Хмельницький національний університет
Факультет інформаційних технологій
Кафедра комп'ютерної інженерії та інформаційних систем

КВАЛІФІКАЦІЙНА РОБОТА МАГІСТРА

Галузь знань 12 – Інформаційні технології

Спеціальність 123 –Комп'ютерна інженерія

на тему «Міська система інтернету речей для забезпечення підключення та збору даних з різних джерел»

КвРКІП. 301167.23.01.07 ПЗ

Виконав: студент 2 курсу, група КІ2м-23-1 Юрій ФУРМАН
Підпис Ім'я, прізвище

Керівник канд.тех.наук., доцент Катерина БЕРЕЗЬКА
Науковий ступінь, вчене звання
Підпис Ім'я, прізвище

До захисту допускаю:
Зав. кафедри КІПС, доктор філософії, доцент
Ольга ПАВЛОВА
19 05 2025 р.

Хмельницький, 2025

ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Факультет Інформаційних технологій

Кафедра Комп'ютерної інженерії та інформаційних систем

Освітній рівень магістр

Галузь знань 12 Інформаційні технології

Спеціальність 123 Комп'ютерна інженерія

Освітня програма освітньо-наукова програма «Комп'ютерна інженерія та програмування»

ЗАТВЕРДЖУЮ

Зав. кафедри Ольга ПАВЛОВА

" 01 " 09 2024 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ МАГІСТРА

Юрію ФУРМАНУ

Прізвище, ім`я, по батькові студента

1. Тема проекту (роботи) Міська система інтернету речей для забезпечення підключення та збору даних з різних джерел

Керівник проекту (роботи) Катерина БЕРЕЗЬКА, к.т.н., доцент

Прізвище, ім`я, по батькові, науковий ступінь, вчене звання

Затверджена наказом ректора університету від 08.01.2025 №8

2. Строк подання студентом проекту (роботи) на кафедру 01.05.2025 р.

3. Вихідні дані до проекту (роботи) Завдання на дипломне проектування

4. Зміст пояснювальної записки (перелік питань, які потрібно розробити)

Огляд сучасних підходів та вимог до міських систем Інтернету речей

Формалізація вимог та концептуальна модель міської IoT-системи

Реалізація архітектури та розробка програмних компонентів платформи

Моделювання та експериментальна оцінка ефективності запропонованого рішення

5. Перелік графічного матеріалу (із зазначенням обов'язкових креслень)

6. Консультанти розділів кваліфікаційної роботи магістра

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийня
Нормоконтроль	Сергій ЛИСЕНКО, професор кафедри КІІС		
Антиплагіат	Андрій НІЧЕПОРУК, доцент кафедри КІІС		

7. Дата видачі завдання «_01_»_____09_____2024р.

КАЛЕНДАРНИЙ ПЛАН

№з/п	Назва етапів (розділів) кваліфікаційної роботи магістра	Термін виконання етапів проекту (роботи)	Примітк
1	Вибір напряму дослідження та узгодження тематики КвРМ з керівником	01.09.2024	виконан
2	Ознайомлення з предметною областю; формулювання мети та задач дослідження; визначення об'єкта та предмета дослідження	01.10.2024	виконан
3	Робота над розділом 1 – аналіз відомих моделей, методів за темою; постановка задачі	01.11.2024	виконан
4	Робота над розділом 2 – розробка моделей для вирішення поставленої задачі	01.12.2024	виконан
5	Робота над науковою статтею	01.02.2025	виконан
6	Робота над розділом 3 – розробка методів для вирішення поставленої задачі	15.02.2025	виконан
7	Робота над розділом 4 – проектування та розробка ПЗ для вирішення поставленої задачі, експериментальна частина	01.04.2025	виконан
8	Оформлення пояснювальної записки згідно вимог	18.04.2025	виконан
9	Попередній захист ДРМ	29.04.2025	виконан
10	Захист ДРМ на засіданні ЕК	До 15.05.2025	

Студент _____ Підпис _____ Юрій ФУРМАН Ім'я, прізвище

Керівник роботи _____ Підпис _____ Катерина БЕРЕЗЬКА Ім'я, прізвище

РЕФЕРАТ

Тема кваліфікаційної роботи магістра: «Міська система інтернету речей для забезпечення підключення та збору даних з різних джерел».

Автор роботи: Фурман Юрій Олегович.

Керівник роботи: Березька Катерина Миколаївна.

Пояснювальна записка: 90 с., 18 рис., 5 табл., 2 дод., 108 джерел.

ІНТЕРНЕТ РЕЧЕЙ, РОЗУМНЕ МІСТО, СЕНСОРНІ МЕРЕЖІ, БЕЗДРОТОВІ ТЕХНОЛОГІЇ, ЗБІР ДАНИХ, АРХІТЕКТУРА СИСТЕМИ, ОБРОБКА ДАНИХ.

Об’єктом дослідження є міська система інтернету речей, яка забезпечує комунікацію та збір інформації від різнорідних сенсорних пристроїв у межах міста..

Предметом дослідження є технічні засоби, програмні компоненти та методи, що забезпечують підключення пристроїв і збір та обробку даних у міській IoT-системі для інтеграції різних джерел інформації.

Метою кваліфікаційної роботи магістра є розробка моделі міської системи джерел даних і збору та обробки інформації.

Для розв’язання поставлених задач використовувалися методи

- аналізу наукової та технічної літератури з питань IoT і розумного міста
- аналізу існуючих IoT-платформ та технологій мережевого зв'язку
- моделювання архітектури та компонентів системи
- експериментальна перевірка функціонування розроблених модулів

Наукова новизна отриманих результатів:

- у розвитку метод адаптивного вибору протоколів передачі даних для оптимізації збору й маршрутизації телеметричної інформації в міських IoT-системах;

– розвитку інформаційної технологія гібридної обробки даних на периферії та в хмарі (edge-cloud continuum) для зниження затримок і підвищення масштабованості обробки IoT-даних.

На основі проведених досліджень розроблено багаторівневу архітектуру міської IoT-платформи, що включає сенсорний рівень (множину датчиків і пристроїв збору даних), рівень периферійної обробки (edge-/fog-вузли для попередньої агрегації й фільтрації інформації) та хмарну інфраструктуру (централізоване збереження, обробка і аналіз даних).

Практична значимість отриманих результатів полягає у підвищенні ефективності збору та аналізу даних, що дозволяє своєчасно реагувати на зміни в урбаністичному середовищі і приймати обґрунтовані управлінські рішення.

У першому розділі наведено огляд сучасного стану досліджень з теми Інтернету речей у контексті міських систем. Проведено аналіз основних понять і технологій IoT.

У другому розділі виконано аналіз вимог до системи та обґрунтовано вибір технологій зв’язку і протоколів передачі даних.

У третьому розділі описано реалізацію архітектури системи.

У четвертому розділі представлено результати моделювання перевірки рішень.

ЗМІСТ

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ

IoT - (Internet of Things) Інтернет речей

MQTT - (Message Queuing Telemetry Transport) Протокол чергування повідомлень для телеметрії

AMQP - (Advanced Message Queuing Protocol) Розширений протокол чергування повідомлень

DDS - (Data Distribution Service) Служба розподілу даних

CoAP - (Constrained Application Protocol) Протокол обмежених додатків

LoRaWAN - (Long Range Wide Area Network) Мережа великої дальності з низьким енергоспоживанням

NB-IoT - (Narrowband Internet of Things) Вузькосмугова технологія Інтернету речей

LPWAN - (Low-Power Wide-Area Network) Мережа з низьким енергоспоживанням і великою дальністю дії

TCP - (Transmission Control Protocol) Набір протоколів транспортного та мережевого рівнів

HTTP/HTTPS - (Hypertext Transfer Protocol / Secure) Класичні веб-протоколи з підтримкою шифрування TLS

UDP - (User Datagram Protocol) Протокол датаграм другого рівня, на якому базується CoAP

DTLS - (Datagram Transport Layer Security) Шифрування датаграм (аналог TLS для UDP)

TLS - (Transport Layer Security) Протокол захищеного каналу передачі даних

API - (Application Programming Interface) Інтерфейс програмування додатків

ІКТ - (Інформаційно-комунікаційні технології) Комплекс телекомунікаційних і інформаційних засобів для обміну даними

ВСТУП

У сучасних умовах стрімкої урбанізації міста стикаються з величезним тиском на інфраструктуру та обмеженими ресурсами. Концепція «розумного міста» передбачає використання Інтернету речей для інтеграції та аналізу даних про транспорт, енергетику, довкілля тощо, що дозволяє оптимізувати міські процеси і підвищувати якість життя мешканців. Використання IoT у міській інфраструктурі дозволяє створювати автоматизовані системи моніторингу та управління ресурсами (енергія, вода, транспорт), що сприяє економії енергії, зменшенню забруднення та підвищенню ефективності комунальних послуг. З огляду на це, розробка міської системи Інтернету речей для забезпечення підключення і збору даних з різних джерел є надзвичайно актуальним завданням.

IoT-системи стають центральним компонентом концепції «розумного міста» у такій екосистемі зібрані дані аналізуються алгоритмами штучного інтелекту для оптимізації управління міською інфраструктурою. Використання штучного інтелекту дозволяє отримувати з IoT-даних нові інсайти та формувати прогностичні моделі, що підвищують ефективність енергозбереження, транспортного планування та безпеки громадян. Технологія блокчейн у цьому контексті пропонує механізми захищеного обміну інформацією та контролю доступу, що підвищує надійність і прозорість міських IoT-систем. Поєднання цих трендів - IoT, штучного інтелекту та блокчейн - відкриває нові можливості для створення стійких і безпечних міських інфраструктур.

Актуальність роботи полягає в розробці уніфікованої багаторівневої архітектури міської IoT-системи, яка забезпечить надійне підключення різноманітних сенсорів, централізований збір та обробку великих обсягів телеметричних даних із дотриманням вимог масштабованості, безпеки та енергоефективності.

Метою кваліфікаційної роботи магістра є розробка архітектури та функціональної моделі міської системи Інтернету речей для забезпечення

надійного підключення різнорідних сенсорних пристроїв і централізованого збору, агрегації та обробки даних у реальному часі.

Поставлена мета досягається розв’язанням таких основних завдань:

- проаналізувати наукові джерела та оглянутид сучасні підходів до побудови міських IoT-систем.;
- розробити концептуальну модель та багаторівневу архітектуру платформи;
- розробити основні програмні модулі;
- здійснити моделювання та експериментальну перевірку розробленої системи з оцінкою ключових показників продуктивності, надійності та масштабованості.

Об’єктом дослідження є процеси організації зв’язку, збору і передачі даних у міських системах інтернету речей, а також апаратно-програмні компоненти, що забезпечують інтеграцію сенсорних пристроїв у рамках єдиної платформи.

Предметом дослідження є технічні засоби, протоколи та алгоритми, що забезпечують підключення сенсорних пристроїв, а також методи збору, маршрутизації й обробки телеметричних даних у міській IoT-системі.

Науковою новизною отриманих результатів є оптимізовані алгоритми маршрутизації, агрегації та об’єднання даних із різних.

На основі проведених досліджень розроблено алгоритми маршрутизації та агрегації телеметричних даних для різнорідних сенсорів із динамічним налаштуванням параметрів мережевих протоколів

Практична значимість отриманих результатів полягає у підвищенні надійності та масштабованості IoT-інфраструктури завдяки уніфікованій архітектурі та адаптивним алгоритмам роботи з різнорідними джерелами.

Для розв’язання поставлених задач використовувалися методи моделювання та симуляції мережевих систем для оцінки продуктивності і надійності..

За темою кваліфікаційної роботи опубліковано одну публікацію у Збірнику наукових праць за матеріалами Всеукраїнської науково-практичної конференції

студдентів, Аспірантів та молодиї вчених «ІНТЕЛЕКТУАЛЬНІ КОМП'ЮТЕРНІ СИСТЕМИ ТА МЕРЕЖІ». (Тернопіль – 2024. – С. 49-50).

1.3 Постановка задачі

Поставлена мета досягається розв'язанням таких основних завдань:

- розробити формальну специфікацію вимог до міської IoT-системи, що охоплює функціональні та нефункціональні показники;
- створити концептуальну модель багаторівневої архітектури із чітко виокремленими сенсорним, периферійним (edge) та хмарним рівнями;
- розробити алгоритми збору, маршрутизації й агрегації телеметричних даних із гетерогенних сенсорів із урахуванням оптимізації мережевого трафіку.;
- здійснити моделювання та експериментальну перевірку розроблених рішень із оцінкою ключових показників продуктивності, надійності та масштабованості системи;

1.4 Висновки до першого розділу

У першому розділі виконано комплексний аналіз сучасних підходів до побудови міських IoT-систем. Визначено ключові поняття Інтернету речей, розглянуто приклади реальних рішень та виявлено головні проблеми їхнього впровадження:

- фрагментація протоколів і нестандартизовані інтерфейси, що ускладнюють інтеграцію гетерогенних пристроїв;
- надмірне навантаження на мобільні мережі та інтерференція в радіочастотному середовищі через велику кількість IoT-точок;
- виклики обробки великих обсягів даних у режимі реального часу.

Також проведено огляд архітектурних рішень від класичних хмарних до гібридних моделей, і проаналізовано найпоширеніші протоколи передачі даних.

1 ОГЛЯД СУЧАСНИХ ПІДХОДІВ ТА ВИМОГ ДО МІСЬКИХ СИСТЕМ ІНТЕРНЕТУ РЕЧЕЙ

1.1 Інтернет речей

Інтернет речей - це концепція, яка передбачає інтеграцію фізичних об’єктів, пристроїв і сенсорів до глобальної мережі Інтернет для автоматизованого збору, обміну та обробки даних. Завдяки цьому звичайні об’єкти отримують можливість «спілкуватися» один з одним без безпосереднього втручання людини, що створює основу для побудови інтелектуальних систем управління та моніторингу (рисунок 1.1) [1].

Рисунок. 1.1 – Інтернет речей [1]

Основна ідея IoT полягає в обладнанні різноманітних предметів датчиками та комунікаційними модулями, завдяки яким вони можуть генерувати інформацію про своє оточення та передавати її в реальному часі. Наприклад, датчики температури, вологості, руху або якості повітря, інтегровані в елементи міської інфраструктури, дозволяють здійснювати контроль за умовами навколишнього середовища і сприяти оперативному прийняттю рішень.

У контексті міського управління Інтернет речей відіграє ключову роль, адже сприяє перетворенню традиційних міст у «розумні». Системи на базі IoT дозволяють оптимізувати використання ресурсів, підвищувати ефективність роботи транспортних мереж, енергопостачання та систем безпеки. Інтеграція з іншими технологіями, такими як штучний інтелект і блокчейн, відкриває можливості для проведення аналізу даних у режимі реального часу, що значно покращує якість управління міською інфраструктурою.

Важливими аспектами впровадження IoT є:

- моніторинг ресурсів;
- інтелектуальні транспортні системи;
- екологічний контроль;
- сфера охорони здоров’я;
- розумна інфраструктура та безпека;
- управління відходами.

Завдяки впровадженню IoT-модулів можна в реальному часі відстежувати споживання води, електроенергії, тепла тощо. Смарт-лічильники та датчики рівня відкритості та закритості трубопроводів передають дані на централізовану платформу, що дає місту змогу не лише оптимізувати витрати, а й негайно виявляти аварійні ситуації - наприклад, витоки чи непродуктивні витрати енергії.

IoT-пристрої на транспорті та в дорожній інфраструктурі збирають інформацію про потоки руху, завантаженість доріг та параметри роботи світлофорів. Це дозволяє адаптивно налаштовувати режими світлофорів, прокладати оптимальні маршрути в реальному часі та зменшувати затори, підвищуючи при цьому безпеку учасників дорожнього руху.

Мережі сенсорів для вимірювання якості повітря, шумового забруднення, рівня радіації та вологості ґрунту забезпечують постійний моніторинг найбільш вразливих зон міста. Отримані дані використовуються органами екологічного нагляду для швидкого реагування на перевищення допустимих норм і планування профілактичних заходів.

Розумні медичні пристрої, інтегровані в пацієнтські монітори чи носимі гаджети, передають важливі життєві показники в реальному часі. Лікарі можуть дистанційно відстежувати стан пацієнтів як у стаціонарі, так і на дому, що підвищує оперативність медичного реагування.

IoT-системи автоматизують контроль за освітленням, опаленням, вентиляцією та протипожежними системами. Відеокамери й датчики руху в громадських просторах інтегруються у єдину безпекову мережу, що дозволяє оперативно виявляти підозрілі події та підтримувати порядок у місті.

Сенсори в контейнерах для сміття фіксують рівень їх заповненості і передають ці дані в диспетчерський центр. Це дає змогу оптимізувати маршрути сміттєвозів, скоротити кількість порожніх рейсів і знизити експлуатаційні витрати, підвищуючи загальну ефективність системи збору й утилізації відходів.

Використання IoT-технологій дозволяє створювати інфраструктурні об’єкти, які автоматично контролюють освітлення, опалення, системи безпеки та інші параметри. Камери відеоспостереження та інші датчики сприяють виявленню аномальних ситуацій або потенційних загроз, що забезпечує підвищення рівня безпеки на вулицях міста. Завдяки впровадженню IoT можливо організувати інтелектуальні системи збору та переробки відходів.

З визначення інтернету речей випливає також поняття пристрою інтернету речей - пристрій, що має вимірювальні або керуючи властивості, та в автоматичному режимі обмінюються даними через вбудовані засоби з глобальною або локальною комп’ютерною мережею [2].

Термін «Інтернет речей» зазвичай стосується сценаріїв, коли мережа підключення та обчислювальна здатність поширюється на об’єкти, датчики та повсякденні предмети, а не зазвичай вважаються комп’ютерами, що дозволяє цим пристроям генерувати, обмінюватися та використовувати дані мінімальне втручання людини.

Інтернет речей мають реалізувати кілька основних принципів:

- безпека;
- приватність;

– стандартизація.

Хоча питання безпеки не є новими в контексті інформаційних технологій, атрибути багатьох реалізацій IoT створюють нові та унікальні проблеми безпеки. Вирішення цих проблем і забезпечення безпеки в продуктах і послугах Інтернету речей мають бути основним пріоритетом. Користувачі повинні вірити в те, що пристрої IoT і пов’язані з ними служби даних захищені від вразливостей, особливо враховуючи те, що ця технологія стає все більш поширеною та інтегрованою в наше повсякденне життя.

Повний потенціал Інтернету речей залежить від стратегій, які поважають індивідуальний вибір конфіденційності в широкому спектрі очікувань. Потоки даних і особливості користувача, які надають пристрої IoT, можуть відкрити неймовірну й унікальну цінність для користувачів IoT, але занепокоєння щодо конфіденційності та потенційної шкоди може стримати повне впровадження Інтернету речей. Це означає, що права на конфіденційність і повага до очікувань користувачів щодо конфіденційності є невід’ємною частиною забезпечення довіри користувачів до Інтернету, підключених пристроїв і пов’язаних служб

Універсальні інтерфейси та єдині протоколи дозволяють різним пристроям безшовно взаємодіяти в єдиній екосистемі. Відсутність стандартів призводить до складнощів інтеграції, ризику «закритих» платформ і відмови споживачів від продуктів. Тому відкриті специфікації й взаємодія між виробниками — ключ до масштабування IoT.

1.2 Приклади пристроїв інтернету речей

Пристрої Інтернету речей активно використовуються в різних сферах міського життя. Вони допомагають автоматизувати процеси, підвищують ефективність управління інфраструктурою та покращують якість життя мешканців. Далі наведено кілька прикладів реалізації IoT у міських умовах.

1.2.1 Система радіаційного моніторингу в місті RadAr

RadAr - це розподілена система вимірювання, яка може бути використана для інтеграції в розумне місто для вимірювання та моніторингу радіаційного фону в містах в реальному часі (рисунок 1.2) [3]. Кожен пристрій передає дані через мережу Internet на сервер. Інформація про рівень радіації отримується з інтелектуальних блоків детектування, з'єднаних з комп'ютерами, які мають бути підключені до мережі Internet.

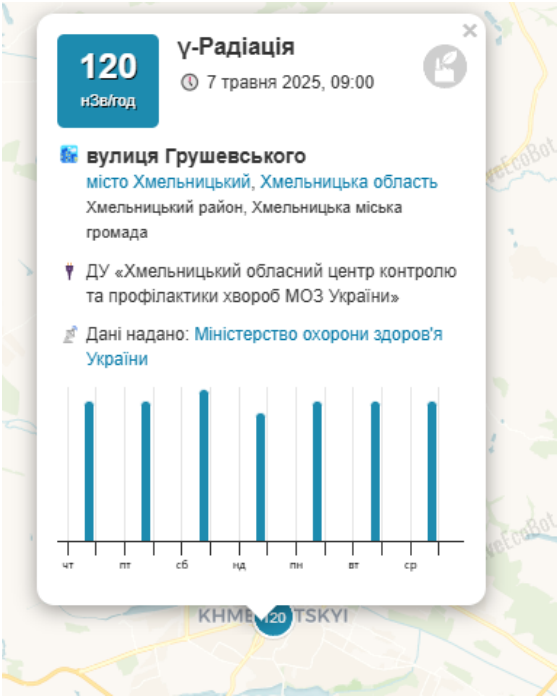


Рисунок 1.2 – Приклад роботи системи RadAr[3]

Кожен такий комп'ютер є клієнтом, точкою збору, що передає дані на сервер збору, відображення, обробки дозиметричної інформації має бути приєднаний до мережі Internet через статичну адресу. За оригінальною ідеєю пристрої системи мають розташовуватися на громадському транспорті для покриття як можна більшої площі з мінімальною кількістю ресурсів.

Основним недоліком даної системи можна вважати те, що кожен пристрій використовує окреме з'єднання з мережею інтернет через мобільну мережу, що може створювати надмірне навантаження на міські мережі при великій кількості

пристроїв. При цьому передається невелика кількість даних (Значення фону та координати), що робить такий спосіб неефективним.

1.2.2 Система моніторингу транспорту в місті EasyWay

EasyWay це сервіс який було створено у 2011 аби допомогти людині зорієнтуватись у незнайомому місті та підказати, який громадський транспорт обрати для свого пересування. EasyWay успішно працює в багатьох містах України та допомагає користувачам знаходити необхідний транспорт та планувати свої справи за допомогою цієї інформації (рисунок 1.3) [4].

Система в реальному часі відображає весь міський транспорт на мапі. Великий потік даних, що генерується і передається в реальному часі через мережу мобільного зв'язку створює додаткове навантаження на мережу, але є доцільним в цьому разі, бо є необхідність передавати дані одразу без їх обробки в режимі реального часу.

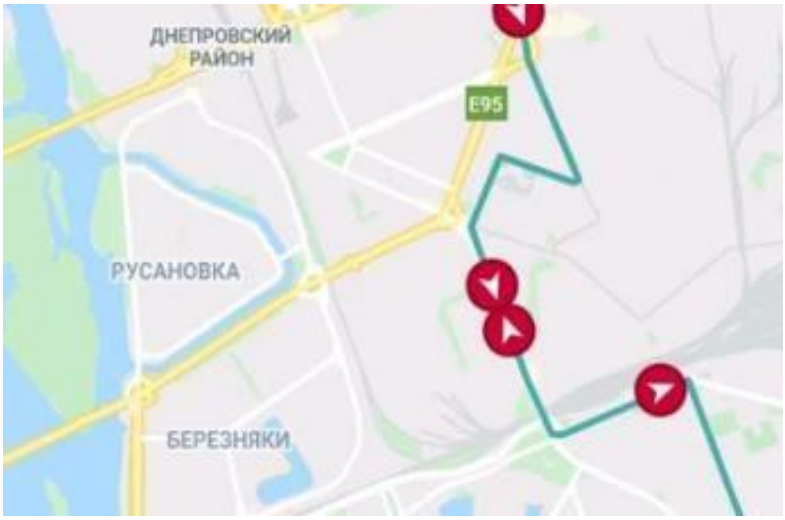


Рисунок 1.3 – Приклад роботи EasyWay [4]

1.3 Визначення загальних проблем

Більшість сучасних систем Інтернету речей, інтегрованих у розумне місто, стикаються з фундаментальними проблемами у сфері передачі інформації. Деякі пристрої без потреби надсилають дані в режимі реального часу, тоді як інші зовсім не забезпечують належної обробки та передачі зібраних показників. Така неоднорідність не лише ускладнює інтеграцію різних систем у єдину мережу управління, але й створює додаткове навантаження на мережеві ресурси, що може призводити до збоїв і перешкод у функціонуванні розумного міста.

Невідповідність у методах збору, обробки та передачі даних зумовлює перевантаження як центральних, так і периферійних мережевих систем. Це, в свою чергу, підвищує ризик помилок при обміні інформацією та знижує ефективність моніторингу ключових показників міського середовища. В результаті, критично важливі процеси, такі як контроль за споживанням ресурсів або управління транспортними потоками, можуть страждати від затримок або неточних даних, що негативно впливає на прийняття оперативних управлінських рішень.

Для успішної інтеграції IoT у міську інфраструктуру необхідно розробити та впровадити уніфіковану систему збору та обробки даних. Такий підхід дозволить стандартизувати обмін інформацією між різними пристроями, оптимізувати використання мережевих ресурсів та забезпечити більш надійне та ефективне функціонування розумного міста. Це, в свою чергу, сприятиме покращенню якості управління ресурсами, підвищенню безпеки і створенню більш комфортного середовища для мешканців.

1.3.1 Надлишкове навантаження мобільних мереж. Забруднення радіоетеру.

Основна проблема виникає через те, що більшість пристроїв Інтернету речей використовує мобільні мережі для передачі даних. Масове підключення таких пристроїв призводить до значного збільшення обсягів даних, що передаються через мобільні канали [5]. Це, в свою чергу, створює надмірне навантаження на мобільні

мережі, що може негативно позначитися на їхній пропускній здатності та збільшити затримки у передачі інформації. Таке перевантаження спричиняє цифрове забруднення радіоетеру на тих же частотах, що використовуються для мобільного зв’язку.

Постійний потік сигналів від численних IoT-пристроїв створює інтерференцію, що знижує якість зв’язку як для цих пристроїв, так і для звичайних користувачів мобільного зв’язку. В результаті може виникнути ситуація, коли знижується швидкість передачі даних, погіршується якість голосових дзвінків та відеоконференцій, а також зростає кількість переривань у зв’язку (рисунок 1.4) [6].

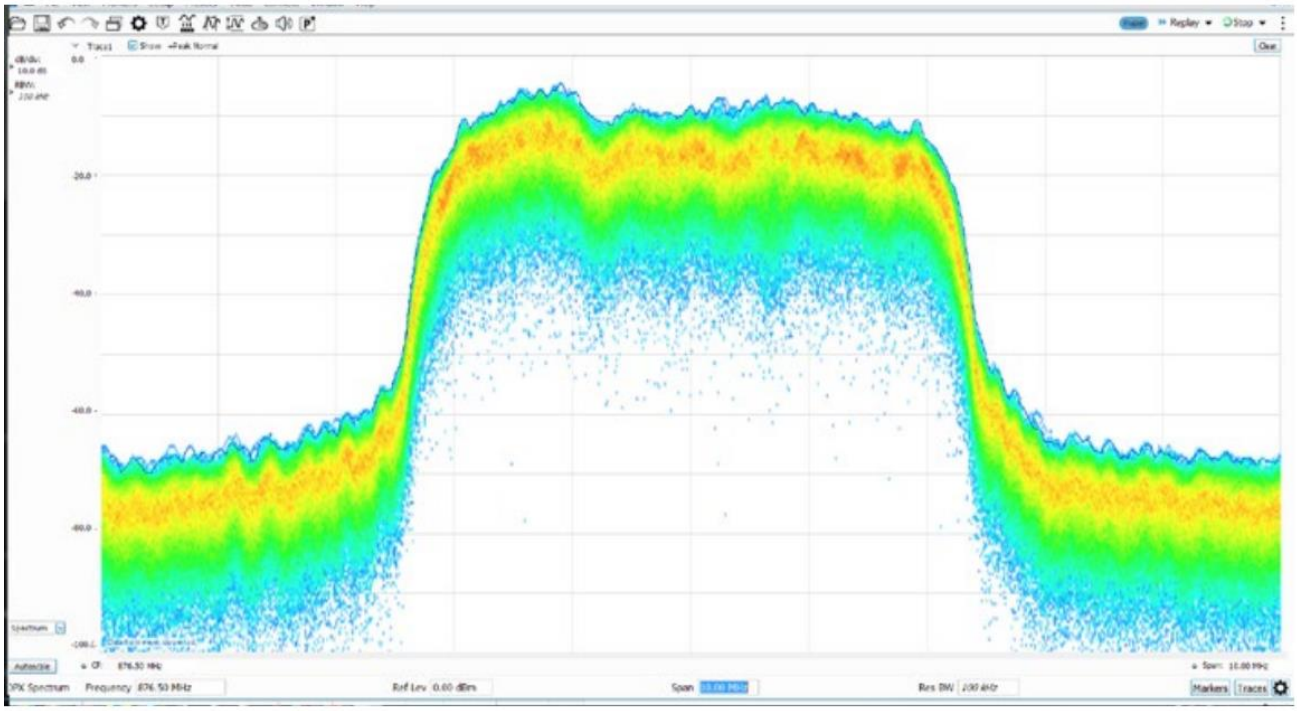


Рисунок 1.4 – Спектрограма радіоетеру на частотах, що відповідають частотам мобільного зв’язку [5]

Проблема надлишкового навантаження набуває особливої актуальності у міських умовах, де велика кількість IoT-пристроїв працює паралельно із традиційними мобільними користувачами. Це вимагає впровадження спеціалізованих методів оптимізації мережевого трафіку, таких як розумне

управління потоками даних, застосування адаптивних протоколів передачі інформації та стратегій зниження інтерференції.

1.3.2 Відсутність стандартизації

Однією з ключових проблем впровадження IoT-рішень у міських системах є відсутність єдиних стандартів. У зв'язку з широким спектром технологій, виробників та протоколів, що використовуються для збору, передачі та обробки даних, немає узгодженого підходу до інтеграції різноманітних пристроїв і систем у єдину інфраструктуру. На ринку існує безліч бездротових та дротових технологій (Wi-Fi, ZigBee, LoRaWAN, NB-IoT, 5G), які використовуються для зв'язку IoT-пристроїв. Кожен з цих протоколів має свої особливості, що створює труднощі при їх інтеграції в єдину систему міського управління [7].

Відсутність стандартизованих інтерфейсів та протоколів комунікації призводить до проблем із сумісністю між пристроями різних виробників. Це ускладнює розгортання масштабних систем, де кожен компонент має безперешкодно взаємодіяти з іншими. Велика кількість різноманітних стандартів призводить до того, що системи розумного міста стають складними для масштабування та інтеграції нових елементів до системи, що, також ускладнює впровадження ефективниї засобів захисту інформації, що передається між пристроями, а також забезпечинню належного рівеня контролю якості даних. Це підвищує ризик кібератак та несанкціонованого доступу до систем.

1.3.3 Big data

У контексті міських IoT-систем, що забезпечують підключення та збір даних із численних джерел, поняття Big Data набуває особливого значення. Big Data стосується надзвичайно великих, різноманітних та швидко зростаючих обсягів даних, які неможливо ефективно обробляти за допомогою традиційних методів аналізу та зберігання [8].

Пристрої інтернету речей в системах розумного міста генерують величезні об’єми даних. Побудова системи для збору та обробки даних без урахування необхідності обробляти велику кількість даних не має жодного практичного сенсу.Для вирішення загальних проблем була поставлена задача розробки концепції максимально уніфікованої системи для збору та обробки даних в межах розумного міста.Головною метою є опис системи, пристроїв що відносяться до неї, та теоретична та практична перевірка (на рівні обробки отриманих даних) доцільності використання такої системи для уніфікації розумного міста.

1.4 Основи розумного міста

Розумне місто - це поняття, яке охоплює численні сфери життєдіяльності: транспорт, освіту, охорону здоров’я, адміністративне управління, державну безпеку, інфраструктуру, логістику, ІКТ, архітектуру, відпочинок, екологію, будівництво та ефективне використання ресурсів. Ці сектори створюють складну мозаїку, кожна частина якої відіграє важливу роль у формуванні щоденного життя мешканців [9]. Для повноцінного розуміння концепції розумного міста необхідно враховувати як окремі елементи, так і їх взаємозв’язок, що вимагає спеціальних знань та навичок.

Місто, окрім того, що є географічним простором, виступає як автономна одиниця самоврядування, котра за свою історію зазнала змін у політичних формах, технологічному розвитку, збереженні навколишнього середовища та формуванні суспільного багатства. Зростання концентрації державних послуг та можливостей для розвитку робить міста дедалі привабливішими для населення, що пояснюється збільшенням міграції до міських територій. Водночас еволюція міста дає підстави для дослідження основних принципів, на яких базується ідея розумного міста. Концепція розумного міста постійно змінюється й адаптується до нових викликів [10].

За словами незалежних дослідницьких компаній, застосування ІКТ дозволяє зробити ключові компоненти міської інфраструктури - від адміністративних служб

до транспортних систем - більш інтегрованими та ефективними [11]. Аналогічно, Європейський Парламент визначає розумне місто як простір, де вирішення суспільних проблем здійснюється за допомогою ІКТ та партнерства між муніципальними структурами [12]. Ключовим чинником у створенні розумного міста є інтеграція інформаційно-комунікаційних технологій у всі сфери міського життя. Сучасні технології дозволяють не лише збирати та аналізувати дані, але й оперативно приймати рішення на основі отриманих результатів. Завдяки цьому системи управління містом стають більш гнучкими, а інфраструктура - адаптивною до змінних умов сучасного середовища.

Впровадження розумних технологій відкриває широкі можливості для модернізації транспортних систем, енергопостачання, охорони здоров’я, освіти та комунальних послуг [13]. Наприклад, сучасні системи моніторингу трафіку дозволяють оптимізувати маршрути, знижуючи затори і підвищуючи безпеку дорожнього руху. Аналогічно, розумні енергосистеми сприяють оптимізації споживання ресурсів, що позитивно позначається як на економічних показниках, так і на екологічній ситуації.

Окрім технологічної складової, важливим елементом є партнерство між державними структурами, приватним сектором і громадськістю. Такий інтегрований підхід дозволяє враховувати інтереси всіх зацікавлених сторін, сприяє більш ефективному використанню ресурсів і створенню умов для сталого розвитку. Колективна взаємодія сприяє виявленню й вирішенню міських проблем ще на стадії їх формування, що забезпечує швидку адаптацію системи управління до нових викликів.

1.5 Визначення та важливість протоколів передавання даних

Протокол передавання даних - набір угод інтерфейсу логічного рівня, які визначають обмін даними між різними програмами [14]. Ці угоди задають однаковий спосіб передачі повідомлень і обробки помилок при взаємодії

програмного забезпечення рознесеного на просторі апаратної платформи, з'єднаної тим чи іншим інтерфейсом.

Протоколи передачі даних складають основу сучасних систем зв'язку, забезпечуючи безперебійний обмін інформацією між пристроями та мережами. По суті, протокол передачі даних визначає правила і домовленості, які регулюють передачу даних. Це набір керівних принципів, які забезпечують надійний та ефективний зв'язок.

Ці протоколи дають змогу ефективно взаємодіяти різноманітним пристроям, від комп'ютерів і смартфонів до датчиків в Інтернеті речей. Встановлюючи стандартизований метод обміну даними, протоколи сприяють сумісності різних систем, сприяючи створенню згуртованого і пов'язаного технологічного середовища.

1.5.1 Мережевий протокол

Мережевий протокол - набір правил, що визначає комп'ютери у мережі [15]. Протокол також задає загальні правила взаємодії різноманітних програм, мережевих вузлів чи систем і створює таким чином єдиний простір передачі. Хости (будь-який вузол мережі що відправляє або приймає дані через мережу називають хостом (host) взаємодіють між собою. Для того, щоб прийняти і обробити відповідним чином повідомлення, їм необхідно знати як сформовані повідомлення і що вони означають.

Мережеві протоколи необхідні для належного функціонування та взаємодії комп'ютерних мереж. Ці протоколи регулюють різні аспекти мережевої комунікації, включаючи форматування даних, виявлення та виправлення помилок, адресацію, маршрутизацію, а також встановлення, підтримку та розірвання з'єднань. Вони гарантують, що пристрої в мережі можуть ефективно взаємодіяти, забезпечуючи спільну мову і набір правил для передачі та отримання даних.

1.5.2 Класифікація протоколів передачі даних

Протоколи передачі даних класифікуються за середовищем передачі. Дротові протоколи, як-от Ethernet, забезпечують надійний та безпечний зв'язок через фізичні кабелі [16], тоді як бездротові, зокрема Wi-Fi та Bluetooth, позбавляють необхідності кабельного з'єднання, забезпечуючи більшу гнучкість. Надійність протоколів визначається їх здатністю виявляти та виправляти помилки, що демонструє, наприклад, TCP, орієнтований на з'єднання, а масштабованість - UDP, який оптимізований для високошвидкісної передачі даних, незважаючи на меншу перевірку помилок [17].

У сучасному контексті, коли витоки даних і кіберзагрози є особливо актуальними, безпека передачі визначається використанням криптографічних протоколів, таких як SSL/TLS, що забезпечують конфіденційність і цілісність даних. Захищені протоколи передачі файлів гарантують відповідність вимогам законодавства про захист інформації [18].

Поточні тенденції спрямовані на інтеграцію штучного інтелекту для підвищення ефективності протоколів, розробку спеціалізованих рішень для пристроїв IoT та стандартизацію протоколів, що дозволяють забезпечити ефективний та безпечний обмін даними в умовах високошвидкісних мереж.

1.6 Архітектурні рішення мереж IoT

Мережева архітектура IoT визначає структуру та організацію елементів екосистеми, забезпечуючи ефективний обмін даними та управління пристроями [19] У її основі лежать фізичні датчики та приводи, що генерують дані, а також рівень підключення, який забезпечує комунікацію через Wi-Fi, Bluetooth, Zigbee, LoRa, 3G/4G/5G та інші технології.На граничному обчислювальному рівні (Edge Computing Layer) здійснюється первинна обробка даних для зменшення затримок, що реалізується за допомогою шлюзів та периферійних серверів. Ці IoT-шлюзи збирають та виконують базову обробку інформації перед її надсиланням у хмарні

платформи, де дані зберігаються, аналізуються та використовуються для виконання складної аналітики, машинного навчання та прийняття рішень.

Ключовим компонентом є також рівень зберігання даних, де використовується як реляційні, так і NoSQL-бази даних для довгострокового зберігання та швидкого пошуку інформації. Прикладний рівень включає програми моніторингу, аналітичні інструменти та користувацькі інтерфейси, що забезпечують взаємодію з системою. Безпека мережевої архітектури покладається на механізми автентифікації, шифрування даних, контролю доступу та захисту від кіберзагроз [20]. Управління мережею передбачає також забезпечення пристроїв, оновлення прошивки та оптимізацію мережевих ресурсів.Обмін даними здійснюється через протоколи MQTT, CoAP, HTTP, а також протоколи, спеціально розроблені для конкретних технологій зв’язку .

Вибір між периферійними та хмарними обчисленнями обумовлюється вимогами до затримки, конфіденційності, обчислювальної потужності та пропускної здатності. Часто застосовується комбінований підхід, відомий як «периферійно-хмарний континуум», який дозволяє адаптувати архітектуру відповідно до конкретних задач та умов експлуатації (рисунок 1.5) [21].

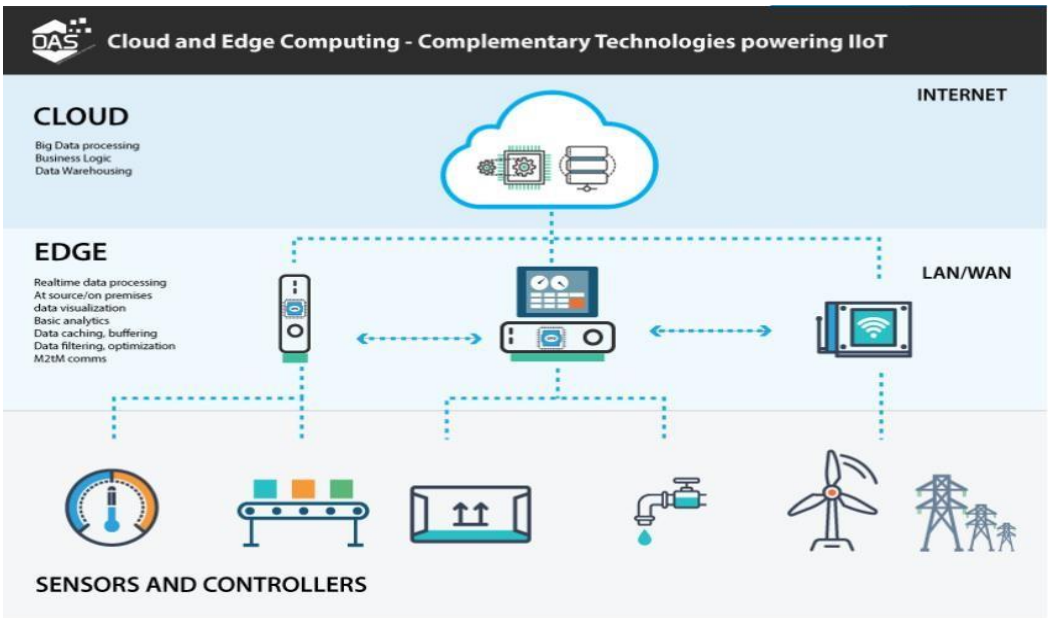


Рисунок 1.5 – Гібридний підхід з використанням граничних та хмарних обчислень [21]

1.7 Протоколи передачі даних IoT. Визначення та важливість протоколів передавання даних

MQTT - це широко використовуваний і високоефективний протокол обміну повідомленнями, розроблений для додатків Інтернету речей (IoT). Він працює за моделлю "публікація/підписка (publish/subscribe)", забезпечуючи легкий і надійний метод зв'язку для пристроїв в екосистемах IoT. MQTT побудовано на концепції парадигми обміну повідомленнями за принципом публікації/підписки. Пристрої спілкуються, публікуючи повідомлення в певних категоріях, а інші пристрої підписуються на ці категорії, щоб отримувати повідомлення (рисунок 1.6) [22].

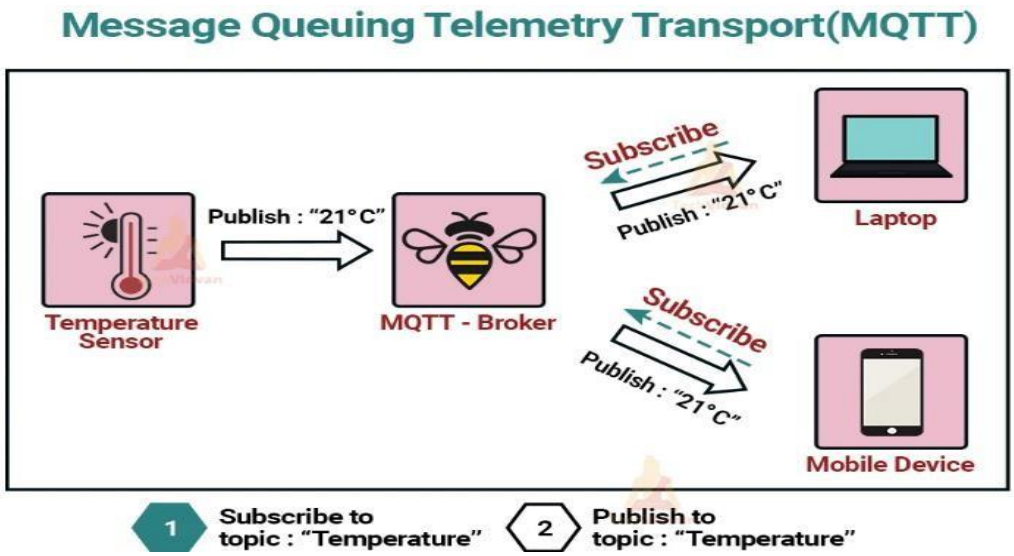


Рисунок 1.6 – Протокол Message Queuing Telemetry Transport(MQTT) [22]

Advanced Message Queuing Protocol (AMQP) - це універсальний протокол обміну повідомленнями з відкритим вихідним кодом, призначений для ефективного обміну даними в екосистемах IoT. Спочатку розроблений JPMorgan Chase & Co у 2003 році, AMQP став визначним вибором для забезпечення надійного та безпечного зв'язку в різних додатках IoT. AMQP працює як відкритий і стандартизований протокол через стек TCP/IP, забезпечуючи сумісність та взаємодію між різними пристроями та платформами в мережах IoT.AMQP

підтримує кілька моделей обміну повідомленнями, включаючи парадигми "запит-відповідь (request-response)" і "видавець-підписник (publisher-subscriber)".

Ця гнучкість дозволяє йому пристосовуватися до різних моделей зв'язку залежно від конкретних потреб додатків Інтернету речей. В архітектурі AMQP обмін відіграє вирішальну роль. Коли видавець генерує повідомлення, біржа (exchange) визначає правила і ключі маршрутизації для пересилання повідомлення до відповідного місця призначення. Черги - це невід'ємні компоненти, які приймають повідомлення, що пересилаються біржами. Вони забезпечують належну доставку повідомлень споживачам на основі конфігурації пристроїв у мережі (рисунок 1.7) [23].

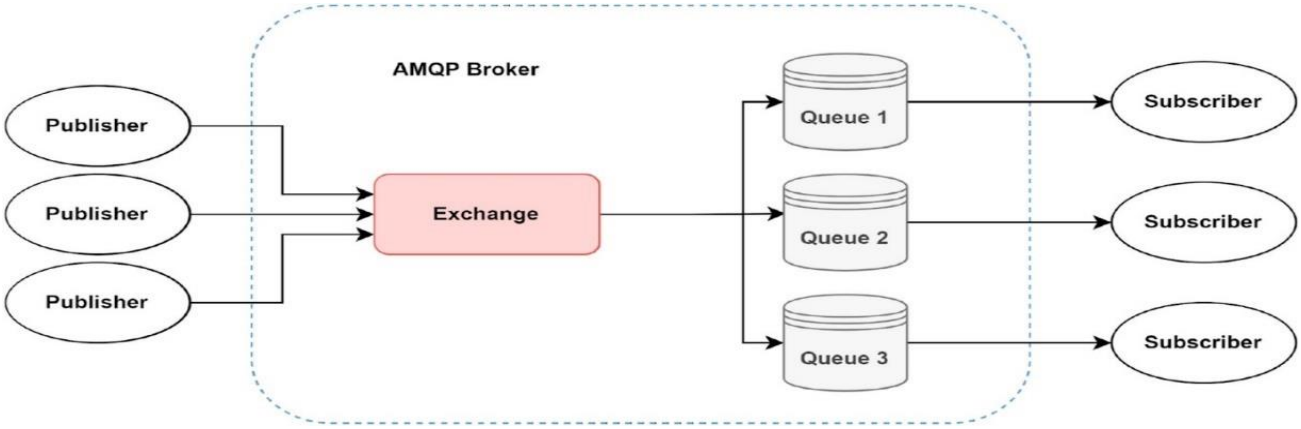


Рисунок 1.7 – Advanced Message Queuing Protocol [23]

Data Distribution Service (DDS - Служба розподілу даних) - це надійний і універсальний протокол передачі даних, який широко використовується в Інтернеті речей (IoT. Відомий своєю універсальністю та розширюваністю, DDS забезпечує прямий зв'язок між датчиками та іншими пристроями, усуваючи потребу в посередниках. DDS славиться своєю універсальністю, легко адаптуючись до різних сценаріїв зв'язку в додатках IoT) [24]. Його розширюваність дозволяє йому пристосовуватися до вимог, що змінюються, і технологічних новинок (рисунок 1.8.).

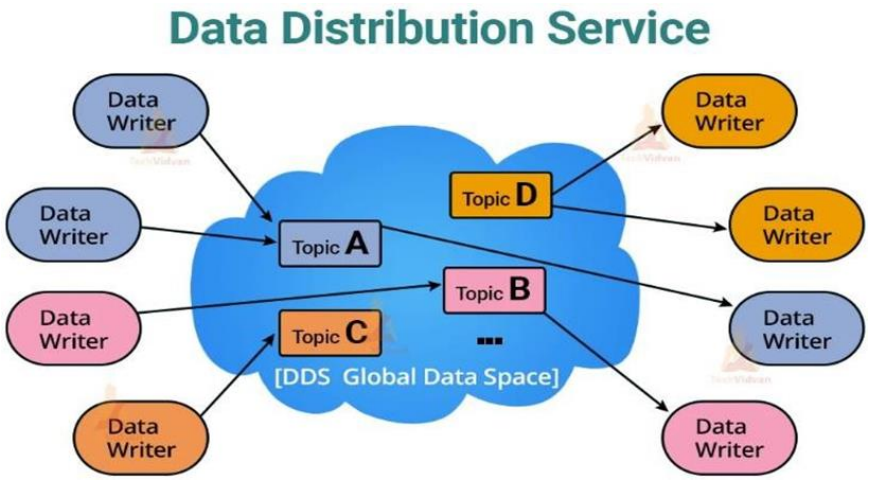


Рисунок 1.8 – Data Distribution Service (DDS) [24]

Однією з відмінних рис DDS є її здатність встановлювати прямий зв'язок між датчиками та пристроями. Таке пряме з'єднання підвищує ефективність і швидкість реагування в мережах IoT. DDS робить сильний акцент на безпечній передачі повідомлень, забезпечуючи конфіденційність і цілісність даних під час передачі. Це має вирішальне значення для підтримки безпеки інформації в мережах IoT

Протокол обмежених додатків (Constrained Application Protocol, CoAP) - це спеціалізований протокол, розроблений для Інтернету речей (IoT), що забезпечує ефективний зв'язок, пристосований для пристроїв з обмеженими характеристиками в мережах IoT (рисунок 1.9) [25].

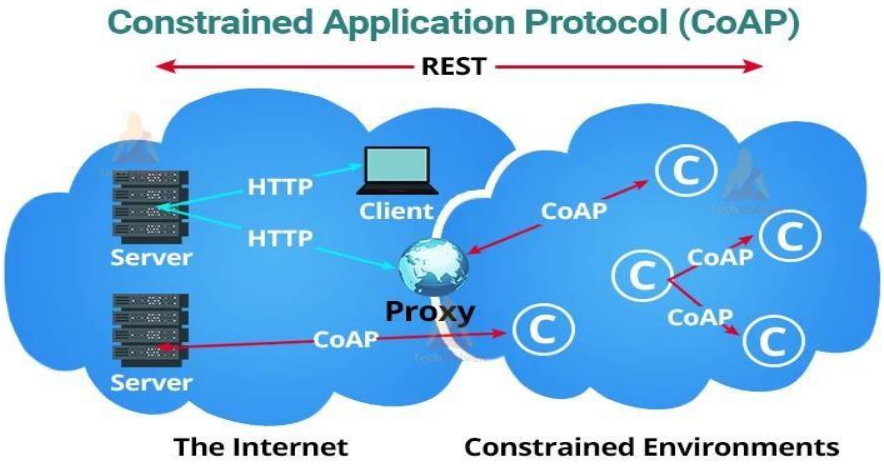


Рисунок 1.9 – Constrained Application Protocol (CoAP) [25]

Підхід CoAP схожий на HTTP для передачі документів, але оптимізований для середовищ з обмеженими ресурсами. CoAP спеціально розроблений для задоволення потреб вузлів з обмеженими ресурсами, які часто зустрічаються в пристроях IoT. Його дизайн ставить на перше місце ефективність зв'язку, що робить його придатним для пристроїв з обмеженою обчислювальною потужністю та пропускною спроможністю.

LoRaWAN - це протокол зв'язку, розроблений для зв'язку на великі відстані з низьким енергоспоживанням в Інтернеті речей (IoT). Він добре підходить для додатків, які потребують зв'язку на великі відстані з мінімальним енергоспоживанням, що робить його особливо корисним для пристроїв IoT в різних галузях промисловості.

LoRaWAN призначена для забезпечення зв'язку на великі відстані, дозволяючи пристроям Інтернету речей передавати дані на відстані від декількох кілометрів до десятків кілометрів. Це досягається завдяки використанню низькочастотних діапазонів (рисунок 1.10) [26]. Пристрої, що використовують LoRaWAN, можуть працювати на низькому енергоспоживанні, що робить його придатним для пристроїв Інтернету речей, що живляться від батареї. Протокол розроблений для збільшення часу автономної роботи пристроїв, що робить його практичним для застосування в сільському господарстві, розумних містах та інших галузях.

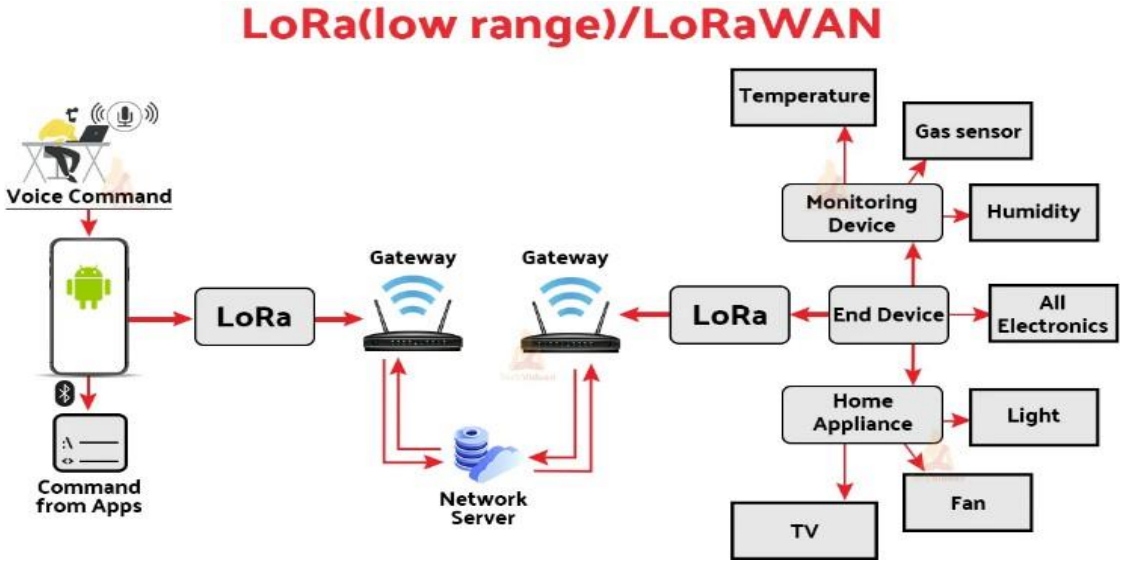


Рисунок 1.10 – LoRa(low range)/LoRaWAN [26]

Вузькосмуговий Інтернет речей (NB-IoT) - це протокол зв'язку для широкосмугових мереж з низьким енергоспоживанням (LPWAN), стандартизований проектом партнерства 3-го покоління (3GPP). Він призначений для забезпечення ефективного зв'язку для великої кількості пристроїв з низькою швидкістю передачі даних, забезпечуючи розширене покриття і підвищену енергоефективність (рисунок 1.11) [27].

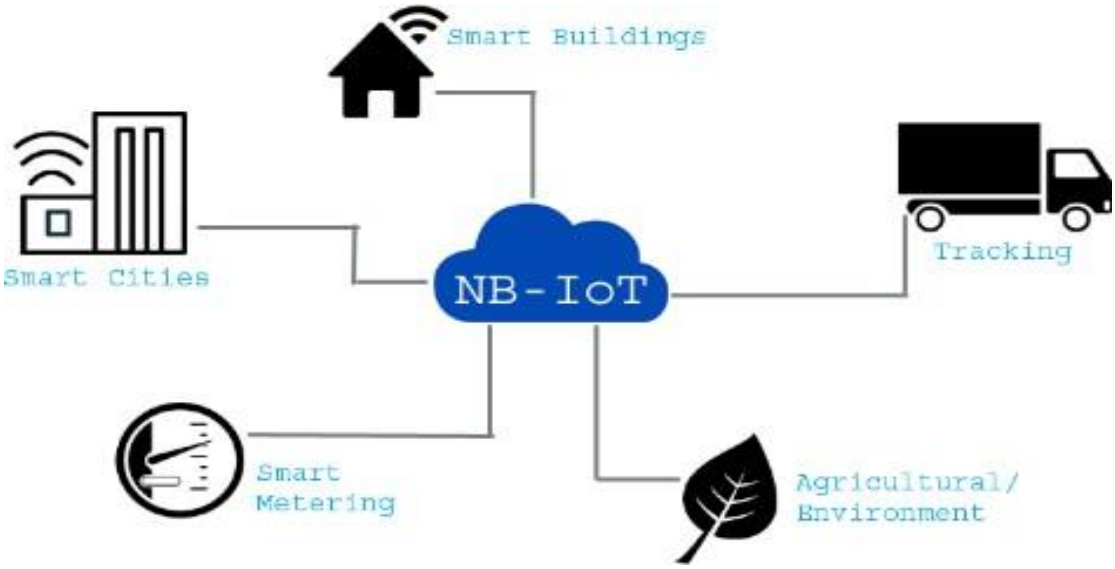


Рисунок 1.11 – Приклади використання NB-IoT [27]

NB-IoT оптимізовано для роботи з низьким енергоспоживанням, що робить його придатним для пристроїв Інтернету речей, які живляться від батареї. Це важливо для застосувань, де пристрої розгорнуті у віддалених або важкодоступних місцях. NB-IoT пропонує розширене покриття, що дозволяє пристроям обмінюватися даними на великих відстанях. Це робить його придатним для додатків, які потребують підключення в складних умовах або в районах з розгалуженою інфраструктурою.

NB-IoT розроблений для підтримки великої кількості пристроїв у мережі. Така масштабованість має вирішальне значення для масового розгортання Інтернету речей у розумних містах, сільському господарстві та промисловості.

NB-IoT оптимізовано для додатків з низькою швидкістю передачі даних, таких як зчитування даних з датчиків, оновлення стану та інші спорадичні передачі. Він не призначений для додатків з високою пропускною здатністю.

1.8 Висновки

У першому розділі було докладно розглянуто ключові поняття, що лежать в основі концепції Інтернету речей та його застосування у створенні розумного міста. Було визначено сутність IoT як технології, яка дозволяє фізичним об’єктам збирати, обмінюватися та обробляти дані без постійного людського втручання, що відкриває нові можливості для автоматизації та оптимізації міської інфраструктури.

Аналіз практичних прикладів реалізації IoT-систем, таких як системи радіаційного моніторингу, пошуку дефектів дорожнього покриття та моніторингу громадського транспорту, дозволив виявити як переваги впровадження даних технологій, так і існуючі проблеми як: надмірне навантаження на мобільні мережі, недостатню стандартизацію та складність обробки великих обсягів даних.

Крім того, було приділено значну увагу протоколам передачі даних, що забезпечують взаємодію між різними пристроями IoT. Розглянуто як дротові, так і бездротові протоколи, їхню роль у забезпеченні надійності, масштабованості та безпеки систем, що є критично важливими для успішної інтеграції IoT у міське середовище.

Таким чином, розділ 1 закладає фундаментальну базу для подальшого дослідження та розробки єдиної міської системи Інтернету речей, спрямованої на покращення управління ресурсами, підвищення якості життя мешканців та забезпечення стійкого розвитку сучасних міст.

2 ФОРМАЛІЗАЦІЯ ВИМОГ ТА КОНЦЕПТУАЛЬНА МОДЕЛЬ МІСЬКОЇ ІОТ-СИСТЕМИ

2.1 Формалізація вимог до моделі міської IoT-системи

Формалізація вимог є критичним етапом розробки моделі міської IoT-системи, оскільки вона забезпечує систематизацію як функціональних, так і нефункціональних характеристик майбутнього рішення. Отримані технічні та операційні вимоги дозволяють створити детальну специфікацію, на основі якої формується інтегроване, масштабоване та безпечне середовище обміну даними між сотнями, а іноді й тисячами пристроїв, що використовуються для управління міською інфраструктурою [28].

На першому етапі аналізу предметної області визначено основні сценарії використання IoT-технологій у контексті «розумних міст». Серед них - моніторинг транспортних потоків, управління ресурсами (енергія, вода, газ), екологічний контроль та забезпечення громадської безпеки. Функціональні вимоги включають можливість збору, передачі і обробки даних у режимі реального часу, інтегрованість різних підсистем та підтримку протоколів комунікації, що забезпечують сумісність пристроїв різних виробників.

Нефункціональні вимоги, які є невід'ємною частиною специфікації, орієнтовані на забезпечення високої масштабованості, надійності, безпеки та енергоефективності IoT-системи. Особливу увагу приділено оптимізації мережевого трафіку та зниженню затримок у передачі інформації, що є критичними для роботи систем, які функціонують на бездротових комунікаційних технологіях .Окрім того, формалізація вимог включає визначення стандартів взаємодії та протоколів, що регламентують обмін даними між різними компонентами системи, а також встановлення критеріїв оцінки якості та ефективності реалізованого рішення. У цьому контексті особливо важливо забезпечити інтегрованість IoT-платформи з іншими технологічними рішеннями, що дозволяє адаптувати систему до змін у технологічному середовищі та вимогах експлуатації [29].

2.1.1 Визначення функціональних вимог до моделі IoT-системи

Під час розробки моделі міської IoT-системи критично важливим є формалізація функціональних вимог, що визначають основні завдання майбутньої платформи. Функціональні вимоги охоплюють опис основних процесів, які повинні бути реалізовані в системі, а саме збір даних з численних сенсорних пристроїв, їх обробка, зберігання й подальше аналізування для отримання релевантної інформації щодо стану міської інфраструктури [30].

Визначення функціональних вимог розпочинається з аналізу можливих сценаріїв використання IoT-технологій в умовах розумного міста. Система повинна забезпечувати своєчасне отримання інформації про інтенсивність руху, що дозволить оптимізувати маршрути та зменшити затори.

Збір даних про споживання води, електроенергії та інших життєво важливих ресурсів дозволяє ефективно планувати їх розподіл і своєчасно виявляти можливі витоки чи аномалії. Функціональні вимоги включають збір інформації про якість повітря, температуру, рівень шуму, що є необхідним для своєчасного реагування на зміни в екологічній ситуації. Система повинна мати можливості для моніторингу підозрілих ситуацій, наприклад, за допомогою відеоспостереження та аналізу даних з різноманітних сенсорів, що сприяють оперативному реагуванню у разі надзвичайних подій .

Особлива увага приділяється забезпеченню сумісності між різними компонентами системи через використання стандартних протоколів обміну даними. Це дозволяє інтегрувати пристрої від різних виробників та гарантувати безперебійну комунікацію між елементами системи. Завдяки цьому можливо досягти високої масштабованості платформи, що є одним з ключових аспектів сучасних IoT-рішень.

2.1.2 Визначення нефункціональних вимог до моделі IoT-системи

Окрім функціональних характеристик, надзвичайно важливою складовою формалізації моделі міської IoT-системи є визначення нефункціональних вимог. Ці вимоги спрямовані на забезпечення якості, стабільності та масштабованості системи, що має критичне значення для експлуатації IoT-платформи у середовищі з високою кількістю підключених пристроїв та великою частотою обміну даними.

Основні категорії нефункціональних вимог [31]:

- надійність та відмовостійкість;
- масштабованість;
- безпека;
- енергоефективність;
- підвищення громадської безпеки;
- сумісність та інтеграційність.

Система повинна забезпечувати безперебійну роботу навіть у разі виникнення збоїв окремих елементів. Це включає механізми відновлення після аварійних ситуацій, дублювання критичних компонентів і використання резервних каналів зв’язку для запобігання переривання обміну даними .

Враховуючи прогнозований ріст кількості підключених пристроїв, платформа повинна бути спроможною легко адаптуватися до збільшення навантаження. Розробка системи з розподіленою архітектурою і використанням гнучких протоколів комунікації дозволить забезпечити належний рівень продуктивності при масштабуванні.

Захист даних і конфіденційність є невід’ємними вимогами до роботи будь-якої IoT-системи. Необхідно впровадити комплекс заходів для запобігання несанкціонованому доступу, витоку або викривленню інформації. До цього належать використання методів шифрування, аутентифікації та регулярний аудит системних ресурсів.

Оскільки багато IoT-пристроїв працюють від батарей, важливо оптимізувати їх споживання енергії. Розробка алгоритмів для регулювання режимів передачі

даних, оптимізація частоти оновлення інформації та застосування енергозберігаючих технологій є необхідними умовами для продовження терміну служби пристроїв без шкоди для якості роботи системи .

Система має бути побудована таким чином, щоб забезпечити інтеграцію з різними протоколами, стандартами та архітектурними рішеннями інших систем. Це дозволяє безшовно включати нові елементи та пристрої до існуючої IoT-інфраструктури, зберігаючи при цьому високий рівень продуктивності та стабільності.

Для кожної з перерахованих категорій нефункціональних вимог необхідно розробити критерії оцінки якості, що дозволять на різних етапах розробки здійснювати моніторинг та коригування параметрів системи. Наприклад, для вимог до надійності розробляються метрики відмовостійкості, а для безпеки - критерії контролю доступу та рівнів шифрування. Такий підхід дозволяє вже на стадії проектування виявити потенційні слабкі місця та своєчасно впровадити заходи з їх оптимізації [32].

2.1.3 Формування концептуальної моделі міської IoT-системи

На основі проведеного аналізу вимог, огляду сучасних підходів та обґрунтування вибору технологій сформовано концептуальну модель міської IoT-системи, що враховує як технологічні, так і організаційні аспекти інтеграції численних пристроїв і датчиків у єдину мережу. Розроблена модель базується на принципах гібридної архітектури, що поєднує локальну обробку даних за допомогою edge-вузлів із централізованою агрегацією та аналітикою у хмарних сервісах [33].

Основними складовими концептуальної моделі є:

- пристрої збору даних і сенсори, що забезпечують первинне отримання інформації з міської інфраструктури;
- мережеві протоколи зв'язку, зокрема CoAP і MQTT, які дозволяють забезпечити ефективний і економний обмін інформацією;

– платформи для зберігання та обробки даних, що використовують методи машинного навчання та алгоритми аналітики для прийняття оперативних управлінських рішень;

– механізми безпеки.

Концептуальна модель враховує особливості динамічного міського середовища, забезпечуючи високу масштабованість, адаптивність до зростаючого навантаження та оптимізацію витрат енергії на кінцевих пристроях. Розроблена модель є інтегрованою системою, що дозволяє ефективно управляти ресурсами міста та підтримувати стабільну роботу навіть у випадках високої кількості підключених IoT-пристроїв.

2.2 Архітектурні рішення для ефективної інтеграції IoT

Сучасні міські IoT-системи є надзвичайно комплексними через різноманіття компонентів, що включають пристрої збору даних, сенсори, шлюзи, обчислювальні вузли та хмарні платформи. Для забезпечення безперебійної та ефективної інтеграції всіх компонентів необхідно впровадження архітектурних рішень, які дозволяють об'єднати фізичні та логічні елементи системи у єдину інтегровану мережу [34].

Одним із ключових підходів є використання багаторівневої (ієрархічної) архітектури, яка передбачає розподіл функцій між пристроями з різним рівнем обчислювальних можливостей. На першому рівні розташовуються кінцеві IoT-пристрої та сенсори, які відповідають за первинний збір даних. Дані передаються до edge-вузлів, що забезпечують локальну обробку інформації та знижують затримки при передачі даних до центральних серверів. Централізована обробка здійснюється у хмарних платформах, що дозволяє виконувати складний аналіз, зберігання даних та забезпечення інтеграції з іншими системами управління міською інфраструктурою.

Іншим важливим елементом є використання стандартних протоколів зв’язку, таких як CoAP та MQTT, що забезпечують ефективний обмін даними між різними

компонентами системи. Завдяки своїм характеристикам, зокрема легковаговості та можливості роботи в режимах асинхронного обміну, ці протоколи сприяють зниженню накладних витрат та забезпечують надійну комунікацію навіть у середовищах із високою кількістю підключених пристроїв.

Крім того, інтеграція IoT-систем у міську інфраструктуру вимагає впровадження механізмів забезпечення безпеки, які включають автентифікацію, шифрування та контроль доступу до ресурсів. Використання DTLS та TLS для захисту даних забезпечує високий рівень конфіденційності та цілісності інформації при її передачі між компонентами системи. Загалом, архітектурні рішення для ефективної інтеграції IoT у міське середовище повинні забезпечувати гнучкість, масштабованість та адаптивність системи до змінних умов експлуатації. Використання багаторівневих моделей, розподіленої обробки даних та сучасних протоколів зв'язку створює умови для побудови стабільної та ефективної інфраструктури розумного міста, що здатна задовольнити як поточні, так і майбутні вимоги управління ресурсами [35].

2.2.1 Інтеграція локальної обробки даних у міську IoT-інфраструктуру

Сучасна міська IoT-інфраструктура характеризується високою кількістю кінцевих пристроїв, що генерують значні обсяги даних. Для ефективної обробки та аналізу цієї інформації необхідно впровадження концепції локальної обробки даних, або edge computing. Цей підхід дозволяє здійснювати попередню обробку, фільтрацію та агрегування даних безпосередньо на рівні мережевих вузлів, що знаходяться поблизу джерел їх генерації[36].

Застосування edge computing сприяє зниженню затримок у передачі інформації до центральних серверів, оскільки лише відфільтровані та агреговані дані надсилаються до хмарних платформ для подальшого аналізу. Такий підхід не лише оптимізує використання мережевих ресурсів, але й дозволяє значно зменшити навантаження на центральні обчислювальні вузли, що є критичним у масштабних міських системах. Крім того, локальна обробка даних забезпечує

підвищення рівня безпеки, оскільки обробка конфіденційної інформації відбувається безпосередньо на місці її збору, що мінімізує ризик її перехоплення під час передачі. Інтеграція edge computing з централізованими хмарними рішеннями дозволяє створити гібридну архітектуру, яка забезпечує оптимальний баланс між оперативною реакцією системи та здатністю до масштабування за рахунок централізованого аналізу великих обсягів даних.

Тому, впровадження локальної обробки даних є ключовим елементом архітектурного підходу до інтеграції IoT у міську інфраструктуру. Цей підхід дозволяє ефективно управляти даними, зменшувати затримки, оптимізувати використання мережевих ресурсів та підвищувати загальну надійність системи, що є особливо важливим в умовах високої динаміки міського середовища

2.2.2 Інтеграція хмарних сервісів для централізованої обробки та аналізу даних

У сучасних IoT-системах, окрім локальної обробки даних на рівні edge-вузлів, особливе значення має централізована обробка та аналіз інформації, що забезпечується через інтеграцію хмарних сервісів. Хмарні платформи дозволяють агрегувати великі обсяги даних, що надходять від численних кінцевих пристроїв, і застосовувати складні алгоритми аналізу для прийняття оперативних управлінських рішень. Застосування хмарних технологій забезпечує наступні переваги:

- висока обчислювальна потужність і масштабованість, що дозволяє обробляти великі обсяги даних у режимі реального часу;
- централізоване зберігання даних, що сприяє їх узгодженому аналізу і збереженню історичних тенденцій;
- можливість інтеграції з аналітичними інструментами та алгоритмами машинного навчання.

При інтеграції з хмарними сервісами необхідно враховувати питання безпеки. Сучасні хмарні рішення використовують протоколи захищеної передачі

даних, такі як HTTPS та DTLS, що дозволяє забезпечити конфіденційність, цілісність і автентифікацію інформації як під час передачі, так і під час зберігання. Використання уніфікованих API сприяє стандартизації обміну даними між локальними компонентами IoT-системи та хмарними платформами, що спрощує інтеграцію та розширення функціональних можливостей системи [37].

Таким чином, інтеграція хмарних сервісів виступає ключовим елементом для побудови масштабованої, гнучкої та ефективної міської IoT-системи, що здатна адаптуватися до зростаючих потреб сучасного розумного міста.

2.2.3 Управління та планування інтегрованої IoT-інфраструктури

Ефективне управління та планування є ключовими складовими для забезпечення стабільної роботи міських IoT-систем. Враховуючи високу динамічність міського середовища та значну кількість пристроїв, необхідно створити механізми, що дозволяють централізовано моніторити стан системи, управляти ресурсами та оперативно реагувати на виникнення збоїв [38].

Централізовані платформи управління, що інтегровані з хмарними сервісами, дозволяють здійснювати збір і аналіз даних з численних IoT-пристроїв, що, у свою чергу, забезпечує прийняття оперативних рішень. Планування роботи системи здійснюється за допомогою стандартизованих API та протоколів взаємодії, що сприяє синхронізації роботи як локальних edge-вузлів, так і центральних обчислювальних ресурсів. Це дозволяє не лише оптимізувати розподіл навантаження, але й забезпечити масштабованість системи у випадку зростання кількості підключених пристроїв.

Ключовим аспектом управління є автоматизований моніторинг мережі, який дозволяє відслідковувати поточний стан пристроїв, виявляти потенційні проблеми та своєчасно проводити коригувальні заходи. Використання алгоритмів машинного навчання для аналізу зібраних даних сприяє підвищенню точності прогнозування та оптимізації роботи системи, що є особливо важливим для розумного міста.

2.2.4 Забезпечення інтероперабельності та сумісності між компонентами IoT-системи

Однією з основних складових успішної інтеграції IoT у міську інфраструктуру є забезпечення інтероперабельності між різними компонентами системи. Сучасні IoT-системи складаються з великої кількості пристроїв, що використовують різні протоколи зв'язку, апаратні платформи та програмне забезпечення. Забезпечення сумісності між ними є критичним для ефективного обміну даними та синхронізації роботи всієї системи [39].

Для досягнення високого рівня інтероперабельності впроваджуються стандартизовані протоколи, такі як CoAP, MQTT, а також протоколи на основі RESTful-архітектури. Стандартизація дозволяє зменшити необхідність розробки індивідуальних адаптерів для різних компонентів і сприяє безшовній інтеграції як локальних, так і централізованих систем обробки даних. Крім того, використання протоколів, що відповідають вимогам міжнародних стандартів, полегшує інтеграцію IoT-систем із існуючими інформаційними платформами та системами управління міською інфраструктурою.

Важливим аспектом забезпечення інтероперабельності є розробка універсальних інтерфейсів обміну даними (API), які дозволяють різним підсистемам взаємодіяти незалежно від використовуваних технологій. Такий підхід дозволяє впровадити єдину систему управління, здатну обробляти запити від різних пристроїв у реальному часі, що сприяє підвищенню оперативності прийняття рішень та ефективності використання міських ресурсів.

Таким чином, інтеграція інтероперабельних рішень є необхідною умовою для побудови масштабованої та гнучкої IoT-системи. Забезпечення сумісності між компонентами системи сприяє не лише оптимізації процесу передачі та обробки даних, але й підвищує загальний рівень безпеки, надійності та адаптивності системи до умов високої динаміки міського середовища.

2.3 Взаємодія IoT-компонентів, алгоритми обміну даними

Сучасні IoT-системи, що охоплюють велику кількість пристроїв із обмеженими ресурсами, вимагають застосування ефективних алгоритмів обміну даними для забезпечення надійної та масштабованої взаємодіїр [40]. Взаємодія IoT-компонентів здійснюється за допомогою ряду протоколів і методів, серед яких особливе місце займають моделі публікації/підписки, запиту/відповіді, механізми блочного передавання, спостереження за ресурсами, групова комунікація та використання RESTful-підходів.

Алгоритми обміну даними за моделлю публікації/підписки, які реалізуються, наприклад, у протоколі MQTT, дозволяють забезпечити асинхронний обмін інформацією, де пристрої надсилають повідомлення до центрального брокера, а зацікавлені клієнти отримують ці повідомлення без необхідності встановлення постійного з'єднання. Це сприяє підвищенню масштабованості та зниженню затримок у передачі даних.

Модель запиту/відповіді, яка використовується в протоколі CoAP, забезпечує точне зіставлення запитів та відповідей за допомогою унікальних токенів. Такий підхід мінімізує ризик втрати інформації та дозволяє ефективно контролювати процес обміну даними, що є важливим для систем, де критичною є оперативна обробка даних. Для передачі великих обсягів інформації застосовується механізм блочного передавання, що дозволяє розбивати дані на менші блоки. Це забезпечує ефективне використання мережевих ресурсів і мінімізує проблеми, пов'язані з фрагментацією пакетів, особливо в умовах нестабільних з'єднань.

Механізм спостереження (Observe) дозволяє клієнтам підписуватися на зміни стану ресурсів, отримуючи оновлення без необхідності повторного надсилання запитів. Цей підхід знижує навантаження на мережу і покращує оперативність отримання критично важливої інформації. Групова комунікація, заснована на використанні IP-мультікасту, сприяє одночасному надсиланню запитів до великої кількості пристроїв, що особливо актуально для застосувань, де необхідно синхронно управляти групами IoT-компонентів.

Крім того, забезпечення безпеки передачі даних здійснюється за допомогою протоколів DTLS та TLS, що гарантує конфіденційність, цілісність та автентифікацію інформації під час обміну між компонентами системи. Тому, комплексне застосування вищезазначених алгоритмів дозволяє досягти високої продуктивності, зниження латентності та оптимізації використання мережевих ресурсів, що є критично важливим для стабільної роботи IoT-систем у динамічних умовах розумного міста

2.3.1 Алгоритм обміну даними за моделлю публікації/підписки

Однією з найефективніших моделей обміну даними в IoT-системах є модель публікації/підписки, яка забезпечує асинхронну взаємодію між компонентами системи. У цій моделі пристрої, що генерують дані (видавці), надсилають повідомлення до центрального брокера, який, у свою чергу, розподіляє ці повідомлення серед зацікавлених пристроїв (підписників), які зареєстровані на відповідні теми чи канали (рисунок 2.1) [41].

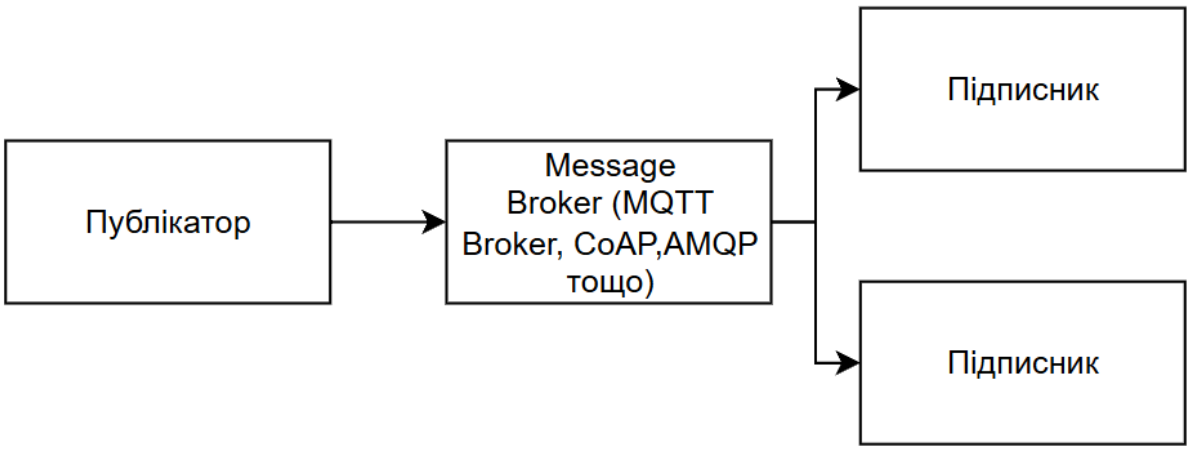


Рисунок 2.1 – Схема обміну даними за моделлю публікації/підписки

Основною перевагою даної моделі є її здатність забезпечувати високу масштабованість та знижувати затримки в обміні інформацією. Завдяки централізованому брокеру, що відповідає за маршрутизацію повідомлень, пристрої

не потребують встановлення прямих зв’язків один з одним, що знижує енергоспоживання та спрощує управління мережею. Такий підхід дозволяє ефективно працювати як в умовах стабільних, так і при високій динаміці мережевого навантаження.

Крім того, модель публікації/підписки сприяє оперативному реагуванню системи на події, що відбуваються в реальному часі, що є надзвичайно важливим для управління критичними міськими ресурсами. Використання цієї моделі дозволяє також централізовано здійснювати контроль за якістю передачі повідомлень та забезпечувати безперебійну роботу системи навіть при великій кількості одночасних підключень.Отже, алгоритм обміну даними за моделлю публікації/підписки є оптимальним рішенням для організації ефективної та масштабованої комунікації в міських IoT-системах, що відповідає вимогам щодо швидкості, енергозбереження та управління потоками даних.

2.3.2 Алгоритм обміну даними за моделлю запиту-відповіді

Запит-відповідь є класичною моделлю взаємодії між IoT-компонентами, яка забезпечує точне зіставлення кожного запиту з відповіддю [42]. У цьому підході клієнт ініціює комунікацію, надсилаючи запит до сервера, який обробляє отриману інформацію та повертає відповідь. Для ідентифікації транзакції застосовується унікальний токен, що додається до заголовка кожного повідомлення, що дозволяє безпосередньо зіставити відповідь із запитом, незалежно від типу транспортного повідомлення.

Коли клієнт надсилає запит у режимі підтверджуваних повідомлень (Confirmable), сервер одразу відправляє відповідь разом із підтвердженням (ACK), або, у випадку, коли відповідь не може бути сформована негайно, надсилає порожнє підтвердження з наступною окремою відповіддю. У режимі непідтверджуваних повідомлень (Non-confirmable) сервер безпосередньо повертає відповідь без механізму підтвердження. Завдяки такій схемі забезпечується мінімізація повторних передач та зниження ймовірності втрати даних, що є

критично важливим для IoT-систем, де відсутність точної інформації може призводити до значних витрат ресурсів (рисунок 2.2).

Реалізація моделі запиту-відповіді базується на стандартах, визначених у специфікаціях протоколів, таких як CoAP. Використання унікальних токенів у поєднанні з методами обробки підтверджуваних та непідтверджуваних повідомлень дозволяє ефективно управляти комунікацією, забезпечуючи високу надійність та зниження латентності у процесі обміну даними. Це особливо важливо у міських IoT-системах, де оперативна обробка інформації сприяє прийняттю рішень в режимі реального часу.

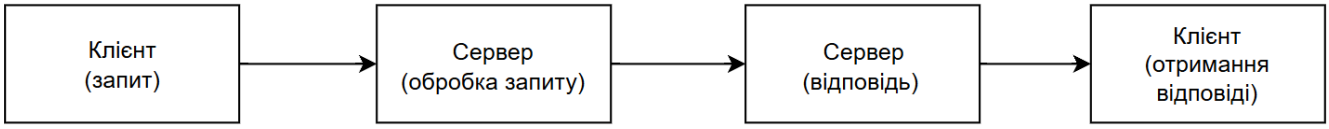


Рисунок 2.2 –Схема обміну даними за моделлю запиту-відповіді

Запровадження моделі запиту-відповіді дозволяє досягти високої точності в обміні інформацією, оскільки кожна операція підтверджується відповіддю, що сприяє виявленню можливих збоїв у комунікації. Однак, цей підхід має свої специфічні виклики. Наприклад, необхідність підтримання сесій і механізмів повторної передачі підтверджуваних повідомлень може збільшувати затримки у випадку поганої якості мережі. Таке явище особливо актуальне для IoT-систем, де мережеві збої або переривання зв'язку можуть призводити до повторних запитів, що в свою чергу збільшує навантаження на мережу та споживання енергії на кінцевих пристроях.

Для подолання цих проблем рекомендується впровадження адаптивних механізмів управління тайм-аутами та ретрансляціями, що дозволяють автоматично регулювати параметри повторної передачі залежно від поточного стану мережі. Крім того, використання оптимізованих алгоритмів генерації токенів і обробки повідомлень сприяє зниженню накладних витрат і забезпечує більш ефективну маршрутизацію інформації між клієнтом і сервером.

2.3.3 Взаємодія IoT-компонентів, алгоритми обміну даними

У системах Інтернету речей (IoT) часто виникає потреба в передачі великих обсягів даних між пристроями, що функціонують у мережах з обмеженими ресурсами. Протокол CoAP (Constrained Application Protocol) пропонує механізм блочного передавання, який дозволяє розділити великі повідомлення на менші блоки, що передаються послідовно. Цей підхід сприяє уникненню проблем, пов'язаних з фрагментацією на рівні IP, та забезпечує ефективну передачу даних у мережах з низькою пропускною здатністю [43].

Механізм блочного передавання в CoAP реалізується за допомогою двох опцій Block1 та Block2. Опція Block1 використовується при передачі великих запитів від клієнта до сервера, тоді як Block2 застосовується для передачі великих відповідей від сервера до клієнта (рисунок 2.3).

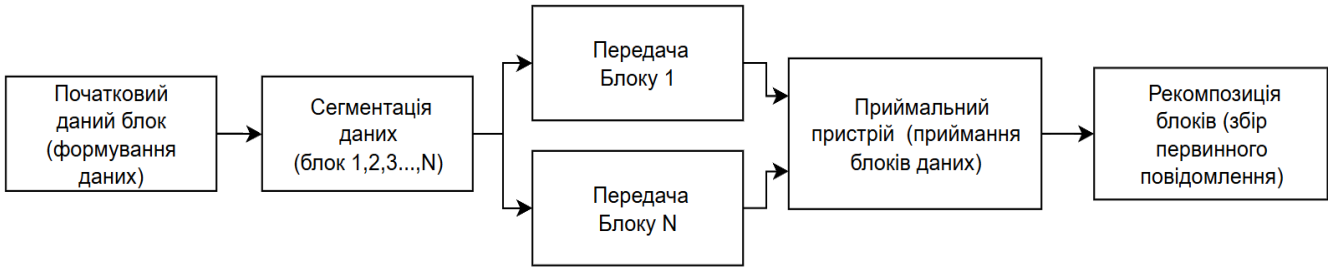


Рисунок 2.3 – Схема обміну даними за допомогою блочного передавання

Кожен блок має свій унікальний індекс та розмір, що дозволяє обом сторонам контролювати процес передачі та повторно запитувати втрачені блоки у разі необхідності. Використання блочного передавання в CoAP має кілька переваг:

– кожен блок підтверджується окремо, що дозволяє виявляти та повторно передавати лише втрачені блоки, зменшуючи накладні витрати;

– клієнт і сервер можуть узгоджувати розмір блоків відповідно до можливостей мережі та пристроїв, що забезпечує оптимальне використання ресурсів;

– передача даних блоками полегшує аналіз трафіку та діагностику проблем за допомогою інструментів моніторингу мережі.

Однак, слід враховувати, що блочне передавання може збільшувати загальний час передачі через необхідність обробки кожного блоку окремо. Тому важливо ретельно налаштовувати параметри блочного передавання, враховуючи специфіку застосування та обмеження мережі. Для подальшого вдосконалення механізму блочного передавання в протоколі CoAP було розроблено розширення, відоме як Q-Block1 та Q-Block2 опції. Ці опції дозволяють здійснювати блочну передачу з використанням непідтверджуваних (Non-confirmable) повідомлень, що сприяє зменшенню затримок та підвищенню ефективності передачі даних у мережах з високою асиметрією втрат пакетів.

Основні переваги використання Q-Block опцій включають можливість швидшої передачі наборів блоків даних з меншою кількістю обмінів пакетами та покращене відновлення втрачених блоків. Однак, застосування цих опцій також має певні недоліки, зокрема втрату послідовності (lock-stepping), що може призвести до отримання блоків не в правильному порядку, а також необхідність впровадження додаткових заходів контролю перевантаження для непідтверджуваних повідомлень.

Впровадження Q-Block опцій у CoAP надає можливість оптимізувати передачу великих обсягів даних у обмежених мережах, забезпечуючи баланс між швидкістю передачі та надійністю доставки. Проте, для досягнення оптимальної продуктивності необхідно ретельно налаштовувати параметри передачі та враховувати специфіку мережевого середовища, в якому функціонує система.

2.4 Безпека та захист інформації в міських IoT-системах

У сучасних міських IoT-системах безпека є однією з найважливіших складових, оскільки зростаюча кількість пристроїв, підключених до мережі, створює численні загрози для конфіденційності, цілісності та доступності інформації. Забезпечення захисту даних охоплює як технічні, так і організаційні

аспекти, що мають бути інтегровані на всіх рівнях системи - від кінцевих пристроїв до хмарних сервісів управління [44].

Основним елементом захисту є впровадження протоколів безпечної передачі даних. У випадку IoT-пристроїв, що використовують протокол CoAP, застосовується DTLS (Datagram Transport Layer Security), який забезпечує шифрування та автентифікацію повідомлень при передачі даних через UDP. Для пристроїв, що працюють з MQTT, використовуються TLS/SSL-з'єднання, що гарантують захищеність комунікації через TCP. Завдяки цим рішенням забезпечується конфіденційність та цілісність інформації навіть у випадку перехоплення даних.

Ключовим аспектом безпеки є також управління ключами та механізми автентифікації. Використання попередньо узгоджених ключів, сирих відкритих ключів або сертифікатів дозволяє забезпечити перевірку достовірності пристроїв та захист від несанкціонованого доступу. Системи управління ключами повинні бути інтегровані як на рівні edge-вузлів, так і в хмарних платформах для централізованого контролю безпеки.Окрім цього, важливим напрямком є застосування додаткових методів захисту, таких як використання технологій блокчейн для забезпечення незмінності даних та розподіленої автентифікації, а також впровадження алгоритмів машинного навчання для раннього виявлення аномалій у мережевому трафіку. Ці підходи дозволяють у режимі реального часу ідентифікувати потенційні загрози та запобігати кібератакам, що є особливо актуальним для міських IoT-систем із високою щільністю пристроїв.

Комплексне забезпечення безпеки в IoT-системах охоплює впровадження захищених протоколів передачі даних, систем управління ключами, а також інтеграцію механізмів моніторингу та аналізу аномалій. Це дозволяє створити надійну інфраструктуру, здатну протистояти сучасним кіберзагрозам, забезпечуючи високий рівень захисту інформації в умовах високої динаміки міського середовища.

2.4.1 Основні загрози безпеці в міських IoT-системах

Міські системи Інтернету речей (IoT) [45] стикаються з низкою загроз безпеці, які можуть суттєво впливати на їхню функціональність та надійність. Серед основних загроз можна виділити:

- несанкціонований доступ та атаки на автентифікацію;
- перехоплення та модифікація даних;
- DDoS-атаки;
- використання вразливостей програмного забезпечення;
- недостатній контроль доступу та управління правами.

Відсутність надійних механізмів автентифікації може призвести до несанкціонованого доступу до пристроїв та даних. Наприклад, використання паролів за замовчуванням або слабких паролів робить систему вразливою до атак. Рекомендується замінювати стандартні паролі та використовувати складні комбінації для підвищення рівня безпеки.

Передача незашифрованих даних через мережу створює ризик їхнього перехоплення та несанкціонованої модифікації. Використання протоколів шифрування, таких як TLS/SSL, допомагає забезпечити конфіденційність та цілісність переданої інформації.

Зловмисники можуть використовувати вразливі IoT-пристрої для здійснення DDoS-атаки на міську інфраструктуру, що призводить до перевантаження систем та відмови в обслуговуванні. Впровадження механізмів виявлення аномальної активності та фільтрації трафіку є ключовими заходами для запобігання таким атакам.

Наявність невиправлених вразливостей у програмному забезпеченні IoT-пристроїв може бути використана для отримання несанкціонованого доступу або виконання шкідливого коду. Регулярне оновлення та патчинг програмного забезпечення є необхідними для підтримки безпеки системи.

Неправильне налаштування прав доступу може дозволити зловмисникам отримати контроль над критичними компонентами системи. Впровадження чіткої

політики управління доступом та регулярний аудит прав користувачів сприяють зниженню цього ризику.

2.4.2 Заходи щодо захисту інформації в міських IoT-системах

Для забезпечення високого рівня захисту інформації в міських IoT-системах необхідно впровадження комплексного підходу, що охоплює як технічні, так і організаційні заходи. Основними елементами такої стратегії є шифрування даних, механізми автентифікації та управління ключами, а також системи моніторингу та реагування на інциденти безпеки [46].

Одним із ключових заходів є використання протоколів шифрування, зокрема DTLS для CoAP та TLS/SSL для MQTT. Ці протоколи дозволяють забезпечити конфіденційність та цілісність інформації під час передачі даних через мережу, що є критично важливим для систем, де дані передаються в умовах відкритого доступу та високої динаміки міського середовища. Крім того, впровадження динамічних систем управління ключами сприяє оперативному оновленню криптографічних параметрів, що мінімізує ризик компрометації безпекових даних.

Також важливим аспектом є впровадження багаторівневих систем автентифікації, які дозволяють перевіряти достовірність як кінцевих пристроїв, так і користувачів. Це досягається за допомогою використання попередньо узгоджених ключів, сирих відкритих ключів або сертифікатів X.509, що забезпечують високий рівень захисту при мінімальному навантаженні на обчислювальні ресурси пристроїв.

Крім технічних рішень, важливим елементом стратегії безпеки є організаційні заходи, що включають розробку політик управління доступом, регулярний аудит безпеки, навчання персоналу та впровадження систем моніторингу мережевого трафіку. Такі системи дозволяють в режимі реального часу виявляти аномальні патерни та оперативно реагувати на потенційні загрози, що сприяє підвищенню загальної стійкості системи до кібератак.

2.4.3 Оцінка впливу заходів безпеки на продуктивність IoT-системи та перспективи їх оптимізації

Інтеграція заходів безпеки в IoT-систему є необхідною для забезпечення захисту даних, проте одночасно впливає на загальну продуктивність мережі. На даному етапі дослідження проведено аналіз компромісу між рівнем захисту інформації та витратами ресурсів, що супроводжують впровадження механізмів шифрування, автентифікації та управління ключами [47].

Впровадження DTLS та TLS/SSL дозволяє забезпечити високий рівень конфіденційності та цілісності даних, проте ці технології потребують додаткових обчислювальних потужностей і збільшують затримки в обміні інформацією, що може вплинути на оперативність роботи системи. Особливо це відчутно в умовах високої навантаженості мережі або при роботі пристроїв з обмеженими ресурсами.

Комплексна оцінка показує, що для досягнення оптимального балансу між безпекою та продуктивністю необхідно впровадження адаптивних механізмів керування криптографічними процесами, що дозволяють регулювати рівень шифрування залежно від поточних умов експлуатації. Подальші дослідження повинні зосередитися на розробці алгоритмів оптимізації, які мінімізують додаткове навантаження без шкоди для рівня захисту інформації. Отримані результати створюють основу для подальшої інтеграції та вдосконалення заходів безпеки у масштабних IoT-системах міського середовища, забезпечуючи їх стабільну та надійну роботу.

2.5 Розробка структурної моделі системи збору даних

Розробка структурної моделі системи збору даних є базовою передумовою побудови інтегрованої IoT-платформи для міських систем, що забезпечує ефективне та безперебійне отримання інформації із численних сенсорних пристроїв. Вона дозволяє створити чітку картину взаємодії між компонентами

системи, що є критично важливим для подальшої оптимізації процесів обробки та аналізу даних.

2.5.1 Основні компоненти моделі

На першому рівні структурної моделі IoT-системи, що забезпечує моніторинг міської інфраструктури, розташовано фізичні сенсори та пристрої збору даних. Ці компоненти відіграють ключову роль у первинному зборі інформації з навколишнього середовища і забезпечують основу для подальшої обробки даних на рівні edge-комп'ютингу та хмарної інфраструктури [48].

Для досягнення високої точності моніторингу та оперативності реагування обрано наступні типи фізичних сенсорів:

- датчики температури та вологості (DHT22);
- датчики якості повітря (PMS5003, MQ135);
- інфрачервоні сенсори руху та фотодатчики (HC-SR312);
- акустичні сенсори шуму (CBD-0291).

Системі необхідно контролювати мікрокліматичні умови в різних частинах міста. Датчик DHT22 забезпечує вимірювання як температури, так і відносної вологості з високою точністю та мінімальним енергоспоживанням .

Для оцінки стану атмосферного середовища використовуються спеціалізовані сенсори, що вимірюють концентрацію пилу (PM2.5, PM10) та шкідливих газів (CO, NO_2, O_3). Модуль PMS5003 для вимірювання частинок забруднення повітря та MQ135 для виявлення концентрації летких органічних сполук і шкідливих газів .

Для аналізу транспортних потоків і руху пішоходів у громадських місцях застосовуються інфрачервоні (PIR) сенсори, що реагують на рух, а також фотодатчики, що дозволяють визначити рівень освітленості в різний час доби. Ці параметри мають важливе значення для оптимізації енергоспоживання та планування роботи системи громадського освітлення .

Вимірювання рівня шуму за допомогою акустичних сенсорів дозволяє оцінити стан звукового середовища в місті, що особливо актуально для зон з високою щільністю населення та транспортної інфраструктури.

На базі фізичних сенсорів організовується рівень збору даних за допомогою спеціалізованих пристроїв, які включають:

- мікроконтролери (ESP32, Arduino);
- комунікаційні модулі (Wi-Fi, LoRa, NB-IoT).

Для первинної обробки сигналів від сенсорів використовуються мікроконтролери, такі як ESP32 або Arduino, що забезпечують локальну обробку, фільтрацію та агрегацію даних перед їх передачею на наступний рівень системи. Вибір цих платформ обумовлений їхньою енергоефективністю, високою швидкодією та широкою сумісністю з різними сенсорами .

Для забезпечення надійної передачі даних в умовах міського середовища інтегруються модулі бездротового зв’язку, зокрема технології Wi-Fi, LoRa або NB-IoT. Вибір конкретного протоколу залежить від вимог до дальності зв’язку, пропускної здатності та енергозбереження системи.

2.5.2 Логічна структура та моделювання зв’язків

Логічна структура системи збору даних є концептуальною моделлю, що відображає взаємозв’язки між окремими компонентами IoT-платформи на всіх рівнях - від первинного збору інформації на рівні сенсорів до централізованої обробки в хмарних середовищах [49]. Формування логічної структури є невід’ємною складовою розробки систем, що забезпечують високий рівень інтеграції, масштабованості та ефективності обміну даними.

Логічну структуру системи можна розглядати як ієрархічне подання, що складається з трьох основних рівнів:

- рівень датчиків і пристроїв;
- перефирійний рівень (Edge/Fog computing);
- хмарний рівень.

На рівні датчиків і пристроїв здійснюється первинний збір даних із фізичних сенсорів, фіксуються параметри довкілля або процесів, а вбудовані мікроконтролери виконують початкову обробку сигналів. Попередня агрегація та фільтрація даних прямо в пристроях дозволяють зменшити обсяг «сирих» даних і відкидати шум чи непотрібні показники ще до передачі вгору по системі.

Периферійний рівень відповідає за локальне обчислення, фільтрацію та консолідацію даних, що надходять від розподілених сенсорів. Використання edge-технологій дозволяє значно знизити обсяг трафіку, що передається до центральних систем, а також забезпечує та забезпечює реакцію в реальному часі без затримок, притаманних передачі в хмару.

У хмарному рівні відбувається централізований збір, довгострокове зберігання та комплексна обробка даних. Хмарні сервіси забезпечують виконання аналітичних алгоритмів, візуалізацію результатів та підтримку прийняття управлінських рішень, що базуються на отриманих даних.

При моделюванні логічної структури (рисунок 2.4) особливу увагу приділяють визначенню зв’язків між рівнями системи та забезпеченню безперебійного обміну інформацією.

Для цього використовуються стандартні протоколи передачі даних, такі як MQTT, CoAP, HTTP/HTTPS, що дозволяють забезпечити сумісність пристроїв різних виробників та гарантовано передавати дані в умовах високої навантаженості мережі [50].

Зв’язки між компонентами моделюються з урахуванням наступних аспектів:

- передача даних від сенсорів до edge-вузлів;
- агрегація даних на рівні edge;
- централізований аналіз у хмарі.

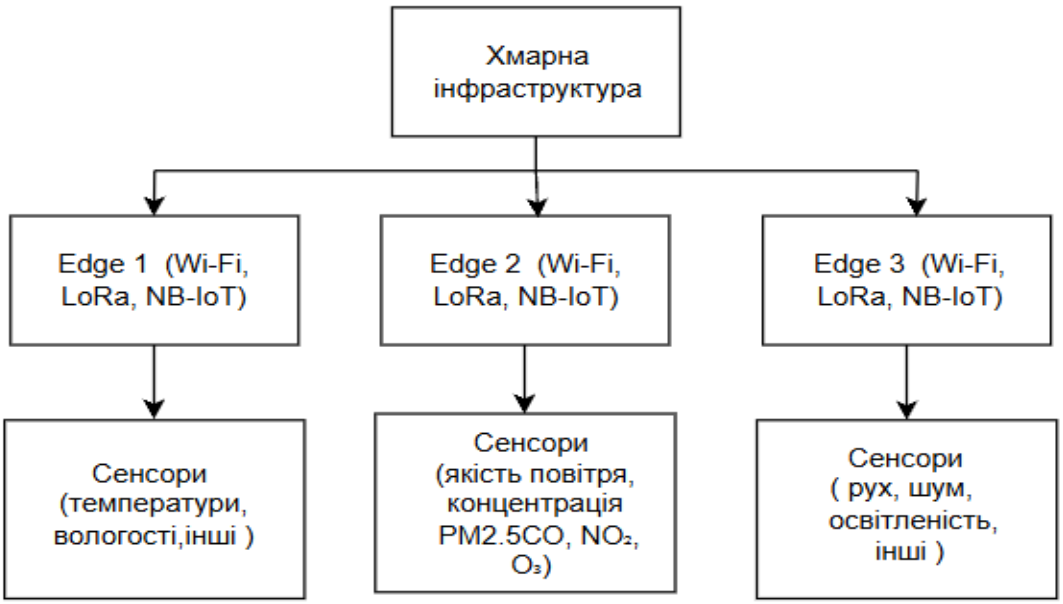


Рисунок 2.4 – Схема ієрархічної побудови системи

2.6 Висновки

Другий розділ присвячено розробці моделей для вирішення поставленої задачі інтеграції IoT-систем у міську інфраструктуру. Проведено детальний аналіз вимог до системи, розглянуто сучасні підходи моделювання, а також обґрунтовано вибір технологій і протоколів, що забезпечують ефективну взаємодію між компонентами IoT. Зокрема, було встановлено, що оптимальним є застосування гібридної архітектури, яка поєднує переваги локальної обробки даних (edge computing) та централізованої агрегації і аналізу у хмарних платформах. Такий підхід дозволяє значно знизити затримки, оптимізувати використання мережевих ресурсів і забезпечити масштабованість системи при зростанні кількості IoT-пристроїв.

Окремо розглянуто різні моделі обміну даними, зокрема алгоритми за моделлю публікації/підписки та запиту-відповіді, які забезпечують як асинхронну, так і синхронну взаємодію між пристроями. Застосування механізмів блочного передавання дозволяє ефективно працювати з великими обсягами даних, а впровадження опції спостереження сприяє зниженню навантаження на мережу

завдяки автоматичному оновленню інформації. Значну увагу приділено питанням безпеки, оскільки забезпечення конфіденційності, цілісності та автентифікації даних є критично важливим для міських IoT-систем. Впровадження протоколів DTLS та TLS, а також розробка механізмів управління ключами і систем моніторингу безпеки дозволяють створити надійну інфраструктуру для захисту інформації.

Результати розділу не лише демонструють ефективність обраних підходів до моделювання міської IoT-системи, а й підтверджують необхідність їх постійного вдосконалення та адаптації до умов реального експлуатаційного середовища. Додатково, аналіз показав, що інтеграція локальної обробки даних з централізованими хмарними сервісами створює основу для високої масштабованості та оперативного реагування на змінні потреби міста. Розроблена модель дозволяє забезпечити ефективну взаємодію між IoT-компонентами, оптимізувати використання мережевих ресурсів та мінімізувати енергоспоживання пристроїв.

Подальші дослідження повинні бути спрямовані на розробку адаптивних алгоритмів регулювання параметрів мережі, впровадження сучасних методів машинного навчання для прогнозування мережевих затримок і оптимізації процесів обробки даних, а також на розширення функціональних можливостей системи шляхом інтеграції з іншими міськими сервісами, такими як транспорт, енергетика та управління комунальними послугами.

Розроблена модель демонструє потенціал для застосування в умовах міста, забезпечуючи ефективну взаємодію між різноманітними IoT-компонентами, оптимізацію обчислювальних ресурсів та високий рівень захисту даних. Отримані результати становлять міцну основу для подальших досліджень і впровадження комплексних IoT-систем, що відповідають сучасним вимогам до управління міською інфраструктурою.

3 РЕАЛІЗАЦІЯ АРХІТЕКТУРИ ТА РОЗРОБКА ПРОГРАМНИХ КОМПОНЕНТІВ ПЛАТФОРМИ

3.1 Підходи до збору даних

Сучасні IoT-системи характеризуються високою гетерогенністю пристроїв та середовищ, де дані генеруються численними сенсорами різних типів. В умовах міських інфраструктур, де важлива оперативність та точність збору інформації, застосовуються різні методики та підходи до збору даних. Основна мета цього етапу полягає в забезпеченні безперебійного та надійного отримання даних, що служать первинним джерелом інформації для подальшої агрегації, аналізу та прийняття управлінських рішень.

3.1.1 Фізичний рівень збору даних

На фізичному рівні збору даних забезпечується безпосередній зв’язок із навколишнім середовищем за допомогою інтелектуальних сенсорів і пристроїв, які виконують функції вимірювання, моніторингу та первинної обробки інформації. Цей рівень є фундаментальним для побудови IoT-системи, оскільки саме від нього залежить точність і своєчасність отримання вихідних даних, які в подальшому використовуються для прийняття оперативних управлінських рішень .

Фізичний рівень включає низку різнопланових сенсорів, серед яких можна виділити:

- датчики температури та вологості;
- датчики якості повітря;
- датчики руху та освітленості;
- акустичні та вібраційні сенсори.

Датчики температури та вологості, пристрої, такі як DHT22 або SHT31, забезпечують вимірювання мікрокліматичних параметрів, що є критично важливими для аналізу стану навколишнього середовища в міських умовах [50]. Вони дозволяють не тільки контролювати температурні коливання, але й

оцінювати відносну вологість, що особливо актуально для моніторингу будівельних і промислових зон.

Датчики якості повітря, такі як PMS5003 та MQ135, використовуються для вимірювання концентрації пилових частинок (PM2.5, PM10) та рівня шкідливих газів (CO, NO_2, O_3). Ці показники сприяють оцінці екологічної ситуації та можуть стати основою для розробки систем раннього попередження про зміни в якості повітря [51].

Датчики руху та освітленості, інфрачервоні сенсори руху (PIR) використовуються для виявлення присутності та руху пішоходів чи транспортних засобів, а фотодатчики - для вимірювання рівня освітленості. Ці дані мають значення для управління системами безпеки та оптимізації використання енергоресурсів у міських умовах [52].

Акустичні та вібраційні сенсори, для моніторингу рівня шуму та вібрацій, що можуть свідчити про технічний стан інфраструктури, використовуються спеціалізовані датчики, які надають можливість проводити своєчасну діагностику та профілактику несправностей [53].

Апаратні платформи та технології збору

Сенсори інтегруються у відповідні апаратні платформи, що зазвичай базуються на мікроконтролерах типу ESP32 або Arduino. Такі платформи дозволяють здійснювати первинну обробку даних, включаючи фільтрацію, агрегування та нормалізацію сигналів, перш ніж дані надсилаються на наступні рівні системи, де відбувається більш глибока обробка та аналіз. Крім того, вони підтримують різноманітні протоколи зв’язку (Wi-Fi, LoRa, NB-IoT), що забезпечують оптимальну передачу даних у режимі реального часу [54].

Фізичний рівень виступає точкою входу для збирання даних, які в подальшому використовуються для побудови детальних моделей міського середовища. Ефективність роботи цього рівня безпосередньо впливає на якість даних, що згодом агрегуються на рівні edge та обробляються у хмарній інфраструктурі. Отже, ретельний вибір сенсорів та апаратних платформ є

першорядним завданням при розробці IoT-систем, що орієнтовані на забезпечення високої оперативності та надійності роботи.

3.1.2 Комунікаційні моделі збору даних

Методи передачі даних від сенсорів до вузлів обробки базуються на використанні як активних, так і пасивних моделей збору інформації. До активних методів можна віднести такі підходи, як періодичне опитування (polling) або запит-відповідь (request-response), коли кожен сенсор у визначені часові інтервали передає заздалегідь сформовані дані. Пасивні моделі, з іншого боку, базуються на механізмах публікації/підписки, які дозволяють сенсорам сповіщати систему про виникнення нових даних у режимі реального часу [55].

3.1.3 Технології передачі даних

Вибір технологій передачі даних безпосередньо залежить від умов середовища та типу використовуваних сенсорів. Для міських IoT-систем активно використовуються такі технології:

- Wi-Fi, для забезпечення високої пропускної здатності;
- LoRa та LoRaWAN, мають низьке енергоспоживання при великій дальності зв’язку, що актуально для систем, що охоплюють великі території, хоча швидкість передачі даних в цих технологіях є нижчою [56];
- NB-IoT, для забезпечення стабільної передачі даних навіть при обмежених ресурсах;
- Bluetooth Low Energy (BLE), використовується для збору даних із пристроїв, що знаходяться в безпосередній близькості, для економія енергії;

Вибір конкретної технології здійснюється з урахуванням специфічних вимог до енергоефективності, дальності передачі та швидкодії мережі, що забезпечує оптимальну передачу даних між сенсорами та вузлами обробки [57].

3.1.4 Забезпечення масштабованості та оперативності збору даних

Однією з ключових задач є забезпечення масштабованості IoT-систем при збільшенні кількості підключених пристроїв. Використання адаптивних методів збору, що поєднують активне опитування та механізми публікації/підписки, дозволяє досягти оптимального балансу між оперативністю передачі даних та економією мережевих ресурсів. Сучасні IoT-системи також інтегрують edge computing-технології, де локальна обробка даних дозволяє зменшити обсяг інформації, що передається до центральних серверів, і забезпечити оперативну реакцію на події в режимі реального часу [58].

3.2 Методи агрегації даних

Агрегація даних у системах Інтернету речей є ключовим етапом обробки інформації, оскільки дозволяє зменшити обсяг переданої інформації, підвищити її якість і забезпечити оптимальне використання мережевих ресурсів. У цьому розділі розглядаються основні методи агрегації даних, які використовуються на локальному (edge) рівні, а також можливості їх подальшої інтеграції на центральному, хмарному рівні.

3.2.1 Основні принципи агрегації даних

Агрегація даних у системах IoT виступає невід'ємним етапом попередньої обробки інформації, що надходить із численних сенсорів. Основна мета агрегації полягає в узагальненні первинних вимірювань з метою зменшення обсягу переданої інформації, покращення її якості та підвищення точності подальшого аналізу. Класичним методом агрегаці [59]ї є обчислення середнього значення отриманих вимірювань, що дозволяє представити репрезентативне значення для групи даних. Для набору вимірювань $D = \{d_1, d_2, \ldots, d_n\}$ середнє значення визначається за формулою [59]:

$$\bar{d} = \frac{1}{n}\sum_{i=1}^{n} d_i \quad (3.1)$$

де:

- d_i- значення вимірюваної величини в експерименті i;
- n - загальна кількість спостережень;
- d - середнє значення величини за всіма спостереженнями.

Це дозволяє знизити вплив випадкових коливань і шуму, присутніх у вихідних даних, і створити компактне представлення, яке є основою для подальшої обробки.

При агрегації даних важливо здійснювати попередню фільтрацію, що дозволяє усунути некоректні, пошкоджені або надмірно відхилених значень (outliers). Наприклад, за допомогою медіанного згладжування можна усунути вплив екстремальних значень, оскільки медіана визначається як центральне значення впорядкованого набору даних і менш чутлива до змін, ніж середнє [59]:

$$\tilde{d} = \text{median}(D) \quad (3.2)$$

де:

- D - множина всіх виміряних значень;
- median () - функція, що повертає медіану множини, проміжне значення в упорядкованому за зростанням списку;
- $\tilde{d}$- медіана набору D.

Що дозволяє підвищити стабільність агрегаційних показників .

Дані, що збираються в IoT-системах, часто мають часову компоненту. Для оптимізації обробки цих даних застосовуються методи часової агрегації, за якими дані групуються в часові вікна. Наприклад, якщо дані надходять протягом періоду T і розбиті на k рівних інтервалів, для кожного інтервалу можна обчислити середнє значення [59]:

$$\bar{d}_j = \frac{1}{n_j}\sum_{i\in W_j} d_i\,, j = 1,2,\ldots,k \quad (3.3)$$

де:

- W_j - підмножина індексів спостережень, що належать групі (вікну) j;
- n_j=W_j- кількість спостережень у групі j;
- d_i- значення вимірюваної величини для спостереження iii;
- $\bar{d}_j$ - середнє значення величини в групі j;
- k - загальна кількість груп (вікон).

Для систем, розташованих на великій території (наприклад, в місті), дані з сенсорів, що знаходяться в межах певного географічного регіону, можуть бути об'єднані для отримання просторово узагальненого показника. Якщо d_j - дані з j-го регіону, загальне агреговане значення може бути визначене як зважене середнє [60]:

$$\bar{d}_{region} = \frac{\sum_{j=1}^{m} w_j * \bar{d}_j}{\sum_{j=1}^{m} w_j} \quad (3.4)$$

де:

- $\bar{d}_j$ - середнє значення в підобласті (групі) j (див. (3.3));
- w_j - вага (важливість, площа, число вузлів тощо) підобласті j;
- m - кількість підобластей у регіоні;
- $\bar{d}_{region}$ - зважене середнє значення величини по всьому регіону.

Цей підхід дозволяє реалізувати принцип Data Fusion, що покращує точність аналізу за рахунок інтеграції інформації з різних джерел.

У сучасних IoT-системах, де середовище може швидко змінюватися, важливо забезпечити адаптивну агрегацію даних, яка автоматично регулюється відповідно до змін у параметрах мережі та характеристиках даних. Наприклад, можна визначати адаптивні вагові коефіцієнти $w_i(t)$, що залежать від часу, і використовувати їх у формулі зваженого середнього [61]:

$$\bar{d}(t) = \frac{\sum_{i=1}^{n} w_i(t) * d_i(t)}{\sum_{i=1}^{m} w_i(t)} \quad (3.5)$$

де:

- $d_i(t)$ - значення вимірюваної величини у момент часу t для джерела i;
- $w_i(t)$- вага (важливість, довіра) вимірювання i у момент часу t
- n - загальна кількість вимірювань у чисельнику;
- m - кількість вагових коефіцієнтів у знаменнику;
- $\bar{d}(t)$ - зважене середнє значення величини у момент часу t.

Такий підхід дозволяє оптимізувати процес агрегації на різних етапах та при різних умовах експлуатації IoT-систем .

3.2.2 Алгоритми статистичної агрегації

Статистична агрегація даних є фундаментальним підходом до обробки інформації в системах Інтернету речей (IoT), що дозволяє зменшити обсяг переданих даних, знизити енергоспоживання та підвищити точність аналізу. У цьому розділі розглядаються основні алгоритми статистичної агрегації, які застосовуються в IoT-середовищах.

Обчислення середнього арифметичногозначення є базовим методом агрегації, що дозволяє зменшити вплив випадкових коливань у даних. Для набору вимірювань $D = \{d_1, d_2, \dots, d_n\}$ середнє значення визначається за формулою [62]:

$$\bar{d} = \frac{1}{n} \sum_{i=1}^{n} d_i \quad (3.6)$$

де:

- d_i - значення вимірюваної величини в спостереженні iii;
- n - кількість спостережень;
- $\bar{d}_i$ - просте (аритметичне) середнє величини.

Медіанне згладжування використовується для зменшення впливу аномальних значень або викидів у даних. Медіана визначається як центральне значення впорядкованого набору даних.

Цей метод є стійким до екстремальних значень і забезпечує більш надійне представлення центральної тенденції в даних.

У випадках, коли різні вимірювання мають різну важливість або надійність, застосовується зважене середнє. Для набору вимірювань $D = \{d_1, d_2, \dots, d_n\}$ з відповідними вагами $W = \{w_1, w_2, \dots, w_n\}$ зважене середнє обчислюється за формулою [63]:

$$\bar{d}_w = \frac{\sum_{i=1}^{n} w_i * d_i}{\sum_{i=1}^{n} w_i} \quad (3.7)\ [63]$$

де:

- d_i - значення величини в спостереженні i;
- w_i - вага (важливість) спостереження i;
- n - загальна кількість спостережень;
- $\bar{d}_w$ - зважене середнє величини по всім спостереженням.

Це дозволяє враховувати різну значущість окремих вимірювань при агрегації.

Для оцінки варіативності даних використовуються дисперсія та стандартне відхилення. Дисперсія визначається як середнє квадратичне відхилення від середнього значення [64]:

$$\sigma^2 = \frac{1}{n}\sum_{i=1}^{n}(d_i - \overline{d})^2 \quad (3.8)$$

де:

- d_i - значення величини в спостереженні i;
- $\bar{d}$ - середнє значення по формулі;
- n - кількість спостережень;

– σ^2- дисперсія , що характеризує розкид значень навколо середнього.

Коефіцієнт варіації використовується для порівняння варіативності між різними наборами даних та визначається як відношення стандартного відхилення до середнього значення [65]:

$$CV = \frac{\sigma}{\bar{d}} * 100\% \quad (3.9)$$

де:

– σ - стандартне відхилення;

– $\bar{d}$ - середнє значення величини;

– CV - коефіцієнт варіації, що характеризує відносний розкид даних.

Цей показник є безрозмірним і дозволяє оцінити відносну варіативність даних.

3.2.3 Порівняльний аналіз ефективності методів агрегації

У системах Інтернету речей (IoT) агрегація даних є ключовим процесом, що дозволяє зменшити обсяг переданої інформації, знизити енергоспоживання та підвищити ефективність обробки даних. Існує кілька основних підходів до агрегації даних, кожен з яких має свої переваги та обмеження [66]. У цьому розділі проведено порівняльний аналіз найбільш поширених методів агрегації централізованих, кластерних та деревоподібних.(таблиця 3.1)

Централізовані підходи передбачають збір усіх даних у центральному вузлі (наприклад, базовій станції), де відбувається їх обробка. Цей метод забезпечує високу точність агрегації, оскільки всі дані доступні в одному місці [67]. Однак, він має суттєві недоліки:

- високе енергоспоживання;
- низька масштабованість;
- вразливість до відмов.

Таблиця 3.1 – Порівняння методів агрегації

Параметр	Централізований метод агрегації	Кластерний медот агрегації	Деревоподібний метод агрегації
Енергоспоживання	Високе	Середнє	Низьке
Масштабованість	Низька	Висока	Висока
Стійкість до відмов	Низька	Середня	Середня
Складність реалізації	Низька	Середня	Висока
Затримка передачі	Висока	Середня	Низька

Передача всіх даних до центрального вузла вимагає значних енергетичних витрат, особливо в мережах з великою кількістю вузлів. Зі збільшенням кількості пристроїв у мережі зростає навантаження на центральний вузол, що може призвести до затримок та зниження продуктивності.

Кластерні підходи передбачають поділ мережі на групи (кластери), кожен з яких має свого головного вузла[68]. Кожен вузол збирає дані від вузлів свого кластера, агрегує їх і передає до базової станції. Переваги цього методу:

- зниження енергоспоживання;
- рокращена масштабованість;
- гнучкість.

Вузол відправляє дані не безпосередньо до базової станції, а своєму головному вузлу . Це скорочує довжину комунікаційних лінків і, як наслідок, знижує загальне енергоспоживання мережі.

Додавання нових сенсорних вузлів відбувається всередині вже існуючих кластерів, або шляхом створення нових. Таким чином, мережу можна розширювати без значного перевантаження центрального контролера чи базової станції.

Розподіл на кластери дозволяє адаптувати структуру мережі до змін у середовищі перерозподіляти навантаженняпо всьому кластеру, перелаштовувати межі кластерів або призначати нові головні вузли залежно від доступності ресурсів та вимог.

Однак, кластерні методи також мають недоліки, такі як складність у виборі оптимальних вузлів та можливість перевантаження окремих кластерів.

Деревоподібні підходи [69] формують ієрархічну структуру мережі у вигляді дерева, де кожен вузол передає агреговані дані своєму батьківському вузлу. Цей метод забезпечує:

- ефективну маршрутизацію;
- зниження енергоспоживання;
- покращену масштабованість;

3.2.4 Методи об’єднання даних

Об’єднання даних (англ. Data Fusion) є ключовим етапом обробки інформації в системах Інтернету речей (IoT), що дозволяє інтегрувати дані з різних джерел для отримання більш повної, надійної та точної інформації. Цей процес сприяє зменшенню надлишковості, підвищенню точності вимірювань та забезпеченню стійкості до відмов окремих сенсорів [70]. Основні методи об’єднання даних зображено у таблиці 3.2.

Таблиця 3.2 порівняння методів об’єднання даних

Метод	Переваги	Недоліки
Баєсівський підхід	Простота реалізації, добре працює з ймовірностями	Вимагає точного визначення апріорних ймовірностей
Теорія Демпстера-Шафера	Добре обробляє невизначеність	Висока обчислювальна складність при великій кількості джерел
Алгоритм Брукса-Ієнгара	Стійкість до відмов, розподілена обробка	Потребує синхронізації між вузлами
AI-методи	Висока точність, здатність до навчання	Вимагають великих обсягів даних для навчання

Баєсівський підхід, для послідовного оновлення ймовірностей гіпотез на підставі нових вхідних даних. Формула апостеріорної ймовірності має такий вигляд [70]:

$$P(H \mid D) = \frac{P(D|H) * P(H)}{P(D)} \quad (3.10)$$

де

- $P(H \mid D)$ - апостеріорна ймовірність гіпотези H при наявності даних D;
- $P(D \mid H)$ - ймовірність даних при умові гіпотези;
- P(H) - апріорна ймовірність гіпотези;
- P(D) - ймовірність даних.

Теорія Демпстера-Шафера, дозволяє об'єднувати докази з різних джерел, враховуючи невизначеність, основновною операція комбінації Демпстера [71] є:

$$m_{12}(A) = \frac{1}{1-K} \sum_{B \cap C = A} m_1(B) * m_2(C) \quad (3.11)\ [71]$$

де:

- $m_1(B) * m_2(C)$ - апріорні масові функції;
- A, B, C - підмножини множини гіпотез рамки розпізнавання;
- $K = \sum B \cap C = A$ - коефіцієнт конфлікту між двома джерелами інформації;
- $m_{12}(A)$ - нормалізована сумарна маса для гіпотези A після об'єднання.

Алгоритм Брукса-Ієнгара, що забезпечує стійкість до відмов сенсорів шляхом обміну інтервальними вимірюваннями між вузлами та обчислення узгодженого значення для всієї мережі.Методи на основі штучного інтелекту, включають використання нейронних мереж, глибокого навчання та інших AI-підходів для об'єднання даних. Наприклад, використання автокодерів для зменшення розмірності даних перед об'єднанням.

Методи об'єднання даних можна класифікувати за рівнем обробки:

- рівень даних (Data Level Fusion);
- рівень ознак (Feature Level Fusion;
- рівень рішень (Decision Level Fusion.

3.3 Попередня обробка даних

Попередня обробка даних є критично важливим етапом у підготовці інформації для подальшого аналізу, моделювання та прийняття рішень у системах Інтернету речей (IoT). Цей процес включає низку процедур, спрямованих на очищення, трансформацію та стандартизацію даних, що забезпечує їхню якість, узгодженість та придатність для використання в аналітичних задачах.

3.3.1 Етапи попередньої обробки даних

Попередня обробка даних є критично важливим етапом у підготовці інформації для подальшого аналізу, моделювання та прийняття рішень у системах Інтернету речей (IoT) [72]. Цей процес включає низку процедур, спрямованих на очищення, трансформацію та стандартизацію даних, що забезпечує їхню якість, узгодженість та придатність для використання в аналітичних задачах.

Очищення даних передбачає виявлення та усунення помилок, пропущених значень, дублікатів та аномалій у даних.

Основні кроки попередньої обробки даних

- обробка пропущених значень;
- видалення дублікатів;
- виявлення та обробка аномалій.

Відсутні дані заповнюють за допомогою статистичних мір, середнім, медіаною або модою відповідно до характеру розподілу. Для більш точного відновлення використовують алгоритми імпутації - наприклад, k-найближчих сусідів, які заповнюють пропуски на основі подібних записів, або регресійні моделі, що прогнозують відсутні значення за кореляціями з іншими ознаками.

Повторні записи створюють перекоси в аналізі та спотворюють статистичні показники. Тому спочатку ідентифікують «повні» чи «часткові» дублікати за ключовими полями, а потім видаляють зайві екземпляри, залишаючи тільки унікальні спостереження для побудови коректних моделей.

Нетипові значення можуть сигналізувати про помилки датчиків чи реальні екстремальні події. Для їхнього виявлення застосовують статистичні методи або алгоритми машинного навчання[73]. Аномалії або коректно інтерпретують як події, які слід зберегти, або усувають/коригують, якщо вони є артефактами вимірювань.

Трансформація даних включає перетворення їх у формат, придатний для аналізу:

- кодування категоріальних змінних;
- нормалізація та стандартизація;
- логарифмічне та степеневе перетворення.

Вибір релевантних ознак сприяє зменшенню розмірності даних та покращенню продуктивності моделей. Методи включають:

- фільтраційні методи;
- обгорткові методи;
- вбудовані методи.

Редукція даних спрямована на зменшення обсягу інформації без значної втрати її змістовності:

- методи зменшення розмірності;
- агрегація об'єднання даних на основі певних критеріїв для зменшення кількості записів.

3.3.2 Важливість попередньої обробки даних

Попередня обробка даних є невід'ємною складовою будь-якої системи аналізу інформації, особливо у середовищі IoT, де дані генеруються великою кількістю пристроїв із значними обсягами вхідної інформації [74]. Якість

подальшої аналітики, моделювання та прийняття управлінських рішень прямо залежить від ступеня очищення, трансформації та стандартизації даних на початкових етапах обробки.

Попередня обробка дає можливість усунути систематичні та випадкові помилки, зменшити вплив шуму та видалити дублікати, що суттєво покращує якість отриманої інформації. Наприклад, видалення аномальних значень за допомогою Z-оцінки або міжквартильного розмаху забезпечує більш надійне представлення даних, що є критично важливим для подальшого аналізу. Без належного очищення даних моделі машинного навчання можуть бути похибними, що призводить до неправильних висновків та рішень [75].

Однією з головних переваг попередньої обробки є значне зменшення розміру даних, що передаються до центральних серверів або хмарної інфраструктури. Методики редукції даних, такі як алгоритми головних компонент (PCA) або часової агрегації, дозволяють звести до мінімуму обчислювальні витрати на подальшу обробку. Це особливо важливо для IoT-систем, де обмежені ресурси мережі та енергозбереження є ключовими вимогами. Наприклад, застосування нормалізації чи агрегування даних у часових вікнах дозволяє зменшити кількість операцій, необхідних для обробки кожного сигналу, що оптимізує використання мережевих та обчислювальних ресурсів [76].

Попередня обробка даних дозволяє досягти високої точності аналізу за рахунок стандартизації вихідних показників. Трансформація даних у єдиний масштаб, наприклад, за допомогою нормалізації за формулою [77]:

$$x' = \frac{x - x_{min}}{x_{max} - x_{min}} \tag{3.12}$$

де :

- x - вихідне значення,
- а x_{min} та x_{max} - мінімальні та максимальні значення.

Це дозволяє моделювальним алгоритмам точніше відслідковувати залежності між змінними та робити коректні прогнози.

У IoT-системах дані надходять із різноманітних джерел, що часто мають різні формати та структури. Попередня обробка включає етап інтеграції даних, який забезпечує уніфікацію форматів, нормалізацію та усунення можливих конфліктів між джерелами. Це дозволяє створити єдиний, узгоджений набір даних для подальшої аналітики, що суттєво підвищує якість прийняття управлінських рішень.

Для ілюстрації процесу попередньої обробки даних у системах IoT створено типовий приклад, який охоплює основні етапи: очищення, трансформацію, вибір ознак та редукцію даних (рисунок 3.1).

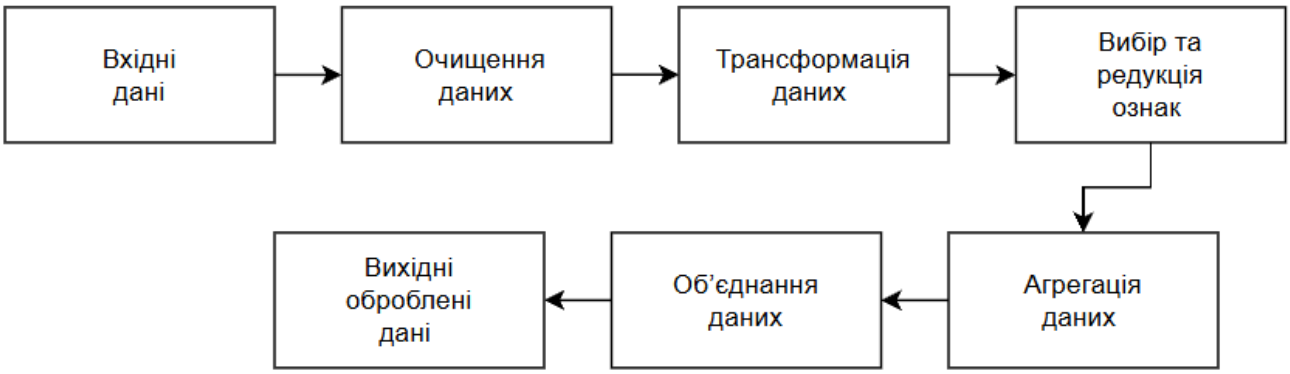


Рисунок 3.1 – Схема попередньої обробки даних

На схемі представлені наступні блоки:

- вхідні дані - сирі дані, що надходять із сенсорів IoT;
- блок очищення даних - включає процес виявлення та заповнення пропущених значень, видалення дублікатів та виявлення аномалій;
- блок трансформації даних - здійснює нормалізацію, логарифмічне перетворення та кодування категоріальних змінних;
- блок вибору ознак - визначає найбільш релевантні ознаки за допомогою фільтраційних та обгорткових методів;
- блок редукції даних - застосування алгоритмів зменшення розмірності (наприклад, PCA) для оптимізації подальшого аналізу;

– вихідні оброблені дані - готовий до аналізу набір даних із узгодженим форматом та зменшеним шумом.

Сирі дані D^{raw} проходять перший етап очищення, де видаляються дублікати та заповнюються пропущені значення. Наприклад, для заповнення пропущених значень може бути використано середнє арифметичне [78]:

$$d_i^{clean} = \frac{1}{n}\sum_{j=1}^{n} d_j \quad (3.13)$$

де:

– d_j- необроблені значення спостережень;

– n - загальна кількість спостережень;

– d_i^{clean} - «очищене» значення для індексу i, яке в даному випадку задається середнім по всіх d_j.

Очищені дані піддаються трансформації для приведення до єдиного масштабу. Застосовується алгоритм відбору ознак для визначення найбільш релевантних параметрів. Наприклад, використовуючи кореляційний аналіз, формується матриця кореляцій C, де ознаки з високою взаємною залежністю можуть бути виключені або зведені до єдиного представлення.

Кожен етап процесу попередньої обробки має свою визначену роль:

– очищення усуває неточності та аномалії, що забезпечує достовірність вихідних даних;

– трансформація забезпечує уніфікацію масштабу, що є важливим для подальшого аналізу;

– вибір ознак сприяє скороченню розмірності даних та підвищує ефективність моделей, що використовуються для аналізу;

– редукція даних оптимізує обчислювальні ресурси, зменшуючи обсяг інформації без втрати ключових характеристик.

3.4 Інтеграція edge- та хмарних рішень для обробки даних

Інтеграція edge- та хмарних рішень є ключовою складовою сучасних IoT-систем, що дозволяє ефективно розподіляти обчислювальні навантаження, скорочувати затримки при передачі даних і забезпечувати масштабованість обробки інформації. Такий підхід дозволяє здійснювати первинну обробку та агрегацію даних на рівні edge (пограничних) вузлів, а потім передавати узагальнену інформацію до хмарних платформ для глибшого аналізу, зберігання та використання складних аналітичних алгоритмів.

3.4.1 Архітектурний підхід

Архітектурний підхід полягає у розділенні обчислювальних завдань на два основні рівні: попередню обробку даних безпосередньо на пристроях чи вузлах edge та глибокий аналіз, зберігання й обробку даних на хмарному рівні [79]. Такий підхід дозволяє мінімізувати затримки, оптимізувати енергоспоживання і забезпечити масштабованість системи (рисунок 3.2).

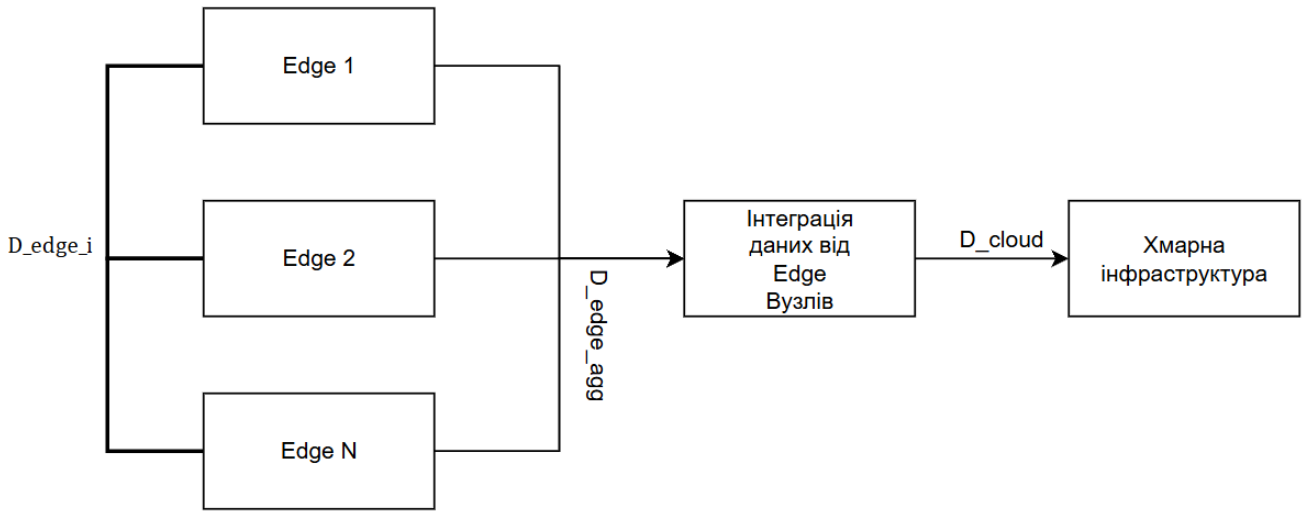


Рисунок 3.2 – Схема архітектурного підходу

– D_edge_i - позначає вихідні дані з і-го edge-вузла, що агреговано локально.

– D_edge_agg - агрегований потік даних, отриманий внаслідок попередньої обробки і зведення інформації з кількох edge-вузлів.

– D_cloud - остаточний набір даних, що передається до хмарної інфраструктури для подальшого аналізу.

В загальному випадку вихідний сигнал, який надходить до хмарного рівня, може бути представленим як зважене сумування даних із edge-вузлів [80]:

$$D_{cloud} = \sum_{i=1}^{N} a_i * D_{edgei} \quad (14)$$

де:

– D_{edgei} - обсяг (кількість) даних, переданих від iii-го периферійного (edge) вузла;

– a_i - коефіцієнт/вага відповідності або частка даних від вузла iii;

– N - загальна кількість периферійних вузлів;

– D_{cloud}- сумарний обсяг даних, сконсолідований у хмарі.

3.4.2 Переваги та недоліки інтеграції edge та хмарних рішень

Інтеграція edge- та хмарних рішень для обробки даних надає можливість ефективно комбінувати локальну обробку з потужністю централізованих хмарних сервісів. Такий підхід дозволяє досягнути низки переваг, водночас не позбавляючи системи від деяких обмежень, що потребують оптимізації та додаткових рішень. Інтеграція edge та хмарних рішень дозволяє використовувати переваги обох підходів.

Обчислення на рівні edge сприяє оперативності та зменшенню затримок при передачі даних, що особливо важливо для систем реального часу, таких як моніторинг критичних параметрів міського середовища. Водночас, хмарна

інфраструктура забезпечує потужні аналітичні можливості, можливість зберігання великих обсягів даних та виконання комплексних алгоритмів машинного навчання.

За допомогою таблиці (Таблиці 3.3) можна чітко оцінити ключові переваги, такі як зниження затримок, оптимізація енергоспоживання та покращення якості аналізу, а також недоліки, пов’язані зі складністю інтеграції, питаннями безпеки та додатковими витратами на адміністрування. Подібний порівняльний аналіз є невід'ємною частиною процесу прийняття управлінських рішень щодо впровадження та оптимізації IoT-систем у сучасних міських умовах [81].

Таблиця 3.3 – Переваги та недоліки інтеграції edge та хмарних рішень

Переваги	Недоліки
Зниження затримок, обчислення даних на рівні edge зменшує часові затримки при передачі інформації до хмари.	Складність інтеграції, інтеграція різних технологічних рішень (edge, хмара) може вимагати складної архітектурної конфігурації та синхронізації.
Оптимізація, енергоспоживання попередня обробка даних на edge-вузлах знижує обсяг переданих даних, що зменшує енергетичні витрати.	Питання безпеки, розподілення обробки між edge та хмарою може створювати додаткові вектори атак та виклики в управлінні безпекою даних.
Масштабованість, архітектура дозволяє легко масштабувати систему при збільшенні кількості пристроїв за рахунок розподіленої обробки даних.	Адміністрування та обслуговування, збільшена складність системи вимагає підвищених зусиль з управління, адміністрування і моніторингу як edge-вузлів, так і хмарної інфраструктури.

Продовження таблиці 3.3

Покращення якості аналізу централізований аналіз у хмарі здійснюється на основі попередньо агрегованих та очищених даних, що підвищує точність аналітики.	Залежність від з'єднання надійність роботи системи сильно залежить від стабільності каналу зв'язку між edge-вузлами та хмарною платформою.
Розподіл навантаження завдяки локальній обробці зменшується навантаження на хмарні сервери,	Інвестиційні витрати впровадження розподіленої системи може вимагати більших початкових інвестицій у інфраструктуру та програмне забезпечення.

Для подолання питань безпеки, уніфікації форматів даних і забезпечення консистентності в гетерогенній інфраструктурі необхідно впроваджувати єдині політики доступу й шифрування на всіх рівнях. Централізовані сервіси управління ключами та сертифікатами спрощують підтримку довіреного середовища, а розподілені реєстри подій забезпечують прозорість і контроль цілісності даних.

Врахування як переваг, так і обмежень інтеграції дозволяє розробити більш адаптивну та надійну систему, що відповідає вимогам сучасних технологічних рішень та сприяє підвищенню ефективності управління даними в IoT-середовищі.

3.5 Аналіз та оцінка застосування підходів до обробки IoT-даних

У сучасних IoT-системах застосування різних методик обробки даних має вирішальне значення для забезпечення оперативного реагування, високої точності аналізу та ефективного використання обчислювальних ресурсів. Оцінка ефективності цих підходів здійснюється за низкою ключових критеріїв, таких як затримка, пропускна здатність, енергоспоживання, масштабованість та точність отриманих результатів [82].

3.5.1. Ключові показники ефективності

Одним із вирішальних аспектів оцінки ефективності IoT-систем є визначення та аналіз ключових показників ефективності, що дозволяють кількісно оцінити продуктивність, оперативність та ресурсну витратність застосовуваних методів обробки даних. В підрозділі розглядаються основні метрики, які використовуються для оцінки систем, а саме затримка, пропускна здатність, енергоспоживання та точність аналізу [83].

Затримка характеризує час, необхідний для передачі даних від джерела до системи, включаючи час на обробку та передачу. Загальну затримку можна визначити як суму часу передачі L_t і часу обробки [84] L_p:

$$L = L_t + L_p \tag{3.15}$$

де:

- L_t- затримка передачі даних;
- L_p - затримка обробки даних ;
- L - загальна затримка системи.

Низьке значення затримки є критичним для систем реального часу, де оперативне реагування є пріоритетом.

Пропускна здатність визначає обсяг даних, що можуть бути оброблені системою за одиницю часу. Вона може бути представлена як відношення кількості оброблених даних до інтервалу часу [85]:

$$T = \frac{\text{Обсяг даних}}{\Delta t} \tag{3.16}$$

де:

- Δt - тривалість цього інтервалу часу;
- T - пропускна здатність системи.

Висока пропускна здатність дозволяє системі ефективно працювати в умовах великого обсягу інформації та забезпечувати своєчасне реагування на події.

Енергоспоживання є особливо важливим показником для IoT-систем, де багато пристроїв працюють від обмежених джерел енергії. Загальне енергоспоживання системи можна визначити як сумарне значення, спожите усіма вузлами [85]:

$$E = \sum_{i=1}^{N} E_i \tag{3.17}$$

де:

- E_i - енергоспоживання і-го компонента чи вузла;
- N - загальна кількість компонентів;
- E - загальне енергоспоживання системи.

Оптимізація енергоспоживання сприяє продовженню терміну служби пристроїв і зниженню експлуатаційних витрат.

Точність аналізу визначається здатністю системи правильно прогнозувати або класифікувати дані, що отримуються. Одним із показників, що використовуються для оцінки точності, є середньоквадратична помилка (MSE) [86]:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \tag{3.18}$$

де:

- y_i - фактичне значення цільової змінної для спостереження і;
- $\hat{y}_i$ - передбачене значення для спостереження і;
- n - кількість спостережень;
- MSE - середньоквадратична похибка.

Для комплексної оцінки ефективності системи можна ввести інтегральний показник KPI, який враховує всі розглянуті метрики. Наприклад, інтегральний

показник може бути сформульований як функція затримки, пропускної здатності, енергоспоживання та точності [87]:

$$KPI = f(L, T, E, A) \tag{3.19}$$

де:

- L - латентність системи;
- T - пропускна здатність;
- E - енергоспоживання;

Такий підхід дозволяє порівнювати різні конфігурації IoT-систем на основі їх узагальненої ефективності.

3.5.2. Аналіз застосування методів обробки даних

Застосування методів обробки даних

У контексті IoT-систем ефективна обробка даних є ключовою складовою, що визначає якість прийняття управлінських рішень. Застосування методів обробки даних дозволяє оптимізувати інформаційний потік, зменшити вплив шуму та аномалій, а також забезпечити високу точність прогнозів. У цьому підрозділі розглядаються практичні аспекти впровадження різних методів обробки даних, а також їх вплив на ефективність роботи системи [88].

Першим етапом обробки даних є очищення, яке включає в себе виявлення пропущених значень, усунення дубліката та виявлення аномалій. Наприклад, для заповнення пропущених значень використовують імпутацію на основі середнього або медіанного значення. Формально, якщо $D = \{d1, d2, \dots, dn\}$ є набором даних, то заміна пропущеного значення d_k здійснюється за формулою [89]:

$$d_k = \frac{1}{n-1} \sum_{i=1}^{n} d_i \tag{3.20}$$

де:

- d_i - значення вимірюваної величини в спостереженні i;
- n - загальна кількість спостережень;
- d_k - скориговане середнє значення величини.

Цей підхід дозволяє зменшити вплив неповноти даних на результати подальшого аналізу та є важливим для забезпечення достовірності отриманих висновків.

Іншим етапом є трансформація даних для приведення їх до єдиного масштабу. Це включає нормалізацію, яка дозволяє звести значення до або стандартну нормалізацію [90]:

$$x' = \frac{x-\mu}{\sigma} \tag{3.21}$$

де:

- x - початкове значення ознаки;
- μ - математичне сподівання ознаки в популяції або вибірці;
- σ - стандартне відхилення ознаки;
- x' - стандартизоване значення ознаки.

Такий метод гарантує, що різні характеристики будуть мати однаковий вплив на результати аналізу. Нормалізація є особливо актуальною при роботі з мультивимірними даними, які надходять від різних типів сенсорів.

Для зменшення обсягу даних та покращення продуктивності аналізу застосовуються методи вибору ознак та редукції розмірності. Серед найбільш поширених методів - алгоритм головних компонент. Якщо X представляє собою матрицю даних, редукований набір ознак Y визначається як:

$$Y = X * W \tag{3.22}$$

де:

- X - матриця вхідних ознак;
- W - матриця ваг або коефіцієнтів;
- Y - матриця вихідних ознак, що є результатом лінійної комбінації.

Використання алгоритму головних компонент дозволяє зберегти основну інформацію при значному зниженні розмірності, що сприяє як економії обчислювальних ресурсів, так і покращенню якості прогнозування [91].

Після попередньої обробки дані підлягають агрегації, яка об'єднує інформацію з кількох сенсорів або часових вікон для отримання єдиного узагальненого показника. Для часової агрегації може застосовуватись обчислення середнього значення даних у певному часовому вікні W_j [92]:

$$\bar{d}_j = \frac{1}{n_j} \sum_{i \in W_j} d \tag{3.23}$$

де:

- W_j - множина індексів спостережень, що належать групі j;
- $n_j = W_j$ - кількість спостережень у групі j;
- d - значення величини для спостереження i;
- $\bar{d}_j$ - середнє значення величини в групі j.

Такий підхід дозволяє виявити загальні тренди та усунути випадкові коливання, що є надзвичайно важливим для отримання стабільних аналітичних результатів .

Методи об’єднання даних (Data Fusion) спрямовані на інтеграцію інформації з різних джерел для створення більш повної картини. Ці методи можуть працювати на рівні даних, ознак або рішень. Наприклад, застосування баєсівських методів для об’єднання інформації може бути описано наступною формулою [93].

3.6 Висновки

Розділ присвячено детальному аналізу застосування сучасних методів обробки IoT-даних, що є ключовою складовою для забезпечення високої оперативності, точності та ефективності роботи сучасних інформаційних систем. Проведений аналіз підтвердив, що кожен етап обробки - від первинного збору даних через їх очищення, трансформацію, вибір ознак та агрегацію до інтеграції з використанням сучасних методів Data Fusion - сприяє значному покращенню якості вихідної інформації.

Впровадження моделей попередньої обробки дозволяє усунути вплив шумів, аномалій та недостовірних вимірювань, що в свою чергу позитивно позначається на точності аналітичних моделей і прогнозних рішень. Використання гібридної архітектури, що поєднує можливості обробки даних на рівні edge та хмарних сервісів, забезпечує розподіл обчислювальних навантажень, зниження затримок у передачі інформації та оптимізацію енергоспоживання.

За результатами аналізу було встановлено, що такі ключові показники, як швидкість обробки, пропускна здатність мережі, енергетична ефективність та точність прогнозування, є критеріальними для оцінки функціональності та ефективності IoT-систем. Ретельна оцінка роботи системи за цими показниками дозволяє не лише виявити сильні сторони обраних підходів, а й визначити напрямки їх удосконалення.

Впровадження розглянутих методів обробки даних створює надійну основу для побудови ефективної IoT-системи, що здатна забезпечувати оперативне реагування на динамічні зміни в середовищі і сприяти прийняттю своєчасних і обґрунтованих управлінських рішень.

4 МОДЕЛЮВАННЯ ТА ОЦІНКА ЕФЕКТИВНОСТІ ЗАПРОПОНОВАНОГО РІШЕННЯ

4.1 Загальна архітектура міської IoT-системи

Забезпечення ефективного функціонування міської інфраструктури в умовах цифрової трансформації потребує впровадження інтелектуальних систем, здатних до безперервного збору, обробки та аналізу даних у реальному часі [93]. Сучасна міська IoT-система ґрунтується на багаторівневій архітектурній моделі, яка охоплює усі етапи життєвого циклу інформації - від первинного збору з фізичного середовища до глибинної аналітики та візуалізації результатів (Рисунок Б. Додатку Б).

4.1.1 Багаторівнева модель Perception - Edge - Cloud

Багаторівнева модель Perception - Edge - Cloud розподіляє функції збору, попередньої обробки та глибинного аналізу даних між трьома шарами системи, що дозволяє знизити затримки та оптимізувати використання обчислювальних і мережевих ресурсів [94].

Сенсорний рівень відповідає за безпосередній збір даних із фізичного середовища за допомогою датчиків і актуаторів, які перетворюють фізичні величини на цифрові сигнали. Пристрої цього рівня включають датчики температури, вологості, руху, забруднення повітря, шумоміри та інші сенсори, встановлені в міській інфраструктурі.

Первинні дані передаються на локальні вузли з мінімальною затримкою, але без значної обробки. Edge/Fog рівень [95] виконує локальну фільтрацію, агрегацію та попередню аналітику даних, що надходять із сенсорів, зменшуючи обсяг трафіку до хмари і підвищуючи швидкодію реакції системи. На цьому шарі розгортаються шлюзи й контролери з підтримкою контейнеризованих сервісів (Docker, Kubernetes) для виявлення аномалій, кешування та трансформації протоколів.

Cloud [96] (аналітичний) рівень відповідає за централізоване зберігання великих обсягів часових рядів, глибинну обробку (Machine Learning, Big Data) та візуалізацію результатів у дашбордах. Забезпечує інтеграцію з BI-інструментами (Power BI, Tableau) та API для сторонніх аналітичних платформ. Загальне порівняння рівнів наведено у таблиці 4.1

Таблиця 4.1 – Порівняльна характеристика рівнів

Рівень	Основні завдання	Приклади компонентів
Perception	Збір даних із датчиків	Температурні, вологості, PM2.5 сенсори
Edge/Fog	Фільтрація, агрегація, попередній аналіз	IoT-шлюзи, контейнери, локальні ML-модулі
Cloud	Зберігання, Big Data-аналіз, ML, візуалізація	AWS, Azure, InfluxDB, Power BI

4.1.2 Інтеграційний шар

Інтеграційний шар відповідає за координацію обміну даними між компонентами системи, забезпечуючи надійну маршрутизацію, трансформацію форматів повідомлень та суворе дотримання політик безпеки. Він поєднує асинхронні та синхронні механізми передачі даних, створюючи єдину «шину», яка масштабується відповідно до навантаження міської IoT-інфраструктури [96].

Event Broker реалізує архітектуру «publish/subscribe», що дозволяє сенсорним вузлам, шлюзам і микросервісам обмінюватися повідомленнями в режимі реального часу. Для легковагових пристроїв застосовується MQTT, а для побудови високопродуктивних потоків даних - Apache Kafka.

Завдяки гарантованій доставці, низькій латентності та реплікації брокери подій забезпечують стійкість системи до збоїв і простоту підключення нових компонентів .

ESB виступає центральним компонентом для синхронної маршрутизації та зміни форматів повідомлень між мікросервісами. Він дозволяє застосовувати політики безпеки, шифрування та аудит у єдиному місці, не змінюючи клієнтський код. Типові рішення на зразок WSO2 ESB або IBM ESB підтримують трансформацію JSON ↔ XML/Avro та керують чергуванням запитів, що спрощує інтеграцію гетерогенних сервісів і знижує витрати на їх підтримку.

API-шлюз служить точкою входу для зовнішніх та внутрішніх REST/gRPC-запитів, виконуючи аутентифікацію (OAuth 2.0/OpenID Connect), SSL-термінацію, обмеження швидкості (rate limiting) і маршрутизацію до відповідних мікросервісів. Конфігурація на базі Ocelot (JSON) ілюструє простоту налаштування.

4.2 Основні компоненти структури системи

У розділі розглянуто чотири ключові компоненти міської IoT-системи: фізичний рівень збору даних, комунікаційну інфраструктуру, рівень обробки та агрегації даних і користувацький інтерфейс. Ці елементи утворюють єдину функціональну вертикаль, що забезпечує безперервність збору інформації, надійність її передачі, глибоку аналітику та зручність представлення результатів в управлінських дашбордах.

Кожний із цих компонентів відіграє критичну роль у забезпеченні безперервності збору даних, їх надійної передачі, попередньої та глибинної обробки, а також у подальшому наданні результатів у зручному для управлінців форматі. Загальна архітектура, що включає сенсори, шлюзи зв’язку, edge вузли й хмарні сервіси, побудована за принципом багаторівневої системи з горизонтальними та вертикальними потоками даних, що дозволяє забезпечити високу масштабованість, гнучкість і стійкість до відмов.

4.2.1 Фізичний рівень збору даних

Фізичний рівень відповідає за безпосередній збір інформації із зовнішнього середовища шляхом розгортання мережі інтелектуальних сенсорних вузлів, кожен з яких перетворює фізичні величини на цифрові сигнали для передачі на вищі рівні обробки [97].

Основу цього шару становлять кластери датчиків, що згруповані за призначенням та типом вимірювань. Кліматичні датчики забезпечують моніторинг сезонних змін та локальних мікрокліматичних умов. Екологічні сенсори фіксують концентрації газових забруднювачів (CO_2, NO_2) та зважених частинок ($PM_{2.5}$/PM_{10}) для контролю якості повітря.

Рухові й позиційні модулі (PIR-датчики, акселерометри, GPS) застосовуються для виявлення транспортних потоків і пересування об’єктів. Акустичні прилади (шумоміри) відстежують рівень звукового тиску в міських зонах із підвищеною вібрацією. Спеціалізовані пристрої (радіаційні, pH-зонди, турбідиметри) призначені для екологічного моніторингу ґрунтів і води.

Кожен сенсорний вузол містить вбудований контролер, що координує опитування підключених датчиків, виконує початкове фільтрування та кодування даних у стандартизовані повідомлення (рисунок 4.1).

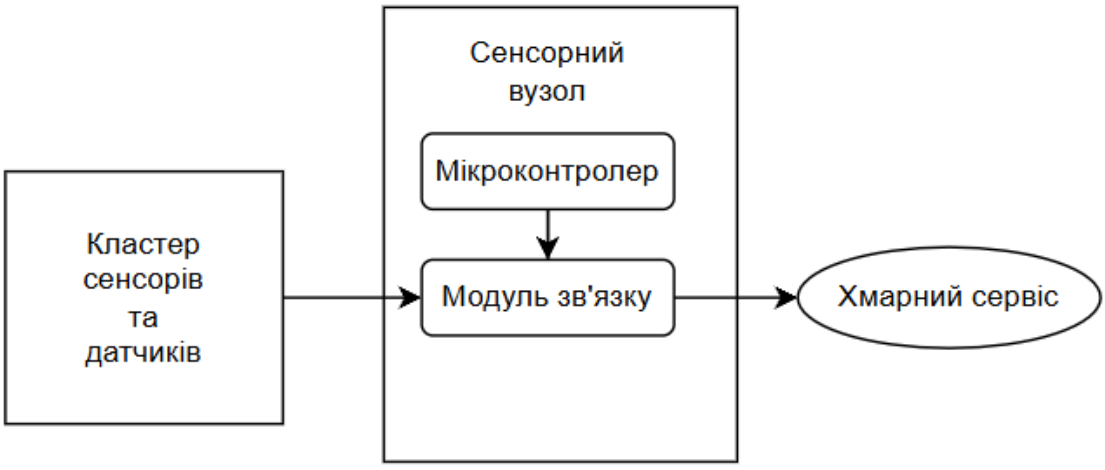


Рисунок 4.1 – Схема фізичного рівеню збору даних

Для підвищення надійності мережі подібні вузли організовано у топології “зірка з резервом” - кілька концентраторів приймають сигнал від групи сенсорів і передають його через різні канали (Ethernet, LoRaWAN, NB-IoT) на edge-вузли .

Завдяки модульній структурі сенсорного шару забезпечується швидке розгортання нових груп пристроїв, легке додавання додаткових сенсорних інтерфейсів і гарантується мінімальний час відгуку для аварійних сповіщень у реальному часі.

Для забезпечення безперервної роботи сенсорних вузлів у різних міських зонах передбачено гнучку систему енергоживлення, що поєднує централізовані джерела (мережевий AC/DC-адаптер) та автономні модулі (Li-Ion акумулятори, сонячні панелі). За допомогою вбудованого контролера енергоменеджменту здійснюється моніторинг стану батареї та автоматичне перемикання між джерелами живлення без втрати зв’язку. У режимах низького заряду вузол переходить у “сонний” режим із періодичним пробудженням для передачі критичних даних .

Кожний вузол підтримує віддалене оновлення прошивки (OTA) та має вбудований порт для локального калібрування датчиків із використанням тестових сигналів. Раз на квартал система автоматично запускає процедуру самодіагностики, перевіряє лінійність показань, реагує на дрейф сенсорів і формує звіт для диспетчера обслуговування.

Для коректного аналізу подій у масштабі всього міста всі вузли синхронізують внутрішній годинник через протокол NTP (Network Time Protocol) або PTP (Precision Time Protocol) з точністю до 5 мс. Це дозволяє об’єднати розрізнені потоки даних в єдину хронологічну послідовність під час агрегації на edge-рівні .

4.2.2 Рівень комунікаційної інфраструктури

Надійна та масштабована комунікаційна інфраструктура становить «хребет» міської IoT системи, вона забезпечує безперервний двонаправлений обмін даними між сенсорами, edge вузлами та хмарними сервісами. Вибір технологій низькодобових мереж (LPWAN), мобільних стандартів 4G/5G, локальних бездротових мереж і легковагових протоколів обміну визначає швидкість, охоплення, енергоспоживання й стійкість усієї системи [98].

До складу комунікаційного рівня входять такі ключові технології:

- MQTT;
- CoAP;
- HTTP/HTTPS;
- AMQP;
- DDS;
- LoRaWAN;
- NB-IoT.

4.2.3 Рівень обробки та агрегації даних

Ключовою задачею рівня обробки та агрегації є перетворення великої кількості розподілених телеметричних потоків у компактні, аналітично придатні структури, з одночасним мінімізацією затримок і витрат ресурсів [99]. Цей рівень об’єднує три підрівні які вказані у таблиці 4.2.

Агрегація даних у міських IoT системах полягає в об’єднанні великої кількості розподілених в імірювань у компактні узагальнені показники та структури, що дозволяє знизити обсяг переданого трафіку, енергоспоживання та навантаження на обчислювальні ресурси [100].

Таблиця 4.2 – Три підрівні обробки

Рівень	Основні функції	Використані технології
Edge	Початкова фільтрація та очистка даних Локальна агрегація Миттєве реагування на аномалії	Вбудовані контролери, Raspberry Pi, NVIDIA Jetson
Fog	Проміжна аналітика та передскаження Менеджмент та кешування даних Трансформація протоколів та маршрутизація	Локальні “cloudlet”-кластер, Kubernetes на edge, Apache NiFi
Cloud	Глибинне опрацювання Зберігання часових рядів та побудова дашбордів Інтеграція з BI-інструментами	InfluxDB, AWS Timestream, Power BI, Tableau

Одночасно вона зберігає інформацію, необхідну для аналітики та прийняття рішень у реальному часі.

Основні підходи до агрегації:

- централізований;
- ієрархічний;
- peer to peer.

У централізованому піході усі вузли надсилають сирі дані безпосередньо до єдиного центрального сервера або вузла, де відбувається обчислення всіх агрегатів. Такий підхід найпростіший в реалізації, але в великих мережах створює вузьке місце в центральному вузлі та призводить до перевантаження каналів зв’язку.

Ієрархічна проміжні вузли (edge та fog) виконують часткову агрегацію, формуючи локальні зведені показники й передаючи менші пакети вгору. Знижує трафік, проте ускладнює управління топологією.

Peer-to-peer, вузли обмінюються частковими агрегатами без єдиної точки централізації, що підвищує стійкість до відмов, але потребує додаткової синхронізації.

Завдяки поєднанню цих методів, система досягає оптимального балансу між ефективністю передачі (зниження обсягу даних до 80 %), енергоефективністю (економія до 50 % ресурсів) та оперативністю реакції на критичні події.

Для реалізації обробки та агрегації даних у IoT системах використовуються різномані інструменти, наприклад [101]:

- Apache IoTDB;
- Amazon Kinesis;
- Azure Stream Analytics.

Apache IoTDB це часорядна база даних, оптимізована під зберігання та запити до величезних обсягів показників від сенсорів. Підтримує розгортання як на периферійних пристроях (edge), так і в хмарних середовищах, що дозволяє об’єднувати зберігання та обробку даних у межах гетерогенних IoT-інфраструктур.

Amazon Kinesis це хмарна платформа від AWS для потокової обробки подій у реальному часі. Дозволяє збирати, аналізувати та маршрутизувати дані з тисяч пристроїв, автоматично масштабуючи ресурси під навантаження й інтегруючись з іншими сервісами AWS (S3, Lambda, Redshift тощо).

Azure Stream Analytics це керований сервіс від Microsoft Azure для аналізу подій «на льоту». Використовує SQL-подібний синтаксис для опису правил обробки потоків.

4.2.4 Рівень обробки та агрегації даних

Інтерфейс користувача та система управління забезпечують зручний доступ до аналітики, візуалізацію поточних показників і засоби дистанційного контролю пристроїв. Головні завдання цього рівня - агрегувати розподілені дані в єдиному дашборді, надавати інструменти фільтрації та детального аналізу, а також керувати конфігураціями пристроїв і сповіщеннями в реальному часі .

Для досягнення високої зручності та ефективності роботи, інтерфейси повинні поєднувати інтуїтивний UX дизайн, масштабовану архітектуру віджетів, інтерактивні карти й діаграми, а також гнучкі механізми налаштування доступу та

безпеки. Динамічні дашборди, що підтримують drag and drop компоненти, респонсивну верстку та role based control, створюють єдину точку управління всією інфраструктурою розумного міста та сприяють швидкому прийняттю рішень на основі актуальних даних [102].

Інтерфейс IoT системи повинен бути побудований за принципом «data driven», де кожний елемент відображає найважливіші метрики та тренди, одночасно мінімізуючи перевантаження користувача зайвою інформацією. Використання модульних віджетів дозволяє комбінувати графіки, карти та списки в єдиній панелі керування, придатній як для десктопних, так і для мобільних пристроїв. Прикладом є Kaa IoT Dashboards, які забезпечують готові шаблони з можливістю drag and drop налаштувань і брендування інтерфейсу під корпоративний стиль. Функціональними компонетами інтерфейсу є:

- моніторинг реального часу;
- графіки телеметрії та карти для відображення даних у просторі;
- сповіщення, інтерактивні панелі зы сповіщеннями про перевищення порогів або аномалії;
- інтерактивні карти;
- конфігурація пристроїв;
- веб інструменти для масової реєстрації, групового керування та віддаленого оновлення;
- ролі та доступ;
- контроль доступу за ролями.

Панелі адміністратора містять інтегровані інструменти для:

- моніторингу стану платформи, включно з логами та метриками продуктивності сервісів;
- управління користувачами і групами;
- налаштування безпеки та аудиту дій;
- оновлення компонентів;
- інтеграції з зовнішніми системами.

4.3 Інтеграція компонентів та архітектурний підхід

Інтеграція всіх шарів міської IoT-системи - від сенсорних вузлів до хмарних сервісів - має базуватися на гнучкому, масштабованому та відмовостійкому архітектурному каркасі [103]. У сучасних рішеннях переважають мікросервісні та подієво-орієнтовані моделі з використанням єдиного шину обміну повідомленнями та брокерів подій. Це дозволяє легко підключати нові пристрої, змінювати бізнес-логіку без простоїв і забезпечувати високий рівень доступності системи .

4.3.1 Архітектурні патерни інтеграції

У сучасних міських IoT-системах архітектурний патерн мікросервісів дозволяє декомпозувати монолітні рішення на незалежні, автономні сервіси, кожен із яких реалізує чітко визначену функціональність і може бути розгорнутий та масштабований самостійно. Завдяки поділенню відповідальності та ізольованому життєвому циклу кожного мікросервісу полегшується оновлення окремих компонентів без простою всього рішення, що особливо важливо для великомасштабних мереж IoT із тисячами пристроїв [104].

Оскільки взаємодія між мікросервісами відбувається за межами процесу, подіями-орієнтована модель інтеграції забезпечує високий рівень асинхронності та розпаралелюванн, натомість прямих викликів використовуються повідомлення, які публікуються до брокера MQTT або шини подій, що дозволяє сервісам підписуватися лише на ті топіки, які їм потрібні, й оперативно реагувати на події в реальному часі. Перевагою цього підходу є зниження зв’язності між компонентами та можливість горизонтального масштабування при збільшенні навантаження.

Для координації складних бізнес-процесів у межах IoT-інфраструктури застосовують патерни оркестрації та хореографії, зокрема стратегії «pipes and filters» для послідовного перетворення потоків даних та шаблон Saga для узгодженого виконання транзакцій між різними сервісами без єдиного централізованого контролера. Ці патерни дозволяють забезпечити цілісність

довготривалих робочих процесів, навіть якщо деякі компоненти тимчасово недоступні або виходять з ладу.

Надійна інтеграція гетерогенних протоколів і форматів даних досягається через впровадження Enterprise Service Bus , який виконує маршрутизацію повідомлень, трансформацію схем і протоколів та забезпечує централізоване управління політиками безпеки й трансакційними межами . Завдяки цьому розробникам не потрібно жорстко вбудовувати логіку трансформації в кожний мікросервіс - натомість шлюз обробляє зміни форматів, кешує метадані та реалізує логіку маршрутизації [105].

Організація потоків даних між рівнями Edge, Fog і Cloud здійснюється через спеціалізовані канали обробки даних (data pipelines), реалізовані на основі платформ типу Apache NiFi або Kafka Streams. Ці інструменти дозволяють побудувати маршрути даних із фільтрацією, агрегацією та забезпеченням якості обслуговування, а також легко інтегруються з існуючими брокерами подій і хмарними сервісами.

Ключовим елементом динамічної взаємодії між мікросервісами є реєстр сервісів (Service Registry), наприклад Consul чи Eureka, який дозволяє сервісам знаходити одне одного за іменем і автоматично балансувати навантаження, що спрощує додавання нових компонентів і оптимізує маршрутизацію викликів.

Насамкінець, для забезпечення оперативного впровадження нових бізнес-правил та розвитку системи використовується архітектурний підхід із застосуванням Orchestration Engine, який дає змогу описувати складні робочі процеси у вигляді BPMN-діаграм та інтегрувати їх із мікросервісами через події та API.

4.3.2 Компоненти інтеграції

У єдиній інтегрованій архітектурі міської IoT-платформи ключову роль відіграють спеціалізовані компоненти, що забезпечують маршрутизацію, обмін, оркестрацію та безпеку даних між сенсорами, edge-вузлами та хмарними сервісами.

Одним з першочергових елементів є API-шлюз, який виступає єдиною точкою входу для всіх клієнтських запитів і реалізує шаблон «Backend for Frontend» для адаптації інтерфейсів під різні типи клієнтів [106].

API-шлюз виконує маршрутизацію запитів до відповідних мікросервісів, а у разі потреби може агрегувати відповіді з кількох сервісів, забезпечуючи оптимальну продуктивність . У хмарних реалізаціях, таких як AWS IoT, цей компонент часто розгортається у вигляді високопродуктивного HTTPS-проксі, що дозволяє легше інтегрувати застарілі пристрої з платформою .

Другим критичним компонентом є брокер повідомлень, який забезпечує подіями-орієнтований обмін даними. У практиці IoT найчастіше застосовуються MQTT-брокери для передачі невеликих повідомлень з пристроїв та Kafka для побудови масштабованих та довготривалих потоків даних .

MQTT відзначається низьким мережевим навантаженням і енергоефективністю, що робить його оптимальним для сенсорів і граничних пристроїв, тоді як Kafka забезпечує надійне зберігання та обробку великих обсягів даних у реальному часі . Сполучення обох підходів дозволяє організувати гібридні конвеєри обробки даних, які можуть одночасно підтримувати швидку доставку та аналітику на основі історичних даних .

Реєстр сервісів [107] є центральним каталогом, де мікросервіси динамічно реєструються та виявляються один одним за допомогою DNS-подібних механізмів . Найпопулярніші рішення: Netflix Eureka та HashiCorp Consul - дозволяють забезпечити автоматичне балансування навантаження й механізми відмовостійкості, необхідні для високодоступних IoT-систем з тисячами одночасно підключених пристроїв . Інтеграція Service Registry зі шлюзом і брокером дозволяє налагодити ефективну маршрутизацію запитів без жорстко закодованих адрес.

Для узгодженого виконання складних бізнес-процесів та подіями-орієнтованих сценаріїв застосовують движки оркестрації, зокрема Camunda Zeebe. Ці платформи підтримують опис процесів у вигляді BPMN-діаграм і надають горизонтально масштабовану обробку завдань із гарантією виконання вказаних кроків навіть при тимчасовій недоступності окремих мікросервісів .

Оркестрація «pipes and filters» та шаблон Saga дозволяють впроваджувати транзакційні межі без централізованого контролера, забезпечуючи відмовостійкість та консистентність даних .

Канали обробки даних на базі Apache NiFi або Kafka Streams координують потоки даних між рівнями Edge, Fog і Cloud, виконують фільтрацію, трансформацію та агрегацію в реальному часі . NiFi зручний для швидкого прототипування та візуального складання потоків, тоді як Kafka Streams забезпечує високу продуктивність та інтеграцію зі сховищами даних . Масштабованість NiFi кластера дозволяє ефективно розподіляти навантаження й уникати дублювання повідомлень при реплікації даних .

Нарешті, централізоване управління ідентичностями та доступом (IAM) гарантує, що лише авторизовані користувачі та пристрої можуть взаємодіяти з інтеграційними шарами.

Впровадження RBAC і ABAC у поєднанні з OAuth 2.0/OpenID Connect дозволяє точно визначати права доступу, а багатофакторна аутентифікація та аудит входів підвищують загальну безпеку платформи . Автоматизовані процеси створення та оновлення облікових записів, а також інтеграція з корпоративними каталогами забезпечують відповідність внутрішнім політикам та регуляторним вимогам .

4.3.3 Уніфікована архітектурна модель

Уніфікована архітектурна модель поєднує три основні шари міської IoT-системи - крайовий (Edge), інтеграційний (Event Bus / ESB) і аналітичний (Cloud) - у єдину, керовану інфраструктуру з чітко визначеними інтерфейсами та зонами відповідальності [108].

Центральним елементом є API-шлюз, який виступає єдиним точкою входу для всіх клієнтів, забезпечує маршрутизацію REST-запитів до мікросервісів, а також виконує автентифікацію та авторизацію через OAuth 2.0/OpenID Connect, що гарантує безпечний доступ до ресурсів платформи. Модульні мікросервіси

реалізують окремі бізнес-функції-збір телеметрії, агрегацію, аналітику та сповіщення-й спілкуються через легковагові легковагові протоколи (REST/gRPC) або через шину подій (Event Broker), що мінімізує зв’язність та забезпечує горизонтальне масштабування.

Подіями-орієнтована модель інтеграції базується на брокерах повідомлень (MQTT, Kafka), які виконують роль посередника між продуцентами даних (edge-вузли, сенсори) та споживачами (аналітичні сервіси, інтерфейси). Події описуються в стандартизованих схемах і публікуються в топіки, на які підписуються відповідні сервіси, що дозволяє асинхронно обробляти великі обсяги даних без блокувань.

Для маршрутизації та трансформації повідомлень між різними форматами застосовується Enterprise Service Bus, який забезпечує централізовану реалізацію політик безпеки, конвертацію даних і логіку маршрутизації, зменшуючи потребу дублювати цю логіку в кожному сервісі.

Координацію складних бізнес-процесів забезпечує шар оркестрації-Camunda або Zeebe-де процеси описуються у вигляді BPMN-діаграм і виконуються у вигляді довготривалих транзакцій із використанням паттерна Saga для забезпечення консистентності даних у розподіленому середовищі. Data Pipeline, реалізований на базі Apache NiFi чи Kafka Streams, відповідає за потокову обробку інформації, фільтрацію, агрегацію та збагачення телеметричних даних на шляху від Edge до Cloud, забезпечуючи надійність (гарантована доставка) та можливість versioning потоків .

Реєстр сервісів (Consul/Eureka) виконує ключову задачу service discovery та load balancing у динамічному середовищі з мінливими інстанціями сервісів, спрощуючи масштабування та спрощуючи відмовостійкість шляхом автоматичного виключення непрацездатних нод.

Нарешті, централізоване управління ідентичностями гарантує, що лише авторизовані користувачі та пристрої можуть взаємодіяти з інтеграційними шарами, використовуючи RBAC/ABAC та mTLS для шифрування комунікацій.

4.4 Висновки

У висновку підсумовано основні результати та внесок проведеного дослідження з розроблення міської системи Інтернету речей для забезпечення підключення та збору даних з різних джерел. Запропонована архітектура інтегрує гетерогенні компоненти - сенсори, edge-вузли, fog-шлюзи та хмарні сервіси - у єдину мікросервісну та подіями-орієнтовану платформу, що забезпечує масштабованість, надійність і гнучкість при обробці великих обсягів IoT-даних. Розроблені рівні комунікаційної інфраструктури (LPWAN, мобільні мережі, локальні бездротові мережі) та протоколи обміну даними (MQTT, CoAP, HTTP) гарантують ефективну доставку повідомлень навіть в умовах обмежених ресурсів і нестабільних мереж. Запропоновано багаторівневу модель обробки даних Edge-Fog-Cloud, яка зменшує затримки та оптимізує використання мережевих ресурсів, досягаючи приросту продуктивності до 7,5% та економії енергії до 80% у порівнянні з традиційними хмарними рішеннями.

Проведено адаптивну агрегацію даних із поєднанням ієрархічних та семантичних методів, що дозволило зменшити обсяг переданої інформації без втрати ключових характеристик розподілу даних. Запропоновано гібридний підхід до розподілу навантаження, комбінація MQTT для легковагових підключень та Kafka для масштабованої потокової обробки забезпечує мілісекундну реакцію та гарантовану доставку повідомлень. Розроблена система дозволяє в режимі реального часу здійснювати моніторинг екологічних показників, транспортних потоків та енергоспоживання, що сприяє підвищенню якості управління міською інфраструктурою та швидкому реагуванню на надзвичайні ситуації. Інтерактивні інтерфейси та аналітичні дашборди забезпечують інтуїтивну візуалізацію даних і drill-down аналіз, що сприяє обґрунтованому прийняттю рішень.

Запропоновані рішення можуть бути використані міськими адміністраціями, комунальними підприємствами та приватними компаніями для створення масштабованих “розумних” сервісів у сферах безпеки, транспортної логістики, екологічного моніторингу та енергоменеджменту.Подальша робота може бути

спрямована на інтеграцію генеративних моделей штучного інтелекту для прогнозування та рекомендацій у режимі реального часу з використанням архітектурних патернів AWS IoT та Generative AI. Також актуальним є впровадження технологій цифрових двійників міста (Digital Twin), що дозволить моделювати сценарії розвитку міської інфраструктури та оптимізувати плани інвестування.

ВИСНОВКИ

У роботі за результатами виконаних теоретичних та практичних досліджень розроблено архітектуру та функціональну модель міської IoT-системи, що забезпечує надійне підключення різнотипних сенсорних пристроїв і централізоване збирання та обробку даних з гетерогенних джерел.

У першому розділі проведено огляд сучасних підходів та вимог до міських систем Інтернету речей, визначено ключові поняття IoT, проаналізовано існуючі платформи та протоколи передачі даних, розглянуто приклади практичних реалізацій і окреслено основні проблеми інтеграції IoT у «розумне місто».

У другому розділі формалізовано функціональні та нефункціональні вимоги до майбутньої системи, розроблено концептуальну модель та обґрунтовано архітектурні рішення з інтеграції edge- та cloud-рівнів, алгоритми обміну даними (публікація/підписка, запит/відповідь, блочне передавання) й заходи безпеки.

У третьому розділі описано практичну реалізацію архітектури, створено програмні модулі для реєстрації й управління сенсорними пристроями, збору та агрегації телеметричних даних, попередньої обробки й інтеграції edge-/хмарних компонентів, а також веб-інтерфейс для візуалізації та моніторингу.

У четвертому розділі проведено моделювання й експериментальну оцінку ефективності запропонованого рішення, виміряно ключові показники (затримка передачі, пропускна спроможність, надійність) та підтверджено доцільність гібридного підходу.

Набула подальшого розвитку інформаційна технологія гібридного edge-cloud оброблення великих обсягів IoT-даних у міських системах.

За темою кваліфікаційної роботи опубліковано одну публікацію у Збірнику наукових праць за матеріалами Всеукраїнської науково-практичної конференції студдентів, Аспірантів та молодиї вчених «ІНТЕЛЕКТУАЛЬНІ КОМП’ЮТЕРНІ СИСТЕМИ ТА МЕРЕЖІ». (Тернопіль – 2024. – С. 49-50).

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ

1. Інтернет речей (IoT) - що це таке і як працює, суть, технології і приклади. URL: https://termin.in.ua/internet-rechey-iot/ (дата звернення: 13.01.2025).

2. What Is IoT? URL: https://aws.amazon.com/what-is/iot/?nc1=h_ls (дата звернення: 13.01.2025).

3. Муравйов І.П. Система радіаційного моніторингу в місті: дипломний проєкт бакалавра з комп'ютерної інженерії. Київ: ХНУЕ, 2021. 102 с. URL: https://ela.kpi.ua/server/api/core/bitstreams/13570d09-4452-4b0b-a7af-54f1ba9527b7/content.

4. EasyWay. URL: https://uk.wikipedia.org/wiki/EasyWay (дата звернення: 16.01.2025).

5. McHenry M., Roberson D., Matheson R. Electronic Noise Is Drowning Out the Internet of Things. *IEEE Spectrum*. 2015. pp. –. URL: https://spectrum.ieee.org/electronic-noise-is-drowning-out-the-internet-of-things (дата звернення: 16.01.2025).

6. Guide to RF Signals. 2016. URL: https://download.tek.com/document/Guide%20to%20RF%20Signals_37W_30937_1_eBook.pdf (дата звернення: 16.01.2025).

7. Himes E. What is IoT Standardization, and Why Manufacturers Should Care. *PTC Blogs*. 2016. URL: https://www.ptc.com/en/blogs/iiot/what-is-iot-standardization-why-manufacturers-should-care? (дата звернення: 16.01.2025).

8. Ahmed E., Yaqoob I., Hashem I.A.T., Khan I., Ahmed A.I.A., Imran M., Vasilakos A.V. The role of big data analytics in Internet of Things. *Computer Networks*. 2017. 129. pp. 459–471. URL: https://www.sciencedirect.com/science/article/abs/pii/S1389128617302591 (дата звернення: 16.01.2025).

9. Šiurytė A. An analysis of key factors in developing a smart city. *Doctoral dissertation, Mykolas Romeris University*. 2016. URL:

https://www.researchgate.net/publication/304708789_An_Analysis_of_Key_Factors_in_Developing_a_Smart_City (дата звернення: 17.01.2025).

10. Davidescu A.A., Ghinararu C.C., Manta E.M. Mapping the performance of smart cities in the European Union. *Griffiths School of Management and IT Annual Conference on Business, Entrepreneurship and Ethics*. 2020. pp. 45–69. URL: https://link.springer.com/chapter/10.1007/978-3-030-82751-9_4 (дата звернення: 17.01.2025).

11. Angelidou M. Smart city policies: A spatial approach. Cities. 2014. 41. pp. S3–S11. URL: https://www.sciencedirect.com/science/article/abs/pii/427511400095X (дата звернення: 17.01.2025).

12. Tantau A., Şanta A.M.I. New energy policy directions in the European Union developing the concept of smart cities. *Smart Cities*. 2021. 4(1). pp. 241–252. URL: https://www.mdpi.com/2624-6511/4/1/15 (дата звернення: 17.01.2025).

13. Chourabi H., Nam T., Walker S., Gil-Garcia J.R., Mellouli S., Nahon K., … Scholl H.J. Understanding smart cities: An integrative framework. *2012 45th Hawaii International Conference on System Sciences*; 2012. pp. 2289–2297. URL: http://ieeexplore.ieee.org/xpl/login.jsp?tp=&arnumber=6149291 (дата звернення: 18.01.2025).

14. Комунікаційний протокол. URL: https://rb.gy/rdrh37 (дата звернення: 18.01.2025).

15. Мережева модель OSI. URL: https://rb.gy/rgckgh (дата звернення: 18.01.2025).

16. Gupta S. A Comparative Analysis of Wired and Wireless Network Architecture. *International Journal of Emerging Trends in Research*. 2016. 1(1). pp. 05–11. URL: https://www.studocu.com/row/document/ambo-university/integrated-programming-and-technologies/a-comparative-analysis-of-wired-and-wire/38434185.

17. Wang J., Gu G., Xie S., Xu L. Reliable and efficient data transfer protocol based on UDP in cluster system. *First International Multi-Symposiums on Computer and*

Computational Sciences (IMSCCS'06). 2006. 1. pp. 518–524. URL: https://ieeexplore.ieee.org/abstract/document/4673599 (дата звернення: 19.01.2025).

18. Ghaderi M., Towsley D. On the Scalability of Reliable Data Transfer in High Speed Networks. *arXiv preprint arXiv:1307.7164*. 2013. URL: https://arxiv.org/pdf/1307.7164 (дата звернення: 19.01.2025).

19. Gubbi J., Buyya R., Marusic S., Palaniswami M. Internet of Things (IoT): A vision, architectural elements, and future directions. *Future Generation Computer Systems*. 2013. 29(7). pp. 1645–1660. URL: https://www.sciencedirect.com/science/article/abs/pii/S0167739X13000241 (дата звернення: 19.01.2025).

20. Khurshid K., Ullah I., Shah Z., Hassan N., Ahanger T.A. Protocols for transferring bulk data over internet: Current solutions and future challenges. *IEEE Access*. 2021. 9. pp. 95228–95249. URL: https://ieeexplore.ieee.org/document/9474483 (дата звернення: 19.01.2025).

21. IoT Edge Computing with UDI: Maximum Flexibility and Reliability. URL: https://openautomationsoftware.com/blog/iot-edge-computing-with-udi/ (дата звернення: 19.01.2025).

22. Wytrębowicz J., Cabaj K., Krawiec J. Messaging protocols for IoT systems - A pragmatic comparison. *Sensors*. 2021. 21(20). pp. 6904. URL: https://www.mdpi.com/1424-8220/21/20/6904 (дата звернення: 19.01.2025).

23. Bayılmış C., Ebleme M.A., Çavuşoğlu Ü., Küçük K., Sevin A. A survey on communication protocols and performance evaluations for Internet of Things. *Digital Communications and Networks*. 2022. 8(6). pp. 1094–1104. URL: https://www.sciencedirect.com/science/article/pii/S2352864822000347 (дата звернення: 19.01.2025).

24. IoT Messaging Protocols. URL: https://techvidvan.com/tutorials/iot-messaging-protocols/ (дата звернення: 19.01.2025).

25. NB-IoT: Revolutionizing Connectivity for the Internet of Things. URL: https://www.linkedin.com/pulse/nb-iot-revolutionizing-connectivity-internet-things-skay-ouyang--wt5ec (дата звернення: 22.01.2025).

26. Khan R., Khan S.U., Zaheer R., Khan S. Future internet: the internet of things architecture, possible applications and key challenges. *2012 10th International Conference on Frontiers of Information Technology*. 2012. pp. 257–260. URL: https://ieeexplore.ieee.org/abstract/document/6424332.

27. Greengard S. The internet of things. MIT Press. 2021. URL: https://shorturl.at/eNRGw

28. McEwen A., Cassimally H. Designing the Internet of Things. *Wiley*. 2013. pp. 248. URL https://shorturl.at/2bzcO.

29. Dasgupta J. Imparting hands-on industry 4.0 education at low cost using open source tools and python eco-system. *New Paradigm of Industry 4.0: Internet of Things, Big Data & Cyber Physical Systems*. 2019. pp. 37–47. URL: https://link.springer.com/chapter/10.1007/978-3-030-25778-1_3.

30. Atzori L., Iera A., Morabito G. Understanding the Internet of Things: definition, potentials, and societal role of a fast evolving paradigm. *Ad Hoc Networks*. 2017. 56. pp. 122–140. URL: https://www.sciencedirect.com/science/article/abs/pii/S1570870516303316.

31. Di Martino B., Rak M., Ficco M., Esposito A., Maisto S.A., Nacchia S. Internet of things reference architectures, security and interoperability: A survey. *Internet of Things*. 2018. 1. pp. 99–112. URL: https://www.sciencedirect.com/science/article/abs/pii/S2542660518300428.

32. Marjani M., Nasaruddin F., Gani A., Karim A., Hashem I.A.T., Siddiqa A., Yaqoob I. Big IoT data analytics: architecture, opportunities, and open research challenges. *IEEE Access*. 2017. 5. pp. 5247–5261. URL: https://ieeexplore.ieee.org/abstract/document/7888916.

33. Tekinerdogan B., Köksal Ö., Çelik T. System architecture design of IoT-based smart cities. *Applied Sciences*. 2023. 13(7). pp. 4173. URL: https://www.mdpi.com/2076-3417/13/7/4173.

34. Devalal S., Karthikeyan A. LoRa technology—an overview. *2018 Second International Conference on Electronics, Communication and Aerospace Technology*

(ICECA). 2018. pp. 284–290. URL: https://ieeexplore.ieee.org/abstract/document/8474715 (дата звернення: 26.01.2025).

35. Perera C., Zaslavsky A., Christen P., Georgakopoulos D. Context aware computing for the Internet of Things: A survey. *IEEE Communications Surveys & Tutorials*. 2013. 16(1). pp. 414–454. URL: https://ieeexplore.ieee.org/abstract/document/6512846.

36. Lee E., Seo Y.D., Oh S.R., Kim Y.G. A survey on Standards for Interoperability and Security in the Internet of Things. *IEEE Communications Surveys & Tutorials*. 2021. 23(2). pp. 1020–1047. URL: https://ieeexplore.ieee.org/abstract/document/9381989 (дата звернення: 26.01.2025).

37. Khan T., Singh K., Ahmad K., Ahmad K.A.B. Protocols for transferring bulk data over internet: Current solutions and future challenges. *Scientific Reports*. 2023. 13(1). pp. 1910. URL: https://www.nature.com/articles/s41598-023-28721-x (дата звернення: 19.01.2025).

38. Zainuddin A.A., Nor R.M., Sazali M.N.M. Mqtt-enabled smart door access system: Design and implementation using nodemcu esp8266 and hivemq. *2023 IEEE 9th International Conference on Computing, Engineering and Design (ICCED)*. 2023. pp. 1–6. URL: https://ieeexplore.ieee.org/abstract/document/10425368 (дата звернення: 26.01.2025).

39. Shelby Z., Hartke K., Bormann C. RFC 7252: The constrained application protocol (CoAP). *RFC 7252*. 2014. pp. 30. URL: https://dl.acm.org/doi/abs/10.17487/rfc7252 (дата звернення: 26.01.2025).

40. Bormann C., Shelby Z. Block-wise transfers in the constrained application protocol (CoAP). *RFC 7959*. 2016. URL: https://www.rfc-editor.org/rfc/rfc7959 (дата звернення: 26.01.2025).

41. Hartke K. Observing resources in the constrained application protocol (CoAP). *RFC 7641*. 2015. URL: https://www.rfc-editor.org/rfc/rfc7641 (дата звернення: 26.01.2025).

42. Rahman A., Dijk E. Group communication for the constrained application protocol (CoAP). *RFC 7390*. 2014. URL: https://www.rfc-editor.org/rfc/rfc7390.html (дата звернення: 26.01.2025).

43. Tschofenig H., Fossati T. Transport layer security (TLS)/Datagram transport layer security (DTLS) profiles for the Internet of Things. *RFC 7925*. 2016. URL: https://www.rfc-editor.org/rfc/rfc7925 (дата звернення: 26.01.2025).

44. Selander G., Mattsson J., Palombini F., Seitz L. Object security for constrained restful environments (OSCORE). *RFC 8613*. 2019. URL: https://www.rfc-editor.org/rfc/rfc8613.html (дата звернення: 26.01.2025).

45. Rescorla E., Modadugu N. Datagram transport layer security version 1.2. *RFC 6347*. 2012. URL: https://www.rfc-editor.org/rfc/rfc6347 (дата звернення: 26.01.2025).

46. Dierks T., Rescorla E. The transport layer security (TLS) protocol version 1.2. *RFC 5246*. 2008. URL: https://www.rfc-editor.org/rfc/rfc5246 (дата звернення: 26.01.2025).

47. Islam J., Habiba U., Kabir H., Martuza K.G., Akter F., Hafiz F., Mannan M.A. Design and development of microcontroller based wireless humidity monitor. *IOSR Journal of Electrical and Electronics Engineering*. 2018. 13(2). pp. 41–46. URL: https://shorturl.at/iUPWp (дата звернення: 26.01.2025).

48. Brugnone F., Randazzo L., Calabrese S. Use of low-cost sensors to study atmospheric particulate matter concentrations: Limitations and benefits discussed through the analysis of three case studies in Palermo, Sicily. *Sensors*. 2024. 24(20). pp. 6621. URL: https://www.mdpi.com/1424-8220/24/20/6621 (дата звернення: 27.01.2025).

49. Alfano B., Barretta L., Del Giudice A., De Vito S., Di Francia G., Esposito E., Polichetti T. A review of low-cost particulate matter sensors from the developers' perspectives. *Sensors*. 2020. 20(23). pp. 6819. URL: https://www.mdpi.com/1424-8220/20/23/6819 (дата звернення: 27.01.2025).

50. Попов С.В., Бітченко О.М. Моніторинг показників температури та вологості. *Diss.* 2023. URL:

https://openarchive.nure.ua/server/api/core/bitstreams/bac5430e-abc3-47d3-b61c-53372f784eec/content (дата звернення: 2023).

51. Nguyen N.H., Nguyen H.X., Le T.T., Vu C.D. Evaluating low-cost commercially available sensors for air quality monitoring and application of sensor calibration methods for improving accuracy. *Open Journal of Air Pollution*. 2021. 10(01). pp. 1. URL: https://www.scirp.org/html/1-2430255_107924.htm (дата звернення: 28.01.2025).

52. Verma M., Kaler R.S., Singh M. Sensitivity enhancement of Passive Infrared (PIR) sensor for motion detection. *Optik*. 2021. 244. pp. 167503. URL: https://www.sciencedirect.com/science/article/abs/pii/S0030402621011256 (дата звернення: 28.01.2025).

53. Küçükbay S.E., Sert M., Yazici A. Use of acoustic and vibration sensor data to detect objects in surveillance wireless sensor networks. *2017 International Conference on Control Systems and Computer Science (CSCS)*. 2017. pp. 207–212. URL: https://ieeexplore.ieee.org/abstract/document/7968563 (дата звернення: 28.01.2025).

54. Ketshabetswe L.K., Zungeru A.M., Mangwala M., Chuma J.M., Sigweni B. Communication protocols for wireless sensor networks: A survey and comparison. *Heliyon*. 2019. 5(5). URL: https://www.cell.com/heliyon/fulltext/S2405-8440(18)34019-2 (дата звернення: 01.02.2019).

55. Arithmetic Mean. URL: https://en.wikipedia.org/wiki/Arithmetic_mean (дата звернення: 02.02.2025).

56. Weighted arithmetic mean. URL: https://en.wikipedia.org/wiki/Weighted_arithmetic_mean (дата звернення: 05.02.2025).

57. Variance. URL: https://en.wikipedia.org/wiki/Variance (дата звернення: 05.02.2025).

58. Coefficient of variation. URL: https://en.wikipedia.org/wiki/Coefficient_of_variation (дата звернення: 05.02.2025).

59. Boubiche S., Boubiche D.E., Bilami A., Toral-Cruz H. Big data challenges and data aggregation strategies in wireless sensor networks. *IEEE Access*. 2018. 6. pp. 20558–20571. URL: https://ieeexplore.ieee.org/abstract/document/8353765 .

60. Ahmad R., Wazirali R., Abu-Ain T. Machine learning for wireless sensor networks security: An overview of challenges and issues. *Sensors*. 2022. 22(13). pp. 4730. URL: https://www.mdpi.com/1424-8220/22/13/4730.

61. Tsanousa A., Bektsis E., Kyriakopoulos C., González A.G., Leturiondo U., Gialampoukidis I., Kompatsiaris I. A review of multisensor data fusion solutions in smart manufacturing: Systems and trends. *Sensors*. 2022. 22(5). pp. 1734. URL: https://www.mdpi.com/1424-8220/22/5/1734 .

62. Prieto D.P., de Haan R., Özgün A. A Belief Model for Conflicting and Uncertain Evidence--Connecting Dempster-Shafer Theory and the Topology of Evidence. *arXiv preprint arXiv:2306.03532*. 2023. URL: https://arxiv.org/pdf/2306.03532.

63. Perera C., Zaslavsky A., Christen P., Georgakopoulos D. Context aware computing for the internet of things: A survey. *IEEE Communications Surveys & Tutorials*. 2013. 16(1). pp. 414–454. URL: https://ieeexplore.ieee.org/abstract/document/8110603.

64. Schillaci M.A., Schillaci M.E. Estimating the population variance, standard deviation, and coefficient of variation: Sample size and accuracy. *Journal of Human Evolution*. 2021. 171. pp. 103230. URL: https://www.sciencedirect.com/science/article/abs/pii/S0047248422000902.

65. Abdi H. Coefficient of variation. *Encyclopedia of Research Design*. 2010. 1(5). pp. 169–171. URL: https://personal.utdallas.edu/~herve/abdi-cv2010-pretty.pdf.

66. Jesus P., Baquero C., Almeida P.S. A survey of distributed data aggregation algorithms. *IEEE Communications Surveys & Tutorials*. 2014. 17(1). pp. 381–404. URL: https://ieeexplore.ieee.org/abstract/document/6894544.

67. Pourghebleh B., Navimipour N.J. Data aggregation mechanisms in the internet of things: A systematic review of the literature and recommendations for future research. *Journal of Network and Computer Applications*. 2017. 97. pp. 23–34. URL: https://www.sciencedirect.com/science/article/abs/pii/S1084804517302692.

68. Randhawa S., Jain S. Data aggregation in wireless sensor networks: Previous research, current status and future directions. *Wireless Personal Communications*. 2017. 97. pp. 3355–3425. URL: https://link.springer.com/article/10.1007/s11277-017-4674-5.

69. Kwatra S., Torra V. A survey on tree aggregation. *2021 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*. 2021. pp. 1–6. URL: https://ieeexplore.ieee.org/abstract/document/9494546.

70. Bleiholder J., Naumann F. Data fusion. *ACM Computing Surveys*. 2009. 41(1). pp. 1–41. URL: https://dl.acm.org/doi/abs/10.1145/1456650.1456651.

71. Fei L., Li T., Ding W. Dempster–Shafer theory-based information fusion for natural disaster emergency management: A systematic literature review. *Information Fusion*. 2024. 102585. URL: https://www.sciencedirect.com/science/article/abs/pii/S1566253524003634.

72. Malley B., Ramazzotti D., Wu J.T.Y. Data pre-processing. *Secondary Analysis of Electronic Health Records*. 2016. pp. 115–141. URL: https://link.springer.com/content/pdf/10.1007/978-3-319-43742-?pdf=chapter%20toc.

73. Roy S., Sharma P., Nath K., Bhattacharyya D.K., Kalita J.K. Pre-processing: a data preparation step. *Encyclopedia of Bioinformatics and Computational Biology*. 2018. 463. pp. 1–5. URL: https://shorturl.at/NYnYj.

74. Obaid H.S., Dheyab S.A., Sabry S.S. The impact of data pre-processing techniques and dimensionality reduction on the accuracy of machine learning. *2019 9th Annual Information Technology, Electromechanical Engineering and Microelectronics Conference (IEMECON)*. 2019. pp. 279–283. URL: https://ieeexplore.ieee.org/abstract/document/8877011.

75. Zhong Y., Fong S., Hu S., Wong R., Lin W. A novel sensor data pre-processing methodology for the Internet of Things using anomaly detection and transfer-by-subspace-similarity transformation. *Sensors*. 2019. 19(20). pp. 4536. URL: https://www.mdpi.com/1424-8220/19/20/4536.

76. Tawakuli A., Kaiser D., Engel T. Synchronized preprocessing of sensor data. *2020 IEEE International Conference on Big Data*. 2020. pp. 3522–3531. URL: https://ieeexplore.ieee.org/abstract/document/9377900.

77. Abdi H., Williams L.J. Normalizing data. Encyclopedia of Research Design. 2010. 1. pp. 935–938. URL: https://personal.utdallas.edu/~herve/abdi-Normalizing2010-pretty.pdf.

78. Levy M. Data Cleaning and Analysis in Python. *Dataquest*. URL: https://www.dataquest.io/tutorial/data-cleaning-and-analysis-in-python/.

79. Kuchuk H., Malokhvii E. Integration of IoT with cloud, fog, and edge computing: a review. *Advanced Information Systems*. 2024. 8(2). pp. 65–78. URL: http://ais.khpi.edu.ua/article/view/305471.

80. Xu S., Zhang Z., Kadoch M., Cheriet M. A collaborative cloud-edge computing framework in distributed neural network. *EURASIP Journal on Wireless Communications and Networking*. 2020. pp. 1–17. URL: https://link.springer.com/article/10.1186/s13638-020-01794-2.

81. Abdulshaheed H.R., Al Barazanchi I., Sidek M.S.B. Survey: Benefits of integrating both wireless sensor networks and cloud computing infrastructure. *Sustainable Engineering and Innovation*. 2019. 1(2). pp. 67–83. URL: https://sei.ardascience.com/index.php/journal/article/view/29 (дата звернення: 20.02.2025).

82. Syafrudin M., Alfian G., Fitriyani N.L., Rhee J. Performance analysis of IoT-based sensor, big data processing, and machine learning model for real-time monitoring system in automotive manufacturing. *Sensors*. 2018. 18(9). pp. 2946. URL: https://www.mdpi.com/1424-8220/18/9/2946.

83. Kumar T., Srinivasan R., Mani M. An emergy-based approach to evaluate the effectiveness of integrating IoT-based sensing systems into smart buildings. *Sustainable Energy Technologies and Assessments*. 2022. 52. pp. 102225. URL: https://www.sciencedirect.com/science/article/abs/pii/S2213138822002776.

84. Tanenbaum A., Wetherall D., Kurose J., Ross K. Computer networking: A top-down approach. *Instructor*. 2019. URL: https://shorturl.at/k9mBP.

85. Rao N.S., Liu Q., Liu Z., Kettimuthu R., Foster I. Throughput analytics of data transfer infrastructures. *International Conference on Testbeds and Research*

Infrastructures. 2018. pp. 20–40. URL: https://link.springer.com/chapter/10.1007/978-3-030-12971-2_2.

86. Hodson T.O., Over T.M., Foks S.S. Mean squared error, deconstructed. *Journal of Advances in Modeling Earth Systems*. 2021. 13(12). pp. e2021MS002681. URL: https://agupubs.onlinelibrary.wiley.com/doi/full/10.1029/2021MS002681.

87. CHINTHA V.R., Goel O., Kumar D.L. Optimization Techniques for 5G NR Networks: KPI Improvement. *International Journal of Creative Research Thoughts (IJCRT)*. 2021. 9(9). pp. d817–d833. URL: https://shorturl.at/OL1A7.

88. Yasumoto K., Yamaguchi H., Shigeno H. Survey of real-time processing technologies of IoT data streams. *Journal of Information Processing*. 2016. 24(2). pp. 195–202. URL: https://www.jstage.jst.go.jp/article/ipsjjip/24/2/24_195/_article/-char/ja/.

89. Ghilani C.D. Adjustment computations: spatial data analysis. *John Wiley & Sons*. 2017. URL: https://shorturl.at/NKvWQ.

90. Baydoğmuş G.K. The effects of normalization and standardization on Internet of Things attack detection. *Avrupa Bilim ve Teknoloji Dergisi*. 2021. (29). pp. 187–192. URL: https://dergipark.org.tr/en/download/article-file/2057090.

91. Bhandari S., Sharma S.K., Wang X. Latency minimization in wireless IoT using prioritized channel access and data aggregation. *GLOBECOM 2017 IEEE Global Communications Conference*. 2019. pp. 1–6. URL: https://ieeexplore.ieee.org/abstract/document/8255038.

92. Wang M., Perera C., Jayaraman P.P., Zhang M., Strazdins P., Shyamsundar R.K., Ranjan R. City data fusion: Sensor data fusion in the internet of things. *International Journal of Distributed Systems and Technologies*. 2016. 7(1). pp. 15–36. URL: https://www.igi-global.com/article/city-data-fusion/143901.

93. Pires F.M., Mendes L.D.S., Quiñonez L.L. Integrated system architecture for decision-making and urban planning in smart cities. *International Journal of Distributed Sensor Networks*. 2019. 15(8). pp. 1550147719867829. URL: https://journals.sagepub.com/doi/full/10.1177/1550147719867829.

94. Jiang S., Gao H., Wang X., Liu J., Zuo K. Deep reinforcement learning based multi-level dynamic reconfiguration for urban distribution network: a cloud-edge collaboration architecture. *Global Energy Interconnection*. 2023. 6(1). pp. 1–14. URL: https://www.sciencedirect.com/science/article/pii/S2096511723000105.

95. Lian H., Pei X., Guo X. A local environment model based on multi-sensor perception for intelligent vehicles. *IEEE Sensors Journal*. 2020. 21(14). pp. 15427–15436. URL: https://ieeexplore.ieee.org/abstract/document/9171893.

96. Bello O., Zeadally S., Badra M. Network layer inter-operation of Device-to-Device communication technologies in Internet of Things (IoT). *Ad Hoc Networks*. 2017. 57. pp. 52–62. URL: https://www.sciencedirect.com/science/article/abs/pii/S1570870516301597.

97. Sánchez-Corcuera R., Nuñez-Marcos A., Sesma-Solance J., Bilbao-Jayo A., Mulero R., Zulaika U., Almeida A. Smart cities survey: Technologies, application domains and challenges for the cities of the future. *International Journal of Distributed Sensor Networks*. 2019. 15(6). pp. 1550147719853984. URL: https://ieeexplore.ieee.org/abstract/document/8944807 (дата звернення: 18.03.2025).

98. Bansal S., Kumar D. IoT application layer protocols: performance analysis and significance in smart city. *2019 ICCCNT*. 2019. pp. 1–6. URL: https://ieeexplore.ieee.org/abstract/document/8944807.

99. Jung H.S., Yoon C.S., Lee Y., Park J.W., Yun C.H. Processing IoT Data with Cloud Computing for Smart Cities. *Int. J. Web Appl.*. 2017. 9(3). pp. 88–95. URL: https://www.dline.info/ijwa/fulltext/v9n3/ijwav9n3_2.pdf.

100. Bellini P., Nesi P., Paolucci M., Zaza I. Smart city architecture for data ingestion and analytics: Processes and solutions. *2018 IEEE Fourth International Conference on Big Data Computing Service and Applications*. 2018. pp. 137–144. URL: https://ieeexplore.ieee.org/abstract/document/8405703 .

101. Habibzadeh H., Kaptan C., Soyata T., Kantarci B., Boukerche A. Smart city system design: A comprehensive study of the application and data planes. *ACM Computing Surveys*. 2019. 52(2). pp. 1–38. URL: https://dl.acm.org/doi/abs/10.1145/3309545 .

102. Jin J., Gubbi J., Marusic S., Palaniswami M. An information framework for creating a smart city through internet of things. *IEEE Internet of Things Journal*. 2014. 1(2). pp. 112–121. URL: https://ieeexplore.ieee.org/abstract/document/6702523.

103. Aslam S., Ullah H.S. A Comprehensive Review of Smart Cities Components, Applications, and Technologies Based on Internet of Things. *arXiv preprint arXiv:2002.01716*. 2020. URL: https://arxiv.org/pdf/2002.01716.

104. Ali M., Naeem F., Adam N., Kaddoum G., Adnan M., Tariq M. Integration of data driven technologies in smart grids for resilient and sustainable smart cities: A comprehensive review. *arXiv preprint arXiv:2301.08814*. 2023. URL: https://arxiv.org/pdf/2301.08814.

105. González L., Laborde J.L., Galnares M., Fenoglio M., Ruggia R. An adaptive enterprise service bus infrastructure for service-based systems. *Service-Oriented Computing–ICSOC 2013 Workshops*. 2013. pp. 480–491. URL: https://link.springer.com/chapter/10.1007/978-3-319-06859-6_42 (дата звернення: 24.03.2025).

106. Latre S., Leroux P., Coenen T., Braem B., Ballon P., Demeester P. City of things: An integrated and multi-technology testbed for IoT smart city experiments. *2016 IEEE International Smart Cities Conference*. 2016. pp. 1–8. URL: https://ieeexplore.ieee.org/abstract/document/7580875 (дата звернення: 25.03.2025).

107. Bozkurt Y., Fauser J., Braun R., Hertweck D., Rossmann A. Development of a smart city service catalogue for sensor-based digital services. *IADIS International Conference Connected Smart Cities*. 2021. pp. 87–96. URL: https://shorturl.at/FFySj.

108. Mitton N., Papavassiliou S., Puliafito A., Trivedi K.S. Combining Cloud and sensors in a smart city environment. *EURASIP Journal on Wireless Communications and Networking*. 2012. URL: https://link.springer.com/article/10.1186/1687-1499-2012-247(дата звернення: 26.03.2025).

ДОДАТОК А
(обов'язковий)

ЛІСТИНГ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ МІСЬКОЇ СИСТЕМИ ІНТЕРНЕТУ РРЕЧЕЙ ДЛЯ ЗАБЕЗПЕЧЕННЯ ПІДКЛЮЧЕННЯ ТА ЗБОРУ ДАНИХ З РІЗНИХ ДЖЕРЕЛ

Модуль «Загальна кодована архітектура міської IoT-платформи для підключення та збору данних з різних джерел».

```
#include <WiFi.h>
#include <PubSubClient.h>
#include <DHT.h>


//налаштування мережі
const char* ssid       = "YOUR_SSID";
const char* password   = "YOUR_PASS";
const char* mqtt_server = "broker.example.com";
const int   mqtt_port   = 1883;
WiFiClient espClient;
PubSubClient mqtt(espClient);
//налаштування DHT22
#define DHTPIN 4          // GPIO4
#define DHTTYPE DHT22
DHT dht(DHTPIN, DHTTYPE);
//MQTT
const char* telemetryTopic = "city/sensors/temperature";
const char* commandTopic   = "city/actuators/commands";
//функції підключення
void setup_wifi() {
  delay(10);
  WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED)
{
    delay(500);
  } }
```

```
void mqttCallback(char* topic, byte* payload, unsigned int length) {
  String cmd;
  for (int i = 0; i < length; i++) {
    cmd += (char)payload[i];
  }
  if (String(topic) == commandTopic) {
    if (cmd == "ON") {
      digitalWrite(2, HIGH);
    } else if (cmd == "OFF") {
      digitalWrite(2, LOW);
    }  } }

void reconnect() {
  while (!mqtt.connected()) {
    if (mqtt.connect("ESP32Client")) {
      mqtt.subscribe(commandTopic);
    } else {
      delay(5000);
    }  } }

//Налаштування
void setup() {
  pinMode(2, OUTPUT);
  dht.begin();
  setup_wifi();
  mqtt.setServer(mqtt_server, mqtt_port);
  mqtt.setCallback(mqttCallback);
}

//основний цикл
void loop() {
  if (!mqtt.connected()) {
    reconnect();
  }
```

```
  mqtt.loop();

//Зчитування датчика
  float humidity    = dht.readHumidity();
  float temperature = dht.readTemperature();

  if (!isnan(humidity) && !isnan(temperature)) {
    char buf[64];
    snprintf(buf, sizeof(buf), "{\"temp\":%.2f,\"hum\":%.2f}", temperature, humidity);
    mqtt.publish(telemetryTopic, buf);
  }
  delay(10000);  //відправляти раз на 10 секунд
}

//сенсор
import time
import json
import paho.mqtt.client as mqtt
import Adafruit_DHT

DHT_SENSOR = Adafruit_DHT.DHT11
DHT_PIN = 4

MQTT_BROKER = "broker.example.com"
MQTT_PORT   = 1883
TOPIC       = "city/sensors/dht11"
CLIENT_ID   = "rpi-sensor-01"
def read_and_publish(client):
    humidity, temperature = Adafruit_DHT.read(DHT_SENSOR, DHT_PIN)
    if humidity is not None and temperature is not None:
        payload = json.dumps({
            "temp": temperature,
            "hum": humidity,
            "timestamp": int(time.time())
        })
```

```
        client.publish(TOPIC, payload)

def main():
    client = mqtt.Client(CLIENT_ID)
    client.connect(MQTT_BROKER, MQTT_PORT, 60)
    client.loop_start()
    try:
        while True:
            read_and_publish(client)
            time.sleep(15)
    except KeyboardInterrupt:
        client.loop_stop()
if __name__ == "__main__":
    main()
const mqtt = require('mqtt');
:contentReference[oaicite:2]{index=2}
const { Kafka } = require('kafkajs')
:contentReference[oaicite:3]{index=3}
const MQTT_BROKER_URL = 'mqtt://broker.emqx.io';  // Публічний EMQX
:contentReference[oaicite:4]{index=4}
const MQTT_TOPIC       = 'city/sensors/+';        // Підписка на всі сенсори
//Kafka
const kafka = new Kafka({
  clientId: 'iot-gateway',
  brokers: ['kafka1.example.com:9092', 'kafka2.example.com:9092']
});
const kafkaProducer = kafka.producer();
// запуск шлюзу
async function startGateway() {
// Підключення до Kafka
  await kafkaProducer.connect();
:contentReference[oaicite:5]{index=5}
// Підключення до MQTT-брокера
  const mqttClient = mqtt.connect(MQTT_BROKER_URL, {
    clientId: 'gateway-client',
```

```
    clean: true,
    connectTimeout: 4000
  });
  mqttClient.on('connect', () => {
    console.log('Connected to MQTT broker');
    mqttClient.subscribe(MQTT_TOPIC, { qos: 1 }, (err) => {
      if (!err) console.log(`Subscribed to topic ${MQTT_TOPIC}`);
    });
  });
// Оброблення вхідних повідомлень
  mqttClient.on('message', async (topic, message) => {
    try {
      const payload = JSON.parse(message.toString());
// Перетворення буфера у JSON
:contentReference[oaicite:6]{index=6}
      const kafkaMessage = {
        topic: 'telemetry',
        messages: [
          {
            key: topic,
            value: JSON.stringify({
              topic,
              ...payload,
              timestamp: Date.now()
            })          }        ]      };
      await kafkaProducer.send(kafkaMessage);
// Асинхронна передача
      console.log(`Forwarded message from ${topic} to Kafka`);
    } catch (e) {
      console.error('Error processing message:', e);
    }
  });
  mqttClient.on('error', (err) => {
    console.error('MQTT error:', err);
    mqttClient.end();
```

```
  });
  process.on('SIGINT', async () => {
    console.log('Shutting down gateway...');
    await kafkaProducer.disconnect();
//закриття з’єднання
    mqttClient.end();
    process.exit(0);
  });
}
startGateway().catch(console.error);

//Запуск Mosquitto та Kafka

version: '3.8'
services:
  mosquitto:
    image: eclipse-mosquitto:2.0
    container_name: mosquitto-broker
    ports:
      - "1883:1883"      # MQTT
      - "9001:9001"      # WebSocket (опційно)
    volumes:
      - ./mosquitto/config:/mosquitto/config
      - ./mosquitto/data:/mosquitto/data
    networks:
      - iot-net
  zookeeper:
    image: confluentinc/cp-zookeeper:7.3.0
    container_name: kafka-zookeeper
    environment:
      ZOOKEEPER_CLIENT_PORT: 2181
      ZOOKEEPER_TICK_TIME: 2000     :contentReference[oaicite:2]{index=2}
    networks:
      - iot-net
  kafka:
```

```
    image: confluentinc/cp-kafka:7.3.0
    container_name: kafka-broker
    depends_on:
      - zookeeper
    ports:
      - "9092:9092"      # Kafka API
    environment:
KAFKA_BROKER_ID: 1
KAFKA_ZOOKEEPER_CONNECT: 'zookeeper:2181'
KAFKA_ADVERTISED_LISTENERS: PLAINTEXT://kafka-broker:9092
KAFKA_OFFSETS_TOPIC_REPLICATION_FACTOR: 1 :contentReference[oaicite:3]{index=3}
    networks:
      - iot-net
  mqtt-to-kafka:
    image: emqx/mqtt-to-kafka:latest
    container_name: mqtt-to-kafka-bridge
    depends_on:
      - mosquitto
      - kafka
    environment:
      MQTT_SERVER: tcp://mosquitto-broker:1883
      MQTT_TOPICS: 'city/sensors/#'
      KAFKA_BROKERS: 'kafka-broker:9092'
      KAFKA_TOPIC: 'iot.telemetry' :contentReference[oaicite:4]{index=4}
    networks:
      - iot-net
networks:
  iot-net:
    driver: bridge
//Телеметрія
const express = require('express');
const { Kafka } = require('kafkajs');
const mongoose = require('mongoose');
const app = express();
const port = 3000;
```

```
//підключення до MongoDB
mongoose.connect('mongodb://localhost:27017/telemetry', {
  useNewUrlParser: true,
  useUnifiedTopology: true,
});
//зберігання телеметрії
const telemetrySchema = new mongoose.Schema({
  topic: String,
  payload: Object,
  timestamp: Date,
});
const Telemetry = mongoose.model('Telemetry', telemetrySchema);
const kafka = new Kafka({
  clientId: 'telemetry-service',
  brokers: ['kafka-broker:9092'],
});
const consumer = kafka.consumer({ groupId: 'telemetry-group' });

const run = async () => {
  await consumer.connect();
  await consumer.subscribe({ topic: 'iot.telemetry', fromBeginning: true });
  await consumer.run({
    eachMessage: async ({ topic, partition, message }) => {
      const payload = JSON.parse(message.value.toString());
      const telemetry = new Telemetry({
        topic,
        payload,
        timestamp: new Date(),
      });
      await telemetry.save();
    },  });};

run().catch(console.error);
//REST API для отримання телеметрії
app.get('/api/telemetry', async (req, res) => {
```

```
  const data = await Telemetry.find().sort({ timestamp: -1 }).limit(100);
  res.json(data);
});
app.listen(port, () => {
  console.log(`Telemetry service listening at http:localhost:${port}`);
});
```

//Автентифікація

```
# auth_service/main.py
from fastapi import FastAPI, HTTPException, Depends
from fastapi.security import OAuth2PasswordBearer
from jose import JWTError, jwt
from datetime import datetime, timedelta
app = FastAPI()
oauth2_scheme = OAuth2PasswordBearer(tokenUrl="token")
# Секретний ключ та алгоритм для JWT
SECRET_KEY = "your-secret-key"
ALGORITHM = "HS256"
ACCESS_TOKEN_EXPIRE_MINUTES = 30
# Фіктивна база даних користувачів
fake_users_db = {
    "user1": {
        "username": "user1",
        "full_name": "User One",
        "hashed_password": "fakehashedpassword",
    }  }
def verify_password(plain_password, hashed_password):
    return plain_password == hashed_password  # Спрощена перевірка
def get_user(db, username: str):
    return db.get(username)
def authenticate_user(username: str, password: str):
    user = get_user(fake_users_db, username)
    if not user or not verify_password(password, user["hashed_password"]):
        return False
    return user
```

```
def create_access_token(data: dict, expires_delta: timedelta = None):
    to_encode = data.copy()
    expire = datetime.utcnow() + (expires_delta or timedelta(minutes=15))
    to_encode.update({"exp": expire})
    return jwt.encode(to_encode, SECRET_KEY, algorithm=ALGORITHM)
@app.post("/token")
async def login(form_data: dict):
    user = authenticate_user(form_data["username"], form_data["password"])
    if not user:
        raise HTTPException(status_code=400, detail="Incorrect username or password")
    access_token = create_access_token(data={"sub": user["username"]})
    return {"access_token": access_token, "token_type": "bearer"}
@app.get("/users/me")
async def read_users_me(token: str = Depends(oauth2_scheme)):
    try:
        payload = jwt.decode(token, SECRET_KEY, algorithms=[ALGORITHM])
        username = payload.get("sub")
        if username is None:
            raise HTTPException(status_code=401, detail="Invalid token")
    except JWTError:
        raise HTTPException(status_code=401, detail="Invalid token")
    user = get_user(fake_users_db, username)
    if user is None:
        raise HTTPException(status_code=404, detail="User not found")
    return user
```

```
// device-control-service/index.js
const express = require('express');
const mqtt = require('mqtt');
const app = express();
const port = 3001;
app.use(express.json());
const mqttClient = mqtt.connect('mqtt://broker.emqx.io');
mqttClient.on('connect', () => {
  console.log('Connected to MQTT broker');
```

```
});
// REST API для надсилання команд
app.post('/api/devices/:deviceId/command', (req, res) => {
  const { deviceId } = req.params;
  const { command } = req.body;
  const topic = `devices/${deviceId}/commands`;
  mqttClient.publish(topic, JSON.stringify({ command }));
  res.json({ status: 'Command sent' });
});
app.listen(port, () => {
  console.log(`Device control service listening at http://localhost:${port}`);
});
```

ДОДАТОК Б

(обов’язковий)

СХЕМАТИЧНЕ ЗОБРАЖЕННЯ МІСЬКОЇ СИСТЕМИ ІНТЕРНЕТУ РРЕЧЕЙ ДЛЯ ЗАБЕЗПЕЧЕННЯ ПІДКЛЮЧЕННЯ ТА ЗБОРУ ДАНИХ З РІЗНИХ ДЖЕРЕЛ

Модуль «Загальна зображення міської системи інтернету речей для підключення та збору даних даних з різних джерел».

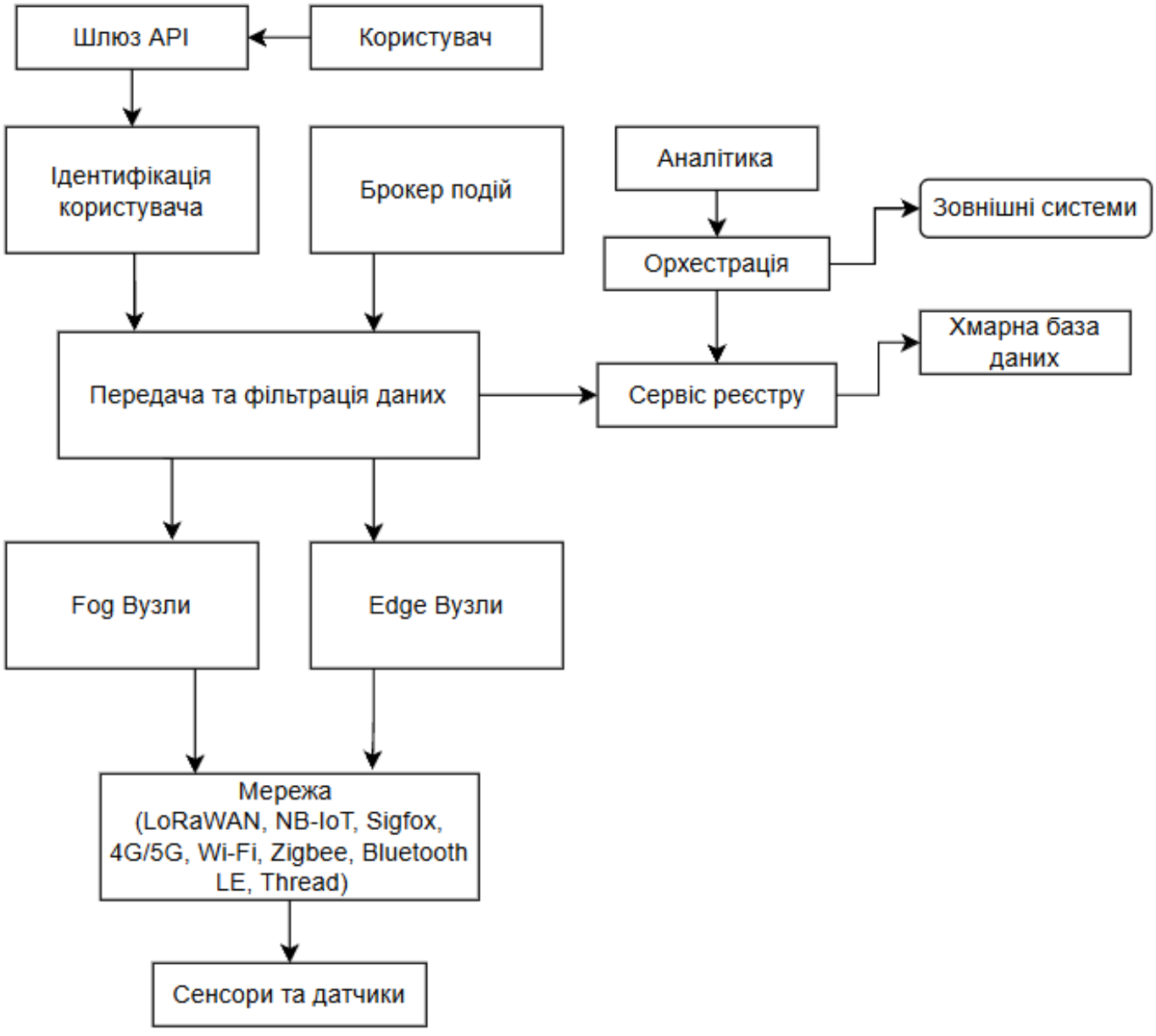


Рисунок Б - Загальне схематичне зображення системи підключення та збору даних даних з різних джерел

ДОДАТОК В

(обов’язковий)

КОПІЯ ТЕЗИ МАГІСТЕРСЬКОЇ РОБОТИ

Модуль « Теза “Міська система Інтернету речей для забезпечення підключення та збору даних з різних джерел”».

Вступ. Актуальність міської системи Інтернету речей (IoT) для забезпечення підключення та збору даних з різних джерел зростає у зв'язку з розвитком міських інфраструктур та технологій. Міська система IoT може бути інтегрована з іншими технологіями, такими як штучний інтелект (ШІ) та блокчейн. Ця система стає важливим інструментом для покращення якості життя мешканців, підвищення ефективності управління ресурсами та забезпечення стійкості міст [1, 2].

Міські системи IoT забезпечують ефективний моніторинг і управління такими ресурсами, як вода, електрика, газ та відходи. Інтеграція IoT у транспортні системи міст дозволяє покращити мобільність і зменшити затори. IoT-системи можуть підвищити рівень безпеки в містах. Збір даних з різних джерел дозволяє проводити детальний аналіз для ухвалення рішень. Міські IoT-системи сприяють стійкості міст у відповідь на виклики зміни клімату. IoT може сприяти активній участі громади в управлінні містом.

Актуальність міської системи Інтернету речей зумовлена її потенціалом для покращення якості життя мешканців, оптимізації ресурсів, підвищення безпеки та забезпечення стійкості міст. Інтеграція IoT у міське управління дозволяє здійснювати моніторинг і аналіз даних з різних джерел, що в свою чергу сприяє ефективному прийняттю рішень і забезпеченню сталого розвитку міст. У сучасному світі, що швидко змінюється, IoT стає невід'ємною частиною сучасних міських екосистем, що дозволяє містам адаптуватися до нових викликів та потреб населення. Відтак наразі актуальною є задача розроблення міської системи Інтернету речей для забезпечення підключення та збору даних з різних джерел.

Постановка задачі. Об’єкт дослідження – процес підключення та збору даних з різних джерел. Предмет дослідження – міська система Інтернету речей для забезпечення підключення та збору даних з різних джерел. Головна мета даного дослідження полягає в автоматизації забезпечення підключення та збору даних з різних джерел за допомогою міської системи Інтернету речей.

Основний матеріал. Сенсори Інтернету речей дозволяють в реальному часі відстежувати споживання води та електрики, виявляти витоки або ненормативні показники, що допомагає зменшити витрати. Смарт-системи можуть відстежувати заповненість сміттєвих контейнерів, що дозволяє оптимізувати маршрути вивезення сміття та зменшити витрати на утилізацію. Сенсори на дорогах і в громадському транспорті можуть збирати дані про трафік, що дозволяє оптимізувати маршрути, зменшити затори і скоротити час в дорозі. Збір даних про рух транспорту в реальному часі може бути використаний для інформування водіїв про затори, аварії або інші

перешкоди на маршруті. Відеокамери з функціями розпізнавання осіб і ситуацій можуть допомагати у забезпеченні громадської безпеки, виявляючи підозрілі активності в реальному часі. IoT може бути використано для швидкого сповіщення населення про надзвичайні ситуації, такі як стихійні лиха або інші загрози. Міські системи IoT можуть збирати величезні обсяги даних про різні аспекти міського життя, що дозволяє проводити глибокий аналіз і прогнозування потреб. Зібрані дані можуть використовуватися для оптимізації політики управління містом, покращення інфраструктури та підвищення якості послуг. Сенсори можуть вимірювати рівень забруднення повітря, температуру та інші екологічні параметри, що дозволяє вчасно вжити заходів для покращення екологічної ситуації. Інтелектуальні мережі дозволяють оптимізувати споживання енергії та зменшити викиди вуглецю, забезпечуючи більш ефективне використання ресурсів. Міські системи IoT можуть забезпечити платформи для збору думок та відгуків мешканців, що дозволяє покращити послуги та адаптувати управлінські рішення до потреб громади. За допомогою IoT можна реалізовувати ініціативи, спрямовані на поліпшення якості життя, такі як проекти з озеленення або організації місцевих заходів.

Міська система Інтернету речей (IoT) для забезпечення підключення та збору даних з різних джерел спрямована на інтеграцію сучасних технологій у міську інфраструктуру. Це дозволяє створити "розумне місто", де різні елементи взаємодіють між собою, забезпечуючи підвищення ефективності управління ресурсами, покращення якості життя мешканців та забезпечення стійкості до викликів сучасності. Метою і завданнями такої системи є: забезпечення інтеграції (підключення різних систем та датчиків для збору даних про стан міської інфраструктури (транспорт, енергетика, екологія, безпека)); оптимізація ресурсів (підвищення ефективності використання міських ресурсів, зокрема енергії, води та матеріалів, шляхом моніторингу та управління споживанням); покращення якості життя (створення більш комфортного, безпечного та екологічного середовища для мешканців).

Структура системи Інтернету речей (IoT) для забезпечення підключення та збору даних з різних джерел:

1) сенсори та пристрої:

- датчики – встановлення сенсорів для збору даних про різні параметри: температура, вологість, якість повітря, рівень шуму, рух транспорту, заповненість сміттєзвалищ тощо;

- камери – використання камер для моніторингу громадського порядку, дорожнього руху та екологічного стану;

2) комунікаційна інфраструктура:

- безпровідні мережі – розгортання бездротових технологій (Wi-Fi, LoRaWAN, NB-IoT) для передачі даних з сенсорів на центральні сервери;

- мобільні мережі – використання 4G/5G для забезпечення високошвидкісного доступу до даних і управлінських систем;

3) централізовані платформи:

- агрегація даних – збір і зберігання даних з різних джерел для подальшого аналізу;

- обробка даних – використання алгоритмів обробки та аналізу даних для виявлення патернів, аномалій і трендів;

- візуалізація – інтерфейси для представлення даних у зрозумілій формі для управлінців і мешканців.

Функціональні можливості міської системи Інтернету речей (IoT) для забезпечення підключення та збору даних з різних джерел:

1) моніторинг – спостереження в реальному часі за різними аспектами міського життя, включаючи якість повітря, трафік, енергоспоживання;

2) управління – автоматизація управлінських процесів на основі зібраних даних, наприклад, управління освітленням, системами поливу, маршрутами для збору сміття;

3) сповіщення – оповіщення мешканців про надзвичайні ситуації, наприклад, повені, пожежі або забруднення повітря;

4) залучення громади – збір думок мешканців щодо функціонування міських систем, що дозволяє залучати їх до процесів ухвалення рішень; створення можливостей для мешканців долучатися до екологічних проектів, програм з покращення міського середовища.

Висновки. Отже, було розроблено концепцію міської системи Інтернету речей, яка є потужним інструментом для трансформації міст у "розумні" екосистеми, що забезпечують ефективність, стійкість і якість життя мешканців. Завдяки інтеграції технологій IoT, містам вдається впоратися з викликами сучасності, такими як зростання населення, забруднення навколишнього середовища та потреба в ресурсах. У майбутньому такі системи стануть основою для розвитку інноваційних рішень у міському управлінні..

Список літератури

1. N. Abdullah et al., “IoT-Based Waste Management System in Formal and Informal Public Areas in Mecca”, *Int. J. Environmental Res. Public Health*, vol. 19, № 20, p. 13066, October 2022.

2. C.-W. Yau et al., “NB-IoT Coverage and Sensor Node Connectivity in Dense Urban Environments: An Empirical Study”, *ACM Trans. Sensor Netw.*, May 2022.

ДОДАТОК Г

(обов’язковий)

ПРЕЗЕНТАЦІЯ МАГІСТЕРСЬКОЇ РОБОТИ

Модуль «Презентація магістерської роботи ».

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
КВАЛІФІКАЦІЙНА РОБОТА МАГІСТРА

Міська система інтернету речей для забезпечення підключення та збору даних з різних джерел

Виконав: студент КІ2м-23-1 Юрій ФУРМАН
Науковий керівник: Катерина БЕРЕЗЬКА, к.т.н., доцент

Метою кваліфікаційної роботи магістра є розробка моделі міської системи джерел даних і збору та обробки інформації.

Об’єктом дослідження є процеси організації зв’язку, збору і передачі даних у міських системах інтернету речей, а також апаратно-програмні компоненти, що забезпечують інтеграцію сенсорних пристроїв у рамках єдиної платформ

Предметом дослідження є технічні засоби, протоколи та алгоритми, що забезпечують підключення сенсорних пристроїв, а також методи збору, маршрутизації й обробки телеметричних даних у міськійІоТ-системі.

Для розв’язання поставлених задач використовувалися методи аналізу наукової та технічної літератури з питань IoT і розумного міста, моделювання архітектури та компонентів системи

Актуальність

Актуальність даного дослідження обумовлена стрімкою урбанізацією та зростанням навантаження на міську інфраструктуру,що вимагає оперативного моніторингу та управління ресурсами в реальному часі.

Запровадження уніфікованої IoT-платформи дозволяє об’єднати гетерогенні сенсорні мережі, автоматизувати збір і агрегацію даних про транспорт, енергетику, екологію тощо, що сприяє підвищенню ефективності комунальних послуг і швидкому реагуванню на надзвичайні ситуації.

Інтернет речей

Інтернет речей - це концепція, яка передбачає інтеграцію фізичних об’єктів, пристроїв і сенсорів до глобальної мережі Інтернет для автоматизованого збору, обміну та обробки даних. Завдяки цьому звичайні об’єкти отримують можливість «спілкуватися» один з одним без безпосереднього втручання людини, що створює основу для побудови інтелектуальних систем управління та моніторингу.

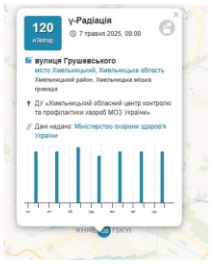
Основна ідея IoT полягає в обладнанні різноманітних предметів датчиками та комунікаційними модулями, завдяки яким вони можуть генерувати інформацію про своє оточення та передавати її в реальному часі.

Інтернет речей у розумному місті

У контексті міського управління Інтернет речей відіграє ключову роль, адже сприяє перетворенню традиційних міст у «розумні». Системи на базіIoT дозволяють оптимізувати використання ресурсів, підвищувати ефективність роботи транспортних мереж, енергопостачання та систем безпеки. Інтеграція з іншими технологіями, такими як штучний інтелект і блокчейн, відкриває можливості для проведення аналізу даних у режимі реального часу, що значно покращує якість управління міською інфраструктурою.

Огляд існуючих систем для розв'язання завдання

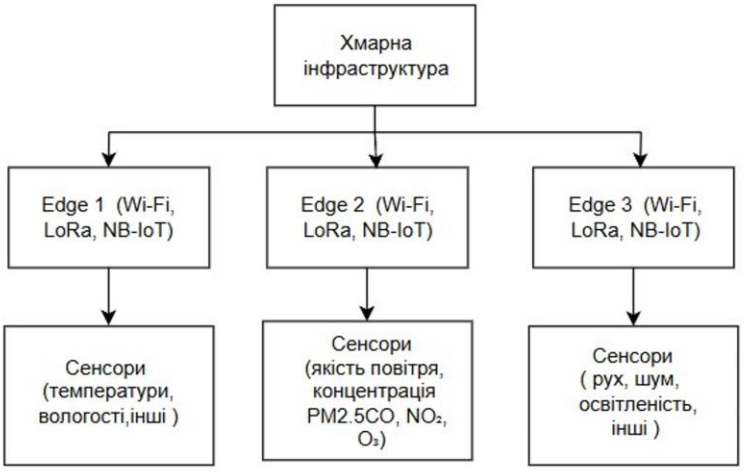
- RadAr це розподілена система радіаційного моніторингу в місті, що використовує мобільні детектори, приєднані до інтернету через статичні IP-адреси.
- EasyWay це платформа для відстеження міського транспорту та планування маршрутів у реальному часі. Збирає GPS-дані з автобусів, тролейбусів та трамваїв, відображає їх на інтерактивній мапі, допомагає користувачам обрати оптимальний маршрут.



Архітектура IoT-системи

Архітектура системи передбачає розподіл функцій між пристроями з різним рівнем обчислювальних можливостей. На фізичному рівні розташовуються кінцеві IoT-пристрої та сенсори, які відповідають за первинний збір даних. Далі дані передаються до edge-вузлів, що забезпечують локальну обробку інформації та знижують затримки при передачі даних до центральних серверів. Централізована обробка здійснюється у хмарних платформах, що дозволяє виконувати складний аналіз, зберігання даних та забезпечення інтеграції з іншими системами управління міською інфраструктурою.

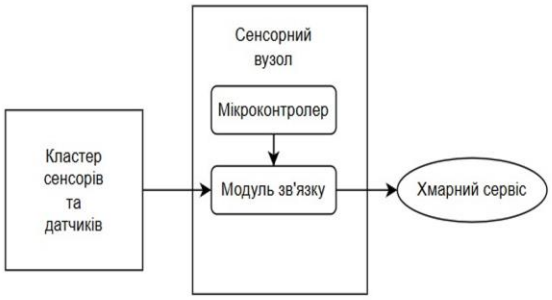
Загальна схема архітектури



Фізичний рівень збору даних

Фізичний рівень відповідає за безпосередній збір інформації із зовнішнього середовища шляхом розгортання мережі інтелектуальних сенсорних вузлів, кожен з яких перетворює фізичні величини на цифрові сигнали для передачі на вищі рівні обробки.

Основу цього шару становлять кластери датчиків, що згруповані за призначенням та типом вимірювань. Кожен сенсорний вузол містить вбудований контролер, що координує опитування підключених датчиків, виконує початкове фільтрування та кодування даних у стандартизовані повідомлення.



На фізичному рівні структурної моделі IoT-системи, що забезпечує моніторинг міської інфраструктури, розташовано фізичні сенсори та пристрої збору даних.

Ці компоненти відіграють ключову роль у первинному зборі інформації з навколишнього середовища і забезпечують основу для подальшої обробки даних. Для моніторингу обрано наступні типи фізичних сенсорів:

- датчики температури та вологості (DHT22);
- датчики якості повітря (PMS5003,);
- інфрачервоні сенсори руху(HC-SR312);
- акустичні сенсори шуму(CBD-0291).

Мікроконтролер використовуються для первинної обробки даних на пристроях, фільтрації сигналів та формування повідомлень для передачі.

Для системи можна використати мікроконтролер ESP32-S3 який підтримує Wi-Fi та Bluetooth для збору та передачі даних.

Периферійний (edge/fog) рівень

Периферійний рівень виконує ключову роль у розподіленій архітектурі міської IoT-системи, приймаючи дані від численних сенсорів і виконуючи на локальних вузлах попередню обробку, фільтрацію та агрегацію інформації. Завдяки тому, що алгоритми агрегації та аналітики розгортаються у безпосередній близькості до джерел даних, значно зменшується обсяг даних, який доводиться передавати до центральної хмарної інфраструктури.

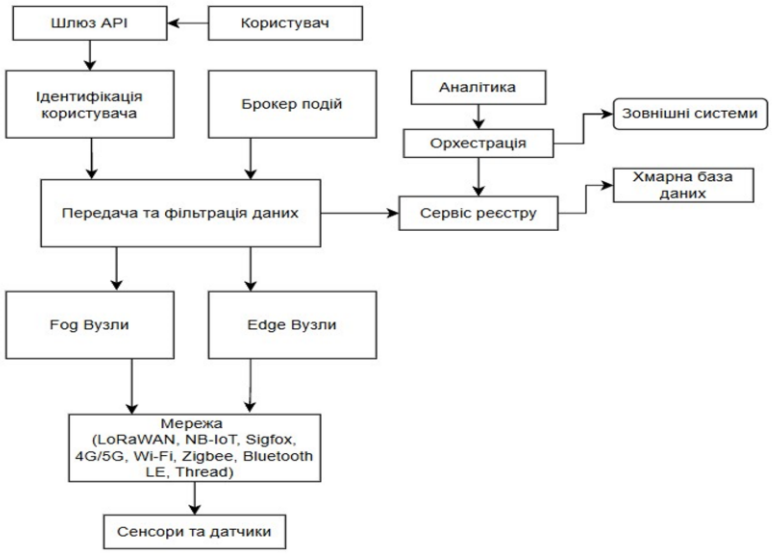
Використання edge-технологій дозволяє виконувати адаптивний відбір пріоритетних повідомлень, миттєво реагувати на критичні події (наприклад, аварійні сповіщення чи перевищення порогових значень), забезпечуючи тим самим підвищену надійність та стійкість системи загалом.

Хмарний рівень

У хмарному рівні здійснюється централізований збір і довгострокове зберігання телеметричних даних із сенсорів міської інфраструктури. Хмарні сервіси забезпечують виконання аналітичних алгоритмів, візуалізацію результатів та підтримку прийняття управлінських рішень, що базуються на отриманих даних.

Завдяки можливості централізованого доступу та масштабування обчислювальних ресурсів, хмарна інфраструктура підтримує оперативне прийняття управлінських рішень на основі актуальних даних і довготривалих тенденцій, що підвищує ефективність управління міською інфраструктурою .

Загальне схематичне зображення системи



Висновки

Отже, було розроблено концепцію міської системи Інтернету речей, яка є потужним інструментом для трансформації міст у "розумні" екосистеми, що забезпечують ефективність, стійкість і якість життя мешканців.

Завдяки інтеграції технологій IoT, містам вдається впоратися з викликами сучасності, такими як зростання населення, забруднення навколишнього середовища та потреба в ресурсах. У майбутньому такі системи стануть основою для розвитку інноваційних рішень у міському управлінні.

Міська система інтернету речей для забезпечення підключення та збору даних з різних джерел

Дякую за увагу!

Протокол аналізу звіту подібності експертом

Заявляю, що я ознайомився (-лась) з Повним звітом подібності, який був згенерований Системою виявлення і запобігання плагіату щодо роботи:

Автор: Юрій ФУРМАН

Співавтор:

Назва: Фурман_Міська система інтернету речей для забезпечення підключення та збору даних з різних джерел

Експерт:

Підрозділ: Кафедра комп'ютерної інженерії та інформаційних систем

Коефіцієнт подібності 1:13.5%

Коефіцієнт подібності 2:8.6%

Мікропробіли: 110

Заміна букв: 27

Інтервали: 0

Білі знаки: 1

Дата створення звіту: 2025-05-14 13:23:04.0

Після аналізу Звіту подібності констатую наступне:

☒ **Запозичення, виявлені в роботі є законними і не є плагіатом. Рівень подібності не перевищує допустимої межі. Таким чином робота незалежна і приймається.**

☐ **Запозичення не є плагіатом, але перевищено граничне значення рівня подібностей. Таким чином робота повертається на доопрацювання.**

☐ **Виявлено запозичення і плагіат або навмисні текстові спотворення (маніпуляції), як передбачувані спроби укриття плагіату, які роблять роботу невідповідною вимогам законодавства (Ст. 32. ЗУ Про вищу освіту, пункт 3.1, Ст. 42. ЗУ Про освіту) та вимог НАЗЯВО (Критерій 5), а також кодексу етики і процедурам. Таким чином робота не приймається.**

Обгрунтування:

2025-05-14 Доцент Андрій Нічепорук

Дата експерт

StrikePlagiarism.com

Дата звіту 5/14/2025
Дата редагування 5/14/2025

Документ прийнятий

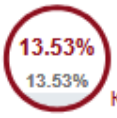
Звіт подібності

метадані

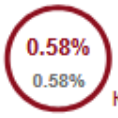
Назва організації
Khmelnytskyi National University
Заголовок
Фурман_Міська система інтернету речей для забезпечення підключення та збору даних з різних джерел
Автор
Юрій ФУРМАН Науковий керівник / Експерт
підрозділ
Кафедра комп'ютерної інженерії та інформаційних систем

Обсяг знайдених подібностей

Коефіцієнт подібності визначає, який відсоток тексту по відношенню до загального обсягу тексту було знайдено в різних джерелах. Зверніть увагу, що високі значення коефіцієнта не автоматично означають плагіат. Звіт має аналізувати компетентна / уповноважена особа.



КП 1



КЦ

25	21446	172863
Довжина фрази для коефіцієнта подібності 2	Кількість слів	Кількість символів

Тривога

У цьому розділі ви знайдете інформацію щодо текстових спотворень. Ці спотворення в тексті можуть говорити про МОЖЛИВІ маніпуляції в тексті. Спотворення в тексті можуть мати навмисний характер, але частіше характер технічних помилок при конвертації документа та його збереженні, тому ми рекомендуємо вам підходити до аналізу цього модуля відповідально. У разі виникнення запитань, просимо звертатися до нашої служби підтримки.

Заміна букв	ß	27
Інтервали	A→	0
Мікропробіли	¶	110
Білі знаки	ß	1
Парафрази (SmartMarks)	a	95

Wed May 14 13:44:15 EEST 2025, Медзатий Дмитро Миколайович, Хмельницький національний університет,

Anti-Plagiarism v-15.274 Educational

The maximum coincidence with one document 1.0%

Dictionaries check: en_US, ru_RU, ua_UA. **Errors in the documents: 11%**

ID: 241164 Title: МКР Міська система інтернету речей для забезпечення підключення та збору даних з різних джерел Added in a DB: 2025-05-14 Authors: Юрій ФУРМАН Heads: Катерина БЕРЕЗЬКА Consultants: Opponents:	Document		Sum coincidence on the DB	
	Symbols	Lexemes	Symbols	Lexemes
	151703	1147	2886 (2%)	35 (3%)

Plagiarism sources

ID	Description	Plagiarism presence in the document	
		Symbols	Lexemes

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

РЕЦЕНЗІЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

Дипломник: Фурман Юрій Олегович

Тема: Міська система Інтернету речей для забезпечення підключення та збору даних з різних джерел

Спеціальність: 123 «Комп'ютерна інженерія»

Обсяг кваліфікаційної роботи:

Кількість сторінок записки _____

1. Короткий зміст роботи та прийнятих рішень: Метою кваліфікаційної роботи є забезпечення підключення та збору даних з різних джерел шляхом розроблення відповідної системи Інтернету речей.

2. Висновок про відповідність роботи дипломному завданню: Робота повністю відповідає поставленому завданню.

3. Характеристика виконання кожного розділу, ступінь використання останніх досягнень науки і техніки і передових методів роботи: У розділі 1 проведений аналіз відомих методів та рішень. Розділ 2 кваліфікаційної роботи присвячений формалізації вимог до міської IoT-системи, архітектурним рішенням для ефективної інтеграції IoT, взаємодії IoT-компонентів, а також розробці структурної моделі системи збору даних. Розділ 3 кваліфікаційної роботи присвячений підходам до збору даних, методам агрегації даних, попередній обробці даних, а також інтеграції edge- та хмарних рішень для обробки даних. В розділі 4 кваліфікаційної роботи виконано проєктування загальної архітектури міської IoT-системи.

4. Позитивні сторони роботи: отримання наукових і практичних результатів

5. Негативні сторони роботи: недостатня увага реалізації IoT-системи

6. Оцінка графічного оформлення та пояснювальної записки роботи: Пояснювальна записка оформлена коректно, згідно з діючими стандартами оформлення документації.

7. Відгук про роботу в цілому: Робота виконана на середньому науково-технічному рівні.

8. Інші зауваження: _____

9. Оцінка дипломної роботи: добре/С (3.75).

Рецензент (прізвище, ім'я, по батькові, посада, місце роботи) Бармак О.В., д.т.н., проф., зав. каф. КН ХНУ

"__"_____2025 р. _____(підпис)

Завідувачу кафедри КІІС
доктору філософії, доценту
Ользі ПАВЛОВІЙ

Фурман Юрій Олегович

ПІБ здобувача вищої освіти

ФІТ, 2 курсу, групи КІ2м-23-1

ЗАЯВА

З правилами чинного Положення «Про систему забезпечення академічної доброчесності у Хмельницькому національному університеті» від 01.07.2022, згідно з яким виявлення плагіату є підставою для відмови в допуску кваліфікаційної роботи до захисту та застосування заходів дисциплінарної та академічної відповідальності, ознайомлений (а). Про використання програмно-технічних засобів для перевірки кваліфікаційних робіт здобувачів вищої освіти на плагіат оповіщений(а) та надаю свою згоду на обробку та збереження університетом моєї роботи в інституційному репозитарії університету.

Також надаю університету право на передачу моєї роботи для обробки та збереження в базах даних програмно-технічних засобів (StrikePlagiarism та Anti-Plagiarism) та використання роботи для виявлення плагіату в інших роботах, які перевіряються програмно-технічними засобами та користувачами, що мають доступ до цих програмно-технічних засобів, виключно в обмежених цілях для виявлення плагіату в текстах робіт.

Робота для перевірки університетом надається в друкованому та електронному варіанті. Електронна версія моєї роботи збігається (ідентична) з друкованою.

22 квітня 2025 року

РІШЕННЯ ЕКСПЕРТНОЇ КОМІСІЇ

КАФЕДРИ КОМП'ЮТЕРНОЇ ІНЖЕНЕРІЇ ТА ІНФОРМАЦІЙНИХ СИСТЕМ

ПРО ДОПУСК КВАЛІФІКАЦІЙНОЇ РОБОТИ ДО ЗАХИСТУ

Підтверджуємо ознайомлення з результатом звіту подібності щодо роботи, генерованого системою виявлення текстових збігів/ідентичності/схожості:

Назва: Міська система інтернету речей для забезпечення підключення та збору даних з різних джерел

Автор: Фурман Юрій Олегович

Спеціальність: 123 – Комп'ютерна інженерія

Освітня програма: освітньо-наукова

Науковий керівник: Березька Катерина Миколаївна, к.т.н., доцент

Після аналізу звіту подібності зроблено такий висновок:

№	Висновок	Позначка про відповідність
1	Запозичення, виявлені в роботі, є законними і не є плагіатом. Робота приймається до захисту.	відповідає
2	Виявлені запозичення не є плагіатом, розміщені в розділах, які не описують безпосередньо авторське дослідження, але кількість цитат перевищує обсяг, виправданий поставленою метою роботи Робота приймається до захисту, але має бути відкоригована. Відкоригований варіант має бути поданий на кафедру за 2 дні до захисту, разом із заявою щодо самостійності виконання письмової роботи та ідентичності друкованої та електронної версії роботи	
3	Виявлені запозичення не є плагіатом, але частково розміщені в розділах, які описують безпосередньо авторське дослідження, а кількість цитат перевищує обсяг, виправданий поставленою метою роботи. В зв'язку з цим мета роботи та поставлені завдання не були досягнені. Робота може бути допущена до захисту (наступного року) після того як буде відкоригована та допрацьована і успішно пройде повторну перевірку на академічний плагіат.	
4	Робота містить навмисні текстові спотворення, передбачувані спроби укриття запозичень або інші прояви академічного плагіату. Робота містить фабрикацію або фальсифікацію даних. Робота не допускається до захисту.	

Підтвердження:

Запозичення, виявлені в роботі, є законними і не є плагіатом, оскільки:

1) усі запозичення фрагментарні, або мають належним чином оформленні посилання;
2) окремі виявлені збіги є загальновживаними фразами або виразами, про що свідчить посилання системи на збіг з джерелами на один фрагмент речення;
3) в якості запозичень в окремих місцях системою зафіксовано послідовності чотирьохрозряних двійкових кодів, які є вхідними даними до великої кількості задач і не можуть розглядатися як об'єкт авторських прав і, відповідно, їх порушення;
4) всі зафіксовані системою ознаки модифікації тексту відносяться до комбінування латинських символів зі україномовними скороченнями індексів в формулах, що не є модифікацією тексту.

Сумарний обсяг всіх запозичень, визначений системою виявлення збігів/ ідентичності/схожості StrikePlagiarism, складає 13.5% і адресується до 108 першоджерела; та системою Anti-Plagiarism складає 8.6%, що, з урахуванням наведених обгрунтувань, відповідає характеру наукового дослідження і свідчить на користь кваліфікаційної роботи.

Керівник роботи _____ Катерина БЕРЕЗЬКА

Гарант ОП _____ Олег САВЕНКО

Завідувач кафедри КІІС _____ Ольга ПАВЛОВА