складності та, за потреби, можливість перенесення окремих оброблюваних компонентів на віддалені обчислювальні вузли або сервери.

Особливу увагу необхідно приділити надійності та стійкості до збоїв. Система повинна передбачати обробку неповних або спотворених даних, відновлення з'єднання після переривання каналу зв'язку, а також механізми збереження результатів і автоматичної фіксації помилок або збоїв у логах.

Крім того, одним із ключових вимог є модульність і розширюваність архітектури. Проєктована система має передбачати можливість майбутнього підключення нових типів сенсорів, зміни протоколів взаємодії, впровадження нових алгоритмів обробки без суттєвої модифікації всієї структури. Це дозволяє адаптувати систему до змін у середовищі або вимогах користувача без необхідності створення нової архітектури з нуля.

У сукупності ці вимоги формують основу технічного завдання та слугують еталоном для подальшого проєктування, реалізації й тестування інформаційної системи, яка повинна відповідати критеріям ефективності, адаптивності та практичної придатності в умовах безпроводного середовища.

2.2 Структурна схема інформаційної системи

Розробка інформаційної системи обробки даних з безпроводної мережі передбачає створення логічної та функціональної архітектури, яка забезпечить ефективну взаємодію між її компонентами, починаючи з джерел даних і закінчуючи виведенням результатів користувачеві. Структурна схема є концептуальним представленням цієї архітектури, що визначає ключові модулі системи, зв'язки між ними, а також напрямки передачі даних.

У центрі архітектури розташовується блок збору даних, який взаємодіє з фізичними або емуляційними сенсорними пристроями через безпроводний інтерфейс. Цей блок відповідає за прийом сигналу, первинне буферування та передачу даних до наступного модуля. Оскільки передавання здійснюється в нестабільному середовищі, цей компонент має враховувати можливість втрат, затримок або переривань зв'язку, тому реалізується із застосуванням механізмів підтвердження прийому, повторної передачі або перевірки цілісності.

Далі дані потрапляють у модуль попередньої обробки, який виконує фільтрацію шумів, виявлення та заміну пропущених значень, нормалізацію діапазонів і, за потреби, усереднення або інтерполяцію. Цей етап необхідний для приведення необроблених даних до форми, придатної для подальшого аналізу. Реалізація цього блоку відбувається засобами Matlab або Python з використанням відповідних бібліотек обробки сигналів.

Аналітичний модуль, який розташовується наступним у структурі, виконує основну обчислювальну роботу. Він включає реалізацію статистичних методів, алгоритмів виявлення аномалій, моделювання трендів

або, у розширеному варіанті, класифікацію та прогнозування. У середовищі Matlab цей модуль може використовувати засоби Signal Processing Toolbox або Statistics and Machine Learning Toolbox, тоді як у Google Colab можлива реалізація на основі бібліотек Pandas, Scikit-learn або TensorFlow.

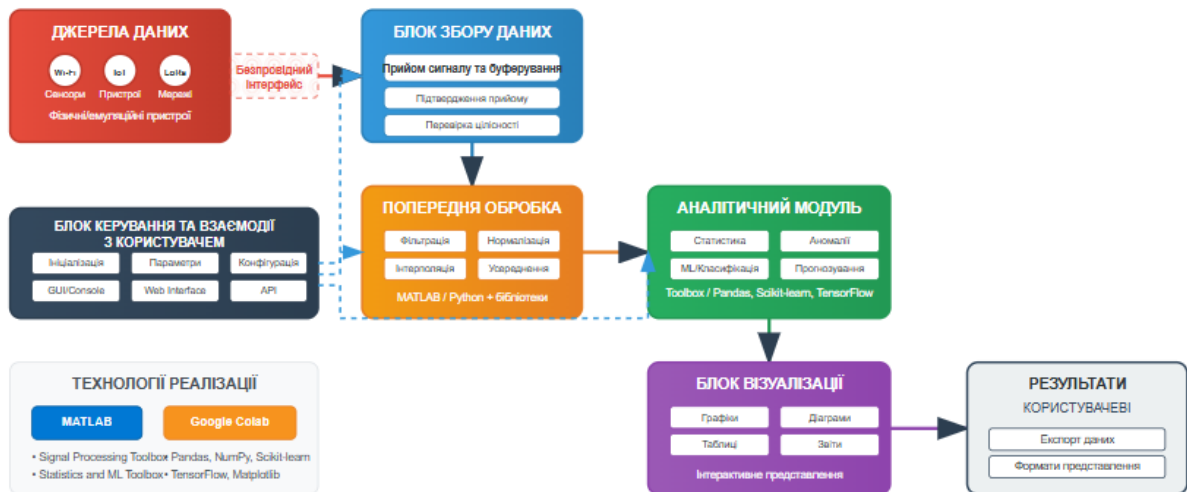


Рисунок 2.2 – Структурна схема інформаційної системи

Результати, сформовані аналітичним модулем, передаються до блоку візуалізації. Він відповідає за побудову графіків, часових діаграм, статистичних таблиць або інтерактивних звітів, що дозволяють користувачеві швидко й зручно оцінити ситуацію. Важливо, що цей блок має бути максимально гнучким – з підтримкою різних форматів представлення, масштабуванням графіків, виділенням аномалій кольором або позначками.

Окремим елементом системи є блок керування та взаємодії з користувачем, який надає інтерфейс для ініціалізації обробки, зміни параметрів аналізу, вибору джерел даних або експорту результатів. Залежно від реалізації, цей інтерфейс може бути консольним (у середовищі Google Colab), графічним (GUI Matlab) або веб-орієнтованим при інтеграції з іншими системами.

Загалом структурна схема системи передбачає послідовний та взаємопов'язаний потік інформації від вхідних пристроїв до виводу даних, при цьому з можливістю незалежної модифікації кожного з модулів, що

відповідає принципам модульності, повторного використання коду і масштабованості. Такий підхід дозволяє не лише реалізувати базову функціональність системи, а й створити платформу для її майбутнього розширення, наприклад, шляхом підключення нових джерел даних або впровадження складніших методів обробки.

2.3 Вибір та обґрунтування апаратних і програмних засобів

Оскільки у даній роботі реалізація інформаційної системи обробки даних з безпроводної мережі виконується без залучення фізичних сенсорів або контролерів, основна увага зосереджується на програмному моделюванні. Для цього застосовується комбінація синтетичних даних та відкритих реальних датасетів, що дозволяє зімітувати типові сценарії функціонування системи без використання апаратної частини. Такий підхід є доцільним у рамках навчального проєкту, оскільки дозволяє зосередитись на аналізі, алгоритмах обробки, візуалізації результатів та структурі самої інформаційної системи.

Основними інструментами реалізації системи виступають два програмних середовища – Matlab та Google Colab, кожне з яких має свої переваги та функціональні можливості.

Matlab є потужним засобом для технічних та наукових обчислень, який широко використовується в обробці сигналів, чисельному моделюванні та візуалізації даних. Його застосування у цьому проєкті зумовлено наявністю вбудованих бібліотек, таких як Signal Processing Toolbox, Statistics and Machine Learning Toolbox, а також зручними механізмами побудови графіків і реалізації складних математичних моделей. У Matlab здійснюється генерація синтетичних даних, їхня попередня обробка (фільтрація, нормалізація, інтерполяція), а також реалізація базового аналізу: обчислення статистичних показників, виявлення трендів та побудова часових діаграм.

Google Colab використовується для розробки і тестування алгоритмів у

середовищі Python, що дозволяє інтегрувати інструменти машинного навчання та роботи з відкритими реальними датасетами. Colab забезпечує доступ до бібліотек NumPy, Pandas, Matplotlib, Scikit-learn, TensorFlow та інших, що є необхідними для аналітичної обробки даних, їх класифікації, прогнозування та візуального представлення. Додатковою перевагою Google Colab є можливість спільної роботи над кодом, автоматичне збереження результатів у хмарі та інтеграція з Google Drive або зовнішніми API.

Таким чином, замість фізичних сенсорів у роботі використовуються штучно згенеровані або імпортовані дані, які мають ті самі структурні характеристики, що й реальні сигнали, одержані з безпроводних мереж. Такий симуляційний підхід дозволяє провести всебічне тестування інформаційної системи, зберігаючи при цьому наукову та методологічну коректність дослідження.

Отже, обрані програмні середовища Matlab і Google Colab у поєднанні з синтетичними або публічними даними створюють повноцінну платформу для реалізації, вивчення та вдосконалення алгоритмів обробки, що відповідає цілям і завданням цієї кваліфікаційної роботи.

3 ПРОГРАМНА РЕАЛІЗАЦІЯ СИСТЕМИ В СЕРЕДОВИЩАХ MATLAB ТА GOOGLE COLAB

Розробку інформаційної системи було розпочато з моделювання вхідних даних, які імітують сигнали безпроводної сенсорної мережі. В умовах відсутності реального обладнання було вирішено згенерувати синтетичний набір даних, що складається з трьох основних параметрів: температура, тиск і вологість. Для кожного з них було реалізовано періодичну компоненту у вигляді синусоїдальної функції, до якої додано випадковий шум, що дозволило створити модель, наближену до реальних сенсорних сигналів. Генерація даних відбувалася з частотою одну точку на секунду, протягом кількох сотень секунд, а часові мітки формувалися динамічно з використанням типу `datetime`.

Після генерації було здійснено первинну обробку даних, зокрема, згладжування за допомогою ковзного середнього та фільтрація шуму. Це дозволило мінімізувати вплив випадкових флуктуацій на результати подальшого аналізу. Всі змінні було інтегровано у структуру типу `table`, що забезпечило гнучкість у подальшій обробці.

Для візуального аналізу було побудовано окремі графіки для кожного параметра. Далі реалізовано виявлення аномалій методом порогового аналізу на основі стандартного відхилення, що дозволило ідентифікувати локальні відхилення, які перевищують статистично обґрунтований рівень. Всі візуалізації виконувалися засобами Matlab, з використанням функцій `plot`, `scatter`, `subplot` та інших засобів графічного відображення. Для подальшого використання дані було збережено у форматі `.csv` для імпорту в Google Colab.

На другому етапі реалізації система була розгорнута у середовищі Google Colab, що забезпечує хмарну інтеграцію з інструментами Python. Першим кроком стала імпортована обробка `.csv` файлу, отриманого з Matlab. Завдяки використанню бібліотеки `pandas` було здійснено завантаження

таблиці та перетворення стовпця з часовими мітками у формат `datetime`, що дозволило коректно працювати з часовими рядами.

Далі було повторно реалізовано функцію згладжування, а також виявлення аномалій на основі двократного стандартного відхилення від середнього значення. Усі знайдені аномальні точки виводились графічно та підтверджували наявність відповідних пікових відхилень у даних. Було реалізовано візуалізацію з використанням `Matplotlib`, зокрема лінійні графіки з маркерами аномалій, порівняння сирих і згладжених даних, а також кольорову сегментацію.

Окремо було проведено кореляційний аналіз між сенсорними параметрами, результати якого візуалізовано у вигляді теплової карти. Низькі коефіцієнти кореляції підтвердили незалежність змінних і виправдали окремий підхід до аналізу кожного сигналу. Також реалізовано спектральний аналіз (FFT), що дозволив виявити домінуючі частоти сигналу, та трендовий аналіз з використанням ковзного середнього великого вікна, який ефективно візуалізував довготривалі зміни у температурі.

3.1 Архітектура інформаційної системи обробки даних з безпроводної мережі

Інформаційна система обробки даних з безпроводної мережі є багаторівневою програмною структурою, яка реалізує збір, обробку, аналіз і візуалізацію даних, зібраних у сенсорному середовищі. У рамках даного проєкту система розподілена між двома основними середовищами: `Matlab` і `Google Colab`, кожне з яких виконує окремі функціональні ролі та взаємодіє через обмін файлами даних та проміжних результатів обробки.

Архітектурна модель, представлена на рисунку 3.1, передбачає, що `Matlab` виконує функції генерації або первинного завантаження даних, їхньої попередньої обробки, базового статистичного аналізу та сигналової фільтрації. Це зумовлено тим, що `Matlab` має розвинуту бібліотечну базу для

роботи з часовими рядами, матричними обчисленнями та сигналами, а також зручні засоби для побудови графіків і реалізації методів попередньої аналітики. У даному середовищі формується базовий оброблений набір даних, який потім експортується у форматі .csv або .mat для подальшої обробки в Colab.

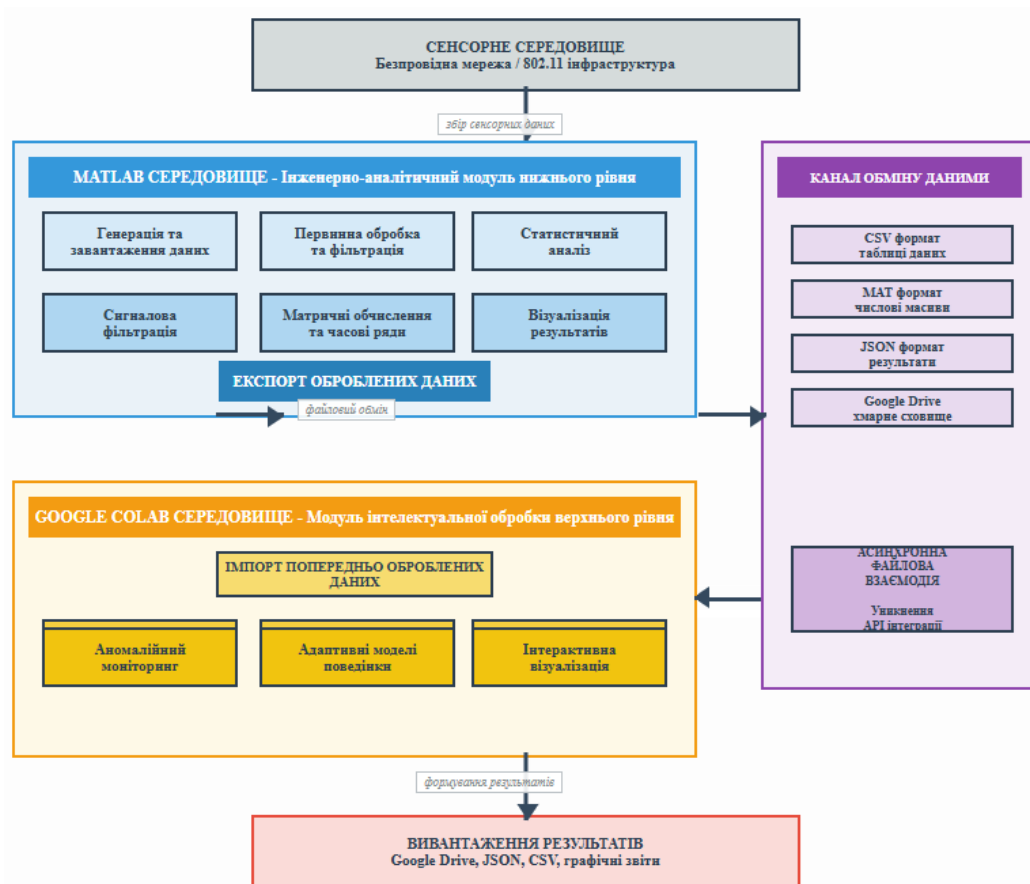


Рисунок 3.1 – Архітектурна схема інформаційної системи обробки даних безпроводної мережі

Google Colab, як хмарна платформа, виступає у ролі розширеної аналітичної платформи. У цьому середовищі система використовує оброблені або частково оброблені дані, збережені в результаті роботи Matlab, і здійснює глибший аналіз із залученням алгоритмів машинного навчання, кластеризації, прогнозування або аномалійного моніторингу. Colab працює у зв'язці з потужним аналітичним стеком Python – Pandas, Scikit-learn, TensorFlow, що дозволяє виводити новий рівень оцінки якості даних,

формувати адаптивні моделі поведінки та автоматично будувати візуалізації з можливістю інтерактивної взаємодії. За потреби, оброблені результати вивантажуються в Google Drive або експортуються у вигляді графічних звітів або JSON/CSV для подальшого використання.

Взаємодія між Matlab та Colab є асинхронною і реалізується через файловий обмін. Основним каналом передачі є спільний формат CSV або MATLAB .mat, що забезпечує сумісність на рівні таблиць даних та числових масивів. Цей підхід дозволяє уникнути складної API-інтеграції, забезпечуючи при цьому ефективний перехід між етапами обробки. У практичному сценарії Matlab може періодично зберігати проміжні результати, які завантажуються в Colab для додаткової обробки або виведення користувачу.

Узагальнено архітектура інформаційної системи функціонує як модульне середовище, де Matlab виконує роль інженерно-аналітичного модуля нижнього рівня, а Google Colab – модуль глибокої інтелектуальної обробки верхнього рівня. Такий підхід забезпечує розділення навантаження, ефективне використання ресурсів кожного середовища та високу адаптивність системи до зміни формату вхідних даних або завдань обробки. У поєднанні ці два компоненти утворюють цілісну інформаційну інфраструктуру, що дозволяє реалізовувати як класичні підходи до обробки сенсорної інформації, так і сучасні інтелектуальні алгоритми аналізу.

3.2 Особливості середовищ Matlab та Colab для розробки інформаційних систем

У сучасній практиці розробки інформаційних систем вибір середовища програмування та обчислень має принципове значення, оскільки саме воно визначає гнучкість, масштабованість та ефективність реалізації проектного рішення. Для завдань, пов'язаних із обробкою даних із безпроводних мереж, доцільним є використання таких програмних платформ, які поєднують засоби чисельного аналізу, обробки сигналів, візуалізації й підтримку роботи

з потоками даних. У цьому контексті середовища Matlab і Google Colab є надзвичайно вдалими з огляду на їхні функціональні можливості, гнучкість і широке застосування в науковій і прикладній діяльності.

Matlab – це високорівневе інтегроване середовище програмування, спеціалізоване на обробці числових і векторно-матричних даних. Воно відзначається багатим набором вбудованих бібліотек, зокрема для роботи із сигналами, статистичного аналізу, моделювання систем і машинного навчання. Його перевагою є висока продуктивність при роботі з великими масивами даних, підтримка інтерфейсів візуального проєктування, а також можливість швидкої побудови прототипів алгоритмів. Matlab також дозволяє імпортувати та експортувати дані у різноманітних форматах (CSV, Excel, JSON, *.mat), що важливо для інтеграції систем з реальними чи симульованими джерелами. Крім того, Matlab підтримує симуляційне моделювання та апаратну взаємодію, що надає можливість розширення системи за межі лише програмного середовища.

Google Colab – це хмарна платформа, заснована на мові Python, яка надає доступ до обчислювальних ресурсів (CPU, GPU, TPU) без потреби локального встановлення середовища. Colab побудований на основі Jupyter Notebook і дозволяє інтерактивно розробляти, документувати та виконувати код, що є особливо зручним для навчання, досліджень і спільної роботи. Colab має прямий доступ до таких популярних бібліотек, як NumPy, Pandas, Matplotlib, Seaborn, Scikit-learn, TensorFlow тощо. Це дозволяє реалізовувати як базові статистичні методи, так і складні аналітичні та нейромережеві моделі. Суттєвою перевагою є також інтеграція з Google Drive, що забезпечує автоматичне збереження результатів та зручний обмін файлами між учасниками проєкту.

Обидва середовища мають свої особливості у підході до розв'язання задач. Matlab, як правило, є більш підходящим для інженерних задач, моделювання та роботи з сигналами. Його синтаксис інтуїтивно зрозумілий для тих, хто працює з матрицями, та дозволяє швидко реалізовувати

аналітичні операції з мінімумом програмістських зусиль. Colab із Python, своєю чергою, надає більше свободи в розробці адаптивних, масштабованих систем, особливо при використанні відкритих даних, побудові web-орієнтованих додатків або інтеграції з API.

Таким чином, вибір Matlab і Google Colab є виправданим з огляду на різноплановість поставлених завдань. Вони не лише доповнюють одне одного, але й забезпечують повний цикл розробки: від генерації або імпорту даних до їхньої аналітичної обробки, візуалізації й збереження результатів. Такий підхід дозволяє створити функціонально повну інформаційну систему без залучення фізичних сенсорів, зосередившись на аналізі алгоритмів, ефективності обробки та зручності інтерфейсу.

3.3 Структура програмного забезпечення в Matlab

Реалізація інформаційної системи обробки даних у середовищі Matlab потребує чіткої структурної організації коду, яка дозволяє забезпечити модульність, логічну послідовність етапів обробки та зручність подальшого супроводу системи. Структура програмного забезпечення, розробленого у цій роботі, орієнтована на поетапну обробку симульованих або реальних даних, їх візуалізацію та подальший аналіз результатів.

Основними функціональними компонентами Matlab-програми є блок генерації або завантаження вхідних даних, модуль попередньої обробки, аналітичний блок, візуалізаційний модуль та загальна управляюча логіка. Така побудова відповідає принципам модульного програмування, коли кожен елемент виконує окреме логічне завдання і може бути незалежно змінений, протестований або розширений.

На першому етапі здійснюється завантаження даних. У цьому блоці передбачено два підходи: генерація синтетичних сигналів із заданими параметрами (частота, шум, пропуски, тренди) або імпорт зовнішнього датасету з формату CSV, Excel чи *.mat. Завдяки вбудованим функціям

readtable, load, xlsread та csvread реалізується гнучке зчитування файлів, а також можливість обробки багатостовпчикових таблиць з часовою прив'язкою.

Наступним є блок попередньої обробки, який реалізує очищення даних, нормалізацію, згладжування та фільтрацію. У Matlab використовуються функції fillmissing, normalize, smoothdata, filter, що дозволяє стандартизувати дані та підготувати їх до подальшого аналізу. У цьому ж модулі можуть застосовуватися алгоритми виявлення викидів або автоматична корекція пропусків на основі інтерполяції.

Аналітичний модуль виконує обчислення статистичних характеристик (середнє, дисперсія, медіана, коефіцієнт кореляції), а також побудову моделей трендів, розподілів та, за потреби, компонентного аналізу. За допомогою функцій mean, std, corrcoef, fit, polyfit, movmean можлива реалізація як одномірного аналізу, так і багатовимірної обробки даних. Для сценаріїв прогнозування використовуються прості методи лінійної регресії або згладжувальні фільтри.

Модуль візуалізації відповідає за графічне представлення результатів. Matlab надає потужний інструментарій для побудови графіків: plot, scatter, bar, histogram, subplot, які дозволяють створити інформативне зображення динаміки даних, порівняння параметрів або оцінку результатів фільтрації. Графіки можуть містити маркери аномалій, автоматичне масштабування та текстові підписи, що значно підвищує зручність користувача у сприйнятті даних.

Загальна управляюча логіка реалізована у вигляді основного скрипту або функції, яка викликає всі інші модулі у відповідній послідовності. При потребі, до структури можна додати графічний інтерфейс користувача за допомогою App Designer, що дозволяє взаємодіяти із системою через елементи керування (кнопки, поля введення, меню).

Завдяки чіткій структурі, система у Matlab залишається зрозумілою, гнучкою та придатною для подальшого розширення. Вона може легко

адаптуватися під нові джерела даних, алгоритми аналізу або способи виведення результатів, що робить її універсальним інструментом для обробки інформації з безпроводних мереж навіть у симульованому середовищі.

3.4 Реалізація обробки даних та візуалізація результатів у Matlab

У середовищі Matlab реалізація обробки даних із безпроводної мережі передбачає побудову алгоритмів, які включають вхідні процедури генерації або імпорту даних, попередню підготовку, аналітичну обробку та візуалізацію результатів. У межах цієї роботи використано симульовані (синтетичні) сигнали, що імітують типові характеристики сенсорних потоків: нестабільність, шум, трендовість, періодичність і пропуски даних.

На етапі генерації даних створюється набір синтетичних сигналів із заданими властивостями. Для цього використовується функція `randn` з додаванням гармонічних компонент і накладанням шуму. При потребі в сигнал вбудовуються аномальні значення (сплески, пропуски) для перевірки стійкості алгоритмів обробки. Дані зберігаються у вигляді масиву або таблиці з часовою прив'язкою, що імітує структуру реального датасету.

Попередня обробка виконується із застосуванням функцій згладжування, фільтрації, інтерполяції пропусків та нормалізації. Наприклад, для сигналу із коливаннями і шумами може бути використаний ковзний середній, або експоненційне згладжування для виділення основної тенденції.

На наступному етапі відбувається статистичний аналіз: обчислюються середнє значення, стандартне відхилення, коефіцієнт варіації, кореляція між каналами (за потреби). Прикладом може слугувати функція `mean` для оцінки загального рівня сигналу, або `movstd` для оцінки локальної змінності.

Для візуалізації результатів обробки використовуються функції `plot`, `area`, `histogram`, `scatter`, `subplot` тощо. Дані можуть бути представлені у вигляді графіків змін у часі, спектральних характеристик або статистичних

розподілів. Наприклад, відображення початкового сигналу разом із фільтрованим дозволяє користувачу візуально оцінити ефективність алгоритму. Крім того, за допомогою маркерів можна позначати виявлені аномалії, піки чи трендові зони.

Окрему увагу приділено автоматизації виводу результатів: усі побудовані графіки мають підписи, осі масштабуються автоматично або вручну, додаються легенди, що підвищує зручність аналізу. При необхідності результати експортуються у вигляді зображень або PDF-файлів для включення в звіти.

Уся логіка обробки структурована у вигляді окремих функцій або скриптів, що дозволяє легко модифікувати параметри сигналу, змінювати методи згладжування або додавати нові етапи аналізу. Такий підхід забезпечує модульність, наочність і гнучкість реалізації, а також відповідає академічним принципам розробки програмного забезпечення.

```

1  %% --- 1. СТВОРЕННЯ СИНТЕТИЧНИХ ДАНИХ ---
2  N = 1000;
3  timestamps = datetime('now') + seconds(0:N-1)';
4  temperature = 22 + 2*sin(2*pi*(0:N-1)/200)' + randn(N,1)*0.5;
5  pressure = 101325 + (0.5*(0:N-1))' + randn(N,1)*20;
6  humidity = 60 + 5*sin(2*pi*(0:N-1)/150)' + randn(N,1)*2;
7
8  sensorsdata = table(timestamps, temperature, pressure, humidity, ...
9      'VariableNames', {'Timestamps', 'Temperature', 'Pressure', 'Humidity'});
10
11  save('sensorsdata.mat', 'sensorsdata');
12  writetable(sensorsdata, 'sensorsdata.csv');
13  disp('✓ Синтетичний датасет створено.')
14  % Завантаження даних
15  load sensorsdata
16
17  % Огляд структури
18  head(sensorsdata)
19  time = sensorsdata.Timestamps;
20  temperature = sensorsdata.Temperature;
21  pressure = sensorsdata.Pressure;
22  humidity = sensorsdata.Humidity;
23
24  %% --- ВІЗУАЛІЗАЦІЯ ПОЧАТКОВИХ ДАНИХ ---
25  figure('Name','Raw Sensor Data','NumberTitle','off')
26  subplot(3,1,1)
27  plot(time, temperature), title('Температура'), ylabel('°C'), grid on
28  subplot(3,1,2)
29  plot(time, pressure), title('Тиск'), ylabel('Pa'), grid on
30  subplot(3,1,3)

```

Рисунок 3.2 – Фрагмент файлу Gotra.m

На рисунку 3.2 представлено код реалізації модуля обробки даних з сенсорів в Matlab. Лістинг коду представлений у додатку Б.1.

3.5 Розробка та реалізація системи обробки даних в середовищі Google Colab

Середовище Google Colab є потужною платформою для хмарної обробки даних, яка поєднує зручність Jupyter-ноутбуків з потужностями Python-бібліотек та інфраструктурою Google. У рамках цієї кваліфікаційної роботи Colab використовується як інструмент для реалізації альтернативного варіанта інформаційної системи обробки даних з безпроводної мережі – з акцентом на використання відкритих бібліотек, машинного навчання та інтерактивної візуалізації.

Першим етапом реалізації є підготовка середовища. У ноутбучі Colab здійснюється імпорт необхідних бібліотек: NumPy для роботи з числовими масивами, Pandas для табличної структури даних, Matplotlib і Seaborn для побудови графіків, Scikit-learn для реалізації алгоритмів аналізу, а також TensorFlow або Keras у разі розширеного прогностного моделювання. Крім того, бібліотека SciPy застосовується для фільтрації сигналів і обробки шуму.

У наступному модулі відбувається генерація або завантаження даних. Для цього можуть бути використані синтетичні генератори або імпорт реальних датасетів з файлів .csv, .xlsx, .json чи з віддалених джерел за допомогою HTTP-запитів. Для тестування роботи з потоковими даними можна імітувати передачу пакетів у циклі з часовими затримками.

Попередня обробка даних включає очистку від пропусків, нормалізацію, згладжування, виявлення викидів або побудову фільтрів Фур'є. Python дозволяє легко працювати з часовими мітками, що є важливим у контексті обробки сенсорних даних із часовою структурою.

У блоці аналітичної обробки застосовуються як класичні статистичні методи, так і алгоритми машинного навчання. Наприклад, для класифікації

аномальних даних може бути використано Isolation Forest, k-Means, Random Forest, або для прогнозування: Linear Regression, ARIMA, LSTM. Colab дозволяє експериментувати з моделями, автоматично зберігаючи результати, графіки та метрики точності.

Візуалізація результатів виконується з використанням Matplotlib, Seaborn або інтерактивних інструментів. Побудова графіків у реальному часі (наприклад, оновлення через цикл із паузами) дозволяє моделювати динаміку надходження сенсорних даних. Можна виводити порівняльні графіки: початковий сигнал, фільтрований варіант, тренд, аномалії або прогнозоване значення.

```
[ ] # Importing libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from scipy.fft import fft
from sklearn.preprocessing import StandardScaler
from google.colab import files
import io
import statsmodels.api as sm
```

```
# Upload CSV file
uploaded = files.upload()
df = pd.read_csv(io.BytesIO(list(uploaded.values())[0]))
df['Timestamps'] = pd.to_datetime(df['Timestamps'])
df.set_index('Timestamps', inplace=True)
df.head()
```

[Огляд...](#) sensorsdata.csv
sensorsdata.csv(application/vnd.ms-excel) - 71704 bytes, last modified: n/a - 100% done
Saving sensorsdata.csv to sensorsdata.csv

Timestamps	Temperature	Pressure	Humidity
2025-07-11 08:29:32	22.435299	101351.112462	59.144514
2025-07-11 08:29:33	22.228202	101345.179391	59.050632
2025-07-11 08:29:34	21.451633	101300.972273	62.270326
2025-07-11 08:29:35	22.962153	101322.904924	60.637687
2025-07-11 08:29:36	21.942348	101312.131871	59.564862

Подальші дії: [Переглянути рекомендовані графіки](#) [New interactive sheet](#)

```
[ ] # Smoothing
df['Temperature_smooth'] = df['Temperature'].rolling(10).mean()
df['Pressure_smooth'] = df['Pressure'].rolling(10).mean()
df['Humidity_smooth'] = df['Humidity'].rolling(10).mean()
```

```
[ ] # Anomaly detection
def detect_anomalies(series):
```

Рисунок 3.3 – Фрагмент коду в Google Colab

Завдяки інтеграції Google Colab із Google Drive, усі дані зберігаються автоматично, а результатами можна ділитися через посилання або експортувати їх у вигляді зображень, таблиць або моделей. Користувач може взаємодіяти з ноутбуком, змінюючи параметри, натискаючи кнопки або завантажуючи власні файли через форму.

4 ТЕСТУВАННЯ ТА АНАЛІЗ РЕЗУЛЬТАТІВ

4.1 Методика тестування інформаційної системи

Тестування інформаційної системи обробки даних є невід'ємною частиною її розробки, оскільки дозволяє оцінити функціональність, надійність, точність алгоритмів і відповідність заданим вимогам. У даному дослідженні тестування проводиться із використанням симульованих даних та готових сценаріїв обробки в середовищах Matlab і Google Colab. Це забезпечує можливість верифікації окремих компонентів системи, а також комплексного оцінювання її роботи в цілому.

Основу методики тестування становить функціональний підхід, за якого система перевіряється на здатність коректно виконувати заявлені функції: завантаження даних, їх попередню обробку, аналітичну трансформацію, побудову візуалізацій та генерацію підсумкових показників. На кожному з етапів перевіряються як позитивні сценарії (дані коректні, повні), так і граничні або помилкові (наявність шуму, пропуски, аномальні значення, розриви в часовій послідовності).

Для тестування вхідного блоку використовуються набори даних різної структури та обсягу. Зокрема, здійснюється перевірка здатності системи обробляти багатовимірні таблиці, неправильно структуровані файли, а також дані з відсутніми часовими мітками. Тестується стійкість до помилок при зчитуванні та поведінка системи у випадках некоректного формату введення.

Блок попередньої обробки тестується на здатність видаляти шум, заповнювати пропуски, згладжувати сигнали та правильно нормалізувати дані. Оцінюється вплив кожного етапу на точність наступного аналізу. Для цього використовуються контрольні сигнали з відомими характеристиками, а також порівнюються оброблені дані з очікуваними результатами.

Аналітичний модуль перевіряється шляхом тестування його здатності

розраховувати статистичні показники, виявляти аномалії та тренди. У Google Colab додатково тестуються моделі машинного навчання – оцінюються точність класифікації, адекватність побудованих прогнозів і стійкість моделей до змін у структурі вхідних даних. У Matlab особлива увага приділяється точності фільтрації та статистичному узагальненню результатів.

Візуалізаційний блок проходить тестування на правильність побудови графіків, відповідність шкал і підписів, коректне відображення виявлених аномалій або трендів. Перевіряється здатність системи до масштабування графіків, зміни параметрів виводу та збереження зображень у різних форматах.

Також проводиться часова оцінка продуктивності: заміряється тривалість обробки даних різного обсягу, зокрема в порівнянні між Matlab і Google Colab. Це дозволяє визначити сильні та слабкі сторони кожного середовища з точки зору ефективності виконання.

4.2 Аналіз отриманих результатів в Matlab

Представлене на рисунку 4.1 зображення містить графічне зображення зміни трьох основних параметрів, що характерні для безпроводної сенсорної мережі: температури, тиску та вологості. На часовій шкалі в межах приблизно 20 хвилин спостерігаються періодичні коливання температури та вологості, що свідчить про їхню синусоїдальну природу, можливо, зумовлену моделюванням або контрольованими лабораторними умовами. Тиск демонструє помітну зростаючу тенденцію, що може бути інтерпретовано як зміна атмосферного або внутрішньо системного параметра внаслідок зовнішнього впливу або імітації тренду. Загальна форма сигналів відповідає характеру даних, які могли бути отримані у результаті моделювання синтетичного середовища, що підтверджує їх застосування як тестового набору для подальшого аналізу, фільтрації та виявлення аномалій у межах інформаційної системи.

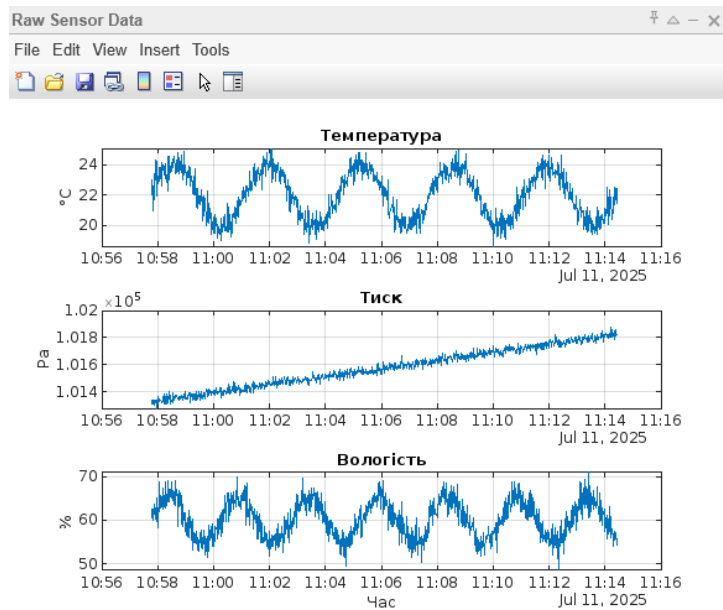


Рисунок 4.1 – Результати в Matlab

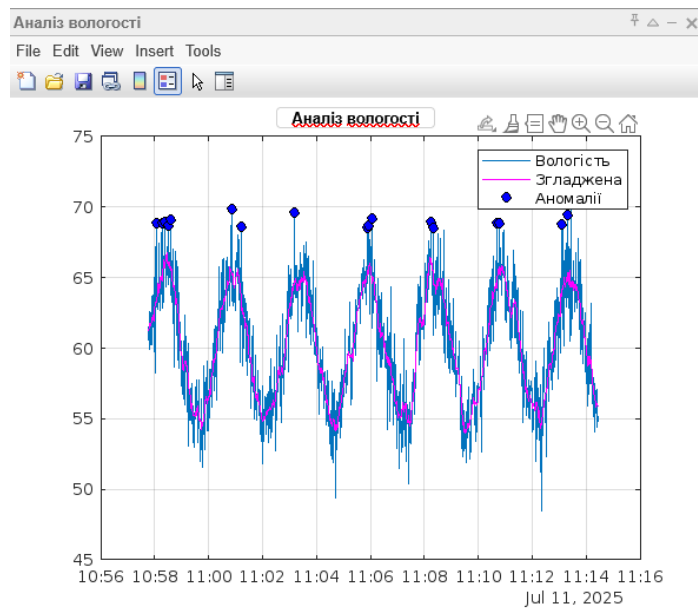


Рисунок 4.2 – Результати в Matlab

На графіку рисунка 4.2 зображено результат аналізу вологості у часі, який включає як необроблені дані, так і їх згладжену версію, отриману методом ковзного середнього. Виділення аномалій здійснено на основі статистичного порогу, ймовірно, у вигляді відхилень за два стандартних відхилення від середнього значення. Виявлені аномалії позначені маркерами й демонструють локальні екстремуми, що суттєво перевищують типовий

амплітудний діапазон. Такий підхід дозволяє ідентифікувати нетипову поведінку сенсорного сигналу, що може вказувати на збої у вимірювальному устаткуванні або аномальні зміни в навколишньому середовищі. Представлений аналіз є частиною ширшої інформаційної системи моніторингу та є свідченням ефективної реалізації функції детекції аномалій.

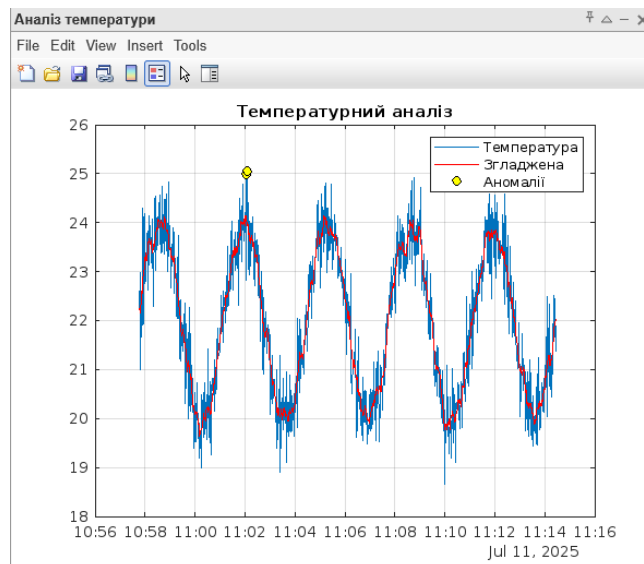


Рисунок 4.3 – Результати в Matlab

На рисунку 4.3 представлено аналіз температурного сигналу, що демонструє періодичну, синусоїдальну форму із характерними флуктуаціями, типовими для сенсорних вимірювань. Синя крива відображає необроблені значення температури, тоді як червона — згладжену версію сигналу, отриману з використанням методу ковзного середнього для придушення високочастотного шуму. Жовтими маркерами позначено виявлені аномальні точки, які виходять за межі статистично очікуваного діапазону, базованого на середньому значенні та стандартному відхиленні. Це свідчить про ефективність реалізованого механізму аномалійного моніторингу в рамках модульної системи обробки даних. Подібна обробка забезпечує достовірність даних для подальшого машинного аналізу або візуалізацій у зовнішніх аналітичних середовищах.

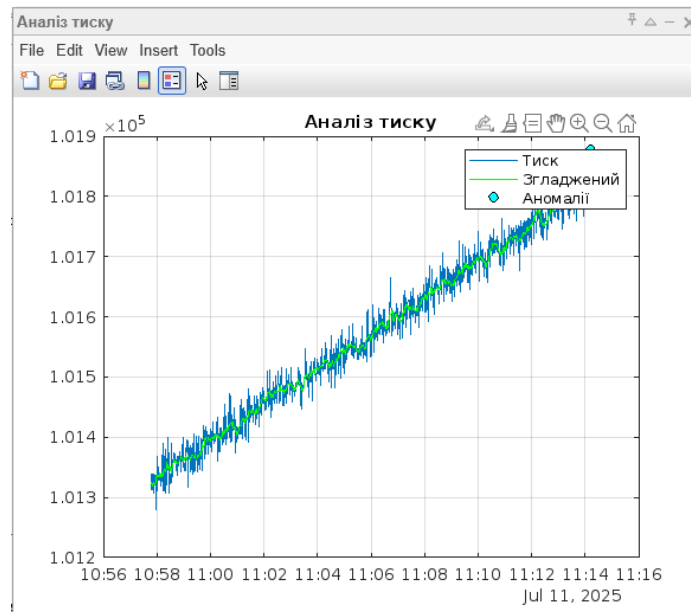


Рисунок 4.4 – Результати в Matlab

Представлений на рисунку 4.4 графік демонструє аналіз зміни тиску в часі в межах сенсорного середовища. Початковий сигнал, зображений синьою лінією, характеризується чітко вираженим зростаючим трендом із додатковими незначними флуктуаціями, що можуть бути зумовлені мікрозмінами або фоновим шумом. Зелена крива ілюструє результат згладжування сигналу, що дозволяє краще простежити загальну тенденцію та зменшити вплив випадкових коливань. Аномальні точки, позначені діамантовими маркерами, виявлено на основі статистичних критеріїв, що вказує на локальні відхилення від передбачуваного розвитку процесу. Така структура аналізу є важливою складовою діагностичного підходу в інформаційній системі, що дозволяє контролювати стабільність параметра в реальному часі та ідентифікувати потенційні збої або техногенні впливи.

4.3 Аналіз отриманих результатів в Google Colab

На рисунку 4.5 представлено теплову карту кореляційної матриці, яка відображає ступінь взаємозв'язку між трьома основними сенсорними параметрами: температурою, тиском і вологістю. Числові значення на

діагоналі дорівнюють одиниці, що вказує на повну автокореляцію кожної змінної із самою собою. Решта значень свідчить про слабкий або відсутній лінійний зв'язок між різними параметрами. Зокрема, кореляція між температурою та тиском становить приблизно -0.15 , між температурою та вологістю -0.051 , а між тиском і вологістю -0.046 .

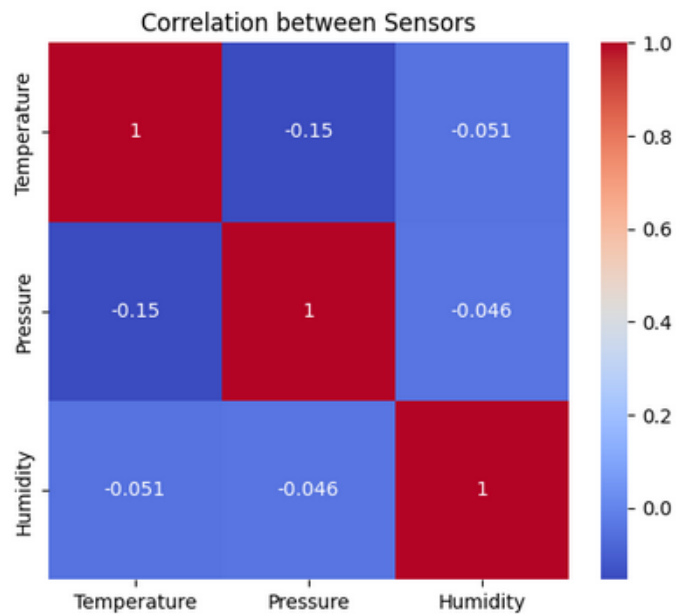


Рисунок 4.5 – Результати в Google Colab

Усі ці значення близькі до нуля, що свідчить про незалежну динаміку кожного сенсорного сигналу в межах аналізованого часового вікна. Така структура кореляцій підтверджує, що в обраному моделюванні відсутній жорсткий фізичний чи функціональний зв'язок між параметрами, що дозволяє аналізувати їх як умовно незалежні в контексті виявлення аномалій та трендів.

На рисунку 4.6 наведено результат спектрального аналізу температурного сигналу, виконаного методом швидкого перетворення Фур'є. Горизонтальна вісь відображає частоту в герцах, тоді як вертикальна вісь – амплітуду відповідних гармонік. Найбільш виражений пік спостерігається в районі нульової частоти, що свідчить про наявність домінуючої сталої або повільно змінної компоненти в температурному сигналі. Решта спектру

містить низькоамплітудні складові, що свідчить про відсутність яскраво виражених гармонічних коливань високої частоти, що типово для згладженого або синусоїдального сигналу з доданим шумом.

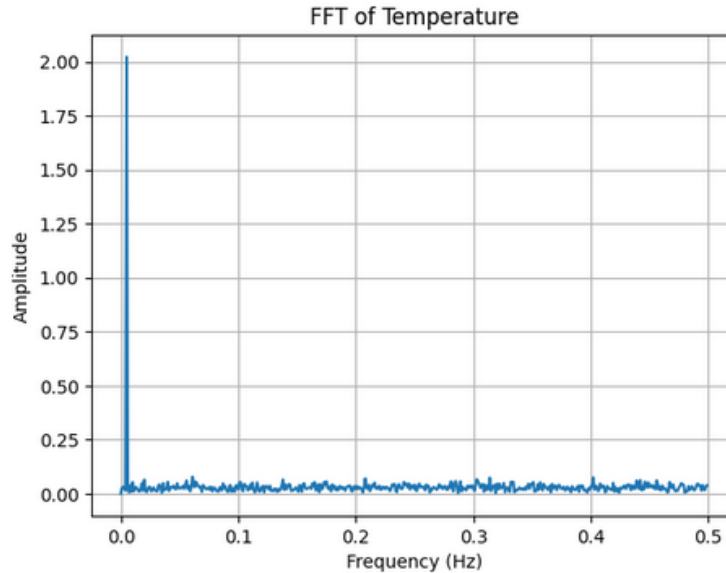


Рисунок 4.6 – Результати в Google Colab

Така спектральна картина підтверджує періодичний, але згасаючий характер температурного коливання, а також дозволяє підтвердити ефективність попередньої фільтрації сигналу.

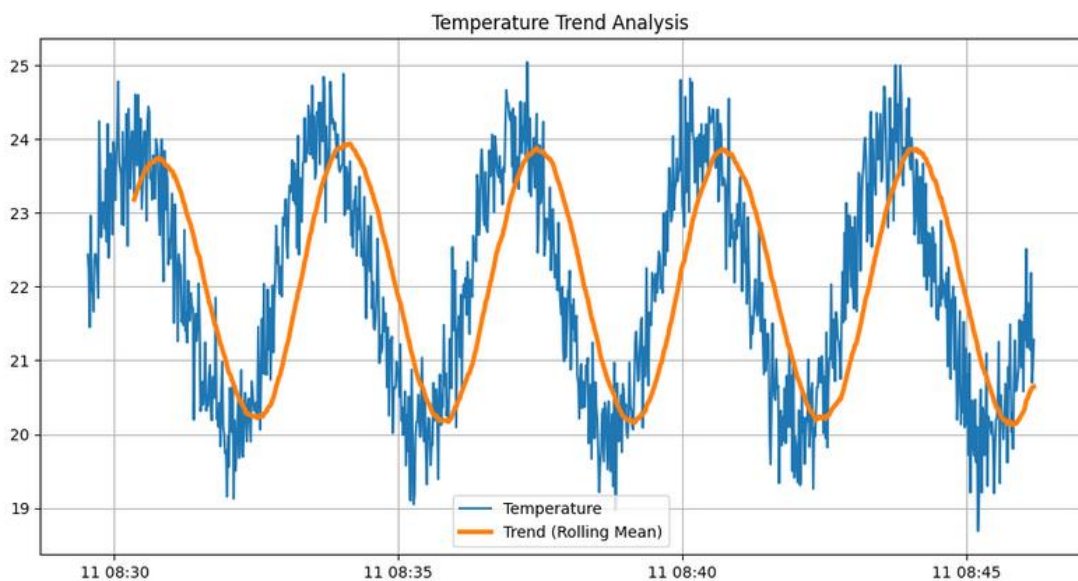


Рисунок 4.7 – Результати в Google Colab

На рисунку 4.7 показано результати аналізу тренду температурного сигналу з використанням ковзного середнього. Синя лінія відображає вихідні сенсорні вимірювання температури, які демонструють характерні коливання із шумовими флуктуаціями. Помаранчева крива — це згладжена траєкторія температури, яка дозволяє виділити основну тенденцію зміни величини в часі. Такий підхід особливо корисний для візуального виділення циклічних закономірностей, сезонності або довготривалих змін фону в даних, які можуть бути прихованими в первинному сигналі через високочастотний шум. Наявність чітко повторюваних піків і спадів підтверджує періодичний характер температурного сигналу, а також обґрунтовує доцільність використання методів трендового аналізу в межах інформаційної системи моніторингу.

ВИСНОВКИ

У ході виконання бакалаврської кваліфікаційної роботи було розроблено та реалізовано інформаційну систему обробки даних з безпроводної сенсорної мережі, яка функціонує на базі двох комплементарних середовищ: Matlab та Google Colab. Архітектура системи побудована з урахуванням функціонального розмежування: Matlab відповідає за моделювання даних, первинну обробку та базову візуалізацію, тоді як Google Colab реалізує розширену інтелектуальну аналітику, включаючи кореляційний аналіз, виявлення аномалій, спектральну декомпозицію та трендову оцінку.

У Matlab було синтезовано набір сенсорних даних, що включає змінні температури, тиску та вологості, з урахуванням імовірнісної природи вимірювань та шумових складових. Проведено згладжування сигналів, статистичну обробку та візуалізацію з метою ідентифікації динаміки змін. Результати демонструють наявність чітких періодичних компонент у температурі та вологості, а також висхідний тренд у тиску.

У Google Colab реалізовано обробку імпортованого CSV-файлу з використанням бібліотек Python. Здійснено виявлення аномалій за допомогою стандартного статистичного критерію, побудовано теплову карту кореляційної матриці, яка підтвердила слабкий лінійний зв'язок між сенсорними параметрами. Проведено швидке перетворення Фур'є температурного сигналу, результати якого засвідчили наявність домінуючої низькочастотної компоненти, що відповідає періодичній природі даних. Окрім того, реалізовано трендовий аналіз із використанням ковзного середнього, що дозволив візуалізувати основну динаміку коливань температури без шумових викривлень.

Отримані результати підтверджують працездатність розробленої інформаційної системи, її здатність адаптуватися до різних джерел даних,

виявляти ключові закономірності та забезпечувати основу для подальшого розширення з метою прогнозування, автоматичного моніторингу або інтеграції в складніші IoT-інфраструктури. Розділення обробки між Matlab та Colab довело свою ефективність і дозволило поєднати інженерну точність із гнучкістю машинного аналізу.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Mohammad Abu Alsheikh, Shaowei Lin, Dusit Niyato, Hwee-Pink Tan. Machine Learning in Wireless Sensor Networks: Algorithms, Strategies, and Applications. arXiv preprint. 2014. 58 p.
2. Joel W. Branch, Chris Giannella, Boleslaw Szymanski, Ran Wolff, Hillol Kargupta. In-Network Outlier Detection in Wireless Sensor Networks. arXiv preprint. 2009. 25 p.
3. Jayant Gupchup, Andreas Terzis, Randal Burns, Alex Szalay. Model-Based Event Detection in Wireless Sensor Networks. arXiv preprint. 2009. 20 p.
4. Antonio Ortega, Pascal Frossard, Jelena Kovačević, José M. F. Moura, Pierre Vandergheynst. Graph Signal Processing: Overview, Challenges and Applications. arXiv preprint. 2017. 30 p.
5. Mohammad Abu Alsheikh et al. Machine Learning in Wireless Sensor Networks... (як джерело методів ML у WSN). arXiv. 2014. 58 p.
6. Tiago Carneiro, Raúl V. Medeiros da Nóbrega, Thiago Nepomuceno, Gui-Bin Bian, Victor H. C. de Albuquerque, Pedro P. Rebouças Filho. Performance Analysis of Google Colaboratory as a Tool for Accelerating Deep Learning Applications. IEEE Access. 2018. 15 p.
7. Murat Dener, Samed Al, Abdullah Orman. STLGBM-DDS: An Efficient Data Balanced DoS Detection System for Wireless Sensor Networks on Big Data Environment. IEEE Access. 2022. 16 p.
8. Vivek Kumar Pandey, Shiv Prakash, Tarun Kumar Gupta, Priyanshu Sinha, Tiansheng Yang, Rajkumar Singh Rathore, Lu Wang, Sabeen Tahir, Sheikh Tahir Bakhsh. Enhancing intrusion detection in wireless sensor networks using a Tabu search based optimized random forest. Scientific Reports. 2025. Article 18634, 12 p.
9. Signal Processing for Intelligent Sensor Systems with MATLAB®,

Second Edition by Douglas Swanson (ed.). Wiley. 2012. 480 p.

10. Intelligent Information Processing with Matlab (Xiu Zhang, Xin Zhang, Wei Wang). Springer. 2024. 256 p.

11. TinyDB: Samuel R. Madden, Michael J. Franklin, Joseph M. Hellerstein, Wei Hong. TinyDB: an acquisitional query processing system for sensor networks. ACM Transactions on Database Systems (TODS), Vol. 30, No. 1. 2005. 26 p.

12. Kiran Maraiya, Kamal Kant, Nitin Gupta. Wireless Sensor Network: A Review on Data Aggregation. International Journal of Scientific & Engineering Research, Vol. 2 Issue 4. 2011. 7 p.