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 «Система постачання туманних послуг для керування даними Інтернету речей»

КвРКІП. 302181.23.02.33 ПЗ

Виконав: студент 2 курсу, група КІ2м-23-2 _____ Богдан КРИВІЦЬКИЙ
Підпис Ім'я, прізвище

Керівник д-р. техн. наук, професор _____ Сергій ЛИСЕНКО
Науковий ступінь, вчене звання Підпис Ім'я, прізвище

До захисту допускаю:
Зав. кафедри КІІС, доктор філософії, доцент
Ольга ПАВЛОВА
06 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

Богдану КРИВІЦЬКОМУ
Прізвище, ім'я, по батькові студента

1. Тема проекту (роботи) Система постачання туманних послуг для керування даними Інтернету речей

Керівник проекту (роботи) Сергій ЛИСЕНКО, д.т.н., професор
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)
Теоретичні основи досліджуваної проблеми
Модель системи постачання туманних послуг для керування даними інтернету речей
Система постачання туманних послуг для керування даними інтернету речей
Реалізація системи постачання туманних послуг для керування даними інтернету речей

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 прийняв
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)	Примітка
1	Вибір напряму дослідження та узгодження тематики КвРМ з керівником	01.09.2024	виконано
2	Ознайомлення з предметною областю; формулювання мети та задач дослідження; визначення об'єкта та предмета дослідження	01.10.2024	виконано
3	Робота над розділом 1 – аналіз відомих моделей, методів за темою; постановка задачі	01.11.2024	виконано
4	Робота над розділом 2 – розробка моделей для вирішення поставленої задачі	01.12.2024	виконано
5	Робота над науковою статтею	01.02.2025	виконано
6	Робота над розділом 3 – розробка методів для вирішення поставленої задачі	15.02.2025	виконано
7	Робота над розділом 4 – проектування та розробка ПЗ для вирішення поставленої задачі, експериментальна частина	01.04.2025	виконано
8	Оформлення пояснювальної записки згідно вимог	18.04.2025	виконано
9	Попередній захист ДРМ	29.04.2025	виконано
10	Захист ДРМ на засіданні ЕК	До 15.05.2025	

Студент _____ Підпис Богдан КРИВІЦЬКИЙ Ім'я, прізвище

Керівник роботи _____ Підпис Сергій ЛИСЕНКО Ім'я, прізвище

РЕФЕРАТ

Тема кваліфікаційної роботи магістра: Система постачання туманних послуг для керування даними Інтернету речей

Автор роботи: Кривіцький Богдан

Керівник роботи: Сергій Миколайович

Пояснювальна записка: 76 с., 15 рис., 9 табл., 2 дод., 90 джерел.

ІНТЕРНЕТ РЕЧЕЙ, ТУМАННІ ОБЧИСЛЕННЯ, ПОСТАЧАННЯ СЕРВІСІВ, ОБРОБКА ДАНИХ, СИСТЕМА КЕРУВАННЯ, СЕМАНТИЧНА ОБРОБКА, РОЗПОДІЛЕНІ СИСТЕМИ, ОБЧИСЛЮВАЛЬНІ РЕСУРСИ, ЗАТРИМКА ДАНИХ, ХМАРНІ ОБЧИСЛЕННЯ.

Об’єктом дослідження є процес оптимізації постачання туманних послуг для керування даними Інтернету речей.

Предметом дослідження є метод та система постачання туманних послуг для керування даними Інтернету речей.

Метою кваліфікаційної роботи магістра є оптимізація постачання туманних послуг для керування даними Інтернету речей.

Для розв’язання поставлених задач використовувалися методи:

- аналіз відомих методів оптимізації постачання туманних послуг для керування даними Інтернету речей;

- розробка моделі системи оптимізації постачання туманних послуг для керування даними Інтернету речей;

- розробка методу оптимізації постачання туманних послуг для керування даними Інтернету речей;

- здійснення дослідження методу оптимізації постачання туманних послуг для керування даними Інтернету речей.

Наукова новизна отриманих результатів:

- удосконалено метод оптимізації постачання туманних послуг для керування даними Інтернету речей, який на відміну від відомих здійснює ізоляцію дані залежно від їхнього призначення, а також виконує семантичну анотацію та

створення запитів до даних із додаванням контекстного значення, що оптимізує постачання туманних послуг для керування даними Інтернету речей;

- удосконалено систему оптимізації постачання туманних послуг для керування даними Інтернету речей.

Практична значимість отриманих результатів полягає у розробленому програмно-технічному засобі постачання туманних послуг для керування даними Інтернету речей.

Для розв’язання поставлених задач використовувалися методи забезпечення функціонування систем з IoT, методи математичного моделювання.

За темою кваліфікаційної роботи опубліковано тези у матеріалах конференції "Актуальні проблеми комп’ютерних наук АПКН-2024"

ЗМІСТ

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ

ПТС – постачальник туманних сервісів

IoT – Інтернет речей

API – Application Programming Interface (інтерфейс прикладного програмування)

MQTT – Message Queuing Telemetry Transport (транспорт телеметрії з використанням черг повідомлень)

HTTP – HyperText Transfer Protocol (протокол передачі гіпертексту)

HTTPS – HyperText Transfer Protocol Secure (захищений протокол передачі гіпертексту)

JSON – JavaScript Object Notation (нотація об’єктів JavaScript)

ВСТУП

Останні двадцять років відбулися значні зміни у тому, як люди використовують інформаційні технології. Мобільні пристрої пройшли шлях від компактних гаджетів із обмеженими можливостями до потужних міні-комп'ютерів. Також спостерігається поява Інтернету речей (IoT) як нової технологічної парадигми. IoT приніс численні переваги, покращивши якість життя та позитивно вплинувши на різні сфери застосування, зокрема промисловість, охорону здоров'я, екологію, транспорт, концепції розумних міст і розумних будинків тощо.

Одним із підходів до підвищення продуктивності IoT-систем стало зменшення навантаження на пристрої шляхом надання їм доступу до віддалених ресурсів у хмарі. Це призвело до появи парадигми хмарних обчислень. Проте, попри значні обчислювальні та сховищні можливості хмар, їхня віддаленість від користувачів ускладнює задоволення потреб IoT-додатків, чутливих до затримок. Подолати це обмеження перенесення сервісів із хмари ближче до краю мережі, що дало б початок парадигмі туманних обчислень (Fog Computing). Туманні обчислення дозволяють розміщувати обчислювальні ресурси поблизу IoT-пристроїв, забезпечуючи швидший доступ до сервісів і вирішуючи проблеми гетерогенності даних та їхньої сумісності.

Актуальність роботи полягає у необхідності підвищення ефективності управління IoT-даними шляхом впровадження адаптивної системи постачання туманних послуг.

Метою кваліфікаційної роботи магістра є оптимізація постачання туманних послуг для керування даними Інтернету речей.

Поставлена мета досягається розв'язанням таких основних завдань:

- проаналізувати відомі методи оптимізації постачання туманних послуг для керування даними Інтернету речей;

- розробити модель системи оптимізації постачання туманних послуг для керування даними Інтернету речей;

- розробити метод оптимізації постачання туманних послуг для керування даними Інтернету речей;

- здійснити дослідження методу оптимізації постачання туманних послуг для керування даними Інтернету речей.

Об’єктом дослідження є процес оптимізації постачання туманних послуг для керування даними Інтернету речей.

Предметом дослідження є метод та система постачання туманних послуг для керування даними Інтернету речей.

Наукова новизна отриманих результатів:

- удосконалено метод оптимізації постачання туманних послуг для керування даними Інтернету речей, який на відміну від відомих здійснює ізоляцію даних залежно від їхнього призначення, а також виконує семантичну анотацію та створення запитів до даних із додаванням контекстного значення, що оптимізує постачання туманних послуг для керування даними Інтернету речей;

- удосконалено систему оптимізації постачання туманних послуг для керування даними Інтернету речей.

Практична значимість отриманих результатів полягає у розробленому програмно-технічному засобі постачання туманних послуг для керування даними Інтернету речей.

Для розв’язання поставлених задач використовувалися методи забезпечення функціонування систем з IoT, методи математичного моделювання.

За темою кваліфікаційної роботи опубліковано тези у матеріалах конференції "Актуальні проблеми комп’ютерних наук АПКН-2024"

1 ТЕОРЕТИЧНІ ОСНОВИ ДОСЛІДЖУВАНОЇ ПРОБЛЕМИ

1.1 Особливості організації туманних послуг

1.1.1 Розуміння концепції Інтернет речей (IoT)

Системи Інтернету речей (IoT) з'єднують фізичний світ з мережею Інтернет [1]. По суті, IoT функціонує шляхом інтеграції реальних об'єктів в Інтернет, використовуючи датчики для збору даних та виконавчі механізми для впливу на навколишнє середовище. Фактично, системи IoT надають технології та інструменти для вимірювання, аналізу та управління фізичним світом [2].

Концепція Інтернету речей (IoT), вперше запропонована Кевіном Ештоном у 1998 році, описує сучасну технологічну тенденцію, яка передбачає об'єднання фізичних пристроїв, транспорту, будівель та інших об'єктів, оснащених датчиками та виконавчими механізмами, для збору та обміну інформацією [1]. Ці "речі" взаємодіють між собою, утворюючи велику систему, що надає нові, постійно доступні обчислювальні послуги. Цей новий підхід змінює наше повсякденне життя, але ці зручності та ефективність мають свою ціну. За прогнозами компанії Cisco, до 2020 року до мережі Інтернет буде підключено понад 50 мільярдів пристроїв [1]. Очікується, що до 2022 року в навколишнє середовище буде інтегровано 1 трильйон мережевих сенсорів, а через два десятиліття їх кількість зросте до 45 трильйонів [3]. В результаті, IoT генеруватиме величезні обсяги даних, що створить значне навантаження на інфраструктуру Інтернету. Тому дослідники активно шукають способи зменшення цього навантаження та вирішення проблеми аналізу та обробки великих даних [4].

Серед доступних рішень, хмарні обчислення стануть ключовим елементом майбутнього розвитку Інтернету речей, особливо з огляду на необхідність спільної роботи всіх підключених пристроїв. Зберігання та обробка даних безпосередньо на пристроях IoT є малоймовірними. Науковці вважають, що хмарні технології можуть служити сполучною ланкою між пристроями та додатками, приховуючи складність управління ресурсами та необхідними функціями [5]. Цей проміжний рівень вплине на майбутню розробку додатків, де збір, обробка та передача даних

створюватимуть нові проблеми. Інтеграція Інтернету речей у хмарне середовище може допомогти вирішити такі проблеми, як продуктивність, надійність, безпека та конфіденційність [6]. Хмара також полегшує обробку потоків даних IoT, забезпечуючи швидке та економічне налаштування та інтеграцію для складного аналізу даних та впровадження.

Основні переваги поєднання Інтернету речей з хмарними обчисленнями полягають у кількох аспектах. По-перше, комунікація: хмарні технології та IoT характеризуються широким застосуванням додатків та обміном даними. Завдяки IoT можлива реалізація повсюдних додатків із використанням автоматизованих засобів зв’язку для ефективної та економічної передачі й збору даних [1]. По-друге, зберігання: величезні обсяги даних, що генеруються мільярдами пристроїв, мають різні формати (структуровані та неструктуровані) та характеристики - різноманітність типів, великий обсяг та високу швидкість генерації. Хмарні рішення вважаються оптимальними для економічного та ефективного зберігання цих "великих даних" [7]. Нарешті, важливою перевагою є обчислювальні потужності: оскільки пристрої IoT мають обмежені обчислювальні ресурси, обробка даних вимагає передавання їх на потужні обчислювальні вузли. Хмарні платформи забезпечують практично необмежені обчислювальні можливості завдяки віртуалізації та моделі "ресурси на вимогу" [8].

Хоча хмарні технології вирішили багато проблем, пов'язаних з IoT, затримка, спричинена географічною віддаленістю хмари, може бути критичною для додатків, що вимагають швидкого реагування, таких як медицина, "розумні міста", "розумні будинки" та відеоспостереження. Для вирішення цієї проблеми виникла нова перспективна концепція, відома як туманні обчислення, або периферійні обчислення [4]. Ця парадигма передбачає обробку даних безпосередньо на периферійних пристроях, а не в хмарі чи віддалених центрах обробки даних. Дані з датчиків та інших підключених пристроїв обробляються миттєво на найближчому периферійному пристрої, наприклад, шлюзі, комутаторі чи маршрутизаторі, який виконує роль "туманного вузла" [9]. Для підтримки додатків, чутливих до часу, ці вузли повинні бути розташовані географічно розподілено. Зростання кількості

пристроїв та обсягу даних, які вони генерують, вимагає створення платформ та інфраструктури, що підтримують різноманітні вимоги IoT-систем. Крім того, локальна обробка даних з різнорідних пристроїв необхідна для зменшення навантаження на мережу та забезпечення ефективної роботи критично важливих додатків. Для цього потрібна уніфікована модель даних [9]. Платформи IoT повинні забезпечувати швидке підключення пристроїв, просте розгортання сервісів та ефективне управління даними. Еластичне надання ресурсів та масштабованість є ключовими вимогами для роботи з великою кількістю пристроїв та динамічними системами IoT [10].

1.1.2 Туманні послуги

Туманні обчислення – це платформа, яка об'єднує безліч різнорідних пристроїв з різними технологіями доступу до мережі (Wi-Fi, стільниковий зв'язок, Lo-Ra, Zigbee, Bluetooth) у децентралізованому середовищі, надаючи їм можливості інтелектуальної обробки даних. Завдання та сервіси виконуються в ізольованому середовищі, що дозволяє вирішувати такі проблеми, як затримки, географічний розподіл, підтримка мобільності, гнучке зберігання та обробка даних [11].

Туманні сервіси, розташовані на межі мережі, зазвичай взаємодіють з хмарними обчисленнями, формуючи трирівневу архітектуру. Ключова відмінність туманних обчислень від інших архітектур полягає в їхній близькості до кінцевих користувачів [12]. Дійсно, однією з головних переваг туманних обчислень є підтримка додатків, що вимагають швидкого реагування, завдяки розміщенню обчислювальних ресурсів на межі мережі, ближче до користувачів. Однак ефективне управління мережами IoT вимагає врахування як обмежень малопотужних пристроїв IoT, так і складності розгортання необхідної комунікаційної інфраструктури [10].

Традиційний підхід до сервісної моделі IoT передбачає надання програмному агенту необроблених даних з датчиків. Ці дані не мають семантичних анотацій і

вимагають значних ручних зусиль для практичного застосування. Така різноманітність створює проблеми з інтероперабельністю [13]. Через приватний характер цих сервісів, IoT перетворився на набір ізольованих вертикальних додатків без горизонтального зв'язку. Відсутність інтероперабельності між незалежними сервісами є загрозою для широкого впровадження IoT, особливо для додатків, які можуть використовувати дані з кількох пристроїв [14]. Для досягнення семантичної інтероперабельності в гетерогенному середовищі IoT необхідне семантичне анотування необроблених даних з використанням онтологій.

Зазвичай, функції семантичної моделі реалізуються централізовано в хмарі [8]. Хоча обчислювальні можливості хмари важливі, час відгуку не завжди відповідає вимогам додатків. Крім того, застосування алгоритмів семантичної анотації до великих обсягів даних на централізованому рівні призводить до значного споживання ресурсів і впливає на продуктивність. Тому необхідна високорівнева архітектура туманних обчислень для роботи з гетерогенними пристроями IoT, що генерують великі та різноманітні дані. Одночасно з цим, потрібно забезпечити єдину модель даних для підтримки інтероперабельності в IoT за допомогою технологій Semantic Web (SW) [15].

1.1.3 Виклики надання туманних послуг

Туманні обчислення – це нова парадигма, що розширює можливості хмарних обчислень для вирішення проблем додатків Інтернету речей (IoT) на периферії мережі. Хоча туманні обчислення мають спільні риси з хмарними, вони характеризуються унікальними особливостями [12]. Архітектура туманних обчислень забезпечує ефективну обробку даних, обчислення, зберігання, мережеві та прикладні сервіси для систем IoT [4]. Проте, окрім переваг, успадкованих від хмарних обчислень, туманні обчислення стикаються з новими викликами [16].

У туманних обчисленнях обчислювальні вузли є гетерогенними та розподіленими, що потребує вирішення проблем, пов'язаних з обмеженими ресурсами [17]. Проблеми туманних обчислень:

Гетерогенність в IoT проявляється як складна екосистема з безліччю різнорідних мереж [8]. Однією з перших проблем, з якою доводиться стикатися в цій сфері, є значна неоднорідність "речей". Ця неоднорідність охоплює кілька рівнів архітектури, від південного до північного шару. На рівні пристроїв різні об’єкти використовують різноманітні стандарти комунікаційних технологій для передачі даних зондування у різних форматах, які потім обробляються та використовуються для прийняття рішень вузлами Fog. Виникає питання, як ефективно керувати цими пристроями інтегровано. На мережевому рівні застосовуються різні топології, такі як зоряна, деревоподібна чи гібридна, що забезпечують передачу даних від джерел до вузлів обробки, тому необхідно розробити ефективну топологію для управління, забезпечення пропускної здатності та контролю енергоспоживання. На рівні обчислювальних вузлів ці вузли можуть мати різну архітектуру та платформи, призначені для розгортання масштабованих, інтелектуальних та інтероперабельних додатків Інтернету речей у різних сферах. Наприклад, у "розумному місті" системи, такі як "розумні будинки", "розумний рух", державні послуги чи "розумна промисловість", можуть генерувати різноманітну контекстну інформацію у різних форматах і специфікаціях, де спільне використання контексту могло б стати доцільним рішенням для забезпечення інтероперабельності в цих вузлах обробки. На прикладному рівні існуючі сервіси та додатки Інтернету речей часто виступають ізольованими вертикальними рішеннями, у яких усі компоненти системи тісно пов’язані з конкретним контекстом застосування, обладнанням чи програмним забезпеченням. У такому випадку ключовою проблемою є уніфікація платформ і проміжного програмного забезпечення, а також забезпечення інтероперабельності програмних інтерфейсів [11-13].

Керованість у цьому контексті розглядається як важливий аспект розгортання, масштабування та інтероперабельності уніфікованої архітектури, що намагається відповісти на питання про те, що таке керованість. Це найбільш практичний спосіб подолання розриву між організаційними та технологічними ізоляторами в IoT, який може застосовуватися в крос-платформних додатках, крос-

платформах чи крос-доменах. Доступність передбачає розробку механізмів для підтримки доступу до даних, що генеруються платформами чи додатками Інтернету речей, наприклад, через веб-сервіси типу RESTful API, які забезпечують доступ до сервісів усередині чи поза архітектурою з використанням механізмів автентифікації та авторизації. Управління послугами означає, що функціональні можливості можуть розроблятися як окремі послуги, які розгортаються незалежно на різних платформах, при цьому враховується механізм комунікації між сервісами для забезпечення внутрішньої та зовнішньої перехресної взаємодії. Слабо зв'язана архітектура, заснована на сервіс-орієнтованому підході з єдиним центром відповідальності та мінімальними залежностями, дозволяє командам працювати й розгортатися незалежно, що також відомо як мікросервіси. На основі стандартів необхідно враховувати управління даними для гетерогенних пристроїв IoT у розподіленій архітектурі. Замість відправлення даних у хмару, ресурси, розташовані близько до джерел даних, можуть використовуватися для локальної обробки, аналізу та швидкого прийняття рішень, що покращує якість надання послуг. Пристрої Інтернету речей є неоднорідними з точки зору протоколів зв'язку, форматів даних і технологій, що створює не лише проблеми інтероперабельності самих пристроїв, а й їхніх даних та семантики [14]. Тому для інтеграції цих джерел даних потрібна модель на основі стандартів, наприклад, RDF, JSON-LD чи онтології [6].

1.2 Особливості керування даними

Керування даними в системах і надання послуг у рамках концепції «Fog» (туманних обчислень) для Інтернету речей (IoT) є надзвичайно важливим і складним завданням [2]. Це зумовлено тим, що такі системи функціонують у розподіленому середовищі, де інформація надходить від величезної кількості різноманітних пристроїв, проходить обробку на проміжних рівнях інфраструктури та зрештою зберігається або в хмарних сховищах, або в локальних системах [10]. У цьому контексті виникають численні ключові виклики, серед яких забезпечення

високої продуктивності обробки даних, зменшення затримок у передачі інформації, гарантування надійного захисту даних і мінімізація обсягів інформації, що передається через мережу. Одним із основоположних принципів «Fog» обчислень є розподілена обробка даних, яка дозволяє виконувати аналітичні операції якомога ближче до джерел інформації – зокрема, на периферійних пристроях (Edge) та на проміжних Керування даними в системах і надання послуг у рамках концепції «Fog» (туманних обчислень) для Інтернету речей (IoT) є надзвичайно важливим і складним завданням [17]. Це зумовлено тим, що такі системи функціонують у розподіленому середовищі, де інформація надходить від величезної кількості різноманітних пристроїв, проходить обробку на проміжних рівнях інфраструктури та зрештою зберігається або в хмарних сховищах, або в локальних системах [14]. У цьому контексті виникають численні ключові виклики, серед яких забезпечення високої продуктивності обробки даних, зменшення затримок у передачі інформації, гарантування надійного захисту даних і мінімізація обсягів інформації, що передається через мережу. Одним із основоположних принципів «Fog» обчислень є розподілена обробка даних, яка дозволяє виконувати аналітичні операції якомога ближче до джерел інформації – зокрема, на периферійних пристроях (Edge) та на проміжних вузлах «Fog» [9]. Такий підхід до організації обробки даних має цілу низку суттєвих переваг, які роблять його незамінним у сучасних IoT-системах.

По-перше, розподілена обробка значно знижує навантаження на основні обчислювальні центри, такі як хмарні сервери. Це досягається за рахунок того, що в хмару передаються лише попередньо оброблені дані або ті, які мають найбільшу цінність для подальшого використання, а не весь необроблений потік інформації [12]. Така оптимізація дозволяє уникнути перевантаження центральних систем і підвищує їхню ефективність. По-друге, наближення обробки до джерел даних суттєво скорочує затримки в передачі інформації, що є критично важливим для IoT-застосунків, які працюють у реальному часі [18]. Наприклад, це стосується таких систем, як відеоспостереження, безпілотні транспортні засоби, медичні пристрої для моніторингу стану здоров’я чи індустріальні сенсори, де навіть мілісекундні затримки можуть мати серйозні наслідки [4]. По-третє, використання «Fog»

обчислень сприяє зменшенню витрат на пропускну здатність мережі, оскільки обсяг переданих даних оптимізується ще на етапі попередньої обробки, до того, як інформація потрапляє на вищі рівні інфраструктури [9].

Важливим аспектом управління даними в IoT-середовищі є локальна фільтрація та агрегація інформації, які дозволяють значно зменшити обсяг даних, що надсилається до центральних систем зберігання [18]. Багато пристроїв Інтернету речей генерують величезні потоки даних, однак лише невелика їхня частка є дійсно корисною чи необхідною для подальшого аналізу [4]. Наприклад, у системах розумного міста відеокамери можуть працювати в режимі безперервного запису протягом усього дня, але лише окремі фрагменти відеопотоку містять важливу інформацію, таку як виявлення порушень правил дорожнього руху чи інших інцидентів [10]. Завдяки застосуванню алгоритмів попередньої обробки на рівні «Fog» обчислень стає можливим аналізувати відеопотік у реальному часі безпосередньо на проміжних вузлах. Це дозволяє виділяти лише значущі кадри чи сегменти, які потім передаються до хмарних серверів або безпосередньо до систем прийняття рішень, замість надсилання повного обсягу запису [4]. Агрегація даних також відіграє ключову роль у зниженні навантаження на мережеву інфраструктуру. Наприклад, замість того, щоб передавати кожне окреме показання температурного сенсора, система може локально обчислювати середнє значення за певний період часу і відправляти лише ці усереднені дані [10]. Такий підхід особливо цінний у ситуаціях, коли ресурси мережі чи обчислювальних систем обмежені, наприклад, у мобільних або вбудованих IoT-пристроях.

У розподілених системах Інтернету речей організація ефективного зберігання даних є ще одним важливим завданням. Для цього зазвичай застосовується ієрархічний підхід, який дозволяє оптимізувати доступ до інформації, мінімізувати витрати на її обробку та підвищити швидкість виконання операцій [7]. На рівні периферійних пристроїв (Edge) зберігаються тимчасові дані, які необхідні для локальної обробки в реальному часі [3]. Наприклад, у системах моніторингу здоров’я інформація про життєві показники може накопичуватися безпосередньо на пристрої користувача, а передача до центрального сервера

відбувається лише у випадку виявлення аномалій, які потребують детальнішого аналізу [10]. На рівні «Fog» обчислень зберігаються середньострокові дані, які можуть бути використані для локального аналізу чи розширеної аналітики в межах певного вузла [11]. Наприклад, система управління розумними світлофорами може аналізувати потік транспорту на основі історичних даних, що зберігаються в локальному сховищі «Fog»-вузла, і на цій основі приймати рішення про коригування тривалості сигналів світлофора [10]. Хмарні системи (Cloud), у свою чергу, використовуються для довготривалого зберігання великих обсягів історичних даних, виконання глибокого аналізу, а також для застосування технологій машинного навчання з метою вдосконалення процесів і прогнозування [1].

Крім того, у «Fog» системах активно застосовується кешування даних, що забезпечує швидкий доступ до інформації, яка часто використовується, без необхідності постійно звертатися до віддалених серверів. Це особливо корисно для застосунків, де низька затримка є критично важливою, таких як безпілотні транспортні засоби чи системи моніторингу стану промислового обладнання [18]. Оскільки IoT-мережі можуть генерувати величезні обсяги даних, ключовим завданням є розмежування критичної та менш важливої інформації [7]. Наприклад, у медичних системах моніторингу дані про пульс, артеріальний тиск чи рівень кисню в крові повинні передаватися з мінімальними затримками, щоб забезпечити своєчасне реагування на загрози здоров'ю, тоді як менш термінові показники можуть оброблятися з певною затримкою [10]. Для цього застосовуються алгоритми управління трафіком, які дозволяють визначати пріоритети передачі даних залежно від їхньої значущості. Динамічне керування потоками також допомагає балансувати навантаження на «Fog»-вузли, запобігаючи перевантаженню мережі та забезпечуючи стабільну роботу системи [13].

Захист даних у «Fog» обчисленнях є одним із найсерйозніших викликів, адже система працює з великою кількістю розподілених вузлів, кожен із яких може стати потенційною точкою вразливості до кібератак [5]. Для забезпечення безпеки застосовуються різноманітні методи, такі як локальне шифрування даних перед

їхньою передачею в хмару, аутентифікація всіх підключених пристроїв і суворий контроль доступу до інформації [7]. Крім того, використовуються техніки анонімізації та агрегації даних, які дозволяють обробляти інформацію без прив’язки до конкретних користувачів, зберігаючи їхню конфіденційність [10]. Наприклад, у розумних містах дані про переміщення людей можуть аналізуватися в узагальненому вигляді для оцінки транспортних потоків, без ідентифікації окремих осіб [10]. Сучасні «Fog» обчислювальні системи також активно інтегрують алгоритми штучного інтелекту для динамічного керування ресурсами та оптимізації роботи [8]. Самоадаптивні системи здатні аналізувати поточне навантаження на «Fog»-вузли, прогнозувати можливі збої чи пікові навантаження та автоматично перерозподіляти ресурси для забезпечення безперебійного функціонування [13]. Застосування машинного навчання дозволяє прогнозувати поведінку IoT-мереж, виявляти аномалії в даних і навіть запобігати кібератакам [1]. Наприклад, у сфері кібербезпеки такі системи можуть ідентифікувати підозрілі патерни в поведінці пристроїв і нейтралізувати потенційні загрози ще до того, як вони призведуть до серйозних проблем [5].

Загалом, ефективне керування даними в «Fog» обчисленнях – це багатогранне завдання, яке охоплює оптимізацію процесів передачі, зберігання й обробки інформації, а також забезпечення її безпеки. Використання інтелектуальних алгоритмів і сучасних технологій дозволяє значно підвищити продуктивність і надійність IoT-інфраструктури, роблячи «Fog» обчислення невід’ємною частиною розвитку Інтернету речей у майбутньому [16].

1.3 Відомі методи постачання туманних послуг. Переваги та недоліки

1.3.1 Визначення RFID

Завдяки стрімкому розвитку мініатюризації та зменшенню вартості RFID, сенсорних мереж, NFC, бездротового зв'язку, технологій та відповідного програмного забезпечення, IoT сьогодні набув колосального значення як для промислових секторів, так і для звичайних користувачів. Найважливішою умовою

для функціонування IoT-додатків є забезпечення тотальної взаємодії [15]. Ці програми мають бути сумісні з широким спектром обладнання та протоколів комунікації. Це передбачає роботу як з крихітними сенсорами, що вимірюють показники навколишнього середовища, так із потужними серверами, що виконують аналіз та обробку отриманих даних [3]. Не менш важливим є узгодження мобільних і периферійних пристроїв, зокрема роутерів та "розумних" центрів управління, а також забезпечення взаємодії з кінцевими користувачами для можливості управління системою [15]. Збирання та аналіз даних за допомогою сенсорів дозволяють негайно реагувати на зміни в реальному світі [5]. Така взаємодіюча та реагуюча мережа відкриває неймовірні перспективи для пересічних громадян, споживачів та бізнесу загалом. Завдяки стрімкому прогресу в мініатюризації, здешевленню RFID, сенсорних мереж, NFC та бездротових технологій, Інтернет речей (IoT) міцно закріпився як в промисловості, так і серед користувачів [3]. Безперебійне з'єднання є ключем до функціонування IoT додатків, що вимагає підтримки широкого спектру пристроїв та комунікаційних протоколів [15]. Це охоплює як мініатюрні сенсори, здатні збирати та передавати інформацію про навколишнє середовище, так і потужні сервери, призначені для аналізу та обробки зібраних даних [3]. Необхідна інтеграція мобільних пристроїв, периферійного обладнання, наприклад роутерів та інтелектуальних хабів, а також участі людей як контролерів [15]. Відстеження фізичного стану об'єктів через датчики та аналіз детальних даних сприяє миттєвій реакції на зміни в реальному світі [5]. Ця комплексна інтерактивна система відкриває безліч перспектив для громадськості, кінцевих споживачів та підприємств [3].

На зорі розбудови Інтернету речей (IoT) технологія RFID вперше привернула до нього пильну увагу. І до сьогодні RFID зберігає ключову роль у розвитку IoT - додатків, бо сама ідея з’єднання всього до Мережі легко втілюється завдяки RFID-міткам [15]. Ці мітки – це компактні мікрочипи, які містять унікальний ідентифікатор та можливість програмування, інтегровані з антеною для бездротового обміну даними зі зчитувачами RFID [15]. Однак з подальшим розвитком технологій, бездротові сенсорні мережі (WSN) та пристрої з підтримкою

Bluetooth значно пришвидшили поширення IoT [5]. Підхід WSN є багатообіцяючою технологією, пов'язаною з розвитком IoT [5]. Попри те, що WSN та IoT можна вважати конкурентами, вони мають багато спільних рис. Сенсорні мережі в WSN складаються з бездротовосполучених датчиків, які мають здатність взаємодіяти один з одним [5]. Стандартним підходом до комунікації в WSN є технологія «точка-точка» (P2P) [15]. Датчики повинні виявляти сусідні датчики та формувати мережу, яка взаємодіє у межах географічно-розподіленої сенсорної мережі [15]. Типовим прикладом такого застосування є використання бездротових датчиків у відділеннях невідкладної допомоги лікарні Джона Гопкінса для моніторингу рівня кисню в крові та частоти серцевих скорочень пацієнтів в реальному часі [17]. У цьому випадку датчики надсилають сигнал тривоги або на проміжне програмне забезпечення для перевірки невідкладної ситуації, або напряму медичному працівнику [17]. Ці технології комунікації та приклади їх використання в IoT активно досліджуються останніми роками [3]. Проте, специфічні характеристики та вимоги, як-от масштабованість, підтримка гетерогенності, повна інтеграція та обробка запитів в режимі реального часу, наразі потребують більшої уваги [18].

1.3.2 Комунікаційні протоколи

Інтернет речей (IoT) організовує обмін інформацією між різними пристроями та користувачами за допомогою різноманітних протоколів зв'язку [15]. За результатами аналізу, ці протоколи можна поділити на три ключові архітектурні категорії, враховуючи різні критерії: ієрархію моделі OSI, стандарти IEEE 802 або типи мереж [15]. Протоколи, що працюють згідно з ієрархією OSI, зазвичай функціонують на прикладному, мережевому та фізичному рівнях. На прикладному рівні використовуються такі протоколи, як CoAP, ISA100.11a, MQTT, SOAP, Web Socket та інші [15].

Мережевий рівень включає набори протоколів SMS/USSD, TCP/UDP, а також протоколи бездротового зондування [15]. На фізичному рівні застосовуються

протоколи ближнього, дальнього стільникового та дальнього нестільникового зв'язку [15].

Протоколи, що базуються на стандарті IEEE 802, включають Wi-Fi, Bluetooth, ZigBee, UWB та інші [15]. Розподіл за типами мереж поділяє IoT-протоколи на чотири основні категорії: персональна мережа (PAN), локальна мережа (LAN), глобальна мережа (WAN) та мобільна мережа [15]. Такий розподіл необхідний для вирішення питань зв'язку на кожному рівні архітектури IoT [15]. Проте, забезпечення взаємодії між різними комунікаційними протоколами залишається значним викликом [15].

В IoT також важливо враховувати протоколи зв'язку з точки зору побудови топології, енергоспоживання, оптимізації, інтеграції та криптографічного захисту [11]. З позиції мереж і комунікацій, IoT можна розглядати як взаємодію різноманітних мереж, зокрема, мобільних (CDMA, 3G/4G/5G і т.д.), WLAN, WSN та MANET [15]. Безперебійний взаємозв'язок є критичним компонентом успішного функціонування IoT [15]. Швидкість, надійність та стабільність мережевого зв'язку безпосередньо впливають на загальний досвід користування IoT-системами [9]. З розвитком високошвидкісних мобільних мереж, таких як 5G, і поширенням протоколів локальних та міських мереж, таких як Wi-Fi, Bluetooth і WiMax, створення взаємопов'язаної мережі об'єктів здається цілком можливим, але робота з численними протоколами, що забезпечують зв'язок між цими середовищами, залишається складною задачею [15]. Засоби комунікації та протоколи залежать від специфікацій пристрою (процесор, об'єм пам'яті, накопичувачі, тривалість роботи від батареї) [15]. Однак, серед найпоширеніших протоколів та стандартів виділяють: RFID (наприклад, серію ISO 18000, яка включає класи файлів і два покоління і охоплює як активні, так і пасивні RFID-мітки) [15]. IEEE 802 WLAN (802.11), Zigbee (802.15.4), Near Field Communication (NFC), Bluetooth (802.15.1) [15]. Стандарти малопотужних бездротових персональних мереж (6LoWPAN) від IETF [15]. M2M протоколи, такі як MQTT і CoAP [15]. Технології IP-рівня, такі як IPv4, IPv6 [15].

1.3.3 Метод function as a service (FAAS)

Хмарні технології невпинно прогресують, еволюціонуючи від базових сервісів, таких як Amazon Web Services (AWS) EC2, до комплексних екосистем із спеціалізованих високорівневих послуг [10]. Початкові хмарні рішення типу "Інфраструктура як послуга" (IaaS) були універсальними системами, що забезпечували лише базову абстракцію обчислювальних ресурсів, найчастіше у вигляді віртуальних машин, якими користувачі керували самостійно [10]. На противагу цьому, нова "безсерверна" концепція прагне повністю усунути операційні клопоти, такі як адміністрування чи масштабування серверів, пропонуючи керовану високорівневу службу з детальною оплатою за використання [10]. Серед спеціалізованих "безсерверних" послуг можна виділити як просте зберігання об’єктів (наприклад, Amazon S3), так і розмовні агенти на основі глибокого навчання (наприклад, Amazon Lex, що лежить в основі Alexa) [10].

Щоб пов’язати різні сервіси між собою потрібен універсальний "безсерверний" інструмент, який виступає "сполучною ланкою", усуваючи розбіжності (у тригерах, форматах даних тощо) між компонентами [2]. Саме для цього з’явилися платформи "Функція як послуга" (FaaS), такі як AWS Lambda, які стали основним рішенням у цій сфері [2].

У рамках FaaS розробники створюють невеликі фрагменти коду (зазвичай на JavaScript або Python) у вигляді функцій із чітко визначеним інтерфейсом [2]. Ці функції активуються подіями-тригерами, такими як HTTP-запити чи додавання даних до сховища [2]. Хмарний постачальник запускає ці функції на вимогу (з подією як вхідними даними), автоматично регулюючи віртуалізовані ресурси для обробки змінних навантажень із різною одночасністю [2].

FaaS застосовується для численних цілей: від ролі "сполучної ланки" у великих "безсерверних" програмах до створення серверної частини для REST-сервісів, а також для задач аналітики даних і машинного навчання [2]. Відтак їхня ефективність є критично важливою для роботи широкого кола хмарних застосунків [2].

1.3.4 Метод peer-to-peer (p2p)

Peer-to-peer, або P2P, - це концепція, яка в буквальному перекладі означає "від людини до людини" чи "від рівного до рівного" [15]. У контексті сучасних технологій і фінансів вона описує децентралізовану модель взаємодії, де учасники обмінюються ресурсами, інформацією чи послугами напряму один з одним, минаючи традиційних посередників, таких як банки, компанії чи централізовані платформи [15]. Ця ідея зародилася в інформаційних технологіях, зокрема в мережевій архітектурі, де комп'ютери могли обмінюватися даними без центрального сервера, як це було, наприклад, у перших файлообмінних мережах типу Napster чи BitTorrent [15]. Згодом концепція перекочувала в інші сфери, зокрема у фінансовий сектор, де вона отримала нове життя у вигляді P2P-кредитування [15].

У фінансовому світі P2P-кредитування - це система, яка дозволяє звичайним людям або бізнесам позичати і надавати гроші один одному через онлайн-платформи [15]. Замість того, щоб звертатися до банку за кредитом чи вкладати гроші в депозит, люди можуть напряму домовитися про позику: один стає позичальником, інший - інвестором [15]. Такі платформи, як LendingClub чи Prosper у США або Funding Societies у Південно-Східній Азії, виступають лише як посередники в технічному сенсі - вони з'єднують сторони, перевіряють дані, беруть на себе частину ризиків і стягують комісію за свої послуги [15]. Наприклад, позичальник може подати заявку на кредит, вказавши суму й мету, а інвестори вирішують, чи хочуть вони профінансувати цю позику, отримуючи відсотки як дохід [15]. Це дає змогу обом сторонам уникнути складних бюрократичних процедур традиційних фінансових установ, а інвесторам - отримати вищий дохід, ніж від банківських вкладів, хоча й із більшим ризиком [15].

Історія P2P-кредитування почалася на початку 2000-х років, коли перші платформи, такі як Zopa у Великобританії (запущена в 2005 році), запропонували альтернативу банківським послугам [15].

1.4 Висновки до першого розділу

У першому розділі досліджено теоретичні засади побудови систем постачання туманних послуг для керування даними Інтернету речей. Розглянуто етапи еволюції інформаційних технологій, зосереджено увагу на зростаючих вимогах до швидкості, обсягу та ефективності обробки даних, що генеруються пристроями IoT. Наведено огляд концепцій хмарних та туманних обчислень, виявлено їх ключові характеристики, переваги та обмеження.

Особливу увагу приділено архітектурним підходам до побудови систем IoT, зокрема з використанням багаторівневої структури з розподілом функціональності між хмарою, туманом та пристроями. Проаналізовано протоколи обміну даними, що відповідають різним рівням моделі OSI, та показано їхнє значення для реалізації надійної комунікації в умовах розподіленого середовища. У результаті аналізу окреслено основні проблеми централізованого підходу до керування IoT-даними, що обґрунтовують необхідність впровадження туманних сервісів.

Сформульовані у розділі теоретичні положення стали основою для подальшого аналізу існуючих моделей та розробки власної архітектури системи постачання туманних послуг.

1.5 Постановка задачі

У сучасному світі Інтернет речей (IoT) відіграє ключову роль у забезпеченні взаємодії між фізичними пристроями, що генерують величезні обсяги даних у реальному часі. Ці дані потребують швидкої обробки, ефективного зберігання та надійного керування, що створює значні виклики для традиційних хмарних обчислень через затримки в передачі даних, високе навантаження на мережу та вимоги до низької латентності. У цьому контексті технологія туманних обчислень (Fog Computing) виступає перспективним рішенням, яке дозволяє розподіляти обчислювальні ресурси ближче до джерел даних, зменшуючи залежність від централізованих хмарних систем і підвищуючи швидкість обробки інформації.

Однак впровадження туманних обчислень для керування даними IoT стикається з проблемами, такими як складність інтеграції різнорідних пристроїв, забезпечення безпеки даних, оптимізація розподілу ресурсів та адаптація до динамічних умов роботи мережі.

Метою даної магістерської роботи є розробка системи постачання туманних послуг для ефективного керування даними Інтернету речей. Для досягнення цієї мети необхідно вирішити низку завдань, які включають аналіз особливостей обробки даних IoT у туманному середовищі, проектування архітектури системи, що враховує розподіленість і гетерогенність пристроїв, розробку механізмів координації між туманними вузлами та хмарними серверами, а також створення алгоритмів для оптимізації обробки й передачі даних. Окремим аспектом є забезпечення безпеки та конфіденційності інформації, що обробляється в системі, враховуючи вразливості, притаманні розподіленим мережам. Крім того, необхідно врахувати адаптивність системи до змін у навантаженні та потребах користувачів, щоб гарантувати її стабільну роботу в реальних умовах.

Таким чином, задачею дослідження є створення теоретично обґрунтованої та практично реалізованої системи постачання туманних послуг, яка забезпечить ефективне керування даними IoT. Це передбачає не лише розробку програмно-апаратного комплексу, але й оцінку його продуктивності через моделювання та експериментальну верифікацію на основі реальних сценаріїв використання. У результаті має бути запропоновано рішення, яке дозволить оптимізувати обробку даних Інтернету речей у туманному середовищі, підвищити швидкість реакції системи та знизити залежність від централізованих обчислень, що сприятиме розвитку IoT-технологій у різних галузях, таких як розумні міста, промислова автоматизація та охорона здоров’я.

2 МОДЕЛЬ СИСТЕМИ ПОСТАЧАННЯ ТУМАННИХ ПОСЛУГ ДЛЯ КЕРУВАННЯ ДАНИМИ ІНТЕРНЕТУ РЕЧЕЙ

2.1 Загальні принципи моделювання туманної системи

Побудова ефективної моделі системи постачання туманних послуг для керування даними Інтернету речей потребує врахування низки міждисциплінарних підходів, які поєднують принципи розподілених обчислень, мережевих технологій, системної інженерії та семантичної обробки інформації [17]. Моделювання в такому контексті не зводиться лише до опису структурної схеми компонентів. Йдеться про формування цілісної концепції, в якій взаємодіють множинні рівні інформаційних потоків, обчислювальних процесів і логіки управління ресурсами.

Основою побудови туманної системи виступає принцип багаторівневої обробки, за якого обчислювальні завдання розподіляються між рівнем збору даних, проміжним обчислювальним рівнем та хмарною інфраструктурою відповідно до складності, терміновості та обсягу оброблюваних даних [30]. Такий підхід дозволяє наблизити обробку до місця генерації даних, що, у свою чергу, забезпечує зменшення затримок, зниження навантаження на канал передачі інформації, а також зменшення залежності від централізованих обчислювальних потужностей, як показано на рисунку 2.1 [20].

У класичних IoT-архітектурах, де усі обчислення та аналітика реалізуються виключно в хмарі, спостерігається надмірне навантаження на мережу, зростання часу відгуку та ризик перевантаження інфраструктури. Туманні обчислення вирішують ці проблеми за рахунок попередньої обробки даних на вузлах, розташованих ближче до пристроїв, що генерують ці дані [18].

$$Dout = f(Dedge, Dfog, Dcloud), \tag{2.1}$$

де:

D_{edge} - оброблені дані на рівні пристроїв збору,

D_{fog} - дані після попередньої обробки на проміжних вузлах,

D_{cloud} - остаточно оброблені та проаналізовані дані у хмарі.

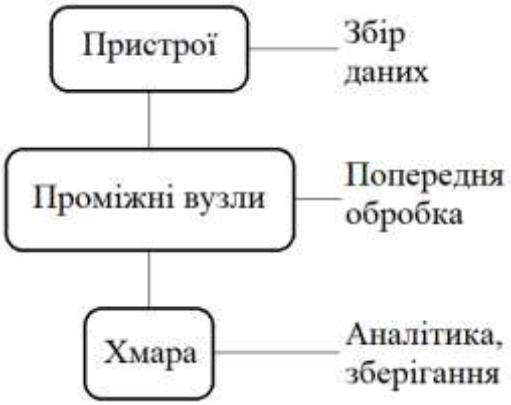


Рисунок 2.1 – Багаторівнева обробка даних у системі Fog-IoT

З методологічної точки зору моделювання туманної архітектури повинно враховувати як фізичні аспекти (апаратна реалізація вузлів, типи сенсорів, канали зв'язку), так і логічні структури (формати даних, семантичні зв'язки, протоколи взаємодії) [13, 46]. Також необхідно передбачати гнучкість конфігурації моделі, що дозволить масштабувати її як у горизонтальному напрямі (шляхом додавання нових вузлів збору або обробки даних), так і у вертикальному (через ускладнення обчислювальних алгоритмів або збільшення обчислювальної потужності проміжного рівня) [14, 25].

Важливою характеристикою моделі є її здатність до динамічної адаптації до змін у середовищі. Йдеться про такі явища, як зміна кількості активних пристроїв, коливання обсягів трафіку, зміни у топології мережі або варіативність якості зв'язку. Ефективна туманна модель повинна забезпечувати стабільне функціонування навіть в умовах непередбачуваного навантаження [42].

Механізм адаптації системи можна описати за допомогою коефіцієнта адаптивності:

$$\alpha = \frac{\mathrm{Cavailable}}{\mathrm{Crequired}}, \tag{2.2}$$

де:

$C_{available}$ - доступна обчислювальна потужність вузла,

$C_{required}$ - необхідна потужність для поточного навантаження.

Реалізація механізмів пріоритезації повідомлень, динамічного масштабування ресурсів проміжних вузлів, а також застосування буферизації й кешування даних дозволяє підвищити стійкість моделі [35].

Ключовою особливістю туманної архітектури є можливість виконання не лише синтаксичної, але й семантичної обробки даних на проміжному рівні. Це передбачає збагачення повідомлень контекстною інформацією - наприклад, зазначенням типу сенсора, місця вимірювання, одиниць виміру або інших параметрів, що забезпечують повноцінну інтерпретацію даних у подальшій обробці, приклади наведено в таблиці 2.1 [27, 49].

Таблиця 2.1 – Приклади контекстного збагачення повідомлень

Атрибут повідомлення	Приклад значення
Тип пристрою	Сенсор температури
Геолокація	Широта: 50.4501, Довгота: 30.5234
Одиниці виміру	°C
Часова мітка	2025-01-23T12:30:00Z

Інтеграція легковагового протоколу обміну повідомленнями публікації–підписки доцільна з точки зору ефективності передачі інформації в середовищах із обмеженими ресурсами. Такий протокол дозволяє працювати в умовах нестабільного з'єднання та мінімізувати затримки, що особливо важливо в реальному часі [43].

Під час моделювання архітектури важливо правильно розподіляти функціональність між різними рівнями, як це відображено в таблиці 2.2 [24, 33].

Таким чином, ефективна модель системи постачання туманних послуг повинна поєднувати архітектурну простоту з гнучкістю, підтримувати масштабування, динамічну адаптацію до змін середовища та забезпечувати високий рівень обробки даних на кожному етапі - від збору до аналітики.

Дотримання цих принципів створює фундамент для розробки продуктивної, надійної та масштабованої системи керування даними Інтернету речей [20].

Таблиця 2.2 – Розподіл функцій між рівнями

Рівень	Основні функції
Edge-рівень	Збір, попередня обробка, форматування даних
Fog-рівень	Фільтрація, агрегація, семантичне збагачення
Cloud-рівень	Довготривале зберігання, аналітика, візуалізація

2.2 Функціональні компоненти моделі

Формування моделі системи постачання туманних послуг для керування даними Інтернету речей передбачає чітке розмежування функціональних ролей між її основними рівнями. Кожен рівень - початковий рівень збору даних, проміжний обчислювальний рівень та хмарна інфраструктура - виконує специфічні завдання, які у своїй сукупності забезпечують повний цикл обробки, транспортування, збереження та представлення даних [17]. Від правильного визначення функціональних меж між цими рівнями залежить ефективність роботи всієї системи, її здатність до масштабування, адаптивність та продуктивність у динамічному середовищі [31].

Початковий рівень (рівень збору даних) представляє найнижчий рівень моделі, де відбувається безпосередня взаємодія з фізичним середовищем. До цього рівня належать сенсорні модулі та обчислювальні пристрої малої потужності, які здійснюють вимірювання, збір первинних даних та їх базову обробку [41]. Основним завданням пристроїв цього рівня є генерація даних з відповідною періодичністю, їх форматування у стандартний вигляд, наприклад у структурі типу JSON, та передача на наступний рівень через мережевий інтерфейс [38].

Важливими характеристиками пристроїв збору даних є енергоефективність, висока швидкість вимірювання та стабільність зв'язку з проміжним обчислювальним рівнем. У розробленій моделі для реалізації цього рівня

використовуються універсальні мікроконтролерні платформи із підключеними сенсорними модулями для вимірювання фізичних параметрів навколишнього середовища. Дані передаються за допомогою протоколу публікації–підписки, що проілюстровано на рисунку 2.2 [45].

Рисунок 2.2 – Взаємодія на рівні збору даних

Проміжний обчислювальний рівень виконує роль посередника між пристроями збору даних і хмарною інфраструктурою. На цьому рівні здійснюється прийом повідомлень, їх фільтрація, перевірка на достовірність, агрегація за певний період часу, а також семантичне збагачення даних [18].

Ці функції реалізуються на обчислювальних модулях середньої потужності, здатних виконувати попередню обробку даних без потреби у потужностях повноцінних серверів. Практична реалізація проміжного рівня включає встановлення брокера повідомлень та середовища розробки потоків обробки даних [39]. Адаптивна обробка повідомлень передбачає виконання функцій, наведених у таблиці 2.3 [40]

Хмарна інфраструктура виконує функції довготривалого зберігання даних, аналітики та візуалізації. Після завершення попередньої обробки на проміжному рівні дані надходять до спеціалізованої системи зберігання часових рядів. Така система оптимізована для роботи з потоковими даними, забезпечуючи високу швидкість запису та зчитування інформації [25, 34].

Таблиця 2.3 – Основні функції проміжного обчислювального рівня

Функція	Опис
Фільтрація	Видалення некоректних або дубльованих повідомлень
Агрегація	Обчислення середніх значень за інтервалами часу
Семантичне збагачення	Додавання контекстної інформації до даних

Візуалізація даних здійснюється за допомогою інструментів побудови дашбордів, які дозволяють створювати адаптивні графічні представлення залежно від типу пристроїв, параметрів вимірювання та часових інтервалів, що узагальнено на рисунку 2.3.

Рисунок 2.3 – Загальна логіка обробки та передавання даних

Таким чином, кожен рівень виконує обмежену, але критично важливу для функціонування системи роль. Завдяки чітко структурованому розподілу обов'язків і злагодженій взаємодії між компонентами досягається стабільність, продуктивність і гнучкість архітектури.

Модель придатна для масштабування і адаптації до специфіки різних сценаріїв застосування - від моніторингу навколишнього середовища до автоматизації промислових процесів.

2.3 Обмін даними та семантична уніфікація

Обмін даними між компонентами системи є ключовим процесом, що забезпечує її цілісне функціонування та узгодженість роботи усіх рівнів. У контексті моделі туманної архітектури передбачається ефективна передача, перетворення та адаптація даних, що надходять від пристроїв збору інформації через проміжні вузли до хмарної інфраструктури [19]. При цьому важливою умовою є не лише технічна можливість передавання даних, а й забезпечення їхньої смислової (семантичної) узгодженості, яка визначає коректну інтерпретацію результатів на вищих рівнях обробки [44].

Передача інформації між рівнями реалізується на основі протоколу публікації–підписки, що дозволяє організувати асинхронну взаємодію за моделлю "publish–subscribe". Пристрої збору даних виступають видавцями повідомлень, які передаються у вигляді структурованих об'єктів (наприклад, у форматі JSON) до проміжного обчислювального рівня, де функціонує брокер обміну повідомленнями [38].

Після надходження повідомлення на проміжному рівні здійснюється синтаксичний розбір його структури, перевірка на коректність даних, виявлення дублікатів, а також агрегація значень за певний період часу. Такі операції дозволяють зменшити кількість шумових записів, оптимізувати навантаження на канали передачі та забезпечити цілісність потоку даних.

Процес агрегації можна формалізувати математично через оператор середнього значення:

$$\bar{x} = \frac{1}{n}\sum_{i=1}^{n} xi, \tag{2.3}$$

де:

x_i – позначає значення параметра, отримане у і момент часу;

n – кількість вимірювань за обраний період.

Особливого значення набуває семантична уніфікація, яка реалізується шляхом доповнення кожного повідомлення додатковими атрибутами, що дозволяють однозначно інтерпретувати його зміст. Як показано в таблиці 2.4, такі атрибути включають унікальний ідентифікатор пристрою, тип сенсора, географічне положення точки збору даних, одиниці вимірювання відповідного параметра, а також уніфіковану часову мітку.

Таблиця 2.4 – Приклад структури семантично збагаченого повідомлення

Атрибут	Опис прикладу
device_id	Унікальний ідентифікатор пристрою
sensor_type	Тип сенсора (температура, вологість тощо)
location	Географічні координати або зона вимірювання
unit	Одиниця виміру (наприклад, °C)
timestamp	Час створення повідомлення

Такий формат дозволяє забезпечити сумісність повідомлень незалежно від типу пристрою, а також надає можливість створювати повноцінні семантичні моделі для автоматизованої обробки, класифікації або виявлення аномалій у потоці даних.

Процес обробки й семантичного збагачення повідомлень на проміжному обчислювальному рівні зображено на рисунку 2.4. Спочатку відбувається прийом повідомлення, після чого здійснюється перевірка його валідності. Далі дані проходять через етапи фільтрації та агрегації, а завершальним етапом є семантичне доповнення повідомлення необхідними атрибутами для подальшої обробки в хмарній інфраструктурі.

Завдяки такій побудові вдається сформувати систему, яка не лише передає дані, але й додає до них смислову інтерпретацію вже на проміжному рівні. Це підвищує якість обробки, зменшує навантаження на хмарні ресурси та створює передумови для розгортання інтелектуальних сервісів, що працюють із даними на основі змістової інтерпретації.

Рисунок 2.4 – Потік обробки та семантичного збагачення повідомлень

2.4 Механізм взаємодії між компонентами

Організація взаємодії між компонентами системи постачання туманних послуг має вирішальне значення для забезпечення узгодженості, безперервності та ефективності обміну інформацією у розподіленому середовищі. У запропонованій моделі обмін даними між рівнями збору інформації, проміжної обробки та хмарного зберігання реалізовано у вигляді асинхронного, подієво-орієнтованого потоку, який базується на концепції публікації–підписки.

На рівні пристроїв збору даних функціонує програмна логіка, яка здійснює періодичне зчитування показників фізичних величин із сенсорних модулів. Після збору інформації вона форматується у вигляді структурованого повідомлення та передається у певну тему за допомогою брокера повідомлень. Передача повідомлень відбувається через легковаговий протокол, здатний ефективно працювати за умов нестабільного з’єднання та мінімального енергоспоживання. Кожне сформоване повідомлення миттєво доставляється до проміжного

обчислювального вузла, який підписаний на відповідну тему через брокер обміну повідомленнями.

На проміжному рівні відбувається інтеграція отриманих даних із різних джерел. Після прийому повідомлень здійснюється їхня перевірка на валідність, агрегація за часовими вікнами та семантичне збагачення шляхом доповнення даних інформацією про місце збору, тип пристрою та одиниці вимірювання. Такі операції дозволяють сформувати уніфіковану структуру даних незалежно від різноманітності початкових повідомлень.

Подальша передача даних до хмарної інфраструктури відбувається через протокол захищеної комунікації із використанням стандартних запитів. У хмарному сховищі дані накопичуються без подальших змін, формуючи часові ряди, що уможливлює їхню аналітику, візуалізацію та довготривале зберігання.

Реалізація асинхронної, подієво-орієнтованої взаємодії між компонентами дозволяє досягти високої надійності роботи системи навіть у разі короткочасних збоїв окремих елементів. Кожен вузол працює автономно в рамках визначеного сценарію, що забезпечує масштабованість архітектури і стійкість до динамічних змін навантаження.

Механізм взаємодії також створює основу для оптимізації обробки даних без перевантаження хмарної частини системи. Завдяки семантичному збагаченню даних на проміжному рівні вдається мінімізувати обсяг переданої інформації, що у свою чергу підвищує продуктивність та знижує затримки.

2.5 Адаптивність та масштабованість моделі

Система постачання туманних послуг для керування даними Інтернету речей має працювати в умовах постійної змінності зовнішнього середовища та динамічного навантаження. Тому важливою властивістю моделі є її здатність до адаптивності та масштабованості без втрати продуктивності, цілісності обробки даних та стійкості комунікації між компонентами [36].

Адаптивність системи визначається її здатністю автоматично змінювати параметри функціонування у відповідь на зміни обставин. У межах запропонованої архітектури адаптивність забезпечується на кількох рівнях. Першим рівнем є пристрої збору даних, які здатні регулювати частоту надсилання повідомлень залежно від доступності мережевих ресурсів або зміни стану вимірюваного середовища [28]. Другим рівнем є проміжні обчислювальні вузли, які можуть варіювати розмір часових вікон агрегації даних, застосовувати різні методи фільтрації та змінювати інтенсивність публікації даних у хмару [35]. На хмарному рівні адаптивність проявляється у динамічному масштабуванні обчислювальних ресурсів та еластичному розподілі навантаження між сервісами [32].

З математичної точки зору механізм адаптації проміжного рівня можна описати через залежність частоти надсилання даних f_{send} від коефіцієнта доступних ресурсів α, який було введено раніше в пункті 2.1. Формально це можна записати так:

$$\text{fsend} = \text{fbase} \times \alpha, \tag{2.4}$$

де:

f_{base} – базова частота надсилання повідомлень у нормальних умовах,

α – коефіцієнт адаптивності системи.

Таким чином, при зменшенні доступних ресурсів відбувається автоматичне зменшення частоти передачі даних, що дозволяє уникнути перевантаження проміжних вузлів та забезпечити стійкість роботи системи в умовах обмежених ресурсів.

Щодо масштабованості, модель повинна підтримувати горизонтальне масштабування на всіх рівнях архітектури. Горизонтальне масштабування означає можливість додавання нових пристроїв збору даних, встановлення додаткових проміжних обчислювальних вузлів або розширення хмарної інфраструктури без суттєвого перегляду існуючих процесів взаємодії. Як показано на рисунку 2.5, завдяки побудові архітектури за принципом publish–subscribe кожен новий

компонент інтегрується у систему через підписку на відповідні теми брокера повідомлень, що не потребує змін у вже існуючих компонентах.

Особливістю розробленої моделі є також можливість вертикального масштабування, тобто збільшення обчислювальних потужностей окремих компонентів без зміни їхньої логічної ролі. Наприклад, проміжний обчислювальний вузол може бути перенесений на більш потужну обчислювальну платформу у разі підвищення обсягів даних або збільшення кількості підключених пристроїв.

Ключовими передумовами забезпечення ефективної адаптивності та масштабованості є дотримання принципів модульності, слабкої зв'язності компонентів.

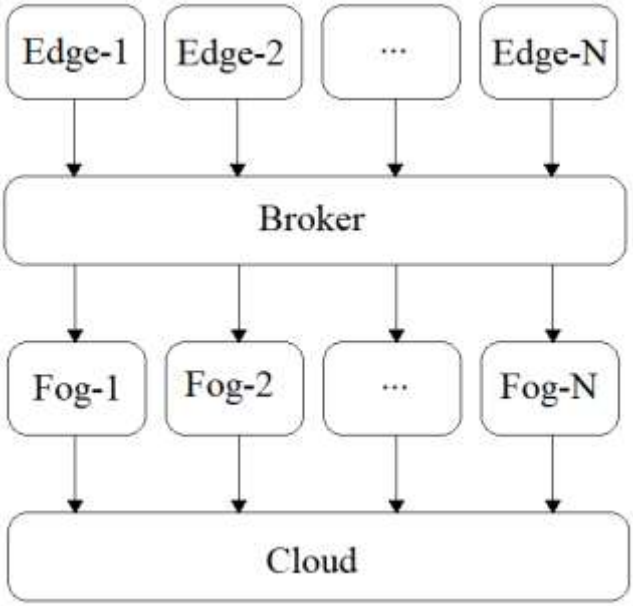


Рисунок 2.5 – Масштабування системи за допомогою додавання нових компонентів

2.6 Забезпечення безпеки та конфіденційності в туманній архітектурі IoT

Одним з ключових викликів при реалізації системи постачання туманних послуг для керування даними Інтернету речей є забезпечення безпеки та конфіденційності інформації. Розподілений характер архітектури, обмежені

ресурси периферійних пристроїв та динамічна структура мережі IoT створюють сприятливі умови для потенційних атак, втрат даних або порушення приватності користувачів.

Захист системи має будуватися на декількох рівнях: на рівні пристроїв збору, на рівні проміжної обробки (Fog), а також на рівні хмарної інфраструктури. Пристрої збору мають бути захищені від несанкціонованого доступу, для чого застосовуються механізми автентифікації та верифікації сертифікатів. Для обміну даними використовуються захищені протоколи з шифруванням (наприклад, TLS), що дозволяє унеможливити перехоплення або модифікацію трафіку під час передачі.

На проміжному рівні важливою є реалізація контролю доступу до даних. Це може досягатися через застосування списків дозволених пристроїв (whitelisting), політик ролей доступу або механізмів OAuth для обмеження функціоналу користувачів залежно від їхніх прав. Шифрування даних перед зберіганням у локальному кеші або перед передачею в хмару є критично важливим кроком для мінімізації ризику витоку інформації.

З математичної точки зору процес шифрування передбачає перетворення відкритого повідомлення M у зашифровану форму C за допомогою ключа K:

$$C = E(M), \tag{2.5}$$

де:

C - зашифроване повідомлення;

E - алгоритм шифрування з використанням ключа K;

M - початкові незашифровані дані.

Відповідне дешифрування відбувається на хмарному рівні або у системі верифікації:

$$M = DK(C), \tag{2.6}$$

Окрім технічних засобів, важливим є впровадження політик захисту конфіденційної інформації. Наприклад, у медичних чи транспортних системах допускається лише анонімізована або агрегована передача даних без персоналізованих ідентифікаторів. Це дозволяє зберігати конфіденційність користувача навіть у разі компрометації вузла.

Узагальнюючи, забезпечення безпеки та конфіденційності в туманній архітектурі IoT є багатошаровим завданням, що включає автентифікацію, шифрування, контроль доступу та захист від вторгнень. Ці заходи мають реалізовуватись на кожному рівні системи з урахуванням її ресурсних обмежень, динамічності та гетерогенності компонентів.

2.7 Моніторинг ефективності та оцінка продуктивності системи Fog-IoT

Одним із важливих етапів впровадження системи постачання туманних послуг для керування даними Інтернету речей є забезпечення постійного моніторингу її ефективності та оцінки продуктивності у різних умовах експлуатації. Це дозволяє виявити вузькі місця в архітектурі, адаптувати параметри обробки даних і підвищити загальну стабільність та надійність системи. Основними показниками продуктивності є середній час обробки повідомлення (від моменту його генерації до надходження у хмару), затримка передачі між рівнями (Edge – Fog – Cloud), пропускна здатність Fog-вузлів, яка вимірюється кількістю повідомлень за одиницю часу, а також навантаження на мережу та обчислювальні ресурси. До цього переліку також належить рівень втрати даних, що може виникати внаслідок збоїв або перевантаження.Для кожного з цих параметрів застосовуються автоматизовані засоби моніторингу, які інтегруються у систему обробки даних. Наприклад, на рівні проміжного вузла вбудовано модулі журналювання подій та метрик, які ведуть облік кількості оброблених повідомлень, часу виконання основних функцій (фільтрація, агрегація, збагачення) та обсягу відправленої інформації.

Оцінка ефективності виконується за допомогою формалізованих метрик. Зокрема, середній час затримки можна описати як:

$$Tavg = (1/n)\sum i = 1(tcloud(i) - tedge(i)), \quad (2.7)$$

де:

- tcloud(i) - момент надходження iii-го повідомлення у хмару;
- tedge(i) - момент його створення на пристрої збору;
- nnn - кількість повідомлень у вибірці.

Крім того, застосовується контроль ефективності агрегації, який визначає співвідношення між кількістю вхідних і вихідних повідомлень на Fog-рівні:

$$\eta a = Nout/Nin, \quad (2.8)$$

де:

- Nin - кількість отриманих повідомлень;
- Nout - кількість сформованих агрегованих повідомлень.

У реальних умовах моніторинг ефективності дає змогу, зокрема, визначити необхідність масштабування системи, адаптувати частоту надсилання повідомлень, оцінити якість обробки на кожному з рівнів, а також запобігти втраті критичних даних.

2.8 Висновки до другого розділу

У цьому розділі розроблено модель системи постачання туманних послуг для керування даними Інтернету речей, що ґрунтується на принципах багаторівневої обробки даних, семантичної уніфікації інформаційних потоків та адаптивного масштабування ресурсів. Особливу увагу приділено забезпеченню стійкості системи до змін навколишнього середовища, варіативності кількості пристроїв та динаміки обсягів даних.

Було визначено функціональні ролі основних компонентів моделі. Пристрої збору даних здійснюють первинне вимірювання та передачу структурованих повідомлень через брокер повідомлень. Проміжні обчислювальні вузли виконують попередню обробку, фільтрацію, агрегацію та семантичне збагачення даних. Хмарна інфраструктура відповідає за довготривале зберігання, аналітичну обробку та візуалізацію інформації. Логіку взаємодії між компонентами побудовано за принципом асинхронної подієво-орієнтованої моделі на базі протоколу publish–subscribe.

Здійснено математичне формалізування процесів агрегації та адаптивної регуляції обробки даних шляхом введення коефіцієнта адаптивності системи, що дозволяє описати поведінку системи у випадках змін доступності ресурсів.

Було запропоновано механізми горизонтального та вертикального масштабування моделі, які забезпечують можливість її розширення без порушення узгодженості взаємодії компонентів. Як показано на рисунку 2.3, завдяки архітектурі на основі підписки-публікації нові пристрої та вузли можуть легко інтегруватися у систему.

Розроблена модель створює передумови для ефективної реалізації систем постачання туманних послуг у реальних умовах, забезпечуючи мінімальні затримки обробки, стійкість до навантажень і можливість розвитку відповідно до змін вимог середовища та зростання обсягу даних.

3 СИСТЕМА ПОСТАЧАННЯ ТУМАННИХ ПОСЛУГ ДЛЯ КЕРУВАННЯ ДАНИМИ ІНТЕРНЕТУ РЕЧЕЙ

3.1 Архітектура системи постачальника туманних сервісів

Архітектура постачальника туманних сервісів (ПТС) розроблена з метою подолання численних труднощів, які виникають у процесі масштабного впровадження додатків Інтернету речей (IoT), коли йдеться про обчислення, зберігання інформації та організацію мережевих взаємодій. Ця архітектура виступає як своєрідна альтернатива традиційним хмарним обчисленням, пропонуючи інший підхід до розміщення та функціонування сервісів. Замість того щоб усі операції з обробки даних, їхнього зберігання чи мережевих функцій відбувалися в централізованих хмарних дата-центрах, ПТС переносить ці можливості ближче до периферії мережі - туди, де дані генеруються та використовуються. Такий розподілений підхід дозволяє ефективніше справлятися з навантаженнями, зменшувати затримки та оптимізувати ресурси, а в деяких випадках навіть передбачає кооперативну взаємодію між різними вузлами мережі, що додає системі гнучкості й адаптивності.

Основою цієї архітектури є концепція автономного управління сервісами, яка спирається на використання мікросервісів.

Мікросервіси - це невеликі, незалежні компоненти, кожен із яких виконує свою чітко визначену функцію, але разом вони утворюють цілісну систему. Такий підхід виявляється надзвичайно зручним і дієвим, коли потрібно вирішувати складні завдання, пов’язані з неоднорідністю IoT-екосистеми. Наприклад, у світі Інтернету речей пристрої можуть суттєво різнитися за своїми характеристиками, протоколами чи способами взаємодії, а ПТС завдяки своїй архітектурі здатен забезпечити сумісність на різних рівнях - чи то мережі, чи то дані, чи то програмне забезпечення. Це дозволяє системі гармонійно працювати з різноманітними технологіями та стандартами, що є критично важливим у сучасних умовах.

Джерела даних у IoT-середовищі вирізняються надзвичайною різноманітністю, і це одна з ключових особливостей, яку враховує архітектура

ПТС. Пристрої можуть підключатися до мережі через численні технології, такі як Wi-Fi, стільникові мережі (наприклад, 4G чи 5G), Lo-Ra, ZigBee чи Bluetooth. Кожна з цих технологій має свої переваги й обмеження, а їхнє співіснування в одній системі створює додаткові виклики. Крім того, для передачі даних між пристроями та сервісами часто застосовуються спеціалізовані протоколи обміну повідомленнями, такі як MQTT, CoAP чи AMQP.

Ці протоколи розроблені з урахуванням потреб IoT, де важливими є низьке енергоспоживання, мала пропускна здатність і здатність працювати в умовах нестабільного з'єднання. Однак усі ці різноманітні технології та підходи ускладнюють управління даними, адже інформація надходить у різних форматах, із різною частотою та обсягом.

Управління даними, які генеруються такими джерелами, залишається одним із найскладніших аспектів у впровадженні IoT-додатків. Оскільки обсяги інформації зростають, а вимоги до швидкості обробки стають дедалі жорсткішими, традиційні хмарні рішення не завжди можуть упоратися з цими задачами оптимально. Саме тут архітектура ПТС демонструє свої переваги, пропонуючи локалізовану обробку даних на периферії, що зменшує потребу в постійній передачі великих обсягів інформації до віддалених серверів.

Водночас кооперативний і розподілений характер цієї моделі дозволяє ефективно розподіляти обчислювальні ресурси між різними вузлами, що сприяє підвищенню продуктивності й надійності системи в цілому.

Таким чином, ПТС не лише вирішує технічні проблеми, а й відкриває нові можливості для масштабування IoT-додатків у реальних умовах.

Архітектура постачальника туманних сервісів (ПТС), побудована на основі мікросервісів, дійсно стала значним кроком уперед у вирішенні проблем, пов'язаних із масштабним розгортанням IoT-додатків, адже вона дозволила усунути багато труднощів, що виникали в традиційних хмарних системах, таких як затримки в обробці даних чи неефективне використання мережевих ресурсів. Проте, разом із цими перевагами з'явилися й нові виклики, які потребують уваги та ретельного опрацювання.

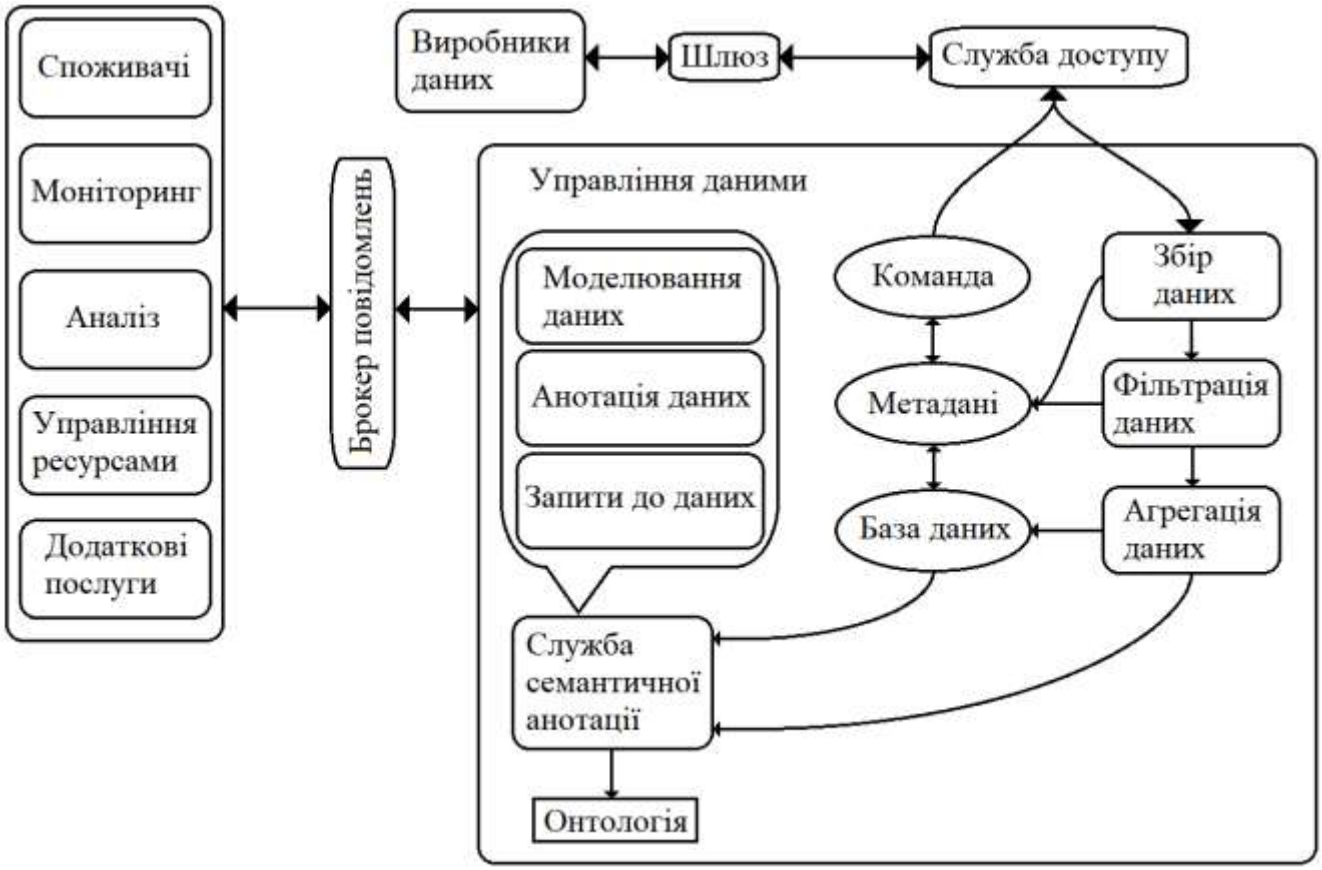


Рисунок 3.1 – Управління даними

Однією з таких проблем є підтримка узгодженості даних між численними сервісами, які функціонують автономно в рамках цієї архітектури. У світі, де мільярди IoT-пристроїв - так званих виробників - генерують величезні потоки даних, повідомляючи про різноманітні події чи аномалії, забезпечення того, щоб усі ці дані залишалися синхронізованими та доступними для використання в реальному часі, стає надзвичайно складним завданням. Ці дані, зібрані з різноманітних джерел, після обробки відіграють ключову роль у вдосконаленні промислових процесів, оптимізації виробництва та підвищенні ефективності систем спостереження, що стало можливим саме завдяки розміщенню обчислень поблизу джерел даних у рамках туманної архітектури.

Архітектура ПТС вирізняється своєю здатністю інтегрувати кілька методів обробки даних, що дозволяє їй справлятися з неоднорідністю, яка є невід'ємною частиною IoT-екосистеми. Йдеться про такі процеси, як збір даних, їхня фільтрація, агрегація та семантична анотація, які разом створюють міцну основу для

забезпечення сумісності між різними пристроями, протоколами й програмами. Завдяки цьому ПТС може підтримувати як синхронні, так і асинхронні протоколи зв’язку, що є критично важливим у середовищах, де пристрої можуть працювати з різними технологіями, такими як Wi-Fi, Lo-Ra чи MQTT. Управління даними в цій архітектурі сприяє не лише підвищенню якості IoT-додатків, але й полегшує їхнє впровадження, адже кожен сервіс створюється незалежно від даних, з якими він працює, що додає системі гнучкості. Однак для того, щоб це управління було дійсно ефективним, необхідно враховувати цілу низку аспектів, які впливають на роботу системи в цілому.

Перш за все, типи даних, які обробляються в рамках ПТС, мають бути чітко визначені та доступні для відповідних сервісів, при цьому вони повинні відповідати нормативним вимогам, що стосуються, наприклад, конфіденційності чи стандартів обробки. Безпека цих даних забезпечується через використання чітко визначених API, які слугують своєрідними воротами для доступу до інформації. Сервіси, які займаються трансформацією даних, обмінюються інформацією через стандартні повідомлення, що базуються на чергах і визначених API, що дозволяє уникнути хаосу в потоках даних і забезпечити їхню структуровану передачу. Політики доступу до даних також відіграють важливу роль - залежно від чутливості інформації, ці політики потрібно зберігати та оновлювати, а сервіси мають бути прив’язані до конкретних ролей у системі, щоб уникнути несанкціонованого використання. При збереженні даних у сховищі враховуються такі фактори, як їхній обсяг, швидкість надходження, спосіб використання, а також вимоги до безпеки та надійності, що робить процес зберігання складним, але необхідним елементом архітектури. Крім того, для забезпечення належного рівня безпеки та прозорості в системі потрібно відстежувати трафік даних, вести облік інформації та контролювати доступ до неї, що додає ще один рівень складності в управлінні.

У цьому контексті мається на увазі підхід до управління даними для постачальника туманних сервісів, який дозволяє ефективно обробляти інформацію від різноманітних IoT-виробників і забезпечує безперебійну взаємодію між споживачами та вузлами ПТС у найрізноманітніших середовищах. У попередніх

описах вже була описана архітектура, заснована на мікросервісах, і пропонували набір сервісів, які відповідають за управління функціями, забезпечення сумісності та масштабованості системи. Однак управління даними та їхній життєвий цикл є тими фундаментальними концепціями, без яких ця архітектура не могла б повноцінно функціонувати. Також був розроблений спеціальний сервіс семантичної анотації даних, який дозволяє ПТС обмінюватися інформацією та забезпечувати її сумісність між різними доменами чи програмами, що робить систему більш універсальною. Унікальність цього підходу полягає в тому, що був створений механізм управління даними, який здатен описувати та зберігати інформацію від гетерогенних IoT-пристроїв, використовуючи технології семантичного вебу, що значно полегшує роботу з різноманітними форматами та джерелами.

Окрім цього, розроблений алгоритм, який дозволяє відображати дані з реляційних баз даних у єдиний формат, що спрощує їхню інтеграцію та подальше використання в системі. Щоб забезпечити ефективне управління процесами реєстрації та доступу пристроїв до архітектури ПТС, також може бути набір процедур, які автоматизують ці процеси, дозволяючи системі швидко адаптуватися до нових виробників чи споживачів. Ці процедури охоплюють усе - від початкової інтеграції пристроїв до обробки даних, що надходять від них, і забезпечують стабільність і надійність роботи ПТС у реальних умовах. Таким чином, цей підхід не лише вирішує наявні проблеми, але й закладає основу для подальшого розвитку туманних обчислень у сфері IoT, пропонуючи гнучке й масштабоване рішення для управління даними в умовах їхньої зростаючої складності та обсягу.

3.2 Управління даними

У сучасному світі, де численні виробники IoT-пристроїв, розподілені по різних куточках планети, від мегаполісів до віддалених промислових зон, створюють надзвичайно різноманітний ландшафт технологій, обсяги даних, що генеруються цими системами, досягають колосальних масштабів. Ці пристрої - від

крихітних сенсорів, що моніторять якість повітря в розумних містах, до складних промислових датчиків, які відстежують роботу обладнання на заводах, - безперервно продукують потоки інформації. У таких умовах туманні обчислення (fog computing) перетворюються з просто корисного інструменту на фундаментальну основу для додатків, які вимагають миттєвої реакції та обробки даних у реальному часі. Їхня роль стає дедалі критичнішою для забезпечення безперебійної роботи систем, таких як автоматичне керування транспортними засобами, моніторинг стану складних механізмів чи оперативне реагування на надзвичайні ситуації, наприклад, природні катастрофи чи аварії на виробництві.

Туманні обчислення вирізняються тим, що переносять обчислювальні потужності максимально близько до джерел даних - самих IoT-пристроїв. Це дозволяє значно скоротити затримки в обробці інформації порівняно з традиційними хмарними системами, де дані змушені долати довгий шлях до віддалених централізованих серверів, проходячи через численні мережеві вузли. У хмарних архітектурах ці відстані можуть призводити до суттєвих втрат часу, що є неприпустимим у сценаріях, де швидкість реакції має вирішальне значення. Наприклад, у системах автономного водіння кожна мілісекунда може означати різницю між безпекою та аварією, а в медичних IoT-додатках - між вчасним попередженням про критичний стан пацієнта та запізнілим реагуванням. Локальна обробка даних у туманних обчисленнях вирішує цю проблему, забезпечуючи швидкість і ефективність там, де традиційні методи вже не справляються.

Однак, попри всі свої переваги, туманні обчислення стикаються з низкою серйозних викликів, які ускладнюють їхнє масштабне впровадження. Перш за все, це пов’язано з величезними обсягами даних, які надходять від IoT-пристроїв. Кількість інформації зростає в геометричній прогресії, і з кожним днем до мережі підключаються все нові й нові пристрої - від побутових гаджетів у розумних будинках до складних систем у промислових комплексах. Ця лавина даних вимагає не лише потужних обчислювальних ресурсів, а й здатності обробляти їх у реальному часі, щоб забезпечити своєчасне виконання запитів і підтримку функціональності систем.

Ще одним викликом є надзвичайна різноманітність даних, що надходять від IoT-пристроїв. Ці дані можуть мати різні формати, синтаксис і структуру, що значно ускладнює їхню обробку та аналіз. Наприклад, один сенсор температури в розумному місті може передавати свої показники у форматі JSON, тоді як інший пристрій, розташований на заводі, використовує XML або навіть бінарний формат. Така гетерогенність створює перепони не лише для інтерпретації даних, а й для забезпечення їхньої сумісності між різними системами та доменами IoT. Уявімо, наприклад, сферу охорони здоров'я, де медичні датчики мають свої специфічні стандарти передачі даних, і порівняймо її з доменом розумних будинків, де пріоритетом є енергоефективність і простота інтеграції. Ці відмінності ускладнюють створення універсальних рішень, які б однаково добре працювали в різних контекстах.

Сумісність у світі IoT - це не просто технічна задача, а багатогранна проблема, що охоплює кілька рівнів взаємодії. Йдеться не лише про сумісність на рівні пристроїв чи мереж, а й про платформи, сервіси та додатки, які повинні безперешкодно працювати разом. Особливу роль відіграє семантична та синтаксична сумісність - здатність систем не лише розпізнавати формати даних, а й правильно інтерпретувати їхнє значення. Наприклад, показник "25" може означати температуру в градусах Цельсія в одному контексті, але в іншому - відсоток вологості чи навіть код помилки. Без уніфікації та чіткого розуміння контексту ці дані втрачають свою цінність.

Таким чином, туманні обчислення, хоча й відкривають нові горизонти для IoT-додатків, потребують постійного вдосконалення. Їхній успіх залежить від здатності адаптуватися до зростаючої складності екосистеми IoT, долати бар'єри гетерогенності даних і забезпечувати стабільну, швидку та безпечну обробку інформації в умовах, коли кожна деталь має значення.

У сучасному світі IoT, де гетерогенність даних від різних виробників створює значні перешкоди для їхньої обробки та використання, була розроблена інноваційна архітектура постачальника туманних сервісів (ПТС). Ця архітектура спеціально спроєктована для того, щоб подолати виклики, пов'язані з

різноманітністю IoT-пристроїв, і пропонує комплексний підхід до управління даними. Її основою є ретельно розроблена інформаційна система, яка не лише впорядковує потоки інформації, а й забезпечує семантичну сумісність між системами. Завдяки цьому дані перестають бути просто хаотичним набором чисел, символів чи сигналів - вони набувають осмисленого значення, стаючи цінним ресурсом для прийняття рішень, обміну інформацією між різними платформами та створення інтелектуальних додатків.

Семантична сумісність, яку підтримує система, дозволяє системі не лише збирати дані, а й глибоко аналізувати їхній зміст. Наприклад, замість того щоб сприймати показник "25" як абстрактне число, система може розпізнати, що це температура в градусах Цельсія, виміряна в конкретному місці - скажімо, у розумному будинку чи на промисловому об’єкті - у певний момент часу. Більше того, ці дані можуть бути пов’язані з іншими параметрами, такими як вологість чи тиск, для формування ширшої картини стану навколишнього середовища чи обладнання. Такий підхід кардинально змінює сприйняття інформації: вона перетворюється з сирих даних на структуровані сутності, які легко інтегруються в різноманітні сценарії використання - від моніторингу здоров’я до управління міською інфраструктурою.

Ключову роль у реалізації цього підходу відіграють семантичні технології, зокрема ті, що базуються на принципах Семантичного Вебу. Ці технології дозволяють комп’ютерам не просто обробляти дані як потік байтів, а й розуміти їхній контекст і значення. У контексті IoT це має вирішальне значення, адже щодня численні пристрої - від сенсорів вологості в сільському господарстві до датчиків руху в системах безпеки - генерують величезні обсяги необробленої інформації. Самі по собі ці дані, такі як показники енергоспоживання чи швидкості вітру, мають обмежену цінність. Проте, коли вони проходять через процес семантичної анотації з використанням онтологій, їхня інформативність зростає в рази. Онтології - це структури знань, які визначають зв’язки між поняттями, наприклад, між "температурою", "місцем розташування" і "часом вимірювання". Завдяки цьому

система може інтерпретувати дані в ширшому контексті, що відкриває двері до створення розумних сервісів і додатків.

У рамках архітектури ПТС центральне місце займає сервіс виробника, який відповідає за первинну обробку необроблених даних і їхню трансформацію в структуровані знання. Цей сервіс використовує спеціалізовані сервіси знань, які допомагають усунути гетерогенність між різними IoT-виробниками та самою системою ПТС. Наприклад, дані, що надходять від сенсора температури через протокол MQTT, можуть суттєво відрізнятися за форматом від інформації, отриманої через CoAP від датчика руху. Однак сервіс виробника здатен привести ці різнорідні потоки до єдиного стандарту, створивши уніфіковану систему даних. Такий підхід не лише спрощує обробку інформації, а й забезпечує її сумісність на всіх рівнях системи, незалежно від типу пристрою чи протоколу передачі.

Окрім цього, сервіс виробника в архітектурі ПТС оснащений API, який забезпечує гнучку взаємодію з різними IoT-пристроями. Це дозволяє системі легко адаптуватися до нових типів виробників, технологій чи навіть змін у стандартах передачі даних. Така гнучкість є критично важливою в умовах швидкого розвитку IoT-екосистеми, де щороку з’являються нові пристрої з унікальними характеристиками та вимогами. Наприклад, якщо завтра з’явиться новий тип сенсора з власним протоколом передачі, ПТС зможе інтегрувати його без необхідності перебудовувати всю архітектуру. Це робить систему не лише ефективним рішенням для сьогодення, а й перспективною основою для майбутніх інновацій.

Таким чином, архітектура ПТС не просто вирішує проблему гетерогенності даних у світі IoT, а й закладає фундамент для створення інтелектуальних систем, які здатні не лише зберігати чи передавати інформацію, а й активно використовувати її для прийняття рішень у реальному часі. Перетворюючи сирі дані на знання, вона відкриває нові можливості для оптимізації процесів, прогнозування подій і створення розумних сервісів, які можуть радикально змінити наше повсякденне життя - від управління енергоефективними будинками до впровадження автономних транспортних систем. Усе це робить ПТС не просто

технологічним рішенням, а кроком до майбутнього, де IoT стане основою для інтелектуального світу.

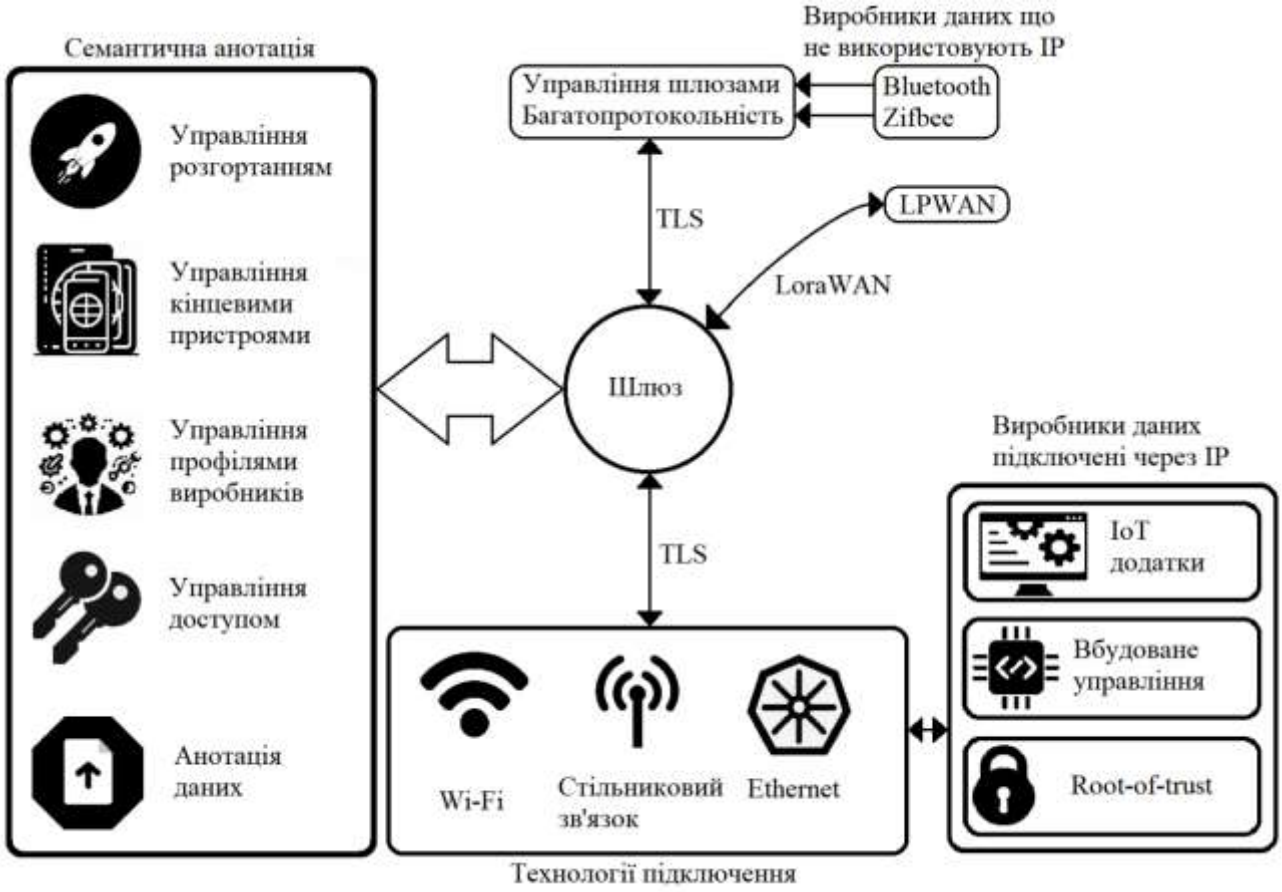


Рисунок 3.2 – Загальна система взаємодії на рівні виробника IoT

Система взаємодії для гетерогенних IoT-виробників, яка детально представлена на відповідному рисунку, є ключовим елементом архітектури постачальника туманних сервісів (ПТС). Вона враховує складність сучасних IoT-систем, де пристрої з різними характеристиками та можливостями об’єднуються в єдину екосистему. Для зручності аналізу та управління ці компоненти з обмеженою обчислювальною потужністю поділяються на три основні категорії: виробники, що використовують IP-протоколи, виробники, які не базуються на IP, а також технології зв’язку та семантична анотація. Такий підхід дозволяє систематизувати різноманітність пристроїв і технологій, що беруть участь у процесі передачі та обробки даних, створюючи основу для їхньої ефективної інтеграції.

Пристрої, що належать до категорій IP та не-IP виробників, зазвичай представляють базовий рівень IoT-екосистеми. Це можуть бути прості сенсори чи датчики з мінімальними обчислювальними ресурсами, які виконують базові функції, такі як вимірювання температури, вологості чи руху. Ці пристрої використовують широкий спектр технологій зв'язку, що відображає їхню різноманітність і адаптивність до різних умов експлуатації. Серед найпоширеніших технологій можна виділити Wi-Fi, який забезпечує швидке підключення в міських умовах, стільниковий зв'язок для віддалених зон, а також низькоенергетичні протоколи, такі як LoRa, ZigBee чи Bluetooth, які ідеально підходять для пристроїв із обмеженим живленням. Кожен із цих протоколів має свої переваги й обмеження, що додає ще один рівень складності до процесу інтеграції.

Для забезпечення безперебійної взаємодії між такими різнорідними виробниками та системою ПТС використовується спеціальний компонент - ПТС-шлюз.

Шлюз відіграє роль посередника, який не лише полегшує передачу даних від IoT-пристроїв до туманної інфраструктури, а й гарантує безпеку та ефективність цього процесу. У контексті IoT-систем, де обсяги даних стрімко зростають, а швидкість їхньої передачі набуває критичного значення, ПТС-шлюз стає незамінним інструментом. Він підтримує різноманітні протоколи обміну повідомленнями, такі як MQTT, CoAP, WebSockets та HTTP/HTTPS, які дозволяють передавати дані від пристроїв до системи ПТС у реальному часі. Наприклад, MQTT забезпечує легку та швидку передачу даних для пристроїв із низьким енергоспоживанням, тоді як HTTP/HTTPS більше підходить для складніших сценаріїв із високими вимогами до безпеки.

Незважаючи на те, що ПТС-шлюз успішно вирішує багато проблем сумісності між гетерогенними виробниками, різноманітність даних, які надходять із різних джерел, залишається значним викликом. Дані можуть відрізнятися за форматом, структурою та семантикою, що ускладнює їхнє повноцінне використання без належного підходу до створення зв'язків між ними. Наприклад, показники температури від одного сенсора можуть надходити у форматі JSON, тоді

як дані про рух від іншого пристрою - у бінарному вигляді. Така неоднорінність перешкоджає глибокому аналізу та обміну інформацією між системами, що обмежує доступ до семантичних даних - тобто даних, які не лише передають значення, а й несуть контекст і зміст.

Для подолання цього бар'єру в архітектурі ПТС передбачена служба семантичної анотації. Ця служба відіграє ключову роль у перетворенні сирих даних на структуровану та осмислену інформацію. Вона використовує онтології - формальні структури знань, які описують зв'язки між поняттями, такими як "температура", "час" чи "місце". Завдяки застосуванню методів Семантичного Вебу, зокрема моделі Linked Data та мови запитів SPARQL, служба семантичної анотації маркує дані, додаючи їм контекстуальне значення. Наприклад, замість того щоб просто отримати число "25", система може визначити, що це температура в градусах Цельсія, виміряна в певній локації, і пов'язати її з іншими параметрами, такими як вологість чи тиск.

Управління даними в рамках ПТС - це багатогранний процес, який охоплює кілька ключових функцій. По-перше, це збирання даних із різноманітних джерел через ПТС-шлюз. Далі йде їхня агрегація, тобто об'єднання в єдину систему для подальшої обробки. Моделювання даних передбачає створення структури, яка відображає їхні характеристики та взаємозв'язки. Відображення та зв'язування даних забезпечують їхню інтеграцію в ширший контекст, а створення запитів - швидкий доступ до потрібної інформації за допомогою SPARQL чи інших інструментів. Усе це разом дозволяє не лише подолати проблему гетерогенності, а й відкриває нові можливості для аналізу та використання даних у розумних IoT-системах, де кожна деталь має значення.

Таким чином, система взаємодії для гетерогенних виробників у рамках ПТС - це не просто технічне рішення, а комплексний підхід, який поєднує апаратне забезпечення, протоколи зв'язку та семантичні технології. Вона забезпечує стабільну основу для роботи з величезними обсягами різноманітних даних, перетворюючи їх на цінний ресурс для інтелектуальних додатків і сервісів майбутнього.

3.2.1 Упорядковування даних

Механізм упорядкування даних відіграє ключову роль у структурі архітектури постачальника туманних сервісів (ПТС), забезпечуючи ефективне управління інформаційними потоками від різноманітних IoT-виробників. Цей механізм охоплює кілька важливих процесів, таких як збирання даних із різних джерел, їхнє зберігання в оптимальній формі, підготовка до обробки та об'єднання для подальшого використання. Усе це разом складає завдання попередньої обробки, яке є фундаментом для якісного аналізу інформації. Попередня обробка спрямована на те, щоб очистити необроблені дані, отримані від виробників, від шумів, дублювань чи нерелевантної інформації, а також привести їх до структурованих форматів, таких як JSON, XML, CSV або навіть реляційні бази даних (RDB).

Такий підхід до впорядкування даних не лише полегшує їхнє подальше використання, а й значно оптимізує обчислювальні ресурси системи. Наприклад, усунення надлишкової інформації дозволяє зменшити навантаження на пам'ять і процесори, що особливо важливо для пристроїв із обмеженими можливостями. Крім того, конвертовані дані стають більш зручними для наступного етапу - анотації, де вони набувають семантичного значення. На початковій стадії управління даними в ПТС ці процеси реалізуються через три основні функції, які детально розглядатимуться далі, забезпечуючи чітку послідовність і логіку в роботі з інформацією.

Функція збору даних у рамках архітектури ПТС відіграє ключову роль у накопиченні інформації, яка надходить від гетерогенних IoT-виробників. Ця функція розроблена так, щоб забезпечити гнучке збирання даних різного типу, походження та структури, дозволяючи пристроям із різними характеристиками легко інтегруватися в систему. Для цього вона підтримує широкий спектр рішень: від традиційних реляційних баз даних (SQL) до сучасних NoSQL-систем, а також документоорієнтованих форматів, таких як CSV, XML і JSON. Така різноманітність підходів гарантує, що інформація від виробників із різними технічними

можливостями може бути зібрана та збережена без втрати її цілісності чи доступності.

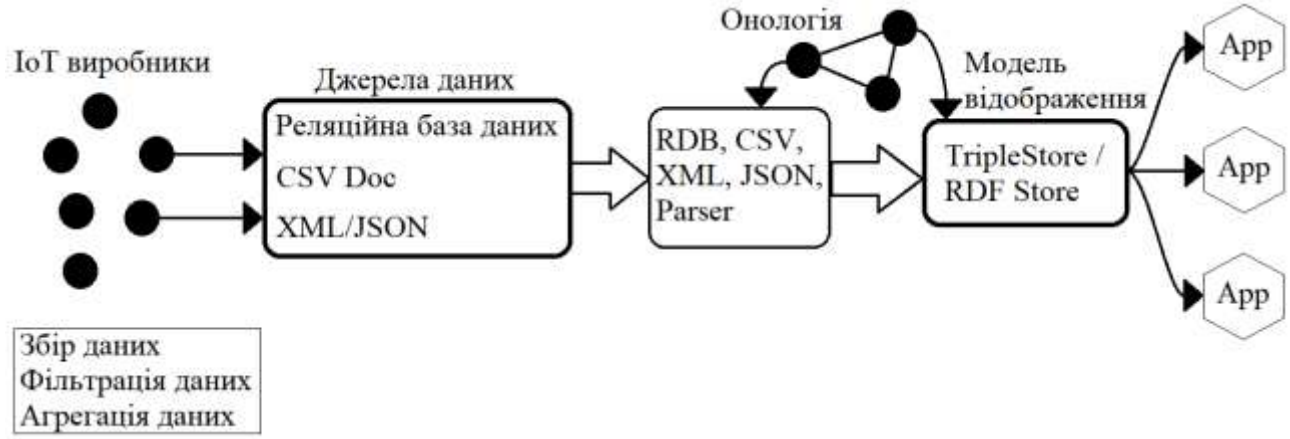


Рисунок 3.3 – Обробка даних

Наступний етап - процес агрегації даних - полягає в отриманні безперервних потоків інформації від IoT-пристроїв і їхньому тимчасовому збереженні для подальшого аналізу. Агрегація дозволяє систематизувати дані, які надходять у реальному часі, створюючи основу для їхньої обробки. Збережені у різних базах даних агреговані дані піддаються ретельній фільтрації, яка базується на налаштованих профілях, таких як producerId, producerName чи producerModel. Ці профілі допомагають ідентифікувати джерело даних і передавати опис виробника до ПТС-шлюзу. Завдяки цьому обсяг інформації, що переміщується між виробниками та шлюзом, значно скорочується, що сприяє економії пропускної здатності мережі та підвищенню ефективності передачі.

Фільтрація даних є важливим кроком, спрямованим на оптимізацію подальших етапів обробки. Її головна мета - зменшити обсяг інформації, яку потрібно буде анотувати, заощаджуючи при цьому обчислювальні ресурси. Цей процес базується на наборі заздалегідь визначених правил, які дозволяють відсіювати дубльовані записи, нерелевантні дані чи надлишкову інформацію, залишивши лише те, що дійсно має цінність для системи. Після фільтрації спеціалізовані парсери для RDB, CSV, XML і JSON беруть на себе обробку відфільтрованих даних. Вони моделюють інформацію, створюючи логічні зв’язки

між об’єктами даних, що готує її до перетворення у формат Resource Description Framework (RDF).

Отримані модельовані дані конвертуються в RDF, який є стандартом для представлення семантичної інформації. Цей формат поєднується з онтологіями, що додають даним семантичні анотації, збагачуючи їх контекстом і значенням. У RDF ресурси описуються через так звані "трійки" або "RDF-твердження", які мають структуру суб’єкт-предикат-об’єкт. Наприклад, суб’єкт може представляти певний об’єкт (скажімо, "ПТС"), предикат - відношення чи дію (наприклад, "є"), а об’єкт - характеристику чи значення ("мікросервісна архітектура"). Така структура дозволяє чітко визначити зв’язки між елементами даних. Візуально трійка зображується як зв’язок типу вузол-дуга-вузол, де напрямок указує від суб’єкта до об’єкта.

Сукупність таких RDF-трійок утворює RDF-граф - структуровану діаграму, яка складається з вузлів (суб’єктів і об’єктів) та спрямованих дуг (предикатів). Наприклад, речення "ПТС є мікросервісною архітектурою" може бути представлене у вигляді трійки й збережене в спеціальному сховищі трійок. У цьому випадку "ПТС" виступає суб’єктом, "мікросервісна архітектура" - об’єктом, а предикат "є" відображає їхній зв’язок. Такий підхід забезпечує не лише структуризацію даних, а й можливість їхнього подальшого аналізу та використання в інтелектуальних IoT-системах, де семантична інформація відіграє вирішальну роль.

Рисунок 3.4 – RDF граф

Суб’єкт – це елемент, який позначає об’єкт, до якого відноситься твердження, і є початком дуги в RDF-графі, наприклад, ПТС. Предикат описує характеристику суб’єкта в твердженні або виступає як властивість у трійці, наприклад, "є". Об’єкт

визначає значення цієї характеристики і може бути представлений як ресурс (URI) або як літерал (значення), наприклад, мікросервісна архітектура.

Твердження розглядаються як ресурси, що зв'язують об'єкти даних у RDF-сховищі трійок і використовуються як універсальний ідентифікатор ресурсів (URI), що є унікальним ідентифікатором у глобальній системі ідентифікації. Дані гетерогенних виробників, які попередньо оброблені та керовані вузлом ПТС, зберігаються у реляційній базі даних (RDB). Сукупність пов'язаних трійок утворює RDF-модель на основі графа, у якій вузли представляють суб'єкти або об'єкти, а ребра – предикати. Отже, при перетворенні даних із різних форматів у формат RDF необхідно забезпечити збереження їхньої цілісності та сенсу.

Нехай D позначає реляційну базу даних, що складається зі скінченної множини відношень. Для кожного відношення $\boldsymbol{\mathcal{T}} \in$ D визначимо:

Нехай κ($\boldsymbol{\mathcal{T}}$) - множина атрибутів, які утворюють первинний ключ таблиці $\boldsymbol{\mathcal{T}}$.

Нехай φ($\boldsymbol{\mathcal{T}}$) - множина атрибутів, що є зовнішніми ключами у $\boldsymbol{\mathcal{T}}$.

Нехай α($\boldsymbol{\mathcal{T}}$) - множина атрибутів, які не входять до первинного або зовнішнього ключа, тобто:

$$\alpha(\mathcal{T}) = Attr(\mathcal{T}) \setminus (\kappa(\mathcal{T}) \cup \varphi(\mathcal{T})), \tag{3.1}$$

де Attr($\boldsymbol{\mathcal{T}}$) - повна множина атрибутів таблиці $\boldsymbol{\mathcal{T}}$.

Нехай ρ($\boldsymbol{\mathcal{T}}$) - множина записів (кортежів) у таблиці $\boldsymbol{\mathcal{T}}$. Для кожного r ∈ ρ($\boldsymbol{\mathcal{T}}$), r[β] позначає значення атрибута β ∈ Attr($\boldsymbol{\mathcal{T}}$).

Визначимо функцію перетворення Г, таку що:

$$\Gamma: D \rightarrow \mathcal{G}, \tag{3.2}$$

де $\boldsymbol{\mathcal{G}}$ - RDF-граф, що складається з множини трійок:

$$\mathcal{G} = \{\langle s, p, o \rangle\}, \tag{3.3}$$

Кожне відношення $\boldsymbol{\mathcal{T}} \in$ D відображається на клас $\boldsymbol{\mathcal{C}\_\mathcal{T}} \in \boldsymbol{\mathcal{G}}$. Кожен запис r $\in$ ρ($\boldsymbol{\mathcal{T}}$) визначає унікальний ідентифікатор суб'єкта s = σ(r), де σ - функція, яка витягує унікальний ідентифікатор із первинного ключа:

$$\sigma(r) \; = \; URI(r[\kappa(\mathcal{T})]), \tag{3.4}$$

Для кожного атрибута β ∈ α($\boldsymbol{\mathcal{T}}$), визначається предикат p_β та відповідне значення об'єкта o_β = r[β], що утворює RDF-трійку:

$$\langle \sigma(r), p\_\beta, o\_\beta \rangle \; \in \; \mathcal{G}, \tag{3.5}$$

Атрибути зовнішніх ключів φ($\boldsymbol{\mathcal{T}}$) створюють додаткові об'єктні зв'язки між класами в RDF-графі, зберігаючи цілісність зв'язків між таблицями.

Перетворення Γ також зберігає функціональні залежності $\mathscr{F}$ у схемі, де:

$$\mathcal{F} \; = \; \{ X \; \rightarrow \; Y \mid X, Y \; \subseteq \; Attr(\mathcal{T}) \}, \tag{3.6}$$

Ці залежності використовуються для забезпечення семантичної узгодженості та можуть бути застосовані для нормалізації даних та побудови обмежень у RDF-моделі.

Таким чином, перетворення Γ формує семантично насичений RDF-граф $\boldsymbol{\mathcal{G}}$, який кодує як структурні, так і семантичні властивості початкової реляційної бази даних D, забезпечуючи її інтеграцію в середовище Семантичного Вебу та можливість подальшого логічного виведення.

3.2.2 Анотація даних

Обробка великого обсягу різнорідних даних у реальному часі відіграє вирішальну роль у розробці інтелектуальних додатків. Для подолання викликів, пов'язаних із великою гетерогенністю та сумісністю, застосовуються семантичні

методи на основі онтологій. Онтологія містить технічні характеристики, необхідні для реєстрації, підключення та передачі даних від виробників, а також може використовуватися як джерело метаданих для інших мікросервісів. Служба семантичної анотації даних (SDA) обробляє інформацію від кожного виробника, отриману від парсерів, як зображено на рисунку 3.5, після чого проводить анотацію, маркуючи ці дані концептами з доменної онтології.

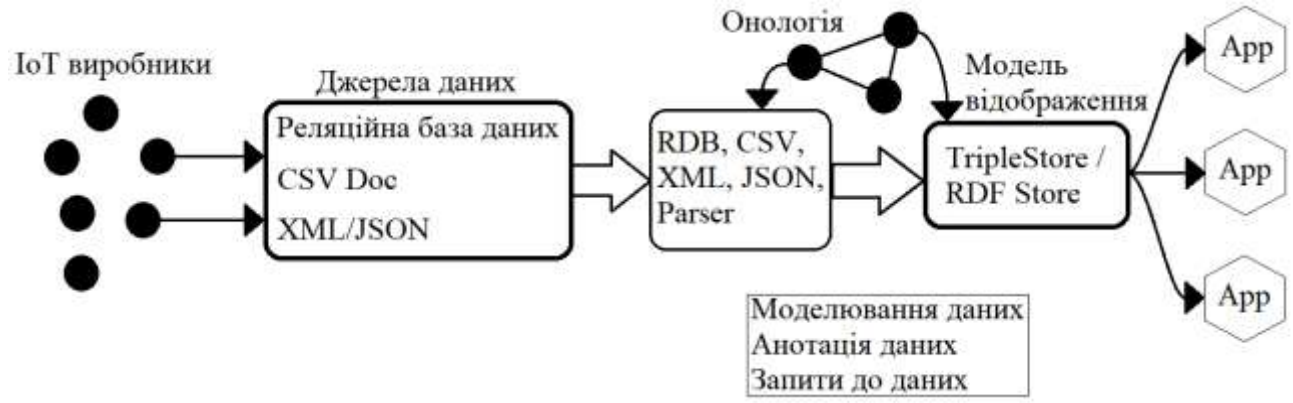


Рисунок 3.5 – Служба семантичної анотації даних (САД)

Зазвичай виробники розташовані на базовому рівні, мають обмежені ресурси й відповідають лише за збір даних та їхню передачу до шлюзу, тоді як виробники на рівні шлюзу мають більші обчислювальні можливості порівняно з рівнем IoT-виробників. Головна задача SDA полягає в аналізі та попередній обробці необроблених даних від виробників шляхом видалення надлишкової інформації та перетворення даних у формати, такі як XML або JSON, що зменшує потребу в обчислювальних ресурсах для анотаційних процесів. На цьому етапі SDA включає три основні функції. Моделювання даних передбачає аналіз відфільтрованих даних, отриманих після етапу фільтрації, за допомогою парсерів, які моделюють ці дані, визначаючи зв’язки з іншими об’єктами даних, після чого дані конвертуються у формат Resource Description Framework (RDF) з використанням онтології. Анотація даних отримує інформацію з попередніх етапів і проводить маркування даних концептами, отриманими через зіставлення доменної та референтних онтологій. Функція запитів до даних дозволяє отримувати явну та неявно виведену

інформацію за допомогою семантичних запитів на основі синтаксичної, семантичної та структурної інформації, що міститься в даних, забезпечуючи точні результати або відповіді на широкі та нечіткі питання шляхом зіставлення шаблонів і логічного виведення. Результатом роботи SDA є файл бази даних у форматі RDF-сховища трійок, де визначено виробників та їхні компоненти разом із функціональностями та виміряними значеннями в онтологічних зв'язках. Онтологія, будучи легким файлом, забезпечує семантичну анотацію даних виробників, що робить дані більш семантичними та сприяє ефективній інтеграції гетерогенних і сумісних даних. Це також дозволяє споживачам отримувати семантично релевантну інформацію, зменшуючи складність для розробників під час інтеграції даних із різних джерел.

3.3 Управління виробниками та споживачами

Сучасна IoT-система, яка претендує на звання складної та ефективної, має бути спроєктована таким чином, щоб забезпечувати підтримку широкого спектру гетерогенних виробників і пропонувати вбудовані інструменти для їхнього управління в рамках своєї архітектури. IoT-пристрої, такі як сенсори, датчики чи виконавчі механізми, нерідко функціонують у складних і навіть екстремальних умовах - наприклад, під впливом високих температур, вологості чи механічних навантажень.

Такі обставини вимагають постійного спостереження за їхнім станом, щоб своєчасно виявляти можливі збої. У разі виникнення проблем може знадобитися заміна пристроїв або їхня модернізація, щоб гарантувати безперебійну роботу в подібних несприятливих середовищах.

Функція управління виробниками в архітектурі ПТС розроблена з урахуванням цих потреб і спрямована на забезпечення безпеки як самих пристроїв, так і даних, які вони генерують. Цей механізм значно полегшує процеси моніторингу, дозволяючи відстежувати працездатність обладнання в реальному часі, а також захищає систему від зовнішніх загроз чи внутрішніх збоїв. Окрім

цього, функції управління надають розробникам IoT гнучкі інструменти для роботи з пристроями. Наприклад, у разі критичних помилок вони можуть повернути пристрій до заводських налаштувань, щоб відновити його базову функціональність, або встановити оновлення програмного забезпечення для усунення вразливостей чи вдосконалення роботи. Такі можливості роблять систему більш адаптивною та стійкою до змін.

У рамках архітектури ПТС активний моніторинг виробників є одним із ключових завдань, яке реалізується за участю так званих споживачів - програмних об’єктів, що відповідають за отримання та обробку даних про стан пристроїв. Ці споживачі виступають посередниками між фізичними виробниками та користувачами системи, забезпечуючи доступ до актуальної інформації. Управління споживачами поділяється на два основні напрями. Перший стосується реєстрації кінцевих користувачів, які отримують доступ до даних, що їх цікавлять, наприклад, показників температури чи вологості з певного пристрою. Другий напрям передбачає створення механізму для фільтрації, обробки та передачі даних зареєстрованим користувачам у зручному вигляді, що оптимізує їхнє використання.

Складність управління IoT-системами зростає пропорційно до кількості підключених виробників, особливо коли вони працюють із різними протоколами (наприклад, MQTT, CoAP) і форматами даних (JSON, XML тощо). Чим більше пристроїв інтегрується в систему, тим критичнішим стає завдання їхнього координованого функціонування. Різноманітність технічних характеристик і стандартів ускладнює моніторинг, обробку даних і забезпечення безпеки, вимагаючи від архітектури ПТС ще більшої гнучкості та надійності. Таким чином, управління виробниками перетворюється на багатогранний процес, який потребує як автоматизованих рішень, так і продуманих стратегій для підтримки стабільності всієї екосистеми IoT.

3.3.1 Служба доступу виробника

Щоб забезпечити управління різнорідною групою виробників із їхніми різними вбудованими протоколами, у рамках архітектури ПТС передбачено сервіс доступу для виробників, який слугує посередником для зв’язку між одним із виробників та архітектурою ПТС. Спочатку виробники мають подати свої профілі для реєстрації у вузлі ПТС. Профіль виробника можна вважати шаблоном або видом класифікації для різнорідних виробників, що включає дані про тип виробника, тип протоколу та тип даних. Докладні відомості про виробників не потрібні, адже вони вже включені до онтології виробника. У таблиці 3.1 представлено набір атрибутів, призначених для опису характеристик виробника.

Таблиця 3.1 – Профіль виробника

Атрибут	Обов’язково/ Необов’язково	Коментар
ID виробника	Обов’язково	Ідентифікатор виробника
Марка	Обов’язково	Марка виробника
Назва	Необов’язково	Назва виробника
Розташування	Необов’язково	Координати виробника
Модель	Необов’язково	Модель виробника
Опис	Необов’язково	Опис виробника

Процес реєстрації виробника відбувається, як зображено на рисунку 3.6, у кілька етапів: спершу виробник звертається за інформацією про доменне ім’я ПТС через DNS-сервіс; після цього DNS-сервіс надає відповідь із IP-адресою вузла ПТС на основі отриманих даних; далі виробник надсилає свій профіль для реєстрації у вузол ПТС, використовуючи цю IP-адресу; вузол ПТС передає профіль до служби семантичної анотації даних для обробки та приведення профілю виробника до потрібного формату; на завершення вузол ПТС підтверджує успішну реєстрацію, після чого виробник може передавати дані до вузла ПТС.

Подібно до згаданого раніше сервісу доступу для виробників, необхідно створити механізм управління для споживачів за допомогою сервісу, відомого як сервіс доступу для споживачів.

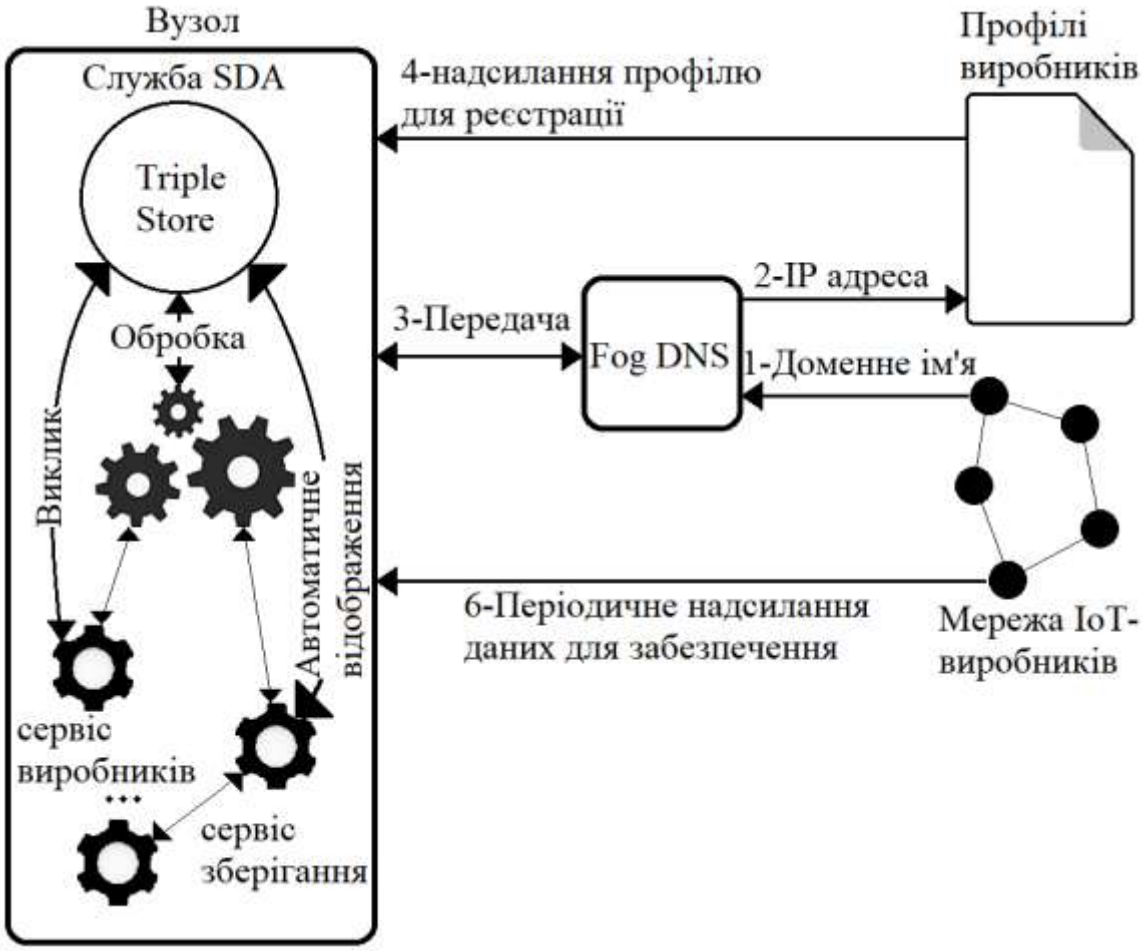


Рисунок 3.6 – Служба реєстрації виробників IoT

На рисунку 3.7 зображена діаграма послідовності, яка охоплює процеси реєстрації та запиту ресурсів у ПТС.

3.3.2 Служба доступу споживача

Застосовується концепція Споживача, визначену в онтології разом із атрибутами та зв’язками з об’єктами даних, що забезпечує доступ до ресурсів вузла ПТС через процедури автентифікації та авторизації.

Таблиця 3.2 – Профіль споживача

Атрибут	Обов’язково/ Необов’язково	Коментар
Назва	Обов’язково	Ідентифікатор виробника
Компанія	Обов’язково	Марка виробника
Електронна пошта	Обов’язково	Назва виробника
Розташування	Обов’язково	Координати виробника
Телефонний номер	Обов’язково	Модель виробника

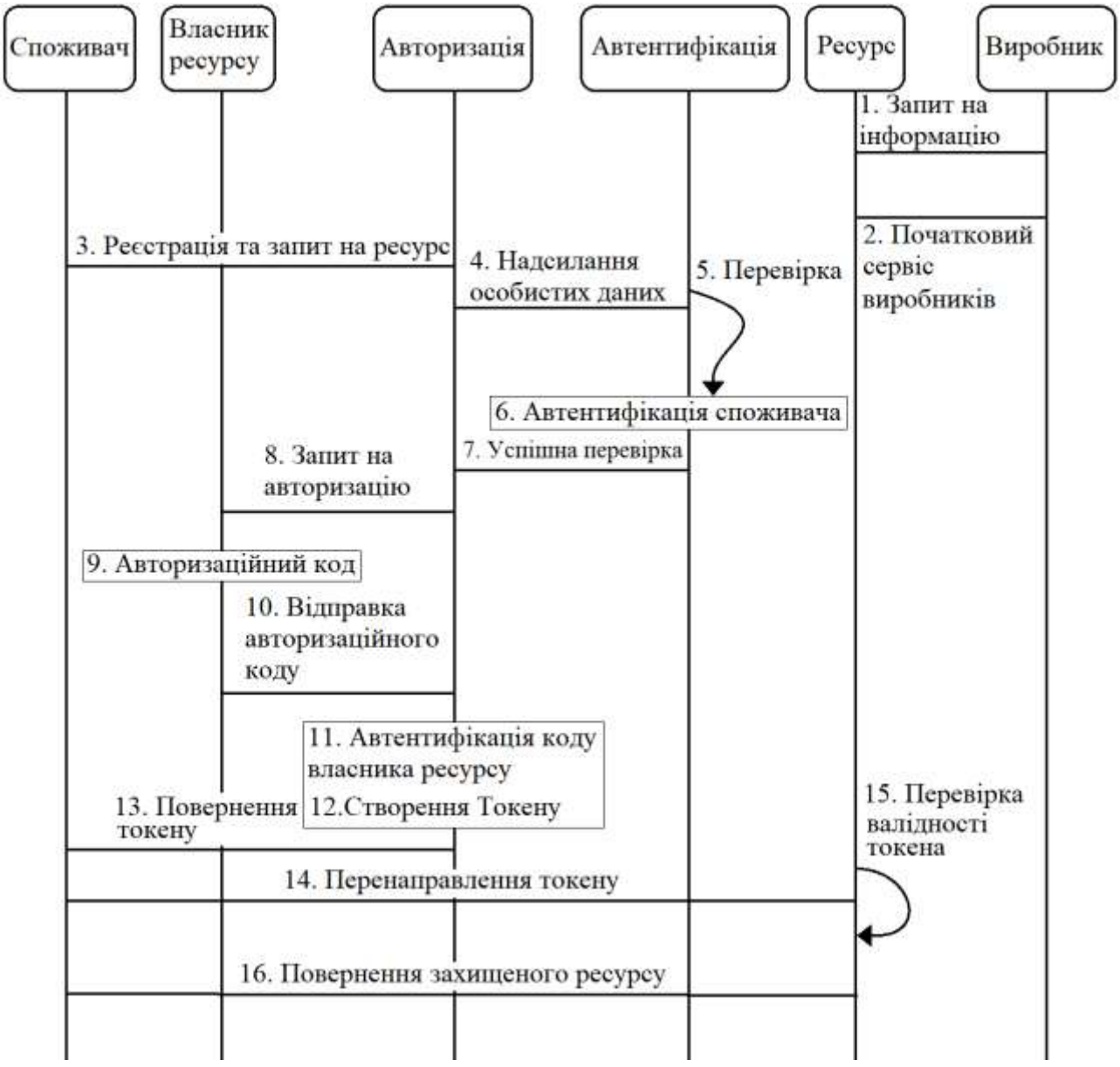


Рисунок 3.7 Процедура реєстрації та доступу для споживачів

Все починається, коли споживач прагне отримати доступ до даних вузла ПТС, пов’язаних із певним виробником.

На етапах із 3 по 13 споживач надсилає запит до служби авторизації для проходження автентифікації.

Служба автентифікації перевіряє легітимність споживача, вимагаючи надати облікові дані, такі як ім’я користувача та пароль.

Ці дані надаються власником ресурсу, тобто кінцевим користувачем, який пройшов реєстрацію через сервіс для споживачів.

Власник ресурсу дозволяє доступ, надсилаючи код авторизації, який передається до служби авторизації для перевірки та видачі токена, що містить інформацію про надану згоду для споживача.

Далі споживач надсилає цей токен до ресурсу ПТС для підтвердження його дійсності та отримання доступу до захищеного ресурсу.

3.4 Висновки до третього розділу

Інтернет речей (IoT) являє собою складну глобальну мережу, що об’єднує різнорідних виробників - пристрої, які виступають джерелами даних, фіксуючи зміни в навколишньому середовищі чи певні події, наприклад, коливання температури, рух об’єктів або рівень енергоспоживання.

Ці дані, що надходять від численних IoT-пристроїв, характеризуються значним обсягом і високою швидкістю генерації, що вимагає їхнього оперативного збору, обробки та використання для прийняття рішень у реальному часі.

Для цього застосовується обчислювальна парадигма, яка дозволяє мінімізувати затримки та оптимізувати ресурси в порівнянні з традиційними централізованими системами. У цьому розділі розглядається метод до управління даними, реалізований у рамках архітектури постачальника туманних сервісів (ПТС), з акцентом на його адаптивність і ефективність.

Запропонований метод до управління даними в ПТС враховує специфіку надання послуг для різних програм, що робить його гнучким і орієнтованим на конкретні потреби. Він базується на двох основних методах організації даних.

Метод передбачає створення окремої бази даних для кожної служби, що дозволяє ізолювати дані залежно від їхнього призначення та підвищити швидкість доступу до них.

Метод використовує спільну базу даних, яка підтримує різноманітні техніки обробки: від збирання та фільтрації необроблених даних до їхньої агрегації, а також моделювання, семантичної анотації та створення запитів до даних із додаванням контекстного значення. Такий двоякий підхід забезпечує як спеціалізацію, так і універсальність у роботі з інформацією.

Для обробки необроблених даних застосовуються процеси збирання, фільтрації та агрегації, які усувають надлишкову інформацію та готують її до подальшого використання.

Водночас для семантичних даних, які потребують глибшого розуміння, використовуються моделювання, анотація за допомогою онтологій і створення запитів, що дозволяють витягувати потрібну інформацію з високою точністю.

Додатково був розроблений спеціальний алгоритм відображення даних, який конвертує різнорідні необроблені дані з різних джерел у компактне й структуроване сховище.

Цей алгоритм спирається на RDF-сховище трійок, що забезпечує ефективні зв’язки між об’єктами даних у форматі суб’єкт-предикат-об’єкт, сприяючи їхній інтеграції та спрощуючи аналіз у реальному часі.

4 РЕАЛІЗАЦІЯ СИСТЕМИ ПОСТАЧАННЯ ТУМАННИХ ПОСЛУГ ДЛЯ КЕРУВАННЯ ДАНИМИ ІНТЕРНЕТУ РЕЧЕЙ

4.1 Постановка експерименту

Розгортання систем постачання туманних послуг у середовищі Інтернету речей супроводжується низкою технічних та організаційних викликів, які пов’язані з необхідністю забезпечення ефективної передачі, обробки та зберігання інформації в умовах значної кількості розподілених джерел даних. Інфраструктура IoT передбачає роботу з великою кількістю гетерогенних пристроїв, що функціонують у режимі реального часу та генерують дані з різною періодичністю, точністю та форматом. За таких умов класичні централізовані підходи, засновані на надсиланні всієї інформації безпосередньо до хмарних обчислювальних ресурсів, не завжди здатні забезпечити задовільні характеристики системи за параметрами швидкості, надійності та продуктивності.

З метою дослідження ефективності застосування туманної архітектури у контексті Інтернету речей було сформульовано експериментальну система, що дозволяє на практиці оцінити потенціал локальної обробки даних на периферії. Основна дослідницька гіпотеза полягає в тому, що інтеграція обчислювального рівня між пристроями збору даних та хмарними сервісами дозволяє суттєво зменшити затримку доставки повідомлень, оптимізувати обсяг мережевого трафіку та знизити навантаження на центральні елементи інфраструктури. Перевірка цієї гіпотези передбачає побудову стенду, в якому порівнюються два сценарії: традиційний, з повною маршрутизацією даних до хмари, та альтернативний, з використанням Fog-вузла для попередньої обробки інформації.

Обраний експериментальний сценарій моделює типову ситуацію, у якій пристрої Інтернету речей здійснюють регулярні вимірювання параметрів навколишнього середовища, наприклад температури та вологості. У класичному випадку ці дані одразу передаються до хмарного серверу, де проходять аналіз та візуалізацію. У розширеній конфігурації дані спочатку надходять до вузла туманної обробки, де здійснюється їх перевірка на коректність, фільтрація,

агрегування за визначеним інтервалом часу, а також семантичне структурування з включенням додаткового контексту. Після завершення попередньої обробки до хмарного середовища передається лише релевантна та оптимізована інформація, що дозволяє суттєво зменшити кількість запитів і підвищити ефективність подальшого аналізу.

У межах дослідження здійснюється оцінка затримки доставки повідомлень від моменту їх генерації на пристрої до моменту реєстрації в базі даних, аналізується обсяг трафіку, що генерується кожним з підходів, а також вивчається поведінка системи за умов зміни навантаження, зокрема при зростанні частоти генерації даних. Важливим аспектом є також стабільність функціонування MQTT-брокера в обох сценаріях, що визначається кількістю активних підключень, частотою обробки повідомлень та часом відповіді.

Для практичної реалізації експерименту було розгорнуто апаратно-програмний комплекс, до складу якого входять мікроконтролери ESP32 із сенсорами DHT22, що формують джерела телеметрії, туманний вузол на базі міні-комп’ютера Raspberry Pi з MQTT-брокером та середовищем Node-RED, а також хмарна інфраструктура у вигляді бази даних InfluxDB та системи візуалізації Grafana. Така конфігурація дозволяє моделювати повний цикл передачі й обробки даних у системах IoT та забезпечує можливість зняття і аналізу всіх необхідних технічних метрик. Додатково у межах дослідження розглядається можливість масштабування архітектури при збільшенні кількості вузлів та навантаження на систему, що дає змогу оцінити її життєздатність у складніших сценаріях використання.

Загалом, запропонована постановка експерименту дозволяє здійснити порівняльний аналіз централізованого і розподіленого підходів до обробки даних у системах Інтернету речей та надати обґрунтовані висновки щодо ефективності впровадження Fog-компонентів як проміжного рівня обчислень у складній мережевій інфраструктурі.

4.2 Побудова експериментального стенду

У рамках реалізації експериментального дослідження було побудовано багаторівневу інфраструктуру, що імітує типову модель функціонування системи керування IoT-даними з використанням туманних обчислень. Запропонована архітектура передбачає послідовний рух даних у напрямку від пристроїв збору інформації до вузлів локальної обробки, а згодом - до централізованого хмарного середовища. Така структура відображає трикомпонентну модель Edge → Fog → Cloud, що забезпечує гнучкість, масштабованість і можливість локального прийняття рішень.

На периферійному рівні, який відповідає за безпосередній контакт із фізичним середовищем, застосовано мікроконтролери ESP32 з підключеним цифровим сенсором температури та вологості DHT22. Ці пристрої функціонували автономно, зчитували показники довкілля з періодичністю в п’ять секунд і надсилали отримані значення до MQTT-брокера у форматі структурованих повідомлень. У якості мережевого середовища використовувалась стандартна Wi-Fi-мережа, що дозволила забезпечити швидкий обмін даними без додаткового мережевого обладнання.

Програмна логіка, реалізована на ESP32, передбачала ініціалізацію сенсора, зчитування поточних значень, формування повідомлення у форматі JSON та його публікацію до MQTT-теми. Нижче наведено фрагмент коду, що ілюструє принцип дії периферійного вузла:

```
String payload = "{\"device\":\"ESP32-01\",";
payload += "\"temperature\":" + String(t) + ",";
payload += "\"humidity\":" + String(h) + ",";
payload += "\"timestamp\":" + String(millis()) + "}";

client.publish("iot/data", (char*) payload.c_str());
```

Сформоване повідомлення містить ідентифікатор пристрою, числові значення температури та вологості, а також часову мітку. Така структура забезпечує зручність подальшої обробки повідомлень, а також їхню семантичну сумісність при інтеграції з іншими вузлами системи або сторонніми сервісами.

Застосування MQTT-протоколу дозволило реалізувати ефективну й надійну передачу даних із мінімальними витратами ресурсів.

Проміжний рівень, відповідальний за обробку, було реалізовано на основі мікрокомп'ютера Raspberry Pi, який виконував функції туманного вузла. На ньому було розгорнуто MQTT-брокер Mosquitto, а також середовище Node-RED, яке використовувалось для побудови потоку обробки повідомлень. У Node-RED реалізовано логіку підписки на MQTT-тему, перевірки коректності отриманих значень, відсіву дубльованих або некоректних записів, агрегації даних за заданий часовий інтервал, а також додавання семантичного контексту. Після завершення локальної обробки оптимізовані повідомлення надсилались через HTTPS-запити до хмарного середовища.

На рівні хмарного зберігання дані записувались до бази InfluxDB, що спеціалізується на обробці часових рядів. Інструмент Grafana, інтегрований із цією базою, забезпечував візуалізацію результатів у вигляді динамічних графіків, що відображають зміни температури та вологості у реальному часі. На рисунку 4.1 наведений спрощений варіант структури стенду.

Рисунок 4.1 – Загальна логічна структура стенду

Реалізований експериментальний стенд дозволив ефективно змоделювати розподілену систему керування даними Інтернету речей, у якій периферійні пристрої функціонують незалежно, а критично важлива обробка здійснюється у вузлі, наближеному до джерел генерації інформації. Такий підхід створює передумови для суттєвого зменшення затримки при передачі даних, оптимізації використання мережевих ресурсів, зниження навантаження на хмарні сервіси, а також підвищення надійності роботи всієї системи у динамічному середовищі.

4.3 Реалізація системи

Реалізація системи постачання туманних послуг для керування даними Інтернету речей базувалась на багаторівневій архітектурі, яка охоплює як пристрої збору даних на периферії, так і проміжну обробку на Fog-рівні та централізоване зберігання у хмарному середовищі. Ключовим аспектом цієї реалізації стало забезпечення повноцінного функціонального ланцюга обробки даних, починаючи від моменту їх генерації на edge-пристроях і закінчуючи візуалізацією та аналізом результатів у хмарі. У такій архітектурі кожен рівень має визначену роль та виконує специфічні завдання, що сприяє гнучкості, масштабованості та стійкості системи. На рисунку 4.2 показаний спрощений варіант передачі даних у системі

На рівні периферії використовувалися мікроконтролери ESP32 із підключеними сенсорами DHT22, які виконували безперервний моніторинг температури та вологості повітря. Дані зчитувались з частотою, визначеною програмно, та передавались до MQTT-брокера у локальній мережі у форматі JSON. Повідомлення включали ключові атрибути, зокрема ідентифікатор пристрою, поточні значення параметрів та часову мітку, що забезпечувало їхній подальший аналіз і сортування.

Для реалізації обміну даними та управління потоками було використано протокол MQTT як один із найефективніших для IoT-середовища завдяки низькому енергоспоживанню, малій затримці та підтримці моделі "publish–subscribe". MQTT-брокер Mosquitto розгортався на мінікомп’ютері Raspberry Pi, який виступав у ролі туманного вузла. Саме на цьому рівні реалізовано базові алгоритми обробки даних, зокрема фільтрацію некоректних повідомлень, агрегацію вимірів за певний часовий інтервал та семантичне збагачення. Для візуального проектування логіки обробки використовувалося середовище Node-RED, яке дозволило гнучко налаштувати послідовність дій: підписку на тему MQTT, розбір JSON-структури, виконання перевірки значень, формування агрегованого запису та передачу його до хмари через HTTP-запити.

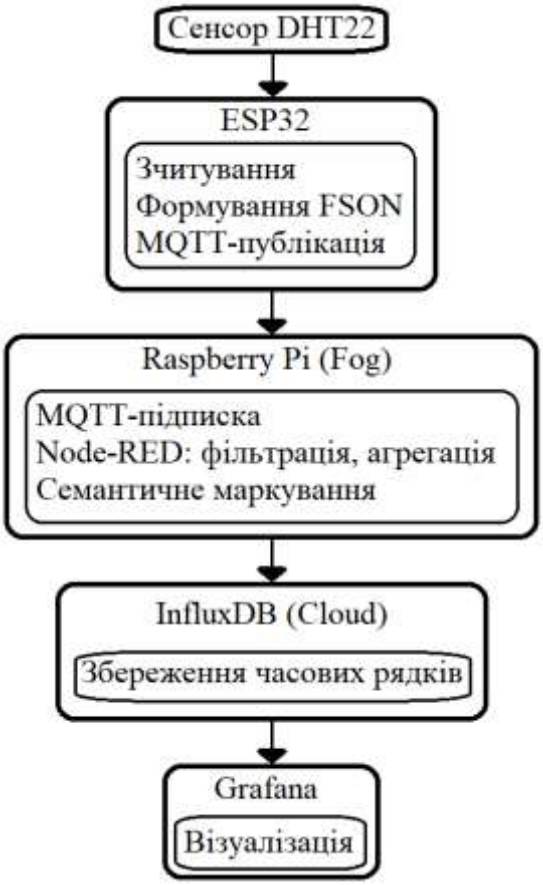


Рисунок 4.2 – Логічний ланцюг передачі даних у системі

Окрему увагу в реалізації було приділено семантичному маркуванню повідомлень. У рамках цієї задачі до структури JSON-повідомлення включались додаткові елементи, що описували одиниці вимірювання, географічне положення, тип сенсора та інші параметри контексту. Такий підхід забезпечує структурованість даних та їхню придатність до обробки не лише в системах візуалізації, а й у більш складних інструментах аналітики.

Після обробки на Fog-рівні повідомлення передавались у хмарну інфраструктуру, де дані записувалися до бази InfluxDB, що оптимізована для роботи з часовими рядами. Надалі ці дані виводилися у системі Grafana, яка дозволила створити адаптивні інформаційні панелі з можливістю відстеження динаміки параметрів у реальному часі. Це забезпечувало повноцінний зворотний зв’язок для користувача або системи автоматизованого управління.

Програмна реалізація системи виконувалася з урахуванням потреб у масштабуванні та сумісності з іншими модулями. Усі компоненти - від

мікропрограмного коду ESP32 до обробки в Node-RED - мали відкриту структуру та могли бути адаптовані для розширення кількості пристроїв або зміни логіки обробки. Завдяки цьому запропонована система є не лише функціонально завершеною, а й придатною до подальшої еволюції відповідно до зростаючих вимог до IoT-інфраструктури.

Таблиця 4.1 – Структура семантично збагаченого JSON-повідомлення

Поле	Опис
device	Унікальний ідентифікатор
temperature	Температура в °C
humidity	Вологість у %
timestamp	Час вимірювання (мс)
unit_temp	Одиниця температури (°C)
unit_humidity	Одиниця вологості (%)
location	Географічне положення вузла
sensor_type	Тип сенсора (DHT22)

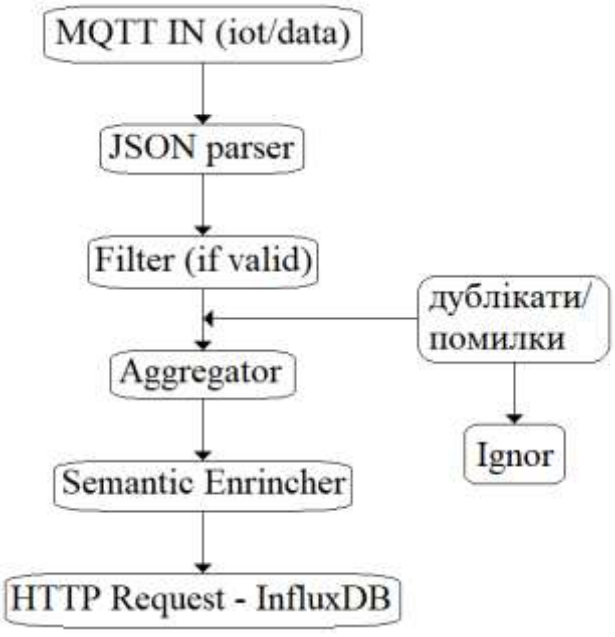


Рисунок 4.3 – Потік обробки повідомлень у Node-RED

Таким чином, реалізована архітектура демонструє здатність туманних обчислень ефективно інтегруватися у середовище Інтернету речей, забезпечуючи локальну обробку даних, зменшення навантаження на хмарні сервіси та загальне підвищення адаптивності та надійності цифрової екосистеми.

4.4 Проведення експерименту

Після розгортання системи та налаштування усіх її компонентів було проведено серію експериментів з метою кількісного і якісного порівняння ефективності функціонування туманної архітектури у протиставленні до традиційної моделі централізованої обробки даних. В основі експерименту лежала задача оцінювання ключових характеристик системи в умовах, що моделюють типове навантаження, притаманне реальному середовищу Інтернету речей.

Для цього було обрано два базові сценарії: перший передбачав надсилання необроблених телеметричних повідомлень безпосередньо з edge-пристроїв до хмарного сховища, другий передбачав попередню локальну обробку даних на рівні Fog-вузла перед їх передачею в хмару.

В обох сценаріях використовувалися ідентичні периферійні пристрої - ESP32 з сенсорами DHT22, які формували телеметричний потік у вигляді структурованих JSON-повідомлень із частотою публікацій раз на п’ять секунд. У першому випадку повідомлення з кожного пристрою надсилались безпосередньо до MQTT-брокера, після чого потрапляли до хмарної бази даних InfluxDB, де зберігалися у вигляді часових рядів. Відповідні графіки виводилися у Grafana без попередньої обробки.

У другому сценарії між джерелом даних та хмарою функціонував туманний вузол на Raspberry Pi, що здійснював попередню перевірку, фільтрацію і агрегацію значень, а також формував узагальнені записи з урахуванням контекстної інформації. До хмари надходили вже оброблені повідомлення, які містили середнє значення параметра за хвилину та супутню метаінформацію.

Під час проведення експерименту фіксувалися три основні метрики: середній час доставки повідомлення від генерації до запису у хмару, обсяг переданих даних

за фіксований часовий проміжок, а також рівень навантаження на MQTT-брокер, визначений шляхом спостереження за кількістю оброблених повідомлень на одиницю часу. Крім того, проводився моніторинг загального функціонального стану системи - наявність затримок, випадків втрати пакетів, спотворення даних або нестабільності при високому навантаженні. У таблиці 4.2 наведені результати проведеного експерименту.

Таблиця 4.2 – Порівняння продуктивності систем

Показник	Без Fog-обробки	З Fog-обробкою
Середня затримка (мс)	342	81
Обсяг переданих даних (МБ)	56.3	11.8
Навантаження на брокер	Високе	Помірне

За результатами дослідження було зафіксовано чітку перевагу архітектури з туманним вузлом за всіма ключовими показниками. У конфігурації без Fog-обробки середній час затримки становив понад 300 мілісекунд, що зумовлювалося необхідністю прямого звернення до хмарного сервера, тоді як при локальній обробці цей показник знижувався до менш ніж 100 мілісекунд. Водночас загальний обсяг переданих даних за десятихвилинний період зменшувався більш ніж у п’ять разів - завдяки попередній агрегації та фільтрації інформації. Навантаження на брокер також знижувалося, що дозволяло системі зберігати стабільність навіть за умов зростання частоти публікацій до одного повідомлення щосекунди.

Крім того, було зафіксовано підвищення стійкості до збоїв у другому сценарії. У момент пікових навантажень у системі без туманного рівня спостерігалися випадки затримок з візуалізацією, тимчасове перевантаження брокера та поява нерівномірних інтервалів у часових рядах. У конфігурації з Fog-обробкою таких ефектів не виявлено, що свідчить про більшу адаптивність системи до зміни умов.

Таким чином, проведений експеримент дозволив не лише продемонструвати принцип дії архітектури з туманною обробкою, але й кількісно підтвердити її

переваги у контексті практичного використання. Результати експерименту лягли в основу подальшого аналізу, спрямованого на оцінку перспектив такого підходу у розподілених цифрових екосистемах.

4.5 Аналіз результатів

Експериментальне дослідження, проведене в межах реалізованої системи постачання туманних послуг для керування даними Інтернету речей, дозволило отримати репрезентативні кількісні та якісні показники, що стали основою для подальшого аналітичного осмислення функціональних властивостей туманної архітектури.

Отримані результати підтверджують попередньо сформульовану гіпотезу щодо переваг використання Fog-рівня в умовах розподіленої обробки даних, особливо у випадках, коли критичною є затримка доставки повідомлень або існують обмеження на пропускну здатність мережевих каналів.

Найбільш показовим параметром виступає середній час затримки між генерацією повідомлення на периферійному пристрої та його записом до хмарного сховища. У конфігурації, що не передбачає використання Fog-вузла, затримка була суттєво вищою, що пов’язано із накопиченням черг на рівні хмарного брокера, особливо у випадках з підвищеним навантаженням. У той час як за умов використання туманної обробки спостерігалося більш рівномірне розподілення трафіку, скорочення кількості транзакцій та, відповідно, зменшення загального часу реакції системи.

Це демонструє ефективність локального фільтрування даних та зменшення впливу зовнішніх чинників, таких як затори у мережі або затримки при маршрутизації пакетів.

Не менш важливим є зменшення обсягу переданої інформації при використанні Fog-підходу. Завдяки агрегації даних, що відбувається на рівні туманного вузла, до хмарного середовища передається лише стисла й опрацьована інформація, яка вже позбавлена повторюваних або зайвих записів. Це дозволяє

значно скоротити навантаження на канали зв’язку та зменшити обсяг дискового простору, необхідного для зберігання великих обсягів необробленої телеметрії.

Порівняльний аналіз показав, що у туманній конфігурації обсяг даних знизився більш ніж у п’ять разів, при цьому інформативність та цінність збережених записів не постраждали.

Таблиця 4.3 – Порівняння стабільності роботи системи

Умова	Без For-обробки	З Fog-обробкою
Затримка при піковому навантаженні	Часті	Мінімальні
Втрата повідомлень	Можлива	Не зафіксовано
Стабільність графіків	Нерівномірна	Плавна та без стрибків
Адаптивність до навантажень	Обмежена	Висока

Окрему увагу заслуговує навантаження на брокер MQTT. У сценарії з відсутністю попередньої обробки брокер приймав усі повідомлення від пристроїв безпосередньо, що призводило до швидкого зростання навантаження на сервер, особливо при збільшенні кількості підключених вузлів. У протилежному випадку, коли трафік агрегувався на Fog-рівні, кількість публікацій у брокері істотно скоротилася, що позитивно вплинуло на стабільність системи. Це, зокрема, підтверджується меншою кількістю помилок, пов’язаних із втратами пакетів, а також стабільним рівнем обробки повідомлень навіть за умов інтенсивного навантаження.

З технічної точки зору також спостерігалося покращення стабільності роботи всієї системи в цілому. Туманна архітектура забезпечила адаптивність до зміни навантаження, у тому числі у випадках, коли частота надсилання повідомлень зростала у кілька разів. У таких умовах традиційна хмарна модель демонструвала тенденцію до уповільнення та накопичення затримок, тоді як за наявності попередньої обробки на периферії система залишалася стабільною та чітко контрольованою.

Не менш важливим фактором є й семантична анотація даних, реалізована на Fog-рівні. Завдяки додатковому структуруванню повідомлень та включенню контекстної інформації, значно спростився процес подальшого аналізу у хмарній частині системи. Стандартизований формат повідомлень дозволив швидко налаштувати дашборди в Grafana та забезпечити гнучке представлення даних за різними ознаками - як у часовому, так і у просторовому вимірі. Це, своєю чергою, підвищує оперативність аналізу та полегшує інтеграцію з іншими аналітичними або автоматизованими компонентами.

Отже, аналіз результатів експерименту дає підстави стверджувати, що туманна архітектура має суттєві переваги в контексті підвищення ефективності обробки IoT-даних. Її використання дозволяє досягти більшої адаптивності системи, знизити затрати на обчислювальні ресурси та підвищити надійність функціонування за умов високої динаміки середовища.

4.6 Висновки до четвертого розділу

Розробка, реалізація та експериментальна перевірка системи постачання туманних послуг для керування даними Інтернету речей дозволили зробити низку обґрунтованих висновків щодо ефективності впровадження Fog-архітектури у розподілені інформаційні системи. Створений експериментальний стенд, що поєднує периферійні пристрої збору даних, проміжний рівень локальної обробки та хмарну інфраструктуру, забезпечив можливість відтворити повноцінну модель функціонування IoT-системи в умовах реального часу. Проведене тестування підтвердило, що застосування туманного обчислювального рівня є доцільним як з погляду технічної реалізованості, так і з огляду на функціональні переваги, які він надає.

Аналіз експериментальних даних показав суттєве зниження затримки при доставці повідомлень у порівнянні з традиційною централізованою моделлю. Завдяки локальній обробці, що передує передаванню до хмари, вдалося мінімізувати час реакції системи на події, що особливо важливо у сценаріях з

високими вимогами до оперативності, таких як моніторинг критичних показників у промисловості або системах охорони здоров’я. Крім того, реалізація механізмів фільтрації та агрегування даних на Fog-рівні дозволила значно зменшити обсяг трафіку, який циркулює у мережі, що своєю чергою сприяє економії ресурсів та підвищенню загальної стабільності функціонування інфраструктури.

Зниження навантаження на MQTT-брокер завдяки зменшенню кількості публікацій також підтвердило ефективність архітектури з проміжною обробкою. Система, що використовує Fog-вузол, виявилася менш чутливою до змін навантаження та здатною функціонувати стабільно навіть за умов різкого зростання інтенсивності передачі даних. Така властивість відкриває перспективи масштабування розробленого рішення на більші системи з десятками або сотнями підключених пристроїв, не ризикуючи втратити цілісність потоку даних або допустити істотні затримки.

Окреме значення має впровадження семантичного підходу до структурування повідомлень, яке здійснюється на рівні туманного вузла. Завдяки цьому стало можливим не лише скоротити обсяг переданої інформації, але й зробити її більш осмисленою з погляду подальшої обробки та аналізу. Контекстні теги, включені до повідомлень, спростили побудову інформаційних панелей і надали змогу більш гнучко оперувати отриманими даними у хмарній інфраструктурі, що є критичним для побудови адаптивних і масштабованих інформаційних систем.

У підсумку можна констатувати, що експериментальна реалізація системи підтвердила життєздатність запропонованої архітектури та довела доцільність впровадження туманних обчислень у сучасні IoT-системи. Побудована модель забезпечує не лише технічну ефективність, але й створює базу для подальшого розвитку в напрямку інтелектуалізації обробки даних, автоматизованого управління та адаптивного реагування у розподілених середовищах. Таким чином, результати даного етапу дослідження становлять вагоме підґрунтя для наступних кроків з удосконалення систем управління потоками IoT-даних у багаторівневих цифрових екосистемах.

ВИСНОВКИ

У роботі за результатами виконаних теоретичних та практичних досліджень розроблено систему постачання туманних послуг для керування даними Інтернету речей, що забезпечує ефективну маршрутизацію, зберігання та семантичну обробку даних у розподіленому середовищі Fog Computing. Запропоновано підхід, який дозволяє оптимізувати використання ресурсів, зменшити затримки обробки інформації та підвищити адаптивність системи до змін у потоці IoT-даних.

У першому розділі проаналізовано поточний стан досліджень у сфері IoT, хмарних та туманних обчислень. Визначено ключові проблеми обробки великих обсягів даних у розподілених середовищах, а також обґрунтовано доцільність використання туманних обчислень для вирішення завдань управління IoT-даними.

У другому розділі розроблено загальну архітектуру системи постачання туманних послуг. Визначено компоненти, їхні функції та взаємозв’язки між ними. Особливу увагу приділено ролі обробників запитів, семантичного модуля, засобів взаємодії з базами даних та механізмів моніторингу.

У третьому розділі запропоновано метод семантичної обробки даних, який включає анотацію вхідних повідомлень та формування семантично релевантних запитів. Також розглянуто механізми адаптивного керування маршрутами даних та процедури оновлення інформації в базі знань відповідно до поточних потреб системи.

У четвертому розділі описано програмну реалізацію розробленої системи. Наведено основні фрагменти коду, схеми обробки запитів та приклади роботи. Проведено тестування ефективності функціонування системи у порівнянні з базовими підходами. Результати експериментів підтверджують зменшення затримки при обробці запитів, покращення точності інформаційної відповідності та ефективне використання обчислювальних ресурсів.

Набула подальшого розвитку інформаційна технологія керування розподіленими даними в умовах туманних обчислень, яка базується на поєднанні методів семантичної обробки, індексації повідомлень та динамічного управління

маршрутами постачання даних. Запропоноване рішення враховує контекст запиту, тип даних і призначення вузла, що дозволяє досягти гнучкості й масштабованості системи.

Впровадження результатів роботи дозволило підвищити ефективність функціонування інформаційних систем у середовищі IoT, зменшити навантаження на хмарну інфраструктуру, покращити швидкодію обробки критичних запитів, а також адаптувати систему до гетерогенного середовища з високою динамікою зміни даних.

За темою кваліфікаційної роботи магістра опублікована одна стаття у фаховому науковому виданні, яка відображає основні наукові результати дослідження та підтверджує їхню актуальність і практичну цінність.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ

1. Das R. Inuwa M.M. A review on fog computing: Issues, characteristics, challenges, and potential applications. Telematics and Informatics Reports. 2023. 10. pp. 100049.

2. Hazra A. Rana P. Adhikari M. Amgoth T. Fog computing for next-generation internet of things: fundamental, state-of-the-art and research challenges. Computer Science Review. 2023. 48. pp. 100549.

3. Costa B. Bachiega Jr J. De Carvalho L.R. Araujo A.P. Orchestration in fog computing: A comprehensive survey. ACM Computing Surveys (CSUR). 2022. 55(2). pp. 1–34.

4. Srirama S.N. A decade of research in fog computing: relevance, challenges, and future directions. Software: Practice and Experience. 2024. 54(1). pp. 3–23.

5. Ometov A. Molua O.L. Komarov M. Nurmi J. A survey of security in cloud, edge, and fog computing. Sensors. 2022. 22(3). pp. 927.

6. Alsadie D. Artificial intelligence techniques for securing fog computing environments: trends, challenges, and future directions. IEEE Access. 2024.

7. Wang Z. Goudarzi M. Gong M. Buyya R. Deep reinforcement learning-based scheduling for optimizing system load and response time in edge and fog computing environments. Future Generation Computer Systems. 2024. 152. pp. 55–69.

8. Dhanalakshmi M. Tamilarasi K. Saravanan S. Sujatha G. Boopathi S. Fog computing-based framework and solutions for intelligent systems: Enabling autonomy in vehicles. Computational Intelligence for Green Cloud Computing and Digital Waste Management. IGI Global. 2024. pp. 330–356.

9. Nazih O. Benamar N. Lamaazi H. Chaoui H. Toward secure and trustworthy vehicular fog computing: A survey. IEEE Access. 2024. 12. pp. 35154–35171.

10. Laroui M. Nour B. Moungla H. Cherif M.A. Afifi H. Guizani M. Edge and fog computing for IoT: A survey on current research activities & future directions. Computer Communications. 2021. 180. pp. 210–231.

11. Singh J. Singh P. Gill S.S. Fog computing: A taxonomy, systematic review, current trends and research challenges. Journal of Parallel and Distributed Computing. 2021. 157. pp. 56–85.

12. Alzoubi Y.I. Osmanaj V.H. Jaradat A. Al-Ahmad A. Fog computing security and privacy for the Internet of Thing applications: State-of-the-art. Security and Privacy. 2021. 4(2). pp. e145.

13. Abd Elaziz M. Abualigah L. Attiya I. Advanced optimization technique for scheduling IoT tasks in cloud-fog computing environments. Future Generation Computer Systems. 2021. 124. pp. 142–154.

14. Songhorabadi M. Rahimi M. MoghadamFarid A. Kashani M.H. Fog computing approaches in IoT-enabled smart cities. Journal of Network and Computer Applications. 2023. 211. pp. 103557.

15. Asghari A. Sohrabi M.K. Server placement in mobile cloud computing: A comprehensive survey for edge computing, fog computing and cloudlet. Computer Science Review. 2024. 51. pp. 100616.

16. Hussein W.N. Hussain H.N. Hussain H.N. Mallah A.Q. A deployment model for IoT devices based on fog computing for data management and analysis. Wireless Personal Communications. 2023. pp. 1–13.

17. Goudarzi M. Palaniswami M. Buyya R. Scheduling IoT applications in edge and fog computing environments: a taxonomy and future directions. ACM Computing Surveys. 2022. 55(7). pp. 1–41.

18. Srirama S.N. A decade of research in fog computing: relevance, challenges, and future directions. Software: Practice and Experience. 2024. 54(1). pp. 3–23.

19. Alwakeel A.M. An overview of fog computing and edge computing security and privacy issues. Sensors. 2021. 21(24). pp. 8226.

20. Ometov A. Molua O.L. Komarov M. Nurmi J. A survey of security in cloud, edge, and fog computing. Sensors. 2022. 22(3). pp. 927.

21. Singh J. Singh P. Gill S.S. Fog computing: A taxonomy, systematic review, current trends and research challenges. Journal of Parallel and Distributed Computing. 2021. 157. pp. 56–85.

22. Kaur J. Verma R. Alharbe N.R. Agrawal A. Khan R.A. Importance of fog computing in healthcare 4.0. Fog Computing for Healthcare 4.0 Environments: Technical, Societal, and Future Implications. 2021. pp. 79–101.

23. Songhorabadi M. Rahimi M. MoghadamFarid A. Kashani M.H. Fog computing approaches in IoT-enabled smart cities. Journal of Network and Computer Applications. 2023. 211. pp. 103557.

24. Rani S. Kataria A. Chauhan M. Fog computing in industry 4.0: Applications and challenges-A research roadmap. Energy Conservation Solutions for Fog-Edge Computing Paradigms. 2022. pp. 173–190.

25. Ahmed K.D. Zeebaree S.R. Resource allocation in fog computing: A review. International Journal of Science and Business. 2021. 5(2). pp. 54–63.

26. Rawat R. Chakrawarti R.K. Vyas P. Gonzáles J.L.A. Sikarwar R. Bhardwaj R. Intelligent fog computing surveillance system for crime and vulnerability identification and tracing. International Journal of Information Security and Privacy (IJISP). 2023. 17(1). pp. 1–25.

27. Rani R. Kumar N. Khurana M. Kumar A. Barnawi A. Storage as a service in fog computing: A systematic review. Journal of Systems Architecture. 2021. 116. pp. 102033.

28. Sarrafzade N. Entezari-Maleki R. Sousa L. A genetic-based approach for service placement in fog computing. The Journal of Supercomputing. 2022. 78(8). pp. 10854–10875.

29. Tiwari A. Sharma R.M. Realm Towards Service Optimization in Fog Computing. Research Anthology on Architectures, Frameworks, and Integration Strategies for Distributed and Cloud Computing. IGI Global. 2021. pp. 1530–1563.

30. Muniswamaiah M. Agerwala T. Tappert C.C. Fog computing and the internet of things (IoT): a review. Proceedings of the 2021 8th IEEE International Conference on Cyber Security and Cloud Computing (CSCloud)/2021 7th IEEE International Conference on Edge Computing and Scalable Cloud (EdgeCom). IEEE. 2021. pp. 10–12.

31. Ogundoyin S.O. Kamil I.A. Optimization techniques and applications in fog computing: An exhaustive survey. Swarm and Evolutionary Computation. 2021. 66. pp. 100937.

32. Kumari N. Yadav A. Jana P.K. Task offloading in fog computing: A survey of algorithms and optimization techniques. Computer Networks. 2022. 214. pp. 109137.

33. Potu N. Jatoth C. Parvataneni P. Optimizing resource scheduling based on extended particle swarm optimization in fog computing environments. Concurrency and Computation: Practice and Experience. 2021. 33(23). pp. e6163.

34. Abd Elaziz M. Abualigah L. Ibrahim R.A. Attiya I. IoT workflow scheduling using intelligent arithmetic optimization algorithm in fog computing. Computational Intelligence and Neuroscience. 2021. 2021(1). pp. 9114113.

35. Wang S. Ruan Y. Tu Y. Wagle S. Brinton C.G. Joe-Wong C. Network-aware optimization of distributed learning for fog computing. IEEE/ACM Transactions on Networking. 2021. 29(5). pp. 2019–2032.

36. Liang Y.C. Li W.D. Lu X. Wang S. Fog computing and convolutional neural network enabled prognosis for machining process optimization. Data Driven Smart Manufacturing Technologies and Applications. 2021. pp. 13–35.

37. Oprea S.V. Bâra A. Edge and fog computing using IoT for direct load optimization and control with flexibility services for citizen energy communities. Knowledge-Based Systems. 2021. 228. pp. 107293.

38. Liu Y. Zhang H. Long K. Zhou H. Leung V.C. Fog computing vehicular network resource management based on chemical reaction optimization. IEEE Transactions on Vehicular Technology. 2021. 70(2). pp. 1770–1781.

39. Singh S.P. Effective load balancing strategy using fuzzy golden eagle optimization in fog computing environment. Sustainable Computing: Informatics and Systems. 2022. 35. pp. 100766.

40. Xiong K. Liu Y. Zhang L. Gao B. Cao J. Fan P. Letaief K.B. Joint optimization of trajectory, task offloading, and CPU control in UAV-assisted wireless powered fog computing networks. IEEE Transactions on Green Communications and Networking. 2022. 6(3). pp. 1833–1845.

41. Karthik S.S. Kavithamani A. Fog computing-based deep learning model for optimization of microgrid-connected WSN with load balancing. Wireless Networks. 2021. 27(4). pp. 2719–2727.

42. Abdel-Basset M. Moustafa N. Mohamed R. Elkomy O.M. Abouhawwash M. Multi-objective task scheduling approach for fog computing. IEEE Access. 2021. 9. pp. 126988–127009.

43. Ghobaei-Arani M. Shahidinejad A. A cost-efficient IoT service placement approach using whale optimization algorithm in fog computing environment. Expert Systems with Applications. 2022. 200. pp. 117012.

44. Saif F.A. Latip R. Hanapi Z.M. Shafinah K. Multi-objective grey wolf optimizer algorithm for task scheduling in cloud-fog computing. IEEE Access. 2023. 11. pp. 20635–20646.

45. Kashani M.H. Mahdipour E. Load balancing algorithms in fog computing. IEEE Transactions on Services Computing. 2022. 16(2). pp. 1505–1521.

46. Ramzanpoor Y. Hosseini Shirvani M. Golsorkhtabaramiri M. Multi-objective fault-tolerant optimization algorithm for deployment of IoT applications on fog computing infrastructure. Complex & Intelligent Systems. 2022. 8(1). pp. 361–392.

47. Liu Y. Fieldsend J.E. Min G. A framework of fog computing: Architecture, challenges, and optimization. IEEE Access. 2017. 5. pp. 25445–25454.

48. Raghavendra M.S. Chawla P. Rana A. A survey of optimization algorithms for fog computing service placement. Proceedings of the 2020 8th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO). IEEE. 2020. pp. 259–262.

49. Goswami A. Modi K. Patel C. Evaluation of optimization algorithm for application placement problem in fog computing: A systematic review. Archives of Computational Methods in Engineering. 2025. pp. 1–29.

50. Qiu M. Kung S.Y. Gai K. Intelligent security and optimization in edge/fog computing. Future Generation Computer Systems. 2020. 107. pp. 1140–1142.

51. Kishor A. Chakarbarty C. Task offloading in fog computing for using smart ant colony optimization. Wireless Personal Communications. 2022. 127(2). pp. 1683–1704.

52. Reyana A. Kautish S. Alnowibet K.A. Zawbaa H.M. Wagdy Mohamed A. Opportunities of IoT in fog computing for high fault tolerance and sustainable energy optimization. Sustainability. 2023. 15(11). pp. 8702.

53. Ogundoyin S.O. Kamil I.A. Optimal fog node selection based on hybrid particle swarm optimization and firefly algorithm in dynamic fog computing services. Engineering Applications of Artificial Intelligence. 2023. 121. pp. 105998.

54. Barzegaran M. Cervin A. Pop P. Performance optimization of control applications on fog computing platforms using scheduling and isolation. IEEE Access. 2020. 8. pp. 104085–104098.

55. Guerrero C. Lera I. Juiz C. Genetic-based optimization in fog computing: Current trends and research opportunities. Swarm and Evolutionary Computation. 2022. 72. pp. 101094.

56. Apat H.K. Nayak R. Sahoo B. A comprehensive review on Internet of Things application placement in Fog computing environment. Internet of Things. 2023. 23. pp. 100866.

57. Chalapathi G.S.S. Chamola V. Vaish A. Buyya R. Industrial internet of things (IIoT) applications of edge and fog computing: A review and future directions. Fog/Edge Computing for Security, Privacy, and Applications. 2021. pp. 293–325.

58. Shukla S. Thakur S. Hussain S. Breslin J.G. Jameel S.M. Identification and authentication in healthcare Internet-of-Things using integrated fog computing based blockchain model. Internet of Things. 2021. 15. pp. 100422.

59. Sadri A.A. Rahmani A.M. Saberikamarposhti M. Hosseinzadeh M. Data reduction in fog computing and Internet of Things: A systematic literature survey. Internet of Things. 2022. 20. pp. 100629.

60. Fersi G. Fog computing and Internet of Things in one building block: A survey and an overview of interacting technologies. Cluster Computing. 2021. 24(4). pp. 2757–2787.

61. Bhardwaj K.K. Banyal S. Sharma D.K. Al-Numay W. Internet of Things based smart city design using fog computing and fuzzy logic. Sustainable Cities and Society. 2022. 79. pp. 103712.

62. Gowda D. Sharma A. Rao B.K. Shankar R. Sarma P. Chaturvedi A. Hussain N. Industrial quality healthcare services using Internet of Things and fog computing approach. Measurement: Sensors. 2022. 24. pp. 100517.

63. Kumar P. Gupta G.P. Tripathi R. A distributed ensemble design based intrusion detection system using fog computing to protect the Internet of Things networks. Journal of Ambient Intelligence and Humanized Computing. 2021. 12(10). pp. 9555–9572.

64. Lai K.L. Chen J.I.Z. Zong J.I. Development of smart cities with fog computing and Internet of Things. Journal of Ubiquitous Computing and Communication Technologies (UCCT). 2021. 3(01). pp. 52–60.

65. Liu Y. Zhang J. Zhan J. Privacy protection for fog computing and the Internet of Things data based on blockchain. Cluster Computing. 2021. 24(2). pp. 1331–1345.

66. Sabireen H. Neelanarayanan V.J.I.E. A review on fog computing: Architecture, fog with IoT, algorithms and research challenges. ICT Express. 2021. 7(2). pp. 162–176.

67. Gulatas I. Kilinc H.H. Zaim A.H. Aydin M.A. Malware threat on edge/fog computing environments from Internet of Things devices perspective. IEEE Access. 2023. 11. pp. 33584–33606.

68. Saeed W. Ahmad Z. Jehangiri A.I. Mohamed N. Umar A.I. Ahmad J. A fault tolerant data management scheme for healthcare Internet of Things in fog computing. KSII Transactions on Internet and Information Systems (TIIS). 2021. 15(1). pp. 35–57.

69. Roy V. An Effective FOG Computing Based Distributed Forecasting of Cyber-Attacks in Internet of Things. Journal of Cybersecurity & Information Management. 2023. 12(2).

70. Xu S. Ning J. Ma J. Huang X. Pang H.H. Deng R.H. Expressive bilateral access control for Internet-of-Things in cloud-fog computing. Proceedings of the 26th ACM Symposium on Access Control Models and Technologies. 2021. pp. 143–154.

71. Mayer A.H. Rodrigues V.F. da Costa C.A. da Rosa Righi R. Roehrs A. Antunes R.S. Fogchain: a fog computing architecture integrating blockchain and Internet of Things for personal health records. IEEE Access. 2021. 9. pp. 122723–122737.

72. Ahlawat C. Krishnamurthi R. Towards smart technologies with integration of the Internet of Things, cloud computing, and fog computing. International Journal of Networking and Virtual Organisations. 2023. 29(1). pp. 73–124.

73. Aljumah A. Kaur A. Bhatia M. Ahamed Ahanger T. Internet of Things-fog computing-based framework for smart disaster management. Transactions on Emerging Telecommunications Technologies. 2021. 32(8). pp. e4078.

74. Alamer A. Security and privacy-awareness in a software-defined fog computing network for the Internet of Things. Optical Switching and Networking. 2021. 41. pp. 100616.

75. Gupta S. Singh N. Toward intelligent resource management in dynamic fog computing-based Internet of Things environment with deep reinforcement learning: A survey. International Journal of Communication Systems. 2023. 36(4). pp. e5411.

76. Ahuja S.P. Deval N. From cloud computing to fog computing: Platforms for the Internet of Things (IoT). Research Anthology on Architectures, Frameworks, and Integration Strategies for Distributed and Cloud Computing. 2021. pp. 999–1010.

77. Islam S. Jamwal S. Mir M.H. Leveraging fog computing for smart Internet of Things crop monitoring farming in COVID-19 era. Annals of the Romanian Society for Cell Biology. 2021. 25(6). pp. 10410–10420.

78. Najafizadeh A. Salajegheh A. Rahmani A.M. Sahafi A. Privacy-preserving for the Internet of Things in multi-objective task scheduling in cloud-fog computing using goal programming approach. Peer-to-Peer Networking and Applications. 2021. 14(6). pp. 3865–3890.

79. Teoh Y.K. Gill S.S. Parlikad A.K. IoT and fog-computing-based predictive maintenance model for effective asset management in Industry 4.0 using machine learning. IEEE Internet of Things Journal. 2021. 10(3). pp. 2087–2094.

80. Malathy N. Revathi T. Opposition-based improved memetic algorithm for placement of concurrent Internet of Things applications in fog computing. Transactions on Emerging Telecommunications Technologies. 2024. 35(2). pp. e4941.

81. Abdulqadir H.R. Zeebaree S.R. Shukur H.M. Sadeeq M.M. Salim B.W. Salih A.A. Kak S.F. A study of moving from cloud computing to fog computing. Qubahan Academic Journal. 2021. 1(2). pp. 60–70.

82. Aiswarya S. Ramesh K. Prabha B. Sasikumar S. Vijayakumar K. A time optimization model for the Internet of Things-based healthcare system using fog computing. In: 2021 International Conference on Innovative Computing, Intelligent Communication and Smart Electrical Systems (ICSES). IEEE. 2021. pp. 1–6.

83. Singh G. Singh J. A fog computing based agriculture-IoT framework for detection of alert conditions and effective crop protection. In: 2023 5th International Conference on Smart Systems and Inventive Technology (ICSSIT). IEEE. 2023. pp. 537–543.

84. Jamil B. Ijaz H. Shojafar M. Munir K. Buyya R. Resource allocation and task scheduling in fog computing and Internet of Everything environments: A taxonomy, review, and future directions. ACM Computing Surveys (CSUR). 2022. 54(11s). pp. 1–38.

85. d'Agostino P. Violante M. Macario G. An embedded low-cost solution for a fog computing device on the Internet of Things. In: 2023 Eighth International Conference on Fog and Mobile Edge Computing (FMEC). IEEE. 2023. pp. 284–291.

86. Mahmud R. Toosi A.N. Con-Pi: A distributed container-based edge and fog computing framework. IEEE Internet of Things Journal. 2021. 9(6). pp. 4125–4138.

87. Telikani A. Shen J. Yang J. Wang P. Industrial IoT intrusion detection via evolutionary cost-sensitive learning and fog computing. IEEE Internet of Things Journal. 2022. 9(22). pp. 23260–23271.

88. Qiu Y. Zhang H. Long K. Computation offloading and wireless resource management for healthcare monitoring in fog-computing-based Internet of Medical Things. IEEE Internet of Things Journal. 2021. 8(21). pp. 15875–15883.

89. Dhingra S. Madda R.B. Patan R. Jiao P. Barri K. Alavi A.H. Internet of Things-based fog and cloud computing technology for smart traffic monitoring. Internet of Things. 2021. 14. pp. 100175.

90. Ogundoyin S.O. Kamil I.A. A trust management system for fog computing services. Internet of Things. 2021. 14. pp. 100382.

ДОДАТОК А
(обов'язковий)

ПУБЛІКАЦІЯ

Сертифікат № 2024-041-1

Міністерство освіти і науки України
Хмельницький національний університет

СЕРТИФІКАТ

Кривіцький Богдан Сергійович

учасник XVI Всеукраїнської науково-практичної конференції
«Актуальні проблеми комп'ютерних наук АПКН-2024»
24 години участі (0,8 ECTS credits)

Голова оргкомітету АПКН-2024

Олег СИНЮК
проректор Хмельницького національного
університету з наукової роботи,
доктор технічних наук, професор

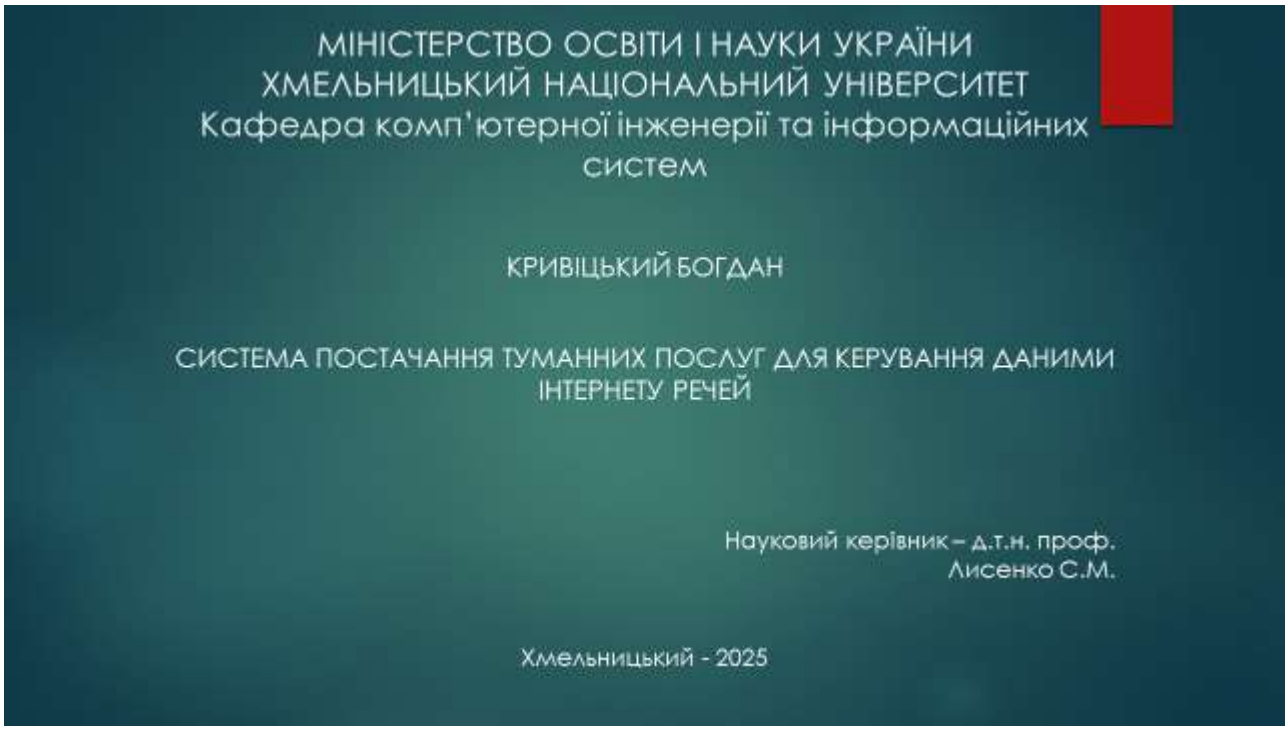
м. Хмельницький
15-16 листопада 2024

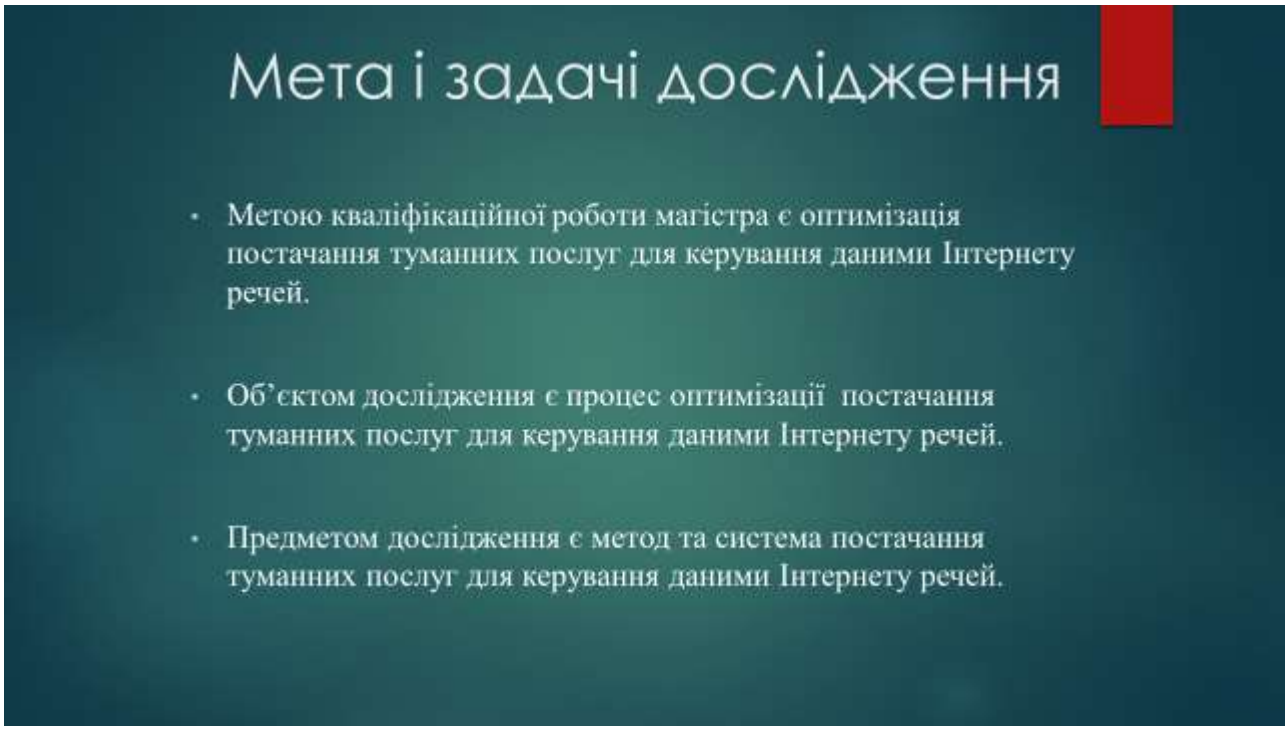
E-mail: apkt.khnu@gmail.com

ДОДАТОК Б

(обов’язковий)

ПРЕЗЕНТАЦІЯ





Мета і задачі дослідження

Для розв'язання поставлених задач використовувалися методи:

- аналіз відомих методів оптимізації постачання туманних послуг для керування даними Інтернету речей;
- розробка моделі системи оптимізації постачання туманних послуг для керування даними Інтернету речей;
- розробка методу оптимізації постачання туманних послуг для керування даними Інтернету речей;
- здійснення дослідження методу оптимізації постачання туманних послуг для керування даними Інтернету речей.

Наукова новизна та практична цінність отриманих результатів

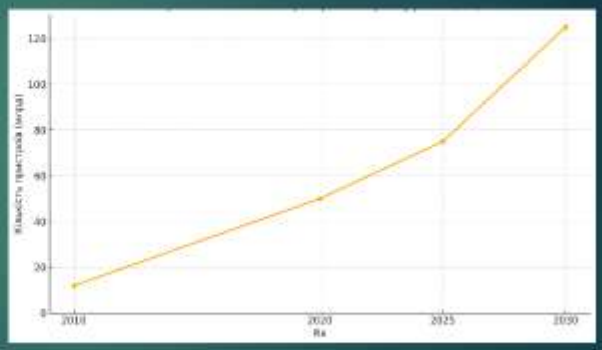
Наукова новизна отриманих результатів:

- удосконалено метод оптимізації постачання туманних послуг для керування даними Інтернету речей, який на відміну від відомих здійснює ізоляцію дані залежно від їхнього призначення, а також виконує семантичну анотацію та створення запитів до даних із додаванням контекстного значення, що оптимізує постачання туманних послуг для керування даними Інтернету речей;
- удосконалено систему оптимізації постачання туманних послуг для керування даними Інтернету речей.

Практична значимість отриманих результатів полягає у розробленому програмно-технічному засобі постачання туманних послуг для керування даними Інтернету речей.

Актуальність дослідження

Актуальність теми зумовлена стрімким зростанням IoT-пристроїв, кількість яких, за даними Cisco, зросла з 50 млрд у 2020 році до прогнозованих 125 млрд у 2030-му. Очікується також інтеграція понад 1 трлн сенсорів у навколишнє середовище. Такий обсяг даних перевантажує інфраструктуру, що потребує переходу до туманних обчислень для зменшення затримок, оптимізації ресурсів і підвищення надійності систем.



Кроки методу

- 1. Збір даних на пристроях IoT з початковою обробкою.
- 2. Передача даних на Fog-вузли через протокол MQTT.
- 3. Попередня обробка: фільтрація, агрегація, семантичне збагачення.
- 4. Передача оброблених даних до хмари для довготривалого зберігання.
- 5. Аналітика, візуалізація та прийняття рішень на основі отриманих даних.

Особливості системи

- Запропонована система має три рівні: пристрої збору даних, проміжні обчислювальні вузли (Fog), хмарна інфраструктура. Забезпечується попередня обробка, агрегація, буферизація та семантична обробка даних, що підвищує адаптивність та ефективність роботи.
- Модель включає буферизацію, кешування та регулювання частоти публікації даних. Проміжні вузли здатні змінювати політики обробки на основі навантаження в режимі реального часу.

Ключові елементи архітектури

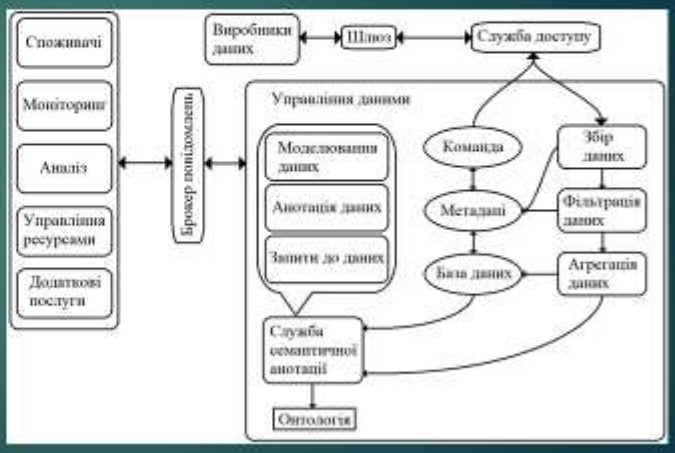
- У моделі реалізовано протокол publish-subscribe для обміну повідомленнями, підтримку JSON-структур для передавання інформації, а також динамічне масштабування, адаптацію до змін трафіку та інтелектуальні механізми фільтрації на всіх рівнях.
- Брокер повідомлень MQTT забезпечує асинхронний обмін даними. Інформація подається у вигляді JSON-структур із додатковими полями для семантичного опису: геолокація, одиниці вимірювання, тип сенсора тощо.

Архітектура ПТС

- Архітектура постачальника туманних сервісів (ПТС) переносить обчислення ближче до джерел даних.
- Вона базується на мікросервісах та підтримує кооперативну взаємодію вузлів.
- Підтримується широкий набір протоколів: MQTT, CoAP, AMQP, HTTP.
- Основні переваги: зменшення затримок, масштабованість, підтримка гетерогенних IoT-пристроїв.

Управління даними в ПТС

- ПТС дозволяє обробляти великі обсяги різноманітних даних в реальному часі.
- Семантична анотація, агрегування, фільтрація, моделювання та RDF-графи формують основу зберігання й аналізу.
- Служба семантичної анотації (SDA) відповідає за глибоку обробку даних із додаванням контексту.



Управління виробниками і споживачами

- У ПТС реалізовані механізми реєстрації, доступу та моніторингу виробників і споживачів.
- Служба доступу виробника: обробка профілів і автентифікація через DNS та API.
- Служба доступу споживача: токени, автентифікація, авторизація доступу до ресурсів.

Атрибут	Обов'язково/ Необов'язково	Коментар
ID виробника	Обов'язково	Ідентифікатор виробника
Марка	Обов'язково	Марка виробника
Назва	Необов'язково	Назва виробника
Розташування	Необов'язково	Координати виробника
Модель	Необов'язково	Модель виробника
Опис	Необов'язково	Опис виробника

Реалізація системи

- Реалізовано трирівневу архітектуру Fog-IoT: рівень збору даних, проміжний обчислювальний рівень та хмара.
- Рівень Edge: сенсори та мікроконтролери, збір та передача первинних даних.
- Fog: MQTT брокер, агрегація, семантичне збагачення, кешування.
- Cloud: довготривале зберігання, візуалізація, аналітика.

Оцінювання ефективності

- Порівняння з базовою хмарною системою показало:

Показник	Без Fog-обробки	З Fog-обробкою
Середня затримка (мс)	342	81
Обсяг переданих даних (МБ)	56.3	11.8
Навантаження на брокер	Високе	Помірне

Умова	Без For-обробки	З Fog-обробкою
Затримка при піковому навантаженні	Часті	Мінімальні
Втрата повідомлень	Можлива	Не зафіксовано
Стабільність графіків	Нерівномірна	Плавна та без стрибків
Адаптивність до навантажень	Обмежена	Висока

Висновки

- На основі проведеного дослідження можна сформулювати такі висновки щодо результатів, отриманих у кожному з розділів кваліфікаційної роботи:
- У першому розділі проведено аналіз наукових джерел і сучасних підходів до організації IoT-архітектур. Визначено недоліки традиційних хмарних рішень, зокрема затримки, перевантаження мережі та слабка масштабованість, що обґрунтовує актуальність застосування туманних обчислень для локалізованої та контекстно-залежної обробки даних.
- У другому розділі сформовано багаторівневу модель системи постачання туманних послуг, яка включає рівні Edge, Fog та Cloud. Визначено функціональні межі між компонентами, описано протоколи обміну, особливості агрегації, кешування та семантичного збагачення даних, що забезпечує гнучкість і ефективність обробки інформаційних потоків.

Висновки

- У третьому розділі описано побудову системи постачальника туманних сервісів, її архітектурні особливості та функціональні складові. Представлено принципи автономного функціонування, підтримку множини протоколів, управління доступом до даних та взаємодію між споживачами й виробниками інформації в межах гетерогенного середовища.
- У четвертому розділі реалізовано функціональні компоненти системи із використанням сучасних технологій (Node-RED, Mosquitto, TimescaleDB, Grafana). Проведено експериментальне дослідження, яке підтвердило ефективність розробленого рішення, зокрема скорочення часу відгуку, зменшення мережевого навантаження та покращення пропускної здатності Fog-рівня.

Tue May 06 21:50:30 EEST 2025, Медзатий Дмитро Миколайович, Хмельницький національний університет, ХНУ

Anti-Plagiarism v-15.274 Educational

The maximum coincidence with one document 2.0%

Dictionaries check: en_US, ru_RU, ua_UA. **Errors in the documents: 9%**

	Document		Sum coincidence on the DB	
ID: 240925 Title: МКР Система постачання туманних послуг для керування даними Інтернету речей Added in a DB: 2025-05-06 Authors: Богдан КРИВІЦЬКИЙ Heads: Сергій ЛИСЕНКО Consultants: Opponents:	Symbols	Lexemes	Symbols	Lexemes
	135546	933	4833 (4%)	59 (6%)

Plagiarism sources

ID	Description	Plagiarism presence in the document	
		Symbols	Lexemes

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: Богдан КРИВІЦЬКИЙ

Співавтор:

Назва: Кривіцький_Система постачання туманних послуг для керування даними Інтернету речей

Експерт:

Підрозділ: Кафедра комп'ютерної інженерії та інформаційних систем

Коефіцієнт подібності 1:8.7%

Коефіцієнт подібності 2:2.9%

Мікропробіли: 2

Заміна букв: 2

Інтервали: 0

Білі знаки: 1

Дата створення звіту: 2025-05-06 22:46:55.0

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:

2025-05-07 Доцент Андрій Нічепорук

Дата експерт

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

РЕЦЕНЗІЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ МАГІСТРА

Здобувач: Богдан КРИВІЦЬКИЙ

Тема: Система постачання туманних послуг для керування даними Інтернету речей

Спеціальність: 123 «Комп'ютерна інженерія»

Обсяг кваліфікаційної роботи магістра:

Кількість листів креслень – ; кількість сторінок записки 76

1. Короткий зміст роботи та прийнятих рішень У роботі запропоновано система постачання туманних послуг для керування даними Інтернету речей

2. Висновок про відповідність роботи дипломному завданню Кваліфікаційна робота магістра відповідає виданому завданню

3. Характеристика виконання кожного розділу, ступінь використання останніх досягнень науки і техніки і передових методів роботи: У першому розділі проведено огляд теоретичних основ досліджуваної проблеми. Досліджено відомі рішення та засоби в цій сфері. У другому розділі запропоновано модель системи постачання туманних послуг для керування даними інтернету речей. У третьому розділі запропоновано система постачання туманних послуг для керування даними інтернету речей. У четвертому розділі запропоновано реалізацію системи постачання туманних послуг для керування даними інтернету речей.

4. Позитивні сторони роботи: Запропонована система постачання туманних послуг для керування даними Інтернету речей, що забезпечує ефективну маршрутизацію, зберігання та семантичну обробку даних у розподіленому середовищі Fog Computing.

5. Негативні сторони роботи: В роботі присутні певні логічні помилки щодо опису модель системи постачання туманних послуг для керування даними інтернету речей.

6. Оцінка графічного оформлення та пояснювальної записки роботи: –

7. Відгук про роботу в цілому: В загальному робота виконана на невисокому рівні.

8. Інші зауваження: –

9. Оцінка кваліфікаційної роботи магістра:

Розглянувши позитивні та негативні сторони представленої кваліфікаційної роботи магістра вважаю, що робота заслуговує оцінки «задовільно» 3.00 (E)

Рецензент (прізвище, ім'я, по батькові, посада, місце роботи) д.т.н., професор, Мартинюк В.В., завідувач кафедри автоматизації, комп'ютерно-інтегрованих технологій та робототехніки

"5 травня" 2025р.

Завідувачу кафедри КІІС
доктору філософії, доценту
Ользі ПАВЛОВІЙ

Кривіцького Богдана Сергійовича
ПІБ здобувача вищої освіти

ФІТ, 2 курсу, групи КІ2м-23-2

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.

6 травня 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: Система постачання туманних послуг для керування даними Інтернету речей

Автор: Богдан КРИВІЦЬКИЙ

Спеціальність: 123 – Комп'ютерна інженерія

Освітня програма: освітньо-наукова

Науковий керівник: Сергій ЛИСЕНКО, д.т.н, професор

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: усі запозичення фрагментарні, або мають належним чином оформленні посилання;

1) окремі виявлені збіги є загальновживаними фразами або виразами, про що свідчить посилання системи на збіг з джерелами на один фрагмент речення;
2) 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 8.7% і адресується до 72 першоджерела; та системою Anti-Plagiarism складає 2.0%, що, з урахуванням наведених обгрунтувань, відповідає характеру наукового дослідження і свідчить на користь кваліфікаційної роботи.

Керівник роботи _____ Сергій ЛИСЕНКО

Гарант ОП _____ Олег САВЕНКО

Завідувач кафедри КІІС _____ Ольга ПАВЛОВА