

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»**

**Навчально-науковий інститут телекомунікаційних систем
Кафедра телекомунікацій**

«На правах рукопису»

УДК _____

«До захисту допущено»

Завідувач кафедри

_____ Сергій КРАВЧУК

« ____ » _____ 2024 р.

**Магістерська дисертація
на здобуття ступеня магістра
за освітньо-професійною програмою «Інженерія та програмування
інфокомунікацій»
зі спеціальності 172 «Телекомунікації та радіотехніка»
на тему: «Алгоритм розгортання шлюзів Інтернету речей для розумного
дому»**

Виконав:

студент II курсу, групи ТЗ-21мп

Пономаренко Іван Олександрович

Керівник:

Доцент кафедри ТК НН ІТС, к.т.н.

Валуйський С.В.

Рецензент:

В.о. завідувача кафедри ІТТ НН ІТС, професор

д.т.н., с.н.с., професор

Скулиш М.А.

Засвідчую, що у цій магістерській
дисертації немає запозичень з праць
інших авторів без відповідних
посилань.

Студент _____

Київ – 2024 року

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
Навчально-науковий інститут телекомунікаційних систем
Кафедра телекомунікацій

Рівень вищої освіти – другий (магістерський)

Спеціальність – 172 «Телекомунікації та радіотехніка»

Освітньо-професійна програма «Інженерія та програмування
інфокомунікацій»

ЗАТВЕРДЖУЮ

Завідувач кафедри

_____ Сергій КРАВЧУК

«__» _____ 2023 р.

ЗАВДАННЯ
на магістерську дисертацію студенту
Пономаренку Івану Олександровичу

1. Тема дисертації «Алгоритм розгортання шлюзів інтернету речей для розумного дому», науковий керівник дисертації Валуйський Станіслав Вікторович, к.т.н., доцент кафедри ТК, затверджені наказом по університету від «31» жовтня 2023 р. № 5075-с.
2. Термін подання студентом дисертації 05.01.2024 р.
3. Об'єкт дослідження процес функціонування мережі Інтернету речей для розумного дому
4. Предмет дослідження алгоритми кластеризації мережі інтернету речей
5. Перелік завдань, які потрібно розробити:
 - Провести літературний огляд за темою дослідження
 - Провести огляд та аналіз існуючих алгоритмів кластеризації, їх переваг та недоліків.
 - Розробити удосконалений алгоритм планування та розгортання шлюзів інтернету речей для розумного дому
 - Розробити стартап-проект

6. Орієнтовний перелік ілюстративного матеріалу:

- Тема, мета, актуальність, задачі, дослідження
- Огляд основних особливостей розумних домів з локальним сервером
- Моделі кластеризації мережі Інтернету речей для мінімізації кількості шлюзів
- Результати

7. Орієнтовний перелік публікацій

- 1) Валуйський С. В., Пономаренко І. О. АЛГОРИТМ РОЗГОРТАННЯ ШЛЮЗІВ ІНТЕРНЕТУ РЕЧЕЙ ДЛЯ РОЗУМНОГО ДОМУ // Сімнадцята міжнародна науково-технічна конференція "Перспективи телекомунікацій" (ПТ-2023), тези доповідей, Київ, с. 332 – 334.

8. Дата видачі завдання “ 7 ” жовтня 2022 р.

Календарний план

№ з/п	Назва етапів виконання магістерської дисертації	Термін виконання етапів магістерської дисертації	Примітка
1	Аналіз технічних характеристик протоколів бездротового зв'язку.	7.10.2022-7.12.2022	виконано
2	Аналіз існуючого програмного забезпечення локальних серверів.	11.12.2022-08.02.2023	виконано
3	Огляд алгоритмів кластеризації (K-Means, LP, Simple PSO та Hybrid PSO).	16.02.2023-05.03.2023	виконано
4	Аналіз моделей кластеризації (K-Means, LP, Simple PSO та Hybrid PSO).	06.03.2023-07.04.2023	виконано
5	Планування та розгортання шлюзів розумного дому.	08.04.2023-12.06.2023	виконано
6	Мінімізація кількості шлюзів інтернету речей для розумного дому.	14.06.2023-10.08.2023	виконано
7	Розробка стартап проекту.	11.08.2023-07.10.2023	виконано
8	Формулювання загальних висновків та оформлення дипломної роботи згідно норм контролю.	13.10.2023-21.12.2023	виконано

Студент

Іван ПОНОМАРЕНКО

Науковий керівник дисертації

Станіслав ВАЛУЙСЬКИЙ

РЕФЕРАТ

Робота містить 94 сторінки, 12 рисунків та 20 таблиць. В процесі написання дисертації було використано 52 джерела.

Актуальність теми: Проблеми, які активно вивчаються в контексті Інтернету речей, включають інтероперабельність та неоднорідність. Для вирішення цих проблем запропоновано розгортання шлюзів IoT, які підтримують різні комунікаційні технології. Розгортання шлюзів IoT в розумних домах вимагає планування, щоб врахувати комунікаційні вимоги пристроїв IoT, такі як пропускна здатність, комунікаційні технології, енергоефективність і інші. Зона покриття бездротових мереж виникає як складне завдання, враховуючи велику кількість факторів, таких як вибір комунікаційної технології, архітектурні особливості оточуючого середовища, мобільність пристроїв та людей, кількість об'єктів, які одночасно підключаються до мережі і багато інших. Існуючі алгоритми розгортання шлюзів Інтернету речей для розумного дому враховують ці фактори лише частково, а отже можуть бути вдосконалені.

Мета роботи: Мета роботи полягає в розробці ефективного алгоритму розгортання шлюзів Інтернету речей для розумного дому із урахуванням оптимізації передачі даних та використання енергоресурсів.

Завдання дослідження:

1. Літературний огляд за темою дослідження
2. Огляд та аналіз існуючих алгоритмів кластеризації, їх переваг та недоліків.
3. Удосконалення алгоритму планування та розгортання шлюзів інтернету речей для розумного дому
4. Розробка стартап проекту на основі проведеного дослідження.

Об'єкт дослідження: процес функціонування мережі інтернету речей для розумного дому.

Предмет дослідження: алгоритми кластеризації мережі інтернету речей.

Наукова новизна отриманих результатів: Наукова новизна магістерської роботи на тему «Алгоритм розгортання шлюзів Інтернету речей для розумного дому» полягає в тому, що розроблено удосконалений алгоритм розгортання шлюзів мережі інтернету речей для розумного дому за критерієм мінімізації кількості шлюзів мережі. Удосконалений алгоритм для розгортання шлюзів Інтернету речей в магістерській роботі спрямований на оптимізацію ефективності системи управління розумним домом. Його основна інновація полягає в здатності мінімізувати кількість необхідних шлюзів в мережі, що призводить до значного зниження витрат та підвищення загальної ефективності системи.

Практична новизна отриманих результатів: Результатом використання удосконаленого алгоритму є зменшення витрат на розгортання та управління шлюзами, а також підвищення ефективності управління домом на показник від 10% до 15%. Це досягається завдяки оптимальному використанню ресурсів мережі та зменшенню перекриття між зонами покриття, що сприяє оптимальній передачі і обробці даних у розумному домі.

Ключові слова : шлюз, протокол, кластеризація, маршрутизація.

ABSTRACT

The work includes 94 pages, 12 figures and 20 tables. 52 were used in the process of writing the dissertation.

Relevance of the topic: The problems actively studied in the Internet of Things (IoT) context include interoperability and heterogeneity. The deployment of IoT gateways supporting various communication technologies has been proposed to address these issues. Deploying IoT gateways in smart homes requires careful planning to consider the communication requirements of IoT devices, such as bandwidth, communication technologies, energy efficiency, and others. The coverage area of wireless networks poses a complex task, considering many factors such as the choice of communication technology, architectural features of the surrounding environment, device, human mobility, the number of objects simultaneously connecting to the network, and many others. Existing algorithms for deploying IoT gateways in smart homes only partially consider these factors and, therefore, may be improved.

The purpose of the work: The purpose of the work is to develop an efficient algorithm for deploying Internet of Things (IoT) gateways in a smart home, considering the optimization of data transmission and the utilization of energy resources.

Research objectives:

1. Literary review on the research topic.
2. Review and analysis of existing clustering algorithms, their advantages, and disadvantages.
3. Enhancement of the planning and deployment algorithm for Internet of Things gateways in a smart home.
4. Development of a startup project based on the conducted research.

Research object: The functioning process of the Internet of Things (IoT) network for a smart home.

Research subject: Clustering algorithms of the Internet of Things network.

Scientific novelty of the obtained results: The scientific novelty of the master's thesis on the topic "Algorithm for deploying Internet of Things gateways for a smart home" lies in the development of an improved algorithm for deploying IoT network gateways based on the criterion of minimizing the number of network gateways. The enhanced algorithm for deploying IoT gateways in the master's thesis is aimed at optimizing the efficiency of the smart home management system. Its main innovation lies in its ability to minimize the number of required gateways in the network, leading to significant cost reduction and overall system efficiency improvement.

Practical novelty of the obtained results: The result of using the improved algorithm is a reduction in deployment and management costs of gateways, as well as an increase in home management efficiency by 10% to 15%. This is achieved through optimal use of network resources and the reduction of overlap between coverage zones, facilitating optimal data transmission and processing in a smart home.

Keywords: gateway, protocol, clustering, routing.

ЗМІСТ

ВСТУП	11
РОЗДІЛ 1 АНАЛІЗ ФУНКЦІОНУВАННЯ ТА ПРИНЦИПІВ ПОБУДОВИ МЕРЕЖІ ІНТЕРНЕТУ РЕЧЕЙ ДЛЯ РОЗУМНОГО ДОМУ	12
1.2 Хмарне середовище для розумного дому:.....	14
1.3 Протоколи бездротового зв'язку розумного дому	15
1.3.1 ZigBee	15
1.3.2 Z-Wave	16
1.3.3 LoRa та LoRaWAN	16
1.3.5 Wi-Fi	17
1.4 Принципи будови складових мереж Інтернету речей.....	21
1.5 Механізм роботи програмного забезпечення OpenHAB та Zabbix. .	27
1.6 Порівняння алгоритмів кластеризації (K-Means, LP, Simple PSO та Hybrid PSO)	32
РОЗДІЛ 2 МОДЕЛІ КЛАСТЕРІЗАЦІЇ МЕРЕЖІ ІНТЕРНЕТУ РЕЧЕЙ ДЛЯ МІНІМІЗАЦІЇ КІЛЬКОСТІ ШЛЮЗІВ	39
2.1 Сутність моделей кластеризації.....	39
2.2 Класифікація на основі виявлення маршруту	45
2.3 Ефективність моделей кластеризації.	55
Висновки	58
РОЗДІЛ 3 АЛГОРИТМ РОЗГОРТАННЯ ШЛЮЗІВ ІНТЕРНЕТУ РЕЧЕЙ ДЛЯ РОЗУМНОГО ДОМУ	60
3.1 Планування та розгортання шлюзів Інтернету речей для розумного дому	60
3.2 Математична постановка задачі мінімізації кількості шлюзів інтернету речей для розумного дому.....	64
3.3 Опис алгоритму розгортання шлюзів інтернету речей для розумного дому	64
Висновки	68

РОЗДІЛ 4 РОЗРОБКА СТАРТАП ПРОЕКТУ	70
4.1 Опис ідеї стартап-проекту.....	70
4.2 Технологічний аудит ідеї проекту.....	72
4.3 Аналіз ринкових можливостей запуску стартап-проекту.....	73
4.4 Розроблення ринкової стратегії проекту	81
4.5 Розроблення маркетингової програми стартап-проекту.....	83
Висновки	86
ЗАГАЛЬНІ ВИСНОВКИ ПО РОБОТІ	87
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	90

ПЕРЕЛІК СКОРОЧЕНЬ

IoT	Internet of Things
BLE	Bluetooth Low Energy
WAN	Wireless Area Network
LAN	Local Area Network
IEEE	Institute of Electrical and Electronics Engineers
LP	Label Propagation
PSO	Particle Swarm Optimization
QoS	Quality of Service

ВСТУП

Інформаційні технології вже стали необхідні у повсякденному житті людей та підприємств, а концепція Інтернету речей (IoT) активно впливає на зміни у їхньому повсякденному функціонуванні. IoT репрезентує комунікаційну парадигму, де об'єкти взаємодіють між собою та з користувачами, використовуючи мережеві комунікаційні технології, переважно бездротові.

Ця парадигма стимулює взаємодію між різними пристроями, такими як електронні пристрої, побутові прилади, транспортні засоби, медичне обладнання, датчики, камери спостереження тощо. Технології, використані в рамках Інтернету речей, сприяють розвитку різноманітних сервісів, пов'язаних із розумними містами та розумними домами.

IoT базується на взаємодії об'єктів із здатністю обмінюватися даними. З огляду на значний обсяг підключених об'єктів та пристроїв, спостерігається стрімкий ріст обсягу передаваних даних через Інтернет. В даній роботі пропонується метод планування, який використовує оптимізаційні моделі для розгортання шлюзів IoT у розумних домах. Ці моделі оцінюють мінімальну кількість шлюзів IoT, необхідних для охоплення бажаної площі, їх розташування та розподіл пристроїв IoT між відповідними шлюзами. Діапазон комунікаційних технологій і використання каналу передачі даних слугують параметрами для моделей оптимізації. Представлено кілька моделей, зокрема з використанням технологій ZigBee, Wi-Fi і BLE.

РОЗДІЛ 1

АНАЛІЗ ФУНКЦІОНУВАННЯ ТА ПРИНЦИПІВ ПОБУДОВИ МЕРЕЖІ ІНТЕРНЕТУ РЕЧЕЙ ДЛЯ РОЗУМНОГО ДОМУ

1.1 Локальний сервер для розумного дому

Розумні дома набувають все більшої популярності завдяки своїй здатності оптимізувати комфорт та безпеку для мешканців. Однак ключовим елементом успіху є програмне забезпечення, яке керує локальними серверами розумного дому.

Локальний сервер для розумного дому є центральним елементом, що забезпечує управління та автоматизацію підключеними розумними пристроями в оселі. Його головна функція полягає в зборі, обробці та виконанні команд, спрямованих на підтримку зручності та ефективності повсякденного життя.

Локальний сервер використовує певну систему автоматизації, яка дозволяє створювати різноманітні сценарії та правила для автоматичного виконання завдань. Це може включати управління освітленням, регулювання температури, контроль безпеки, тощо. [1].

Однією з основних переваг локального сервера є можливість керувати розумними пристроями без необхідності використання хмарних сервісів. Це забезпечує більшу конфіденційність та безпеку даних. Локальний сервер забезпечує вищий рівень безпеки, оскільки дані обробляються локально, а не передаються через зовнішні сервери. Це важливо для захисту особистої інформації та забезпечення конфіденційності користувача.

Локальний сервер може забезпечувати можливість моніторингу роботи системи та збереження логів подій, що полегшує виявлення проблем та налагодження роботи системи [2].

Ключові фактори вибору програмного забезпечення для локального серверу розумного дому:

- *Оптимізована архітектура:* Програмне забезпечення для локальних серверів розумного дому має вдосконалену архітектуру, засновану на мікросервісах. Це дозволяє оптимально керувати різноманітними пристроями та додатками, забезпечуючи стабільну роботу системи.
- *Безпека та конфіденційність:* Велика увага приділяється розробці механізмів захисту даних. Шифрування та ідентифікаційні протоколи використовуються для забезпечення високого рівня конфіденційності особистої інформації користувачів.
- *Сумісність та інтеграція:* Програмне забезпечення підтримує різноманітні стандарти комунікації, такі як Zigbee, Z-Wave, або MQTT, що робить його універсальним для різних типів пристроїв. Це сприяє зручності та сумісності.
- *Ефективне управління енергоспоживанням:* Використання алгоритмів оптимізації енергоспоживання дозволяє не лише забезпечувати стабільну роботу системи, а й раціонально використовувати енергію, що стає важливим фактором для сучасних екологічно-освічених споживачів.
- *Інтуїтивний інтерфейс та зручність використання:* Графічний інтерфейс та мобільний додаток розроблені з фокусом на простоту використання. Це дозволяє користувачам легко керувати своїм розумним будинком та налаштовувати його за своїми потребами.

1.2 Хмарне середовище для розумного дому:

Хмарне середовище для розумного дому представляє собою інноваційну інфраструктуру, що забезпечує централізоване управління та контроль над всіма підключеними пристроями та системами. В основі цього середовища лежить використання хмарних технологій, які дозволяють зберігати та обробляти великі обсяги даних, отриманих від датчиків, камер, домашніх пристроїв та інших елементів інтелектуального дому.

Однією з ключових переваг хмарного середовища є доступність у будь-який час із будь-якого пристрою, підключеного до Інтернету. Користувач може контролювати свій розумний дім через спеціальні додатки на смартфоні, планшеті чи комп'ютері, навіть будучи віддаленим від дому. Крім того, хмарне зберігання дозволяє ефективно аналізувати дані та використовувати штучний інтелект для покращення функціональності системи [4].

Ще однією важливою характеристикою є забезпечення високого рівня безпеки. Дані, що передаються та зберігаються в хмарному середовищі, шифруються та захищаються від несанкціонованого доступу. Це важливо для забезпечення конфіденційності особистої інформації та надійності роботи розумного дому.

Хмарне середовище для розумного дому постійно розвивається, впроваджуючи нові технології та функції, щоб забезпечити користувачам максимальний комфорт, ефективність та безпеку в їхньому повсякденному житті.

Хмарні середовища розумних домів мають свої певні недоліки:

- *Залежність від підключення до мережі Інтернет:* Без доступу до Інтернету функціонал розумного будинку може бути обмеженим або

абсолютно відсутнім. Залежність від хмарних сервісів може викликати затримки у відгуку системи на команди користувача.

- *Вразливість до збоїв:* Хмарні сервіси піддаються ризику відмови служби, і в разі виникнення проблеми з інфраструктурою може бути обмежено функціональне використання розумного будинку. Централізовані системи можуть стати об'єктом кібератак, що призводить до витоку чутливих даних чи навіть контролю над системою.

1.3 Протоколи бездротового зв'язку розумного дому

Найбільш популярні протоколи бездротової передачі даних в основному представляють собою прості методи налаштування обладнання, проте успіх їхнього використання суттєво залежить від відсутності перешкод у мережі [5].

1.3.1 ZigBee

ZigBee - це бездротовий протокол передачі даних, спроектований для низькоспоживаючих пристроїв у мережах «Internet of Things» (IoT). Він використовується для створення надійних та енергоефективних бездротових мереж з невеликою дальністю передачі даних. ZigBee використовує частотний діапазон 2,4 ГГц та пропонує низьку швидкість передачі даних, але забезпечує ефективність енергоспоживання та високу надійність зв'язку, що робить його ідеальним вибором для реалізації розумних систем у домашньому оточенні, промисловості та сенсорних мережах.

ZigBee використовує частотний діапазон 2,4 ГГц та може використовувати мережі типу "зірка", "меш" чи "кластерне дерево". Цей протокол часто використовується для забезпечення безпроводного зв'язку між сенсорами, контролерами та іншими пристроями в різних застосуваннях, таких як домашня автоматизація, промисловість та медичні системи.

Завдяки своїй ефективності та гнучкості ZigBee став популярним вибором для створення розумних мереж, де важлива низька споживана потужність та здатність пристроїв ефективно співпрацювати в обмеженому бездротовому середовищі. ZigBee є Open-Source протоколом.

1.3.2 Z-Wave

Z-Wave - це протокол бездротового зв'язку, призначений для систем "розумного будинку". Розроблений для оптимальної взаємодії та управління різноманітними "розумними" пристроями в домашньому середовищі. Основні особливості Z-Wave включають низьку споживану потужність, що дозволяє пристроям працювати на довгому терміні без заміни батарей, а також високий ступінь стабільності та надійності зв'язку. Важливо відзначити й можливість не взаємодії між різними пристроями для різних країн через використання різних частот ліцензій.

1.3.3 LoRa та LoRaWAN

LoRa - це модуляційна технологія для передачі даних через радіохвилі. Вона використовує частотний діапазон, що лежить в межах від 868 МГц до 915 МГц (залежно від регіону), і забезпечує дуже довгий зв'язок на відстань до кількох кілометрів навіть в міських середовищах.

LoRaWAN - це протокол вищого рівня, який використовує технологію LoRa для побудови довгодіючих мереж IoT. Він визначає спосіб організації та керування мережею, дозволяючи взаємодію різних пристроїв в єдиній інфраструктурі.

LoRa використовує різні частоти в різних країнах, існують обмеження на потужність та час передачі.

1.3.4 Bluetooth та BLE

Bluetooth є бездротовим стандартом передачі даних, розробленим для обміну інформацією між електронними пристроями через короткі відстані. Bluetooth використовує частотний діапазон 2,4 ГГц, розділений на канали для

забезпечення мінімізації взаємодії з іншими бездротовими пристроями. Здатність до передачі даних з різною швидкістю, включаючи 1 Мбіт/с для Bluetooth 4.0 та більш високі швидкості для новіших версій. Можливість використовувати Bluetooth для точного визначення місцезнаходження пристроїв, наприклад, у внутрішніх приміщеннях. Bluetooth найчастіше використовується для прямих точка-точка з'єднань, таких як пара мобільний телефон і гарнітура або ноутбук і бездротова миша.

Bluetooth Low Energy (BLE) - це варіант технології Bluetooth, спрямований на забезпечення низького споживання енергії та ефективності для пристроїв IoT. BLE спроектований для тривалого життя батареї, що робить його ідеальним для сенсорних пристроїв, які працюють в обмежених умовах живлення. Використовує той же 2,4 ГГц частотний діапазон, що й Bluetooth, але з меншою швидкістю передачі даних, оптимізованою для роботи в обмежених умовах. Також підтримує прямі точка-точка з'єднання, але також може працювати в режимі магістраль-вузол, де один пристрій (зазвичай смартфон) взаємодіє з декількома пристроями одночасно. Використовується в різноманітних областях, таких як медицина, фітнес, домашня автоматизація, транспорт та інші сфери IoT.

1.3.5 Wi-Fi

Wi-Fi працює в різних частотних діапазонах, таких як 2,4 ГГц та 5 ГГц. Більш нові стандарти можуть також використовувати частоти вище 5 ГГц для полегшення завантаження мережі та уникнення перешкод. Стандарти Wi-Fi підтримують широкий діапазон швидкостей передачі даних, включаючи 802.11b (11 Мбіт/с), 802.11g (54 Мбіт/с), 802.11n (до 600 Мбіт/с) та 802.11ac (до 1,3 Гбіт/с). Використовує різні протоколи шифрування, такі як WEP, WPA та WPA2 (та WPA3), для захисту від несанкціонованого доступу та зашифрування передачі даних.

Протокол Wi-Fi широко використовується у розумних будинках, особливо для пристроїв, які завжди приєднані до електромережі. Зазвичай

цей протокол застосовується для встановлення зв'язку між смартфонами або планшетами та автоматизованими системами. Зручність управління будинком за допомогою мобільних пристроїв обумовлена їхньою постійною доступністю. Час від часу Wi-Fi використовується для з'єднання з автономними пристроями, що сприяє швидкому та легкому підключенню до технології розумного будинку. Проте, його висока вартість та надмірно велика пропускна здатність не завжди є ефективною для складних систем автоматизації.

ZigBee не завжди справляється з умовами, коли в мережі існують сильні перешкоди, створені іншими пристроями. Завдяки своєму одноканальному підходу ZigBee іноді не може ефективно керувати перешкодами, які часто виникають у зайнятому діапазоні 2,4 ГГц, що також використовується іншими технологіями, такими як Wi-Fi або Bluetooth. Оскільки пропускна здатність смуги 2,4 ГГц постійно зростає, ця проблема може загостритися в найближчому майбутньому [8].

Проблему з ZigBee поглиблює той факт, що стандарт IEEE 802.15.4 визначає фізичний рівень стека протоколів ZigBee, обмежуючи швидкість передачі даних до 250 кбіт/с. Цей стандарт використовується не лише ZigBee, але й іншими десятками різноманітних рішень. Зв'язковий альянс ZWave здатний визначати кожен рівень моделі OSI, що забезпечує централізований контроль за всіма аспектами зв'язку [7].

Щодо аспектів безпеки, ZigBee пропонує різноманітні заходи для забезпечення надійного захисту даних інтелектуальних пристроїв. Застосування 128-розрядного алгоритму AES для шифрування та автентифікації даних, а також трьох видів ключів для управління безпекою, робить зазначену технологію достатньо надійною на перший погляд. Втім час від часу виникають проблеми з безпекою пристроїв, що підтримують ZigBee, зокрема пов'язані із формуванням парних клавів при підключенні нового пристрою до мережі [9].

Один з недоліків цієї технології пов'язаний із діапазоном частот. Вибір низькочастотного, стабільного та систематичного діапазону для пристроїв короткого діапазону, відмінного від популярних і завантажених (2,4 ГГц) пристроїв, виявився «далекоглядним та обґрунтованим рішенням», яке «значно впливає на користувачів розумного будинку, які використовують ZWave, Wi-Fi та інші технології». Проте для пристроїв короткого діапазону використовуються різні частоти в різних країнах, наприклад, в Європі (СЕРТ-країни), Китаї та багатьох країнах Азії - 868,42 МГц. У США та Мексиці ці частоти зайняті технологією GSM, тому Z-хвильові рішення працюють на частотах 908 та 42 МГц - робочий діапазон Z-хвилі - 869,0 МГц. Це означає, що придбання нового продукту за кордоном та його підключення до системи автоматизації будинку вимагає ретельного уваги, оскільки пристрій, призначений для американського ринку, може бути несумісним із пристроями інших країн.

Bluetooth оперує у діапазоні 2,4 ГГц, подібно до інших бездротових технологій, таких як мікрохвильові печі, бездротові телефони та інші. Незважаючи на наявність деяких пристроїв Bluetooth, призначених для уникнення перешкод, використання цього діапазону може мати свої недоліки. Окрім постійних перешкод, смуга частот 2,4 ГГц також має серйозний недолік - радіохвилі цього діапазону втрачають сигнал швидше, ніж частоти нижче 1 ГГц, коли проникають через стіни та інші перешкоди.

Тому діапазон технологій Bluetooth Low Energy не є його сильною стороною. Хоча теоретично можливе досягнення «до 100 метрів у прямому прицілі», четверта версія Bluetooth на практиці може обмежитися відстанню до 10 метрів у приміщенні при взаємодії двох пристроїв. Це ускладнюється тим, що сигнал значно залежить від перешкод, характеристик поширення та конфігурації виробника, який, так само як і Bluetooth Smart, має здатність регулювати потужність сигналу пристрою, включаючи збільшення його потужності та збільшення споживання енергії [10].

Звісно, технології Bluetooth, Wi-Fi та ZigBee мають обмежений радіус зв'язку, і вони витрачають багато енергії, що відрізняє їх від мобільного зв'язку та Wi-Fi. З іншого боку, зв'язок з ZigBee та LoRa відзначається високою енергоефективністю. Ці технології призначені для передачі невеликого обсягу даних, при цьому ZigBee проявляє себе на невеликій відстані, а LoRa спроектована для комунікації на великій відстані, і вони також використовують різні топології мережі.

Можливості використання Bluetooth у мережі з низьким енергоспоживання суттєво різняться від інших технологій. Основна відмінність полягає в методі розподілу повідомлень по мережі. Подібно до маршрутизації на основі джерела (Z-хвиля) та маршрутизації на основі призначення (Sigbee, Thread), кожне повідомлення передається від одного вузла до іншого, доки не досягне пункту призначення. В мережі Bluetooth mesh використовується обмежений підхід до лавинної маршрутизації.

Цей підхід, хоча не є ідеальним з точки зору пропускнуої здатності мережі, не вимагає створення таблиць маршрутизації та складних процедур для відновлення продуктивності мережі. Таким чином, керована лавинна маршрутизація використовує менше пам'яті та обчислювальних потужностей, що позитивно впливає на кінцеву вартість готових рішень. Теоретично, вона може демонструвати кращу стійкість до звукового спектру в діапазоні 2,4 ГГц порівняно з іншими рішеннями, які працюють на цій же частоті, за умови щільного налаштування датчиків в мережі. Однак через високий рівень використання ресурсів, Bluetooth mesh мережі не є ідеальним вибором для складних проектів автоматизації будинку, не говорячи вже про різноманітні комерційні реалізації.

На сьогодні технологія Z-Wave визнана найефективнішою для забезпечення зв'язку між пристроями різних виробників за допомогою єдиного протоколу. Z-Wave охоплює всі рівні моделі OSI, починаючи з розробки як технологія стільникової мережі з високим рівнем стійкості до помилок. При цьому Z-Wave відрізняється тим, що він не працює в

перевантаженому діапазоні частот 2,4 ГГц. З усіма цими перевагами Z-Wave Alliance успішно створила найбільш розвинутий екосистему пристроїв з понад 100 мільйонами реалізацій.

1.4 Принципи будови складових мереж Інтернету речей.

«Інтелектуальний будинок» є передовою концепцією взаємодії між користувачем та житловим простором, при якій власник будинку обирає заздалегідь програмований сценарій, а система автоматичного керування відповідає наявним умовам, налаштовує режими роботи та моніторить параметри всіх електроприладів та інженерних систем будинку.

Ідея «розумного будинку» визначається як перспективний напрямок розвитку комунікаційних та інформаційних технологій. Основна концепція полягає в тому, що це система домашньої автоматизації, яка аналізує інформацію від датчиків та виконує різні завдання повсякденного життя автоматично, без необхідності втручання людини. Такий підхід дозволяє користувачам уникнути виконання щоденних рутинних обов'язків [11].

У сутності, «розумний будинок» представляє собою високотехнологічну концепцію, яка дозволяє ефективно управляти інженерними комунікаціями в житловому просторі. Це досягається через використання програмованої інтелектуальної системи, яка налаштовується під індивідуальні потреби та побажання користувачів.

Структура системи керування основними компонентами «розумного будинку» включає п'ять основних категорій: управління електричною та побутовою технікою; управління мультимедійними системами; управління системою безпеки; управління освітленням; управління мікрокліматом будинку.

Мережа розумного будинку складається з низки ключових компонентів, таких як центральний контролер, який служить головним мозком системи та координує взаємодію всіх підключених пристроїв.

Додатково, сенсори, актуатори, термостати та інші розумні пристрої становлять основні будівельні блоки мережі. Спільно ці компоненти створюють інтелектуальну інфраструктуру, яка адаптується до змінних потреб мешканців

З точки зору топології, мережі для розумного будинку можуть використовувати різноманітні конфігурації, такі як зіркова, шина чи мешканцеві, залежно від обсягу та типу підключених пристроїв. Однак безпека мережі відіграє важливу роль, і вона забезпечується шляхом використання шифрування даних, аутентифікації та авторизації, щоб унеможливити від несанкціонованого доступу та забезпечити конфіденційність користувачів [12].

За допомогою різноманітних протоколів зв'язку, автоматизовані пристрої у розумному будинку встановлюватимуть між собою зв'язок та обмінюватимуться інформацією.

Таблиця 1.1

Основні протоколи для розумного будинку

Протокол	Тип передачі	Швидкість передачі	Примітка
ZigBee	Бездротовий	20-250 кбіт/с	Ефективний для мереж з великою кількістю пристроїв.
Z-Wave	Бездротовий	9.6-100 кбіт/с	Має низьку споживану потужність та добру зв'язку віддаленість.
Wi-Fi	Бездротовий	до 866 Мбіт/с	Широко використовується для підключення до Інтернету.

Наприклад, це може включати викликання різних дій, таких як увімкнення/вимкнення освітлення чи розблокування дверного замка з використанням розумної технології. На ринку існує багато протоколів зв'язку, які використовують дротове підключення, електромережу, гібридні бездротові технології або традиційні бездротові зв'язки. Незважаючи на те, що деякі з цих протоколів залишаються закритими, у більшості випадків

вони пропонують своє API для інтеграції та взаємодії з іншими пристроями. У даній роботі розглядаються найбільш популярні протоколи зв'язку, спрямовані на взаємодію інтелектуальних пристроїв у розумному будинку [13].

Переваги бездротового з'єднання включають:

1. Вартість: зазвичай бездротове обладнання менше коштує в установці та підтримці порівняно із дротовими аналогами.
2. Уникнення передчасного проектування: бездротові технології дозволяють уникнути необхідності визначення точного розміщення кабелів на етапі проектування.
3. Надійність та висока швидкість відгуку: бездротові системи можуть забезпечити високу надійність та миттєвий відгук у порівнянні із деякими дротовими варіантами.
4. Швидкість реалізації: встановлення бездротових з'єднань може бути швидше та менш працевитратним процесом порівняно із дротовими альтернативами.

Особливості бездротової передачі даних у розумному будинку включають:

1. Залежність від якості зв'язку: проходження сигналу може значно впливати на якість зв'язку, особливо при наявності різноманітних перешкод.
2. Обмежений термін служби через наявність батарейок: використання батарейок в обладнанні системи може призвести до обмеженого терміну служби, вимагаючи постійного контролю та обслуговування.

ZigBee, ратифікований ZigBee Alliance у 2004 році, є протоколом верхнього рівня, заснованим на IEEE 802.15.4 для бездротової передачі даних. Існують різні версії протоколу, відомі своїми роками створення.

Перша версія, ZigBee 2004, була випущена в 2004 році, а через два роки з'явилася друга версія - ZigBee 2006 (див. рис. 1.1).

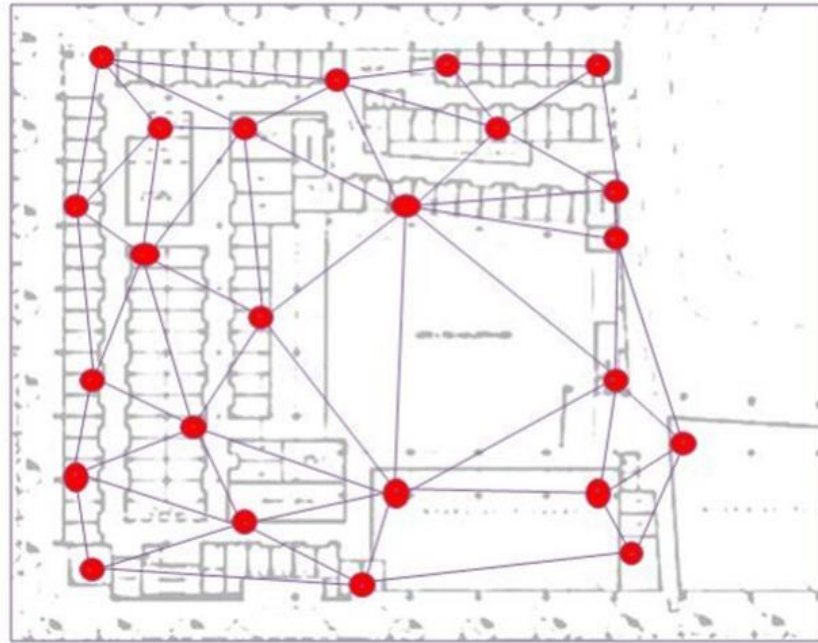


Рис. 1.1 Топологія протоколу ZigBee

Однак значні зміни внесли в 2007 році, коли протокол був розділений на два профілі стека: ZigBee 2007 і ZigBee Pro. ZigBee 2007 підходить для простих мереж в будинках або невеликих комерційних приміщеннях, в то час як ZigBee Pro має розширений функціонал, такий як широкомовлення і симетричне шифрування пакетів [14].

Основна перевага ZigBee полягає в низькому енергоспоживанні. Мережа організована так, що основна частина інформації передається через маршрутизатори, а кінцеві пристрої перебувають у «сплячому» режимі, економлячи енергію. Завдяки цій особливості, пристрої на базі ZigBee можуть працювати протягом двох років на одній батареї AA чи навіть AAA.

У мережах ZigBee надійність зв'язку підвищується завдяки існуванню додаткових зв'язків між пристроями. Усі пристрої, які не переходять в режим сну, виконують функцію роутерів, які відповідають за маршрутизацію мережевого трафіку, вибір оптимального маршруту та інші завдання. Навіть у випадку виходу з ладу пристрою, що був організатором мережі, ZigBee-

мережа продовжить своє функціонування завдяки існуючим дублюючим зв'язкам. Помилка або відмова будь-якого роутера не є критичною завдяки надлишковості зв'язків. Введення додаткових вузлів, які постійно живляться та можуть виконувати функції роутера, покращує надійність мережі [15].

Топологія ZigBee дозволяє підключати маршрутизатори один до одного для поліпшення надійності системи. У випадку відмови будь-якого маршрутизатора пакети даних автоматично перенаправляються через інші доступні маршрутизатори, зберігаючи цілісність інформаційного потоку.

Комунікація в мережі ZigBee реалізується через передачу пакетів даних між пристроями, які належать до трьох категорій [16]:

1. Координатор: цей пристрій ініціалізує мережу і керує її процесами, встановлює та зберігає ключі безпеки пристроїв, встановлює політику безпеки своєї мережі та встановлює з'єднання з іншими мережами. У кожній мережі ZigBee може бути лише один координатор.

2. Маршрутизатор (Роутер): це пристрій, який відповідає за динамічну передачу пакетів даних у мережі. Його можна підключати до інших маршрутизаторів або до координатора в мережі, а до нього - інші маршрутизатори або дочірні пристрої. Маршрутизатори мають стаціонарне живлення і можуть обслуговувати до 32 дочірніх кінцевих пристроїв одночасно, включаючи ті, які знаходяться в режимі «сплячого».

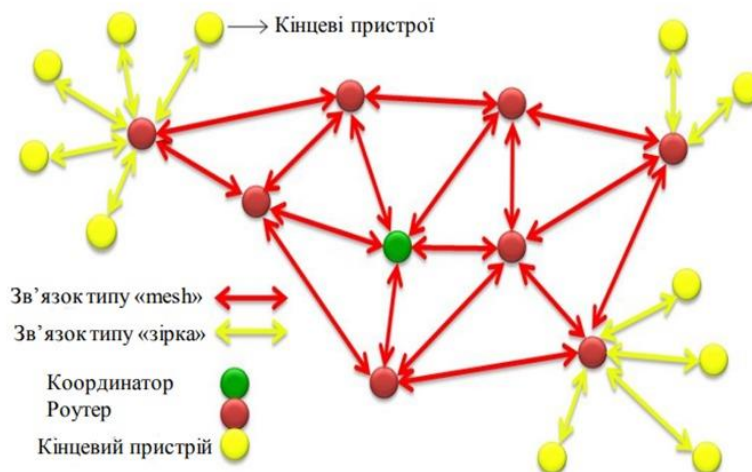


Рис. 1.2. Структура мережі ZigBee

Кінцевий пристрій є складовою системи, яка працює від незалежного джерела енергії і здійснює певні функції за допомогою датчиків або виконавчих механізмів. Цей пристрій підключається до маршрутизатора або координатора, не має дочірніх пристроїв, та може передавати і отримувати пакети даних виключно через маршрутизатор або координатор. Взаємодія з іншими кінцевими пристроями відбувається лише через маршрутизатор або координатор, що призводить до того, що пристрій переважну частину часу перебуває в режимі «сплячого» для збереження заряду [17].

Підсумовуючи, можемо виділити ключові переваги та недоліки технології ZigBee.

Переваги:

- Ефективне використання енергії;
- Значний радіус дії у приміщенні;
- Швидкий перехід з режиму «сплячого» (близько 3 мілісекунди);
- Управління системами за допомогою віддаленого або мобільного пристрою;
- Підтримка різних мережевих топологій, включаючи мережу, здатну відновлювати свою структуру в разі відключення деяких компонентів.

Недоліки:

- Необхідність встановлення вартісних адаптерів на сервері або створення шлюзу між ZigBee і Wi-Fi мережею;
- Низький рівень стандартизації та відсутність єдиної програмно-апаратної платформи для розробки складних додатків;
- Часто недостатня швидкість передачі даних, оскільки більша частина пакетів використовується для передачі службової інформації, такої як адресна інформація та синхронізація. Корисна швидкість передачі складає близько 30 кілобіт на секунду.

1.5 Механізм роботи програмного забезпечення OpenHAB та Zabbix.

Галузь розумних будинків досить давно перебуває у стадії активного розвитку, що призвело до виникнення різноманітних постачальників «розумного» обладнання, такого як освітлення, RGB підсвічування, термостати, приводи для штор/жалюзів, різноманітні датчики тощо, які взаємодіють за різними протоколами. Це призводить до ситуації, коли користувачам, які бажають автоматизувати своє житло, доводиться вибирати між продукцією різних виробників та користуватися різними системами управління для контролю різноманітних пристроїв. Створення повноцінного «Internet of Things» також ускладнюється цим різноманіттям.

OpenHAB вирішує цю проблему, реалізуючи єдину шину, яка дозволяє об'єднати пристрої з різними протоколами в єдину мережу. Таким чином, користувач може використовувати єдиний засіб управління, наприклад, за допомогою додатка на смартфоні, і налагоджувати складну логіку взаємодії між пристроями [18].

Завдяки своїй відкритості, OpenHAB вже наразі підтримує приблизно 50 різних протоколів для «розумних» пристроїв. Серед них такі як ZigBee, Wi-Fi, BLE, ZWave, KNX, EnOcean, аудіо-відео системи, наприклад, Sonos і кінотеатр XBMC, а також пристрої, як Samsung SmartTV та багато інших. Це надає користувачеві можливість вибирати пристрої для свого будинку, враховуючи різноманітні параметри, і практично не обмежується можливостями всієї платформи.

OpenHAB представляє собою спеціалізований сервер, який може працювати на будь-якому комп'ютері, що працює під управлінням однієї з більшості популярних операційних систем. Процес встановлення полягає у розпакуванні дистрибутиву сервера та встановленні Java-машини. Далі розпочинається процес налаштування та налагодження [19].

OpenHAB володіє високою гнучкістю конфігурації, оскільки, базуючись на технології OSGi, він дозволяє змінювати конфігурацію кожного окремого плагіну («Binding») «на льоту», без необхідності перезавантаження серверу.

Кожен пристрій «розумного» будинку має бути налаштований в openHAB як один або кілька елементів (item-ів). Ці елементи описуються у спеціальному файлі в каталозі configuration/items. Нижче наведено приклад одного такого елемента:

Dimmer Light_GF_Living_Table «Table» (GF_Living, Lights)

Описується конкретний елемент з назвою Light_GF_Living_Table і типом Dimmer. Зазначено, що цей пристрій може бути включений, виключений або змінювати своє значення в діапазоні від 0 до 100. Зазвичай Dimmer використовується для управління різними освітлювальними приладами, але може також використовуватися для керування розеткою або іншими функціями [20]. Важливо розуміти, що в конфігурації елемент не обов'язково є окремим пристроєм. Наприклад, його можна налаштувати для керування лише однією функцією складного пристрою, такого як гучність домашнього кінотеатру чи потужність конвекторів. Ось приклад використання Dimmer для управління гучністю.

Dimmer Volume «Volume [%1f %%]»

Як видно, OpenHAB дозволяє використовувати різноманітні item-и як елементи керування для всього, що відповідає будь-яким критеріям (наприклад, може бути включено, вимкнено або приймати значення від 0 до 100). Ім'я item-а дозволяє призначати йому елементи користувацького інтерфейсу в конфігурації та звертатися до нього у коді ваших скриптів і правил. Загалом, синтаксис опису item-а має наступний вигляд.

itemtype itemname [«labeltext»] [] [(group1, group2, ...)] [{bindingconfig}]

Як видно, спочатку вказується тип (Switch, Dimmer, Color, String, Number, Rollershutter, DateTime, Contact, Group), за ним слідує унікальне ім'я в межах

файлу, необов'язковий label (який відображається в користувацькому інтерфейсі), іконка (вибрана з набору openHAB), одна або декілька груп (для групування пристроїв за змістом або розташуванням) і конфігурація самого Біндинга - в фігурних дужках. Біндинг пов'язує item з конкретним пристроєм (або його функцією), що дозволяє openHAB перетворювати абстрактні команди типу On/Off в конкретні повідомлення відповідного протоколу та правильно інтерпретувати повідомлення від пристрою. У кожного Біндинга свій синтаксис конфігурації, який вказується в фігурних дужках. Також важливо зазначити, що у кожного Біндинга існують загальні параметри, які слід визначити в файлі `'configurations/openhab.cfg'`. У каталозі `'configurations/sitemaps'` розташовані файли, в яких описується вигляд користувацького інтерфейсу для керування пристроями, описаними в items. Тут інтерфейс користувача складається лише з одного «фрейму» (не плутати з фреймами в термінах HTML), що містить 4 елементи (два слайдера для управління диммерами, вибір кольору для підсвічування і простий перемикач) [21]. Повний перелік можливих елементів інтерфейсу можна знайти на сторінці Sitemaps.

Zabbix представляє собою інструмент моніторингу мережі, який виконує централізований контроль за доступністю та продуктивністю мережі та пристроїв. У випадку виявлення помилок, адміністратор мережі отримає сповіщення через певний пристрій або електронну пошту. Zabbix - це абсолютно безкоштовний інструмент моніторингу мережі без обмежень у функціональності та кількості контрольованих пристроїв. Користувачі мають офіційне дозвіл на внесення змін до вихідного коду. Крім того, Zabbix підтримує будь-які розміри мережевих інфраструктур, чи то невелика локальна мережа, чи масштабна корпоративна архітектура.

Створений у 1998 році, коли на ринку існували лише HP Open View та IBM BMC, Zabbix вирізняється своєю вартістю та простотою в порівнянні з іншими дорогими та складними в обслуговуванні інструментами моніторингу з відкритим кодом [22].

Zabbix пропонує розширений спектр функцій, підтримуючи як агентні, так і безагентні методи моніторингу для пристроїв, таких як маршрутизатори, комутатори та сервери. Мережеві пристрої повинні підтримувати SNMP для ефективної роботи з Zabbix. Засіб може контролювати доступність пристроїв та їхню продуктивність, а також надає підтримку моніторингу VMware для статистики віртуальних машин. Правила виявлення низького рівня використовуються для виявлення віртуальних машин та гіпервізорів. Zabbix також здійснює моніторинг баз даних та веб-сервісів. У випадку відмови мережі або пристрою Zabbix негайно сповістить системного адміністратора. Механізм виявлення мережевих пристроїв на низькому рівні та їхнє логічне групування також включено. Однак Zabbix не надає можливості прогнозування тенденцій, через що він не може попередити про можливі помилки заздалегідь.

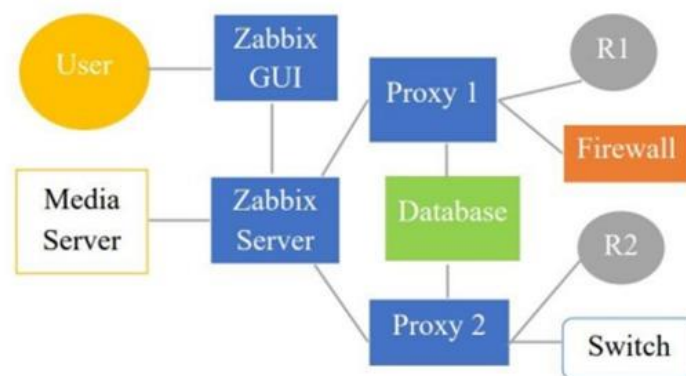


Рис. 1.3 Компоненти Zabbix

Zabbix складається з різних компонентів, таких як: Zabbix Server, Zabbix Proxy, Zabbix Agent та веб-інтерфейс, кожен з яких виконує заздалегідь визначені ролі у моніторингу.

Zabbix Server є основною складовою системи та відповідає за віддалений моніторинг мережі та її компонентів, а також зберігає конфігурації, історичні та експлуатаційні дані. Якщо виникає помилка, Zabbix Server негайно інформує менеджера мережі.

Zabbix Proxy відповідає за збір продуктивнісних даних від імені сервера Zabbix. Його локальний буфер збирає всі дані, які потім передаються на Zabbix Server. Це спрощує обчислювальне навантаження на сервер вводу-виводу [23].

Агент Zabbix відповідає за локальний моніторинг мережевих пристроїв, включаючи відстеження ресурсів, таких як жорсткий диск, пам'ять та статистика процесора. Щоб здійснювати моніторинг ресурсів, агент Zabbix повинен бути встановлений локально на кожному пристрої, працюючи ефективно завдяки власним системним дзвінкам.

Веб-інтерфейс, який також є частиною Zabbix Server, запускається зазвичай на тому ж сервері, що і Zabbix Server, і працює як медіасервер та база даних. Цей інтерфейс забезпечує ефективне читання та запис в базу даних, обходячи Zabbix Server, що покращує продуктивність. З іншого боку, Zabbix Server не може функціонувати без веб-інтерфейсу.

Загалом, ці компоненти дозволяють Zabbix підтримувати три типи моніторингу: проста перевірка, агент Zabbix та зовнішня перевірка. Кожен з них використовується для різних методів контролю доступності та ефективності мережевих пристроїв. Ці процеси також можуть здійснюватися без проксі в залежності від конфігурації та потреб користувача [24].

Zabbix відповідає 90% вимог надійного інструменту моніторингу телекомунікаційної мережі, забезпечуючи як моніторинг на основі агентів, так і без них. Зазначені функції включають виявлення низького рівня, автоматичне виявлення та логічне групування, що робить Zabbix надійним інструментом для будь-якої мережі. Незважаючи на це, важливо відзначити відсутність функції прогнозування тенденцій у Zabbix, оскільки це може вплинути на загальну ефективність інструменту. Тим не менш, Zabbix є надійним та передбачуваним інструментом моніторингу мережі, сповіщаючи користувача про помилки та надаючи впевненість у їх наявності. Ці принципи застосовуються також до процесів відновлення та візуалізації.

Однією з ключових переваг Zabbix є його масштабованість, що робить його застосовним у будь-якому розмірі середовища. Масштабованість застосовується до продуктивності та зручності використання інтерфейсу.

1.6 Порівняння алгоритмів кластеризації (K-Means, LP, Simple PSO та Hybrid PSO)

Кластеризація представляє собою метод машинного навчання, що включає у себе групування точок даних в набори, відомі як кластери. При використанні алгоритму кластеризації для набору точок даних, ми можемо віднести кожен точку до конкретної групи. Точки, що належать до одного кластеру, мають схожі властивості чи особливості, тоді як ті, що належать різним кластерам, мають відмінні властивості [27].

Цей метод належить до категорії неконтрольованого навчання і є широко використовуваною технікою для аналізу статистичних даних. Кластерний аналіз даних базується на двох припущеннях. По-перше, вважається, що ознаки об'єктів піддаються бажаному розбиттю на кластери, і, по-друге, важливе значення має вибір масштабу або одиниць вимірювання ознак.

Застосування кластерного аналізу включає в себе вибір об'єктів для кластеризації, визначення набору змінних (атрибутів), за якими проводиться оцінка об'єктів у вибірці, при необхідності – нормалізація значень змінних, обчислення міри схожості між об'єктами та застосування методу кластерного аналізу для формування груп подібних об'єктів, які називаються кластерами, та представлення результатів аналізу.

Алгоритм k-means спрямований на мінімізацію відстані між точками в кластері та їх центроїдами. Цей метод, ґрунтований на центроїдах, визначає кластери на основі відстані, обчислюючи відстані для визначення присвоєння кожної точки до певного кластеру. У кожному кластері присутній центроїд [28].

Основною метою алгоритму k-means є зниження суми відстаней між точками та їх відповідними центроїдами у кожному кластері.

Щодо критеріїв зупинки кластеризації k-means, існують три основних [29]:

1. Центроїди новоутворених кластерів залишаються незмінними.
2. Точки залишаються у своїх початкових кластерах.
3. Досягнута максимальна кількість ітерацій.

Алгоритм може бути зупинений, якщо центроїди новоутворених кластерів залишаються незмінними. Якщо після декількох ітерацій отримано однакові центроїди для всіх кластерів, це вказує на те, що алгоритм не вивчає нових шаблонів, і процес може бути зупинений. Інший варіант - зупинка, якщо точки залишаються в тому ж кластері після декількох ітерацій. Завершення процесу також можливе, якщо досягнута задана максимальна кількість ітерацій, наприклад, 100.

Алгоритм кластеризації LP (Label Propagation) є методом неконтрольованого машинного навчання, спрямованим на призначення категорій елементам в графі на основі взаємозв'язків між ними. Основна ідея полягає в тому, що елементи в графі отримують мітки (категорії) від своїх сусідів. Процес вирішення міток проводиться ітеративно, де кожен елемент поширює свою мітку на сусідів, враховуючи їхні ваги.

Алгоритм починається з ініціалізації кожного елемента унікальною міткою, а потім мітки розповсюджуються вздовж зв'язків у графі. У кожній ітерації елемент оновлює свою мітку, враховуючи мітки його сусідів та ваги зв'язків. Цей процес триває до збіжності, коли мітки стабілізуються.

Особливість LP полягає в тому, що він не вимагає заздалегідь визначеного числа кластерів і може виявляти структуру кластерів в графі, що робить його корисним для аналізу невизначених даних або графових структур [30].

Алгоритм кластеризації Simple PSO (Particle Swarm Optimization) є методом оптимізації, вдосконаленим для вирішення задач кластеризації. У

цьому алгоритмі кластеризації, група частинок, які представляють елементи даних, рухається у просторі пошуку, прагнучи знайти оптимальне розташування кластерів.

Кожна частинка в Simple PSO представляє кластер та утримує інформацію про своє розташування в просторі кластеризації та якість кластеризації. Частинки взаємодіють між собою, обмінюючи інформацією про кращі рішення. Ця взаємодія сприяє збільшенню ймовірності знаходження оптимального розташування кластерів.

У кожній ітерації алгоритму частинки рухаються в просторі, керуючись своїм поточним положенням та інформацією від найкращих сусідів. Процес триває до досягнення критерію зупинки, такого як фіксована кількість ітерацій або досягнення заданої якості кластеризації. Алгоритм Simple PSO виявляється ефективним для розв'язання задач кластеризації завдяки його здатності уникнути локальних мінімумів та ефективно використовувати простір пошуку [31].

Алгоритм кластеризації Hybrid PSO (гібридна оптимізація рою частинок) представляє собою еволюційний підхід до вирішення задач кластеризації, який комбінує переваги алгоритму PSO та інших методів. У цьому підході, кластеризація здійснюється за допомогою рою частинок, які ефективно пристосовуються до простору пошуку для знаходження оптимального розташування кластерів.

Одна з основних особливостей Hybrid PSO полягає в тому, що він використовує додаткові еволюційні механізми або операції для покращення процесу кластеризації. Це може включати в себе використання генетичних алгоритмів, мутацій або інших технік, які допомагають різноманітнити та вдосконалити рої частинок для знаходження більш точних та ефективних кластерів [32].

Під час кожної ітерації Hybrid PSO, частинки в рої пристосовують свої положення в просторі, оптимізуючи функцію кластеризації. Використання гібридного підходу дозволяє покращити швидкість та точність знаходження

оптимальних рішень, забезпечуючи більш гнучкий та ефективний алгоритм для складних завдань кластеризації.

Таблиця 1.2

Порівняння алгоритмів кластеризації

Характеристика	K-Means	LP	Simple PSO	Hybrid PSO
Тип алгоритму	Центроїдний	Ієрархічний	Еволюційний	Гібридний (еволюційний та частинок)
Стійкість до вихідних даних	Впливає на результат	Залежить від методу об'єднання	Залежить від параметрів ініціалізації	Залежить від параметрів ініціалізації
Залежність від кількості кластерів	Так	Ні	Залежить від параметрів	Залежить від параметрів
Здатність обробляти великі об'єми даних	Так	Залежить від алгоритму	Залежить від параметрів	Так
Вимоги до обчислювальних ресурсів	Низькі	Залежать від кількості об'єктів	Середні	Середні
Стійкість до шуму в даних	Ні	Залежить від алгоритму	Залежить від параметрів	Так
Можливість обробки різних типів даних	Так	Залежать від кількості об'єктів	Так	Так
Використання додаткових еволюційних механізмів	Ні	Ні	Ні	Так
Підтримка гібридних методів	Ні	Ні	Ні	Так

Аналіз алгоритмів кластеризації K-Means, LP (Linkage Probability), Simple PSO (Простий алгоритм оптимізації рою частинок) та Hybrid PSO (Гібридний алгоритм оптимізації рою частинок) у контексті розумного будинку визначається особливостями цих методів та їхньою ефективністю в вирішенні завдань, пов'язаних із кластеризацією даних розумного будинку.

Алгоритм K-Means може бути використаний для кластеризації різних пристроїв у розумному будинку, таких як освітлення, термостати, аудіо-відео системи тощо. Кластери можуть групувати пристрої за типом, локацією чи часовим розподілом, що полегшує керування та автоматизацію.

Алгоритм LP може бути корисним для ієрархічної кластеризації пристроїв у розумному будинку з урахуванням ймовірності зв'язків між

ними. Наприклад, можна визначити, як пристрої взаємодіють між собою для вдосконалення системи автоматизації, забезпечуючи ефективніше споживання енергії та зручне керування.

Алгоритми PSO можуть бути використані для оптимізації режимів роботи пристроїв у розумному будинку. Simple PSO може допомагати в пошуку оптимальних параметрів для забезпечення оптимального споживання енергії. Hybrid PSO може комбінувати переваги PSO з іншими методами для кращої адаптації до різноманітних умов та потреб розумного будинку.

Висновки

Локальний сервер для розумного дому є ключовою складовою інфраструктури, яка дозволяє централізовано керувати та моніторити пристроями вдома. Використання локального сервера дозволяє забезпечити надійність та конфіденційність даних, а також знижує залежність від зовнішніх мереж та сервісів. Хмарне середовище для розумного дому відкриває нові можливості для зручного дистанційного керування та моніторингу пристроїв через Інтернет. Це розв'язання забезпечує гнучкість та доступність, але потребує уважного врахування питань безпеки та конфіденційності.

Принципи будови складових мереж розумного будинку визначають структуру та взаємозв'язки між пристроями. Локальні мережі, хмарні сервіси та принципи взаємодії створюють інфраструктуру, яка забезпечує ефективну роботу системи, враховуючи вимоги щодо швидкості, безпеки та доступності. OpenHAB та Zabbix – це різні системи, проте обидві грають важливу роль у розумних будинках. OpenHAB використовується для забезпечення інтеграції та автоматизації пристроїв, тоді як Zabbix відповідає за моніторинг та управління їхньою продуктивністю. Механізми їхньої роботи ретельно налаштовані для забезпечення стабільності та ефективності системи. Порівняння алгоритмів кластеризації, таких як K-Means, LP, Simple

PSO та Hybrid PSO, вказує на їхню різноманітність та застосування у різних сценаріях розумного будинку.

РОЗДІЛ 2

МОДЕЛІ КЛАСТЕРИЗАЦІЇ МЕРЕЖІ ІНТЕРНЕТУ РЕЧЕЙ ДЛЯ МІНІМІЗАЦІЇ КІЛЬКОСТІ ШЛЮЗІВ

2.1 Сутність моделей кластеризації

Кластеризація, відома також як кластерний аналіз, є статистичною процедурою, що спрямована на розподіл набору об'єктів у групи, які не перетинаються і називаються кластерами. Кожен кластер включає схожі об'єкти, при цьому об'єкти в різних кластерах значно відрізняються один від одного. Кластеризація входить в склад статистичної обробки та розглядається у широкому контексті завдань навчання без учителя. Можна також розглядати кластеризацію як завдання класифікації.

Задача кластеризації схожа до задачі класифікації, оскільки обидві передбачають розділення об'єктів на основі їх подібності. Проте, у випадку кластеризації не визначається приналежність об'єктів до конкретних класів. Це універсальне завдання, яке можна вирішити різними методами. Алгоритми формування кластерів можуть відрізнятися у своїх підходах до визначення того, що включати в один кластер, а також у ефективності пошуку [27].

Методи формування кластерів можуть базуватися на відстані між об'єктами, щільності областей в просторі даних, інтервалах або конкретних статистичних розподілах. Вибір конкретного методу залежить від конкретного набору даних і мети отриманих результатів. Важливо враховувати, що кластерний аналіз є ітераційним процесом, оскільки часто доводиться адаптувати методи обробки даних та параметри моделі, щоб досягти бажаних властивостей.

Вирішення задачі кластеризації є багатозначним з кількох причин. По-перше, відсутність єдиного критерію якості кластеризації призводить до наявності різноманітних ефективних критеріїв і алгоритмів, які можуть приводити до різних результатів. По-друге, часто невідомо точна кількість

кластерів, і ця кількість визначається за допомогою суб'єктивних критеріїв. По-третє, результат кластеризації значно залежить від обраної метрики, вибір якої також є суб'єктивним і залежить від експертної оцінки.

Щодо формального формулювання завдання, ми маємо набір даних X , позначений як $\ell = \{x_1, \dots, x_\ell\} \subset X$, та функцію відстані між об'єктами $\rho(x, x')$. Завданням є розбиття набору на підмножини, які не перетинаються, так, щоб кожен кластер містив об'єкти, близькі за відстанню ρ , і об'єкти різних кластерів суттєво відрізнялись. Кожному об'єкту $x_i \in X_\ell$ призначається мітка (номер) кластера u_i .

Алгоритм кластеризації, позначений як функція $a: X \rightarrow Y$, відображає кожен об'єкт $x \in X$ у відповідну мітку кластера $y \in Y$. У деяких випадках множина міток Y відома заздалегідь, але частіше визначення оптимальної кількості кластерів є завданням з вибору критерію якості кластеризації.

Задача групування об'єктів полягає в тому, щоб об'єкти в одному кластері були більш схожі один на одного, ніж на об'єкти інших кластерів. Рівень подібності визначається числовим показником, який відображає ступінь взаємозв'язку між двома об'єктами. Кластеризація застосовується для аналізу даних і також в різних областях, таких як машинне навчання, розпізнавання образів, аналіз зображень, пошук інформації, біоінформатика, стиснення даних і комп'ютерна графіка. [29].

Існують два типи кластеризації: жорстка і м'яка. У жорсткій кластеризації кожен об'єкт повністю належить або не належить кластеру. У м'якій кластеризації точка або об'єкт даних може з певною ймовірністю належати більше ніж одному кластеру.

Різновиди вхідних даних та питання міри відстані:

1. Кожен об'єкт описується своїми характеристиками, які відомі як ознаки. Ознаки можуть бути числовими чи нечисловими.
2. Матриця відстаней між об'єктами. Кожен об'єкт характеризується відстанями до всіх інших об'єктів у навчальній вибірці.

Дані, які використовуються у кластерному аналізі, можуть мати інтервальний, порядковий або категоріальний характер. Проте, наявність комбінації різних типів змінних ускладнює процес аналізу. Це пояснюється тим, що для проведення кластерного аналізу необхідний конкретний метод вимірювання відстані між спостереженнями, і цей метод залежить від типу даних.

Для вимірювання відстані в разі бінарних та категоріальних даних існують різні альтернативи. Наприклад, для інтервальних даних часто використовується евклідова відстань.

Кластеризація - це завдання, яке має суб'єктивний характер, і може існувати багато правильних алгоритмів для його вирішення. Кожен алгоритм дотримується своїх правил для визначення «подібності» між об'єктами даних. Вибір оптимального алгоритму кластеризації для конкретної задачі часто вимагає експериментального підходу, оскільки немає однозначної математичної підстави для віддачі переваги одному алгоритму перед іншим. Алгоритм може добре справлятися з одним набором даних, але бути менш ефективним для іншого.

Алгоритми кластеризації можна розподілити за кластерною моделлю, що визначає, як вони формують кластери чи групи. Наприклад:

- 1) Кластеризація на основі зв'язку: ідея полягає в тому, що близькі точки в просторі даних є більш спорідненими. Кластери формуються шляхом об'єднання точок відповідно до їх відстані. Різні відстані призводять до формування різних кластерів, які можна представити у вигляді дендрограми (ієрархічна кластеризація). Ці методи створюють ієрархію, а не єдиний розподіл даних, і користувачам все ще слід вибрати відповідні кластери, тобто рівень для кластеризації. Вони також не дуже стійкі до викидів, які можуть спричинити появу додаткових кластерів або навіть об'єднання інших кластерів.

- 2) Кластеризація на основі центроїда: в цьому типі кластери представлені центральним вектором або центроїдом. Центроїд не обов'язково

повинен бути частиною набору даних. Це ітеративні алгоритми, де подібність визначається відстанню точки даних до центроїда кластера. Наприклад, метод k-середніх є формою кластеризації на основі центроїда.

Метод k-середніх широко використовується для кластеризації, націлюючись на зменшення середньоквадратичної відстані між точками в межах кожного кластера. Навіть якщо він не гарантує абсолютної точності, його простота та швидкість компенсують це. Вдосконаленням методу k-середніх з використанням простої рандомізованої техніки висіву отримуємо алгоритм, який конкурує з оптимальною кластеризацією за складністю $O(\log k)$. Експерименти показали, що це поліпшення значно підвищує швидкість і точність методу k-середніх.

Перевагою методу k-середніх є те, що він більш зручний для кластеризації великої кількості спостережень порівняно з методом ієрархічного кластерного аналізу, де дендограми швидко стають перенасиченими і втрачають наочність.

Недоліками методу можна відзначити його чутливість до викидів, які можуть вплинути на середнє значення. Також кількість кластерів (означена тут як k) має бути задана фахівцем перед використанням.

В загальному сценарії кластеризація використовується для отримання додаткової інформації про представлений набір даних. Таким чином, розумно провести процес кластеризації неодноразово, вивчаючи цікаві кластери та записуючи отримані висновки. Кластеризація служить інструментом для кращого вивчення даних, але не завжди повинна використовуватися як автоматичний метод класифікації даних. Це може бути не зовсім надійним, і один лише алгоритм кластеризації не забезпечить всю необхідну інформацію, яку можна отримати з набору даних.

Основна ідея методу k-середніх полягає в тому, щоб визначити кластери таким чином, щоб мінімізувати загальну варіацію між кластерами, тобто сумарну варіацію в межах кожного кластера [33].

Для цього методу існує ряд алгоритмів, проте стандартний визначає загальну варіацію в межах кластера як суму квадратичних евклідових відстаней між елементами та відповідним центроїдом:

$$W(C_k) = \sum_{x_i \in C_k} (x_i - \mu_k)^2 \quad (2.1)$$

Де x — точка даних, яка належить до кластера C_k , μ_k — середнє значення точок, що належать кластеру C_k .

Кожне спостереження x призначається конкретному кластеру таким чином, щоб сума квадратів відстані між спостереженням та призначеним йому центроїдом μ була мінімізована.

Алгоритм має наступний вигляд:

1. Визначається k — кількість кластерів, які будуть створені.
2. Вибираються випадкові k об'єктів з набору даних як початкові центроїди кластерів.
3. Кожне спостереження призначається найближчому центроїду на основі евклідової відстані між об'єктом і центроїдом.
4. Для кожного з k кластерів перераховується центроїд кластера, обчислюючи нове середнє значення всіх точок даних у кластері.
5. Ітеративно мінімізується загальна сума в межах площі. Повторюються кроки 3 і 4 до тих пір, поки центроїди не зміняться або досягне максимальної кількості ітерацій (зазвичай R використовується як максимальна кількість ітерацій, і за замовчуванням це значення становить 10).

Загальна сума або загальна варіація в межах кластера обчислюється наступним чином:

$$\sum_{k=1}^k W(C_k) = \sum_{k=1}^k \sum_{x_i \in C_k} (x_i - \mu_k)^2 \quad (2.2)$$

Алгоритми ієрархічної кластеризації, також відомі як алгоритми таксономії, створюють систему вкладених розбиттів для вибірки, а не одне непересічне розбиття. Результат таксономії зазвичай представляється у вигляді дендрограми, що є деревом подібної ієрархії. Дендрограма дозволяє

відобразити кластерну структуру у вигляді плаского графіка, що залишається незалежним від розмірності початкового простору даних. Існують інші методи візуалізації багатовимірних даних, такі як багатовимірне шкалювання або карти Кохонена, але вони можуть внести штучні спотворення, важко оцінювані в великому масштабі.

Ієрархічні методи поділяються на два типи:

1. Агломеративні методи: нові кластери формуються шляхом поступового об'єднання менших кластерів, створюючи дерево від листя до стовбура.

2. Дивізійні методи: нові кластери виникають шляхом поділу більших кластерів на менші, створюючи дерево від стовбура до листя.

Оцінка подібності між кластерами часто виконується через «неподібність», наприклад, використання евклідової відстані між ними. Іншими словами, чим більше відстань між двома кластерами, тим менше їх подібність.

Ключовим етапом в ієрархічній агломеративній кластеризації є об'єднання двох найближчих кластерів у кожному кроці, проте належність до кластера має бути визначена відповідно до трьох ключових аспектів: як представити кластер із більше, ніж однієї точки; як визначити «близькість» між кластерами; коли припинити об'єднання кластерів.

Зупинка об'єднання кластерів залежить від доступної інформації про дані. Наприклад, при групуванні футболістів на полі, можна зупинитися на лише двох кластерах, оскільки грають лише дві команди.

Процедура методу може бути описана наступним чином:

1. Розрахувати матрицю близькості, яка визначає відстань між кожною парою шаблонів, розглядаючи кожен шаблон як окремий кластер.

2. Знайти найбільш схожу пару кластерів за допомогою матриці близькості, об'єднати їх в один більший кластер та оновити матрицю близькості, відображаючи цю операцію злиття.

3. Якщо всі шаблони вже належать одному кластеру, завершити процес. В іншому випадку перейти до кроку 2.

Ієрархічний алгоритм надає дендрограму, що відображає складне групування шаблонів і рівні схожості, на яких змінюються групування.

Більшість ієрархічних алгоритмів кластеризації є варіаціями однозв'язкового та повнозв'язкового підходів, а також алгоритму мінімальної дисперсії. Однозв'язковий та повнозв'язковий алгоритми є найбільш популярними, різнячись у способі оцінки подібності між кластерами.

У випадку однозв'язкового алгоритму відстань між двома кластерами обчислюється як мінімум відстаней між усіма парами шаблонів, взятих з цих кластерів. У повнозв'язковому алгоритмі відстань між кластерами розглядається як максимум усіх попарних відстаней між шаблонами в обох кластерах. В обох випадках два кластери об'єднуються на основі критерію мінімальної відстані. Повнозв'язковий алгоритм формує щільні та компактні кластери, тоді як однозв'язковий страждає від ланцюгового ефекту, утворюючи кластери, які можуть бути або занадто громіздкими, або занадто витягнутими.

2.2 Класифікація на основі виявлення маршруту

Давайте проведемо аналіз набору даних за допомогою розглянутих раніше методів кластеризації. Наш вибір падає на набір даних, що відомий як «Іриси Фішера», який був використаний Рональдом Фішером для демонстрації роботи свого розробленого методу дискримінантного аналізу. Також відомий як «іриси Андерсона» через їх збір американським ботаніком Едгаром Андерсоном. Цей класичний набір даних часто використовується в літературі для наочного пояснення різних статистичних алгоритмів.

```
library(datasets)
data(iris)
str(iris)
summary(iris)
> str(iris)
'data.frame': 150 obs. of 5 variables:
 $ Sepal.Length: num  5.1 4.9 4.7 4.6 5 5.4 4.6 5 4.4 4.9 ...
 $ Sepal.Width : num  3.5 3 3.2 3.1 3.6 3.9 3.4 3.4 2.9 3.1 ...
 $ Petal.Length: num  1.4 1.4 1.3 1.5 1.4 1.7 1.4 1.5 1.4 1.5 ...
 $ Petal.Width : num  0.2 0.2 0.2 0.2 0.2 0.4 0.3 0.2 0.2 0.1 ...
 $ Species     : Factor w/ 3 levels "setosa","versicolor",...: 1 1 1 1 1 1 1 1 1 1 ...
> summary(iris)
  Sepal.Length    Sepal.Width    Petal.Length    Petal.Width      Species
Min.   :4.300    Min.   :2.000    Min.   :1.000    Min.   :0.100    setosa   :50
1st Qu.:5.100    1st Qu.:2.800    1st Qu.:1.600    1st Qu.:0.300    versicolor:50
Median :5.800    Median :3.000    Median :4.350    Median :1.300    virginica :50
Mean   :5.843    Mean   :3.057    Mean   :3.758    Mean   :1.199
3rd Qu.:6.400    3rd Qu.:3.300    3rd Qu.:5.100    3rd Qu.:1.800
Max.   :7.900    Max.   :4.400    Max.   :6.900    Max.   :2.500
```

Рис. 2.1. Середня частина розробки

Цей відомий набір даних про іриси містить вимірювання в сантиметрах довжини та ширини чашолистка, а також довжини та ширини пелюстки для 50 квітів кожного з трьох видів ірису: *iris setosa*, *versicolor* та *virginica*.

Тепер давайте ретельніше розглянемо цей набір даних і вивчимо наявні кореляції між різними змінними.

```
> cor(iris[1:4])
      Sepal.Length Sepal.Width Petal.Length Petal.Width
Sepal.Length    1.0000000 -0.1175698  0.8717538  0.8179411
Sepal.Width     -0.1175698  1.0000000 -0.4284401 -0.3661259
Petal.Length     0.8717538 -0.4284401  1.0000000  0.9628654
Petal.Width      0.8179411 -0.3661259  0.9628654  1.0000000
```

Рис. 2.2. Зміни в кореляції

Виявляємо наявність відзначених та значущих кореляцій. Наприклад, спостерігається кореляція на рівні 87% між довжинами пелюсток і чашолистків, а також 96% між шириною та довжиною пелюсток.

Деякі аспекти викидів у даних також привертають наше увагу. Аналіз показує, що немає критичних аномалій; виявлені лише невеликі відхилення у змінній довжини чашолистка.

```
boxplot(iris$Sepal.Length, iris$Sepal.Width,  
iris$Petal.Length, iris$Petal.Width)
```

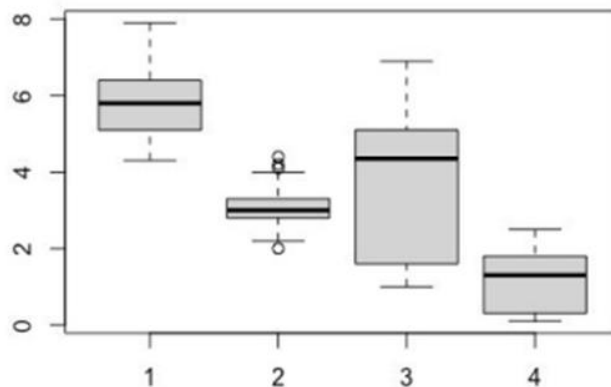


Рис. 2.3. Графік пелюсток та чашолистків

```
g1 <- ggplot(iris, aes(x = Sepal.Length, y = Sepal.Width,  
color = Species)) +  
  geom_point() +  
  labs(x = «Sepal Length», y = «Sepal Width»)  
g2 <- ggplot(iris, aes(x = Petal.Length, y = Petal.Width,  
color = Species)) +  
  geom_point() +  
  labs(x = «Petal Length», y = «Petal Width»)  
grid.arrange(g1, g2, nrow = 1)
```

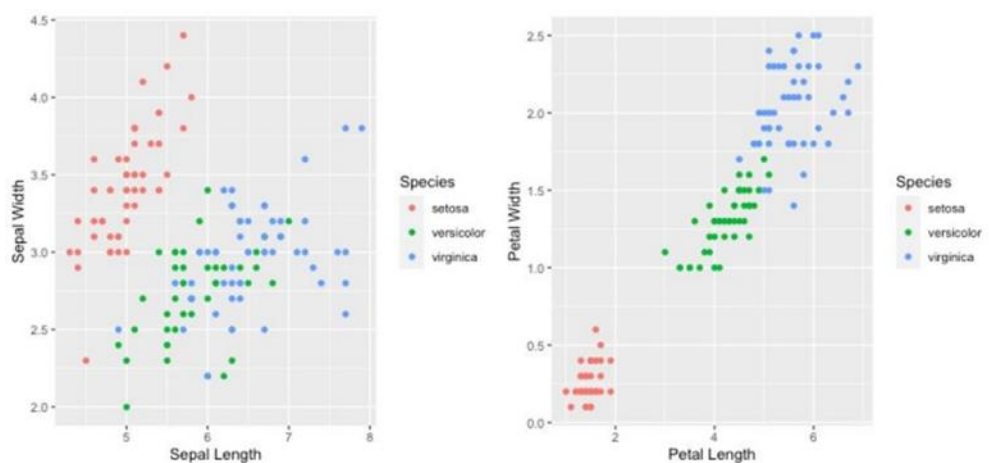


Рис. 2.4. Графік довжини та ширини пелюсток та чашолистків

на це, було виявлено, що точність розпізнавання була високою, навіть якщо вид versicolor був розпізнаний менш точно [34].

Модель кластеризації Simple PSO (Particle Swarm Optimization) - це метод, що базується на ідеях оптимізації з використанням алгоритму зграї частинок. У контексті кластеризації, такий підхід використовує зграї частинок для пошуку оптимального розподілу даних на кластери.

```
def __init__(self, dimensions, min_values, max_values):
    self.position = np.random.uniform(min_values, max_values, dimensions)
    self.velocity = np.random.rand(dimensions)
    self.best_position = np.copy(self.position)
    self.best_fitness = float('inf')

def fitness_function(data, centroids):
    # Реалізуйте функцію оцінки фітнесу, враховуючи відстані між точками і центрами кластерів
    pass

def update_velocity(particle, inertia_weight, personal_coeff, global_coeff, global_best_position):
    # Оновлення швидкості частинки згідно з алгоритмом PSO
    pass

def update_position(particle):
    # Оновлення позиції частинки згідно з алгоритмом PSO
    pass

def simple_pso_clustering(data, num_clusters, num_particles, num_iterations):
    dimensions = data.shape[1] # Кількість визначених рис у кожній точці даних
    min_values = np.min(data, axis=0)
    max_values = np.max(data, axis=0)

    # Ініціалізація зграї частинок
    particles = [Particle(dimensions, min_values, max_values) for _ in range(num_particles)]

    # Знайдення найкращого кластеру
    global_best_particle = min(particles, key=lambda x: x.best_fitness)

    # Головний цикл PSO
    for iteration in range(num_iterations):
        for particle in particles:
            # Оцінка фітнес-функції для поточної частинки
            particle.fitness = fitness_function(data, particle.position)

            # Оновлення найкращого розв'язку для частинки
            if particle.fitness < particle.best_fitness:
                particle.best_fitness = particle.fitness
                particle.best_position = np.copy(particle.position)

            # Оновлення глобального найкращого розв'язку
            if particle.best_fitness < global_best_particle.best_fitness:
                global_best_particle = np.copy(particle)

        for particle in particles:
            # Оновлення швидкості та позиції частинок
            update_velocity(particle, inertia_weight, personal_coeff, global_coeff, global_best_particle.position)
            update_position(particle)

    # Формування кластерів на основі отриманих розв'язків
    centroids = [particle.best_position for _ in range(num_clusters)]
    clusters = [[] for _ in range(num_clusters)]

    for point in data:
        # Призначення кожної точки до найближчого кластеру
        distances = [np.linalg.norm(point - centroid) for centroid in centroids]
        assigned_cluster = np.argmin(distances)
        clusters[assigned_cluster].append(point)

    return clusters
```

Рис. 2.7. Модель кластеризації Simple PSO

```
# Оновлення найкращого розв'язку для частинки
if particle.fitness < particle.best_fitness:
    particle.best_fitness = particle.fitness
    particle.best_position = np.copy(particle.position)

# Оновлення глобального найкращого розв'язку
if particle.best_fitness < global_best_particle.best_fitness:
    global_best_particle = np.copy(particle)

for particle in particles:
    # Оновлення швидкості та позиції частинок
    update_velocity(particle, inertia_weight, personal_coeff, global_coeff, global_best_particle.position)
    update_position(particle)

# Формування кластерів на основі отриманих розв'язків
centroids = [particle.best_position for _ in range(num_clusters)]
clusters = [[] for _ in range(num_clusters)]

for point in data:
    # Призначення кожної точки до найближчого кластеру
    distances = [np.linalg.norm(point - centroid) for centroid in centroids]
    assigned_cluster = np.argmin(distances)
    clusters[assigned_cluster].append(point)

return clusters

# Параметри PSO
inertia_weight = 0.5
personal_coeff = 1.5
global_coeff = 1.5
num_clusters = 3
num_particles = 20
num_iterations = 100
```

Рис. 2.8. Модель кластеризації Simple PSO

Модель кластеризації Hybrid PSO (гібридна модель з використанням алгоритму частинок та іншого методу) може використовувати, наприклад, комбінацію алгоритму частинок (PSO) та методу кластеризації, такого як k-середніх. Основна ідея полягає в тому, щоб об'єднати переваги різних методів для покращення ефективності кластеризації.

Основні характеристики гібридної моделі кластеризації Hybrid PSO з використанням PSO та k-середніх [35]:

1. Ініціалізація:

Згенерувати випадковим чином початкові положення частинок PSO. Ініціалізувати центроїди для k-середніх.

2. Оцінка фітнес-функції:

Визначити функцію фітнесу для кожної частинки PSO, використовуючи наприклад відстані між точками та центроїдами кластерів.

3. Оновлення швидкості та позиції частинок PSO:

Використовувати стандартні правила PSO для оновлення швидкості та позиції частинок.

4. Оновлення центроїдів для k-середніх:

Після кожної ітерації PSO, оновлювати центроїди для k-середніх, використовуючи призначення точок до кластерів.

5. Ітеративний процес:

Повторювати кроки 2-4 протягом заданої кількості ітерацій.

6. Формування кластерів:

Використовувати останні центроїди кластерів після закінчення ітерацій для формування кластерів.

Такий гібридний підхід може дозволити використовувати переваги PSO для глобальної оптимізації та зближення до локальних оптимумів разом з

ефективністю k-середніх у визначенні центроїдів та створенні кластерів. Важливо експериментувати з параметрами, щоб забезпечити оптимальну продуктивність на конкретних даних.

Модель оптимізації рою частинок отримала свою назву внаслідок свого походження від еволюційного алгоритму, який був розроблений на основі вивчення алгоритмів, що моделюють соціальну поведінку тварин, таких як зграї птахів і роїня бджіл.

Ця модель включає параметри, аналогічні до методу K-середніх, такі як кількість шлюзів (c), кількість пристроїв (n), бажана площа ($A_{i,j}$) і положення пристрою ($D_{i,j}$). У випадку моделі PSO, кількість частинок (p) є унікальним параметром, що визначається вхідним значенням. Кожна частинка у популяції представляє одне можливе рішення та є вектором, що містить таку саму кількість позицій, як і шлюзів ($P_i = (P_1, P_2, \dots, P_c)$) [36].

Рух кожної частинки в просторі пошуку здійснюється з певною швидкістю (v), і вони мають "пам'ять" - свою найкращу позицію ($pbest$). Крім того, існує "колективна пам'ять", що враховує найкращу глобальну позицію для всього рою ($gbest$). Фітнес-функція визначає, наскільки кожна частинка відповідає поставленій меті.

Таким чином, модель PSO використовує принципи колективної поведінки для оптимізації пошуку в просторі параметрів.

Ініціалізація частинок проводилася випадковим чином в межах області пошуку. Після ініціалізації, найкраще глобальне положення ($gbest$) оновлювалося з урахуванням положень частинок. Ця модель позиціонує шлюзи, враховуючи найкраще глобальне положення, визначене розташуванням частинок. Відповідно до цього, швидкість частинок оновлювалася відповідно до рівнянь (2.1) і (2.2), які визначали оновлені позиції частинок на певній ітерації.

$$v_i^{(t+1)} = w^t v_i^t + c_1 r_1 (pbest_i^t - p_i^t) + c_2 r_2 (gbest_i^t - p_i^t) \quad (2.3)$$

$v(t+1)_i$ - оновлена швидкість i -тої частинки на ітерації $(t+1)$,

$w(t)$ - коефіцієнт інерції на ітерації t ,

$v(t)_i$ - швидкість i -тої частинки на ітерації t ,

c_2 - когнітивний і соціальний коефіцієнти,

r_2 - випадкові значення між 0 і 1,

$pbest(t)_i$ - найкраща локальна точка для i -тої частинки на ітерації t ,

$x(t)_i$ - положення i -тої частинки на ітерації t ,

$gbest_i$ - найкраща глобальна точка для i -тої частинки.

$$p_i^{(t+1)} = p_i^t + v_i^{(t+1)}$$

де $p(t+1)_i$ вказує на нове положення частинки, $p(t)_i$ є поточним положенням частинки, а $v(t+1)_i$ - новою швидкістю, обчисленою згідно з рівнянням (тут вставте номер рівняння).

$$w^t = w_{initial} - (w_{initial} - w_{final}) \cdot t / m_i$$

Локальне програмування (LP) в контексті кластеризації означає використання методів оптимізації для розбиття набору даних на кластери. LP-моделі кластеризації використовуються для мінімізації об'єктивних функцій, які визначають ступінь подібності у межах кластера та відмінностей між кластерами. Такі моделі можуть враховувати різні аспекти подібності, такі як евклідова відстань, косинусна схожість, або інші метрики в залежності від конкретного завдання.

Однією з популярних LP-моделей кластеризації є метод мінімальної дисперсії (Variance Minimization). У цьому підході кожен кластер визначається своїм центроїдом, а об'єктивна функція мінімізує дисперсію в

межах кожного кластера. Такий підхід особливо ефективний, коли кластери мають компактну структуру [37].

Інша LP-модель, відома як метод максимальної гомогенності (Homogeneity Maximization), спрямована на максимізацію подібності між об'єктами в одному кластері та мінімізацію подібності між різними кластерами. Цей підхід допомагає створювати кластери з однорідними групами об'єктів, що подібні за деяким критерієм.

Нарешті, метод оптимізації суми квадратів (Sum of Squares Optimization) є ще однією LP-моделлю кластеризації, де завданням є мінімізація суми квадратів відстаней між кожним об'єктом та центроїдом його приналежного кластера. Цей підхід допомагає створювати кластери зі збалансованою структурою та компактними групами об'єктів.

LP-моделі кластеризації є потужним інструментом, оскільки вони базуються на математичних методах оптимізації, що дозволяють ефективно розв'язувати завдання кластеризації в різних областях застосування.

Таблиця 2.1

Порівняльний аналіз моделей кластеризації

Модель	Обчислювальна складність	Необхідність апріорних знань	Залежність від ініціалізації	Здатність коректно працювати з викидами
LP	Висока	Невелика	Залежить від методу	Залежить від вибору критерію апроксимації
K-Means	Середня	Невелика	Так	Чутливий, може впливати на результати
Simple PSO	Низька	Мінімальна	Так	Може привести до субоптимальних рішень
Hybrid PSO	Середня	Мінімальна	Так	Залежить від конфігурації гібридизації

- Обчислювальна складність:
 - Висока складність вказує на те, що LP вимагає значних обчислювальних ресурсів, оскільки використовує методи оптимізації для розв'язання задачі кластеризації.
 - K-Means та Simple PSO мають помірну обчислювальну складність, що робить їх придатними для більших наборів даних.
 - Hybrid PSO має середню складність, залежну від конфігурації гібридизації, що може вплинути на обчислювальні витрати.
- Необхідність апріорних знань:
 - LP та K-Means не потребують значних апріорних знань, оскільки вони базуються на математичних методах кластеризації.
 - Simple PSO та Hybrid PSO також не вимагають значних апріорних знань, забезпечуючи простість використання.
- Залежність від ініціалізації:
 - LP має залежність від обраного методу оптимізації та вибору критерію апроксимації, що може впливати на результати.
 - K-Means та Simple PSO залежать від ініціалізації, що може вплинути на кінцевий результат кластеризації.
 - Hybrid PSO також залежить від ініціалізації, але залежність може бути модифікована параметрами гібридизації.

- Здатність коректно працювати з викидами:
 - LP та K-Means можуть бути чутливими до викидів, залежно від обраного критерію апроксимації або метрики подібності.
 - Simple PSO та Hybrid PSO також можуть бути вразливими до викидів, що може впливати на здатність алгоритму досягти оптимальних результатів.

2.3 Ефективність моделей кластеризації.

Ефективний алгоритм кластеризації повинен відповідати наступним критеріям за замовчуванням:

1. Забезпечувати високу подібність об'єктів всередині кластеру та знижену подібність між кластерами, що сприяє чіткому виділенню груп об'єктів.
2. Якість результату кластеризації визначається вибором критерію подібності та конкретною реалізацією методу, а також його здатністю виявляти приховані особливості в даних.

Вимоги до алгоритму кластеризації включають [38]:

1. Масштабованість: алгоритм повинен бути ефективним для роботи з великими базами даних, забезпечуючи швидке та надійне кластеризування.
2. Здатність працювати з різними типами атрибутів: алгоритми повинні адаптуватися до різноманітних даних, включаючи числові, категоріальні та двійкові атрибути.
3. Виявлення кластерів довільної форми: алгоритм повинен бути здатний визначати групи об'єктів різної форми та розміру.
4. Висока розмірність: повинна бути можливість обробки не лише малорозмірних, але й високо розмірних даних.

5. Робота з шумними даними: алгоритм повинен бути стійким до виявлення та обробки шуму, відсутніх або помилкових даних.
6. Інтерпретованість результатів: результати кластеризації повинні бути легко інтерпретовані, зрозумілі та використовувані в контексті завдань аналізу даних.

Для валідації результатів кластеризації використовуються математичні метрики та методи оцінювання, які спрямовані на забезпечення надійності та обґрунтованості отриманих груп.

Індекс Ренда використовується для оцінки відповідності між кластерами, враховуючи інформацію, яка була відома спеціалістам, але не враховувалась при побудові кластерів. Цей індекс широко використовується для оцінки якості алгоритмів кластеризації.

Індекс Ренда обчислюється, як:

$$RI = \frac{a + b}{a + b + c + d} = \frac{a + b}{\frac{n(n-1)}{2}}$$

- a) кількість пар елементів у множині S, які одночасно належать одній підмножині в X і одній підмножині в Y;
- b) кількість пар елементів у множині S, які розташовані в різних підмножинах як в X, так і в Y;
- c) кількість пар елементів у множині S, які належать одній підмножині в X та різним підмножинам в Y;
- d) кількість пар елементів у множині S, які розташовані в різних підмножинах як в X, так і в Y.

Індекс Ренда приймає значення від 0 до 1. Значення RI, близьке до 1, вказує на ідентичне розбиття, тоді як значення, що наближається до 0, вказує на випадкове розбиття.

Взаємна інформація (АМІ) визначається з використанням функції ентропії, розглядаючи розбиття вибірки як дискретні розподіли, де ймовірність віднесення до кластеру визначається часткою об'єктів у ньому. Індекс МІ (взаємна інформація) обчислюється як міра взаємної інформації для двох розподілів, що відповідають розбиттям вибірки на кластери.

Для обчислення взаємної інформації (АМІ) потрібна таблиця спряженості, така ж, як для індексу корегованої випадкової угоди (ARI). Припустимо, що із набору $S = \{s_1, s_2, \dots, s_N\}$ випадковим чином вибирається один об'єкт. Ймовірність його належності кластеру X_i визначається наступним чином:

$$P(i) = \frac{|X_i|}{N}$$

$$p'(j) = \frac{|Y_j|}{N}$$

Значення ентропії, що відповідає розбиттям $\backslash(X\backslash)$ та $\backslash(Y\backslash)$, дорівнюватиме:

$$H(X) = - \sum_{i=1}^r P(i) \log(P(i))$$

$$H(Y) = - \sum_{j=1}^p P'(j) \log(P'(j))$$

Коефіцієнт взаємної інформації (КВІ) між двома розбиттями обчислюється за допомогою наступного виразу:

$$MI(X, Y) = \sum_{i=1}^r \sum_{j=1}^p P(i, j) \log \frac{P(i, j)}{P(i) \cdot P'(j)}$$

де:

$P(i)$ - ймовірність, що об'єкт належить кластеру X_i ;

$P(j)$ - ймовірність, що об'єкт належить кластеру Y_j ;

$P(i, j)$ - ймовірність, що об'єкт належить одночасно кластерам.

Таким чином, формула враховує ймовірності належності об'єктів до кожного кластеру окремо та одночасно до обох кластерів.

$$P(i) = \frac{|X_i \cap Y_i|}{N}$$

$$AMI = \frac{MI - E[MI]}{\max(H(X), H(Y)) - E[MI]}$$

$$0 \leq AMI \leq 1$$

Ефективність моделей кластеризації визначається їхньою здатністю коректно ідентифікувати та групувати схожі об'єкти в наборі даних. Основними показниками ефективності є здатність алгоритму виділяти чітко визначені та репрезентативні кластери, що відповідають структурі даних [40].

Успішна модель кластеризації повинна демонструвати високу подібність в межах кожного кластеру, а також низьку подібність між кластерами. Оцінювальні метрики, такі як індекс Ренда чи скоригована взаємна інформація, можуть використовуватися для кількісної оцінки ефективності моделі на основі фактичних та передбачених кластеризацій.

Крім того, ефективність моделей кластеризації оцінюється за їхню здатність працювати з різноманітними типами даних, великими обсягами і врахуванням шуму чи викидів в даних. Важливо, щоб модель була масштабованою та показувала стабільні результати при зміні параметрів чи умов.

Загалом, ефективна модель кластеризації повинна відповідати конкретним вимогам завдання, забезпечуючи точні та інтерпретовані результати при роботі з реальними даними.

Висновки

Моделі кластеризації є ключовим інструментом у сфері машинного навчання та аналізу даних. Однією з їхніх основних сутностей є групування схожих об'єктів в наборі даних, що дозволяє виявити приховані структури та

закономірності. Моделі, такі як LP, K-Means, Simple PSO та Hybrid PSO, використовують різні підходи та алгоритми для досягнення цієї мети.

Проведений порівняльний аналіз виявив, що кожна з вказаних моделей має свої переваги та обмеження. Метод LP, заснований на лінійному програмуванні, вирізняється ефективністю у вирішенні проблем із залученням обмежень. K-Means стабільно працює з числовими даними, але чутливий до початкового вибору центрів. Simple PSO та Hybrid PSO використовують природні інспіровані методи та можуть покращувати результати за рахунок оптимізації

Ефективність моделей кластеризації визначається їхньою здатністю вирішувати конкретні завдання. При оцінці ефективності слід враховувати такі аспекти, як точність, стійкість до варіацій параметрів та вміння працювати з різноманітними типами даних. Кожна модель має свої переваги та недоліки, і вибір конкретної залежить від конкретного завдання та властивостей даних. Враховуючи це, важливо обрати таку модель, яка найкраще відповідає вимогам конкретного дослідження чи завдання з кластеризації.

РОЗДІЛ 3

АЛГОРИТМ РОЗГОРТАННЯ ШЛЮЗІВ ІНТЕРНЕТУ РЕЧЕЙ ДЛЯ РОЗУМНОГО ДОМУ

3.1 Планування та розгортання шлюзів Інтернету речей для розумного дому

Інтернет речей (IoT) — це концепція, яка передбачає підключення різних пристроїв до мережі Інтернет для обміну даними і управління. Шлюз IoT відіграє ключову роль у встановленні зв'язку між пристроями IoT, такими як сенсори та пристрої, і локальним сервером. Він перетворює протоколи комунікації і фільтрує дані для забезпечення ефективного обміну інформацією.

Розумний дім становить конкретний випадок застосування IoT, який спрямований на автоматизацію та оптимізацію роботи побутових систем і пристроїв за допомогою сенсорної технології. Однак існує проблема, пов'язана з бездротовим з'єднанням, яка ставить обмеження на реалізацію сенсорних мереж. Наприклад, згасання сигналу обмежує відстань передачі інформації [50].

Враховуючи це, зв'язок між потужністю сигналу та відстанню між сенсорним вузлом і шлюзом описується законом зворотного квадрата відстані. Це свідчить про те, що поділ великих відстаней передачі інформації на кілька менших є більш енергоефективним підходом.

Значущість розташування шлюзів у мережі розумного дому визначається ефективністю зв'язку та зниженням втрат сигналу. Оптимальний вибір місць для розташування шлюзів може поліпшити покриття мережі, забезпечити стійкий зв'язок і зменшити споживання енергії. Однак кількість шлюзів у розумному домі також повинна бути оптимізована.

Тема розгортання шлюзів Інтернету речей для розумного дому є актуальною, враховуючи зростаючий інтерес до розумних систем та

підключених пристроїв. Інтернет речей стає все більш поширеним, а розумний дім визначається як одна з його ключових застосувань. Цей концепт дозволяє підключати різноманітні пристрої до Інтернету для обміну даними та інформацією. Шлюзи Інтернету речей, діючи як посередники між пристроями та додатками, забезпечують їх взаємодію та координацію.

Серед ключових аспектів розгортання шлюзів Інтернету речей для розумного дому важливе ефективне використання енергії, особливо оглядати пристрої, що працюють на батарейках чи обмежені енергопостачанням. Для цього використовуються різні алгоритми та підходи. Один з можливих алгоритмів, оптимізація маршрутів передачі даних та комунікації, враховує різні параметри, такі як розташування пристроїв і шлюзів, їхня енергетична потужність та обмеження на передачу даних [48].

Додатково, алгоритми розгортання враховують фактори, такі як відстань між пристроями та шлюзами, інтенсивність комунікації, потенційні перешкоди та затримки, а також вимоги до безпеки та приватності. Використання таких алгоритмів дозволяє досягти оптимального розташування шлюзів та ефективного використання ресурсів для забезпечення зручності, швидкості та енергоефективності в розумному домі. Одним з викликів при розгортанні є підтримка розширюваності та масштабованості системи, що є критично в умовах великої кількості підключених пристроїв. Узагальнюючи, розгортання шлюзів Інтернету речей для розумного дому визначається важливістю для ефективності та функціональності системи, а дослідження цієї області є важливим для розвитку розумних систем у майбутньому.

Планування та розгортання шлюзів Інтернету речей (IoT) для розумного дому з локальним сервером вимагає комплексного підходу. Вибір програмного забезпечення, бездротових технологій та пристроїв пропонується проводити за таким планом:

1. Аналіз вимог та обрання пристроїв IoT:

- Ідентифікація вимог розумного дому:
 - Визначення типів пристроїв (сенсори, виключувачі, камери, тощо).
 - Встановлення зв'язку з об'єктами автоматизації (освітлення, клімат-контроль, безпека).
- Вибір сумісних пристроїв IoT:
 - Врахування стандартів зв'язку (Zigbee, Z-Wave, Wi-Fi, Bluetooth).
 - Вибір пристроїв, які можуть працювати з локальним сервером.

2. Проектування мережі та забезпечення безпеки:

- Мережева інфраструктура:
 - Створення мережевого плану для пристроїв IoT.
 - Використання безпроводних технологій (Wi-Fi, Zigbee) залежно від потреб.
- Захист від кібератак:
 - Застосування шифрування для зберігання та передачі даних.
 - Встановлення механізмів аутентифікації та авторизації.

3. Вибір та налаштування локального серверу:

- Вибір серверного обладнання:
 - Врахування можливостей обробки даних та забезпечення плавної роботи.
- Встановлення серверного програмного забезпечення:
 - Вибір платформи (наприклад, OpenHAB, Home Assistant).
 - Налаштування правил автоматизації та інтеграції.

4. Інтеграція та тестування:

- Підключення пристроїв до локального сервера:
 - Налаштування пристроїв для співпраці з сервером.
- Розробка сценаріїв автоматизації:
 - Створення сценаріїв для взаємодії різних пристроїв.
- Тестування та налагодження:
 - Перевірка роботи сценаріїв в реальних умовах.

5. Забезпечення масштабованості та підтримки:

- Масштабованість системи:
 - Розгляд можливостей додавання нових пристроїв без значного перепроєктування.
- Підтримка та оновлення:
 - Забезпечення регулярних оновлень програмного та апаратного забезпечення.

6. Документація та навчання:

- Створення документації:
 - Опис налаштувань, сценаріїв та інструкцій для користувачів.
- Навчання користувачів:
 - Проведення навчання для мешканців розумного будинку.

Після розгортання системи важливо регулярно відслідковувати нові технології та вдосконалювати систему відповідно до змінних потреб та стандартів.

3.2 Математична постановка задачі мінімізації кількості шлюзів інтернету речей для розумного дому

В останні роки розумні дома, засновані на інтернеті речей (IoT), стали все більш популярними, пропонуючи користувачам широкий спектр автоматизованих функцій для поліпшення комфорту та енергоефективності. Однак важливим викликом є оптимізація інфраструктури IoT, зокрема, мінімізація кількості шлюзів, які обслуговують підключені пристрої. Це стрімке зростання кількості пристроїв у розумних домах робить актуальним питання ефективного використання ресурсів [44].

Мета мінімізації кількості шлюзів полягає в забезпеченні ефективного функціонування мережі IoT, униканні зайвого розходу енергії та оптимізації передачі даних. Мінімізація кількості шлюзів сприяє стабільності зв'язку, зменшує ризик перевантаження мережі та поліпшує загальну продуктивність системи.

Іншим важливим аспектом є економія ресурсів, таких як пропускна здатність та енергія. Мінімізація кількості шлюзів може виявитися ключовим фактором для оптимізації цих ресурсів та забезпечення тривалої та ефективної роботи розумного дому. Розробка ефективних алгоритмів та стратегій розгортання є важливим етапом в досягненні цієї мети.

Загалом, мінімізація кількості шлюзів IoT в розумних домах визначається як важлива задача для підвищення продуктивності та стійкості мережі. Цей підхід спрямований на створення ефективних, енергоефективних та економічно доцільних систем для користувачів розумних технологій.

3.3 Опис алгоритму розгортання шлюзів інтернету речей для розумного дому

Існує кілька технологій бездротового зв'язку, які можуть бути використані в розумному домі, серед яких обрано ZigBee, Wi-Fi та BLE. Задачею планування та оптимізації розгортання IoT-шлюзів для комунікаційних мереж було визначення мінімальної кількості шлюзів, необхідних для охоплення конкретної території, та розподілу пристроїв IoT

між цими шлюзами. Для кількісної оцінки розглядалося діапазон технологій бездротового зв'язку та пропускну здатність каналів зв'язку шлюзів. Загальною метою було зменшити кількість шлюзів при забезпеченні максимального охоплення пристроями в межах бажаної області [49].

Зона покриття розраховувалася на основі властивостей поширення сигналу комунікаційної технології. Кожна технологія мала власні характеристики поширення сигналу та зазначалася різними характеристиками в різних середовищах поширення. Моделі поширення сигналу використовувались для абстрагування від впливу дальності дії технології зв'язку залежно від втрат на тракті.

З урахуванням цілі зменшення кількості шлюзів, визначимо змінні для прийняття рішень:

- $X_{i,j}$ – матриця зв'язків між пристроями та шлюзом;
- Y_i – множина шлюзів, активованих для задоволення потреб пристроїв.

Таблиця 3.1 містить параметри, які використовувалися для оптимізаційної моделі, побудованої на лінійному програмуванні. Враховуючи координати пристроїв Інтернету речей ($D_{i,j}$), оптимізаційна модель передбачала встановлення зв'язків між кожним пристроєм та шлюзом (Y_i). Шлюз повинен бути в межах досяжності певної радіотехнології (r). Ці зв'язки були визначені за допомогою змінних прийняття рішень. Кожному пристрою призначався шлюз, проте з одним шлюзом можливе пов'язання кількох пристроїв. Модель використовувала обмеження для формування асоціацій, що відповідали необхідним критеріям для досягнення поставленої мети. Мета оптимізаційної моделі виражена рівнянням.

$$\min \sum_{i=1}^n Y_i \quad (3.1)$$

Множина Y_i представляє активні шлюзи. Рівняння (3.1) є завданням мінімізації кількості шлюзів (Y_i). Кожен $D_{i,j}$ є потенційним шлюзом.

Таблиця 3.1

Змінні в моделі оптимізації лінійного програмування

Змінна	Опис
r	Діапазон безпроводової технологій
n	Кількість розгорнутих пристроїв
c	Смуга пропускання, яка використовується пристроєм
cb	Загальна пропускна здатність каналу зв'язку
$A_{i,j}$	Всі можливі точки в межах бажаної області
$D_{i,j}$	Координати розгорнутих пристроїв
$Mdist_{i,j}$	Матриця відстані (вимірюється між точкою пристрою та точками інших розгорнутих пристроїв)
$Mativ_{i,j}$	Двійкова матриця (де 0 вказує на те, що відстань між пристроями не знаходиться в межах досяжності, а 1 вказує на те, що відстань між пристроями знаходиться в межах досяжності технології)

Для забезпечення покриття пристроїв Інтернету речей визначимо мінімальну кількість шлюзів, необхідну для цього [50]:

$$\sum_{j=1}^n X_{i,j} = 1, \quad i = 1, \dots, n \quad (3.2)$$

$$X_{i,j} \leq Mativ_{i,j} * Y_i, \quad i, j = 1, \dots, n \quad (3.3)$$

$$\sum_{j=1}^n X_{i,j} = 1 * c - cb \leq 0 \quad (3.4)$$

$$X_{i,j} \geq 0, \quad i, j = 1, \dots, n \quad (3.5)$$

$$Y_i \geq 0, \quad i, j = 1, \dots, n \quad (3.6)$$

Обмеження, викликані рівняннями (3.2) та (3.3), призначені стабілізувати зв'язок між пристроєм та єдиним шлюзом. Рівняння (3.4) регулює кількість пристроїв на кожному шлюзі, враховуючи пропускну здатність каналу зв'язку та не допускаючи перевищення максимальної

пропускної здатності. У випадку перевищення максимальної пропускної здатності, активується інший шлюз для відповіді на вимоги. Рівняння (3.5) та (3.6) встановлюють невід'ємні обмеження. Реалізація оптимізаційної моделі дозволяє визначити активні шлюзи (Y_i) та асоціації між пристроями та відповідними шлюзами ($X_{i,j}$). За цими результатами можна побудувати графік, використовуючи модель для розумного дому або підприємства, на якому відображено розташування шлюзів і відповідних пристроїв Інтернету речей, утворюючи таким чином кластер.

Планування та оптимізація розгортання шлюзу Інтернету речей для мереж зв'язку має на меті розрахувати мінімальну кількість шлюзів, необхідних для охоплення певної території, та кількість пристроїв IoT на кожному шлюзі. Для кількісної оцінки враховується діапазон використовуваної технології бездротового зв'язку і пропускна здатність каналу зв'язку шлюзу. Остаточною метою є мінімізація кількості шлюзів при максимальному охопленні пристроїв у межах бажаної зони.

Кожна пара координат для нанесених точок у певній області використовується як вхідний параметр для моделі оптимізації, що базується на лінійному програмуванні. Запропонована модель передбачає вибір точок встановлення пристроїв IoT випадковим чином, використовуючи початкові значення, але вони також можуть бути визначені відповідно до реальних їхніх позицій. При використанні реальних позицій враховувалися координати місць, в яких повинні бути розгорнуті пристрої.

```
from flask import Flask, request, jsonify
```

```
app = Flask(__name__)
```

```
# Зберігання даних про пристрої IoT
```

```
devices = {}
```

```
# Маршрут для реєстрації нового пристрою
```

```

@app.route('/register', methods=['POST'])
def register_device():
    data = request.get_json()
    device_id = data.get('device_id')
    device_type = data.get('device_type')

    if device_id and device_type:
        devices[device_id] = device_type
        return jsonify({'message': 'Device registered successfully'})
    else:
        return jsonify({'error': 'Invalid data'})

# Маршрут для отримання статусу пристрою
@app.route('/status/<device_id>', methods=['GET'])
def get_device_status(device_id):
    device_type = devices.get(device_id)
    if device_type:
        return jsonify({'device_id': device_id, 'device_type': device_type,
'status': 'online'})
    else:
        return jsonify({'error': 'Device not found'})

if __name__ == '__main__':
    app.run(debug=True)

```

Висновки

Для побудови системи локального дому була обрана модель кластеризації LP. З використанням дистрибутиву Ubuntu операційної системи Linux та програмного забезпечення OpenHAB для управління розумним домом і

Zabbix для моніторингу мережі та потужностей була розроблена система розумного дому.

Мережа побудована з використанням мультипротокольних шлюзів з підтримкою ZigBee, Wi-Fi та BLE. Особливість системи розумного дому полягає в її вищій захищеності порівняно з тими, що базуються на хмарному середовищі, і у можливості працювати при відсутності доступу до мережі Інтернет.

РОЗДІЛ 4

РОЗРОБКА СТАРТАП ПРОЕКТУ

4.1 Опис ідеї стартап-проекту

Таблиця 4.1

Опис ідеї стартапу

Зміст ідеї стартап-проекту	Напрямки застосування	Вигоди для користувача
Впровадження екосистеми розумного дому з локальним сервером	Житлові будинки	Зручне керування всіма аспектами розумного дому
		Енергоефективне управління для зниження витрат
		Підвищена безпека та спокій через системи охорони
		Зручність і комфорт в повсякденному житті
	Офісні приміщення	Підвищена безпека та контроль над приміщеннями
		Забезпечення комфорту та задоволення потреб працівників
	Комерційні об'єкти	Підвищена ефективність управління та економія ресурсів
		Забезпечення безпеки гостей та їх комфорту
		Враження і задоволення гостей через розумне керування

Аналіз потенційних техніко-економічних переваг ідеї стартап-проекту у порівнянні із пропозиціями конкурентів передбачає:

а) визначення техніко-економічних властивостей та характеристик ідеї стартап-проекту;

б) визначення попереднього кола конкурентів, що існують на ринку, та проведення збору інформації щодо техніко-економічних показників для власного стартап-проекту та проектів-конкурентів;

в) проведено порівняльний аналіз показників: для власної ідеї визначені показники.

Таблиця 4.2

Визначення сильних та слабких характеристик

Техніко-економічні характеристики	Мій проект	Fibaro	Koogeek	W	N	S
Використання програмного забезпечення відкритим початковим кодом	+	-	+		+	
Використання локального серверу розумного дому	+	+	-			+
Індивідуальні можливості кастомізації системи	+	-	-			+
Єдина підтримка апаратної та програмної частини	+	+	+		+	
Підтримка декількох безпроводових технологій	+	-	-			+
Складність оновлень програмного забезпечення серверу	+	-	+	+		
Використання додаткових хабів для роботи з бездротовими технологіями	+	-	-	+		

В розділі W (вразлива сторона): особливості, де проект показує менші показники у порівнянні з конкурентами; в розділі N (нейтральна сторона): особливості, де мій проект та конкуренти мають схожі показники; в розділі S (сильна сторона): особливості, де проект демонструє кращі показники, ніж його конкуренти.

Конкуренти володіють лише частковим функціоналом, який реалізований в даному проекті. Серед сильних сторін визначені підтримка декількох бездротових технологій (ZigBee, Z-Wave та інші), які ставляться

перед обраним проектом. Слабкими сторонами програмного забезпечення серверу розумного дому та використання додаткових хабів для роботи з бездротовими технологіями.

4.2 Технологічний аудит ідеї проекту

Таблиця 4.3.

Технологічний аудит проекту

Ідея проекту	Технології реалізації	Наявність технологій	Доступність технологій
Впровадження екосистеми розумного дому з локальним сервером	Локальний сервер розумного дому на основі програмного забезпечення з відкритим початковим кодом OpenHub	Наявна	Доступна
Висновок: реалізувати проект можливо			

В рамках реалізації ідеї проекту, що передбачає впровадження екосистеми розумного дому з локальним сервером, обрано технологічний підхід, що базується на застосуванні локального сервера розумного дому, побудованого на основі програмного забезпечення з відкритим початковим кодом OpenHAB. Дана технологічна концепція передбачає використання існуючих розробок у сфері розумних домів і надає відкритий доступ до свого програмного забезпечення.

Зазначена технологічна інфраструктура вже існує і є доступною для використання. Локальний сервер, заснований на OpenHub, є наявним і готовим до впровадження. Таким чином, можна зробити обґрунтований висновок, що технологічна здійсненність ідеї проекту є можливою, оскільки існують наявні та доступні технології для успішної реалізації екосистеми розумного дому з обраною концепцією локального сервера.

4.3 Аналіз ринкових можливостей запуску стартап-проекту

При дослідженні ринкових можливостей, в першу чергу проведений аналіз попиту: наявність попиту, обсяг, динаміка розвитку ринку. Дані наведені у таблиці нижче (див. табл. 4.4).

Таблиця 4.4

Попередня характеристика потенційного ринку стартап-проекту

Показники стану ринку (найменування)	Характеристика
Кількість головних гравців, од	30
Загальний обсяг продаж	Від користувачів
Динаміка ринку (якісна оцінка)	Зростає
Наявність обмежень для входу	Немає
Специфічні вимоги для стандартизації, специфікації	Немає
Середня норма рентабельності в галузі, %	29%

Середня норма рентабельності в галузі (або по ринку) порівнюється із банківським відсотком на вкладення. За умови, що останній є вищим, можливо, має сенс вкласти кошти в інший проект.

За результатами аналізу можемо зробити висновок, що ринок є привабливим для входження.

Далі визначимо потенційні групи клієнтів та їх характеристики, сформуємо орієнтовний перелік вимог до товару (див. табл. 4.5).

Таблиця 4.5.

Характеристика потенційних клієнтів стартап проекту

Потреба, що формує ринок	Цільова аудиторія (цільовий сегмент ринку)	Відмінності у поведінці різних потенційних цільових груп клієнтів	Вимоги споживачів до системи
Зручність і автоматизація	Власники приватних будинків і квартир	Молоді сім'ї, технологічно орієнтовані користувачі, заможні	Простота використання, мобільний доступ, наявність розумних пристроїв і інтеграція

		клієнти	з ними.
Безпека	Усі категорії власників будинків та квартир	Сім'ї з дітьми, елітні житлові комплекси	Відеоспостереження, датчики виявлення вторгнень, система сигналізації, контроль доступу.
Енергоефективність	Екологічно свідомі	Заможні клієнти,	Автоматичне регулювання

Продовження таблиці 4.5.

Потреба, що формує ринок	Цільова аудиторія (цільовий сегмент ринку)	Відмінності у поведінці різних потенційних цільових груп клієнтів	Вимоги споживачів до системи
	користувачі, енергоефективні будинки	екологічні організації	опалення та освітлення, енергетичний моніторинг, оптимізація споживання енергії.
Комфорт і атмосфера	Всі категорії власників будинків та квартир	Дошкільнята, старші люди, люди з обмеженими можливостями	Регулювання температури, освітлення, звукових систем, створення розслаблюючої атмосфери.
Ефективне управління простором	Компанії різного розміру та галузі діяльності	Корпорації, стартапи, ІТ-компанії, фінансові установи	Інтеграція з розумними пристроями, гнучкість і масштабованість, системи контролю доступу та реєстрації робочого часу.
Енергоефективність та витрати	Офісні приміщення, корпоративні комплекси	Компанії з великим обсягом енергоспоживання, компанії зеленого бізнесу	Автоматичне регулювання освітлення, опалення та кондиціонування повітря, моніторинг та оптимізація споживання енергії.
Безпека та контроль доступу	Офісні будівлі, корпоративні комплекси	Компанії з конфіденційною інформацією, охоронні компанії	Відеоспостереження, системи контролю доступу, автоматичне виявлення вторгнень та пожеж.

Зручність та комфорт працівників	Усі категорії офісів	Молоді співробітники, фрілансери, дистанційні робочі групи	Гнучке управління освітленням, системи розумного офісного обладнання.
----------------------------------	----------------------	--	---

Після визначення потенційних груп клієнтів проведемо аналіз ринкового середовища: складемо таблиці факторів, що будуть сприяти ринковому впровадженню проекту чи перешкоджати (див. табл. 4.6).

Таблиця 4.6.

Фактори загроз

Фактор	Зміст загрози	Можлива реакція компанії
Конкуренція	Поява нових конкурентів у сфері розумного дому	Постійне вдосконалення продукту, розробка унікальних функцій та особливостей, збільшення якості обслуговування, активна маркетингова кампанія, залучення нових клієнтів.
Кібербезпека	Ризик кібератак на систему розумного дому та витіку конфіденційних даних	Розробка та впровадження ефективних заходів кібербезпеки, шифрування даних, перевірка наразісних каналів комунікації, постійне моніторинг і оновлення захисних механізмів.
Сумісність існуючими технологіями	Проблеми інтеграції з іншими розумними пристроями і системами в домі	Посилення досліджень та розробка сумісних інтерфейсів, співпраця з виробниками інших пристроїв, підтримка стандартів комунікації.
Зміни в регуляторному середовищі	Зміни законодавства, правила збереження та обробки даних, вимоги щодо безпеки	Аналіз та впровадження нових нормативних вимог, співпраця зі стейкхолдерами, забезпечення відповідності регулятивам.
Застарілість технологій	Ризик втрати конкурентоспроможності через застарілість технологій	Постійне оновлення технологічного стеку, дослідження та впровадження новітніх

		розробок, залучення технічних експертів, моніторинг технологічних тенденцій.
--	--	---

Таблиця 4.7

Таблиця факторів можливостей

№ п/п	Фактор	Зміст можливості	Можлива реакція компанії
1	Зростання популярності розумних домівок	Збільшення попиту на продукти та послуги розумного дому	Розширення продуктової лінійки, вдосконалення функціоналу, розширення географічного охоплення ринку та маркетингові кампанії
2	Технологічний прогрес	Постійне вдосконалення технологій у сфері розумного дому	Активне дослідження та впровадження новітніх технологій, створення інноваційних рішень та продуктів, партнерство з технологічними компаніями, залучення талановитих розробників.
3	Зріст ринку Інтернету речей (IoT)	Збільшення кількості підключених пристроїв та їх взаємодія	Розширення інтеграційних можливостей, підтримка протоколів IoT, партнерство з виробниками IoT-пристроїв, розробка власного екосистемного підходу.
4	Розвиток штучного інтелекту (AI)	Застосування AI в розумних домівках для автоматизації та оптимізації процесів	Впровадження AI-технологій у систему, розробка інтелектуальних алгоритмів, партнерство з компаніями, що спеціалізуються на штучному інтелекті.

Надалі проводиться аналіз пропозиції: визначаються загальні риси конкуренції на ринку. Аналіз пропозиції необхідно виконати аналізуючи існуючі види конкуренції. Пропозиції повинні відповідати на питання «Як просувати продукт». Аналіз пропозицій зображено на таблиці (див. табл. 4.8).

Таблиця 4.8

Ступеневий аналіз конкуренції на ринку

Особливості конкурентного середовища	В чому проявляється дана характеристика	Вплив на діяльність підприємства (можливі дії компанії, щоб бути конкурентоспроможною)
1. Вказати тип конкуренції монополія/олігополія/чиста /монополістична	чиста	Створення унікальності продукту, розвиток власних технологій, залучення інвестицій для розширення та конкуренції з монополістом укладання прямих договорів з компаніями.
2.3 а рівнем конкурентної боротьби локальний/національний	національний	Ефективна маркетингова стратегія, залучення талановитих фахівців, постійне вдосконалення продукту.
3. За галузевою ознакою- міжгалузева/ внутрішньогалузева	внутрішньогалузева	Акцент на унікальність та інноваційність продукту, партнерські відносини з іншими гравцями на ринку, розробка комплексних рішень.
4. Конкуренція за видами товарів: товарно-видова товарно-родова між бажаннями	товарно-видова	Покращувати наявний функціонал
5. За характером конкурентних переваг - цінова та нецінова	цінова	Конкуренція на основі ціни, а також інших факторів, таких як якість, сервіс, бренд та ін.
6. За інтенсивністю - марочна/немарочна	немарочна	Змагання між компаніями, які пропонують схожі продукти, але з різними брендами та позиціонуванням

Таблиця 4.9

Аналіз конкуренції в галузі за М. Портером

	Потенційні конкуренти	Постачальники	Клієнти	Товари замітники
Складові аналізу	Fibarо, Koogeek	Мінімізація витрат часу постачальників	Контроль якості	Розумний дiм з хмарним середовищем
Висновки	Можливість виходу на ринок, тому що існуючі рішення не надають аналогічних переваг	Постачальники підлаштовують ся під ринок	Клієнти диктують вимоги опираючись на умови експлуатації	Обмеження для роботи через товари замітники

Таблиця 4.10

Обґрунтування факторів конкурентоспроможності

Фактор конкурентоспроможності	Обґрунтування
Обслуговування після продажу та встановлення	Підтримка щодо використання системи після її продажу
Якість продукту/послуги	Висока якість продукту або послуги дозволяє забезпечити задоволення потреб клієнтів та отримати конкурентну перевагу.
Ціна	Конкурентоспроможна ціна дозволяє привернути клієнтів і зайняти певну ринкову позицію.
Інновації	Постійні інновації у продукті або процесах дозволяють привернути увагу клієнтів та забезпечити конкурентну перевагу.
Маркетингова стратегія	Ефективна маркетингова стратегія дозволяє залучити і утримати клієнтів, підвищити в пізнаваність бренду та стати привабливим на ринку.
Доступ до ресурсів	Маючи доступ до необхідних ресурсів (фінансових, технологічних, людських), компанія може розвиватися та стати конкурентоспроможною.
Клієнтське обслуговування	Висока якість обслуговування та підтримки клієнтів забезпечує задоволення та лояльність клієнтів.

Таблиця 4.11

Порівняльний аналіз слабких та сильних сторін

Фактор конкурентоспроможності	Бали 1-20	Рейтинг товарів-конкурентів у порівнянні з Predicty						
		-3	-2	-1	0	+1	+2	+3
Обслуговування після продажу та встановлення	18		+					
Якість продукту/послуги	16						+	
Ціна	15			+				
Інновації	18				+			
Маркетингова стратегія	17				+			
Доступ до ресурсів	15						+	
Клієнтське обслуговування	18		+					

Фінальним кроком ринкового аналізу можливостей впровадження проекту є SWOT-аналіз (матриця аналізу сильних(Strength), слабких (Weakness) сторін, загроз (Threds) та можливостей (Opportunities) (див. табл. 4.11) на основі раніше виділених ринкових загроз, можливостей, сильних та слабких сторін.

Перелік ринкових загроз та можливостей складаємо на основі аналізу факторів загроз та можливостей маркетингового середовища.

Ринкові загрози, ринкові можливості є прогнозованими результатами впливу факторів, які ще не реалізовані на ринку та мають деяку ймовірність здійснення.

Таблиця 4.12

SWOT-аналіз стартап-проекту

Сильні сторони	Слабкі сторони
Інноваційний підхід до розумного дому, що пропонує унікальні функції і можливості. Локальний сервер, що забезпечує високий рівень захисту даних і конфіденційності. Широкий спектр підключених пристроїв і сумісність з різними стандартами. Інтегрована система керування, що дозволяє користувачам зручно керувати всіма аспектами розумного дому. Стратегічне партнерство з провайдером послуг зв'язку та енергетики для розширення покриття і функціоналу	Висока вартість впровадження системи розумного дому може бути бар'єром для деяких користувачів. Потреба в технічній експертизі для встановлення та налагодження системи. Конкуренція зі сторони вже існуючих компаній на ринку розумного дому.
Можливості	Загрози
Ріст популярності розумного дому і підвищений інтерес користувачів до автоматизації та зручності управління своїм житлом. Збільшення попиту на енергоефективні рішення та зниження споживання енергії. Партнерство зі виробниками підключених пристроїв для розширення функціоналу та інтеграції з новими технологіями.	Зростання конкуренції на ринку розумних домівок і поява нових гравців. Загроза кібератак і порушення безпеки даних у зв'язку з підключеними пристроями. Зміни у законодавстві щодо приватності даних та регулювання ринку розумного дому.

Перелік ринкових загроз та ринкових можливостей складаємо на основі аналізу факторів загроз та можливостей маркетингового середовища.

4.4 Розроблення ринкової стратегії проекту

Таблиця 4.13

Альтернативи ринкового впровадження стартап-проекту

Опис профілю цільової групи потенційних клієнтів	Готовність споживачів сприйняти продукт	Орієнтовний попит в межах цільової групи	Інтенсивність конкуренції в сегменті	Простота входу у сегмент
Малий бізнес	Середня	Високий	Високий	Середній
Інноваційні підприємства	Висока	Високий	Низька	Легко
Офісні приміщення	Висока	Високий	Високий	Середній
Молоді сім'ї, які шукають інноваційні рішення для свого житла	Висока	Високий	Високий	Середній
Середньовікові технічно-освічені користувачі	Висока	Середній	Середня	Середня

Таблиця 4.14

Визначення базової стратегії розвитку

Обрана альтернатива розвитку проекту	Стратегія охоплення ринку	Ключові конкурентоспроможні позиції відповідно до обраної альтернативи	Базова стратегія розвитку
Орієнтація поточної моделі на ринок спостереження та охорони природи	Стратегія концентрованого маркетингу	Корпорації, які займаються розробкою систем моніторингу навколишнього середовища	Стратегія спеціалізації (спирається на диференціацію)
Орієнтація поточної моделі на ринок спостереження та охорони людей	Стратегія концентрованого маркетингу	Корпорації, які займаються попередженням катаклізмів, розробкою систем спостереження	Стратегія спеціалізації (спирається на диференціацію)

Залежно від міри сформованості галузевого ринку, характеру конкурентної боротьби, необхідно обрати одну з трьох стратегій конкурентної поведінки: розширення первинного попиту, оборонну або наступальну стратегію або ж застосувати демаркетинг або диверсифікацію (див. табл. 4.15).

Таблиця 4.15

Визначення базової стратегії конкурентної поведінки

Чи є проект «першопрохідцем» на ринку?	Чи буде компанія шукати нових споживачів, або забирати існуючих у конкурентів?	Чи буде компанія копіювати основні характеристики товару конкурента, і які?	Стратегія конкурентної поведінки
Ні	Шукає нових споживачів	Так. Стартап копіюватиме можливість управління розумним домом за допомогою додатку на смартфонах або ПК у локальній мережі з можливістю управління ззовні.	Стратегія заняття конкурентної ніші

З обраних сегментів до стартап-компанії та до продукту розробляється стратегія позиціонування (див. табл. 4.16), що полягає у формуванні ринкової позиції, за яким споживачі мають ідентифікувати проект.

Вибір стратегії зайняття конкурентної ніші виявляється оптимальним для стартапу з ряду обґрунтованих причин. Насамперед, проект не є першопрохідцем на ринку, що дозволяє йому взяти на увагу інновації та навчання з досвіду інших компаній.

Ключовим аспектом стратегії є акцент на залученні нових клієнтів, замість ведення конкурентної боротьби за існуючих споживачів інших компаній. Це надає можливість створювати унікальні характеристики продукту, що відрізняються від пропозицій конкурентів, і привертати увагу нового сегменту ринку.

Такий підхід дозволяє стартапу ефективно використовувати власні ресурси, максимізувати конкурентні переваги та створювати унікальне пропозиційне сприйняття для потенційних клієнтів. Зосередження на інноваціях та приваблювання нових клієнтів створює фундамент для успішного входження на ринок і подальшого росту.

Таблиця 4.16

Визначення стратегії позиціонування

Вимоги до товару цільової аудиторії	Базова стратегія розвитку	Ключові конкурентоспроможні позиції власного стартап проекту	Вибір асоціацій, які мають сформувану комплексну позицію власного проекту (три ключових)
Зручне використання та інтуїтивний інтерфейс	Розширення географії присутності	Зручне використання та інтуїтивний інтерфейс	Користувальницький дружній дизайн, простота використання
Персоналізація та індивідуальний підхід	Розвиток партнерської мережі	Персоналізація та індивідуальний підхід	Стратегія підтримки клієнтів та індивідуального обслуговування

4.5 Розроблення маркетингової програми стартап-проекту

1. Сегменти споживачів:

- 1) Власники квартир
- 2) Власники приватних будинків
- 3) Будівельні компанії що працюють «Під ключ» в елітному сегменті нерухомості
- 4) Організації та компанії

2. Переваги

- 1) Зручність та комфорт
- 2) Економія часу
- 3) Безпека

- 4) Індивідуальний підхід
- 5) Технологічність та престиж

3. Канали

- 1) Реклама
- 2) PR
- 3) Бізнес спільноти
- 4) Соціальні медіа
- 5) Директ-маркетинг
- 6) Рекомендації.

4. Взаємовідносини з клієнтами

- 1) Особисті зустрічі;
- 2))Консультації по телефону та відеозв'язок;
- 3) Месенджери.

5. Потоки доходів

- 1) Планування розміщення апаратного комплексу та дизайн інтер'єру;
- 2) Встановлення апаратного комплексу, його підключення та налаштування;
- 3) Платна підписка на обслуговування та технічну підтримку;

6. Ключові ресурси

- 1) Технічні знання та компетенції фахівців
- 2) Інформаційні ресурси - канали
- 3) Людські ресурси - технічні спеціалісти, маркетологи, піарщики, рекламні агенти, фахівці технічної підтримки, фахівці клієнтської підтримки,

аналітики, дизайнери, проєктувальники, фінансові фахівці, управлінський персонал.

4) Фінансові ресурси - інвестиції в дослідження та розробку, маркетинг та рекламу, фонд заробітної плати, інвестиції в апаратну частину продукту.

5) Інфраструктура - апаратна частина, програмна частина, технології.

7. Ключові дії

- 1) Розробка та виготовлення продукту та послуг
- 2) Маркетингові дії для просування продукту та послуг на ринку
- 3) Надання послуг та продукту
- 4) Підтримка клієнтів після продажу продукту
- 5) Аналіз результатів продажу і відгуків клієнтів для вдосконалення продукту або послуги
 - Прямі продажі
 - Продаж через дистриб'юторів
 - Рекламні канали
 - Партнерські програми

8. Ключові партнери

- 1) Банк
- 2) Постачальники, апаратного забезпечення та комплектуючих
- 3) Будівельні компанії
- 4) Ремонтні бригади що виконують ремонтні роботи «під ключ»
- 5) Агентства з продажу нерухомості
- 6) Агентства дизайну інтер'єру

Висновки

Стартап-проект екосистема розумного дому має потенціал стати конкурентоспроможним на ринку. SWOT-аналіз допоміг визначити сильні та слабкі сторони нашого проекту. Виявлені значні можливості для розвитку, такі як зростаючий попит на розумні системи, зручність використання та інноваційні можливості. Однак, також існують загрози, пов'язані з конкуренцією та технологічними викликами.

Враховані особливості конкурентного середовища та проведений аналіз пропозицій на ринку. Це дозволило визначити ключові конкурентоспроможні позиції стартапу, такі як інтуїтивний інтерфейс, ефективне використання ресурсів, інноваційні рішення, надійність та швидка підтримка.

Для успішного впровадження стартапу, потрібно звернути увагу на вимоги цільової аудиторії та створити стратегію розвитку, спрямовану на задоволення їх потреб. Важливо будувати позицію проекту на асоціаціях, таких як зручність, економія, інновації та надійність.

Аналіз альтернатив ринкового впровадження допоможе визначити оптимальні шляхи проникнення на ринок, враховуючи готовність споживачів та інтенсивність конкуренції у різних сегментах.

Загалом, стартап-проект має потенціал стати успішним гравцем у сфері розумного дому, за умови правильної стратегії розвитку, надійного технічного рішення та ефективного маркетингового просування.

ЗАГАЛЬНІ ВИСНОВКИ ПО РОБОТІ

Впровадження розумного дому включає в себе використання як локальних серверів, розташованих безпосередньо в житловому просторі, так і хмарних середовищ для зберігання та обробки даних. Локальні сервери дозволяють забезпечити низьку затримку та більшу приватність, тоді як хмарні середовища надають доступ до інформації з будь-якого місця та спрощують управління системою. Використання обох підходів може бути ефективним для створення гнучких та доступних систем розумного дому. При побудові складових мереж, незалежно від того, чи це локальний сервер чи хмарне середовище, ключовою проблемою є забезпечення надійності, безпеки та ефективності комунікації. Принципи розподіленої архітектури та використання стандартів забезпечують масштабованість та інтероперабельність, що важливо для успішного функціонування систем розумного дому.

OpenHAB та Zabbix є популярними інструментами для керування та моніторингу розумних систем відповідно. OpenHAB надає різноманітні можливості інтеграції для пристроїв різних виробників, дозволяючи створювати єдине середовище керування. Zabbix, з іншого боку, забезпечує надійний моніторинг стану системи та можливість автоматичного реагування на події. Обидва інструменти взаємодіють, створюючи повноцінну систему керування та моніторингу для розумного дому. Порівняльний аналіз алгоритмів кластеризації, таких як K-Means, LP, Simple PSO та Hybrid PSO, вказує на їхню різнобічність та відмінності в ефективності. Кожен з них має свої переваги та обмеження, що може визначати їхню застосовність в конкретних сценаріях розумного дому. Наприклад, LP може бути корисний для великих обсягів даних, тоді як Hybrid PSO може надати високу точність в складних умовах. Обираючи алгоритм, важливо враховувати конкретні вимоги та характеристики системи розумного дому для досягнення оптимальних результатів.

Моделі кластеризації є важливим інструментом аналізу даних, який допомагає виявляти приховані структури та групи в наборі даних. Кожна модель має свої унікальні особливості та принципи роботи. Наприклад, алгоритм k-середніх розділяє дані на кластери, максимізуючи подібність об'єктів у межах кластера та мінімізуючи подібність між кластерами. У порівнянні з ієрархічними методами, які створюють дендрограму вкладених розбиттів, вони різняться своєю прямолінійністю та лаконічністю. Порівняльний аналіз моделей кластеризації дозволяє краще розуміти їхні переваги та обмеження. Метод кластеризації LP використовує лінійне програмування для знаходження оптимальних розбиттів, що робить його ефективним для обробки великих обсягів даних. K-Means використовує ітеративний підхід для мінімізації середньоквадратичної відстані між точками та центрами кластерів. Simple PSO та Hybrid PSO базуються на методі оптимізації частинць, проте Hybrid PSO комбінує різні підходи для підвищення точності та швидкості. Ефективність моделей кластеризації залежить від їхньої здатності адаптуватися до різноманітних умов та вимог завдань. У порівнянні з лінійним програмуванням, алгоритми, такі як K-Means, дозволяють працювати з різними типами даних та формами кластерів. Simple PSO та Hybrid PSO відрізняються у рівні складності та швидкості збіжності, що може визначати їхню придатність для конкретних завдань. Загальною вимогою є здатність до виявлення структур у даних та адаптація до різноманітних умов, що робить аналіз та вибір моделі важливим завданням для успішної кластеризації.

У сучасних розумних домах ключову роль відіграє правильне планування та розгортання шлюзів Інтернету речей (IoT). Враховуючи різноманітні пристрої та датчики, які використовують технології IoT, ефективне розташування шлюзів є вирішальним для забезпечення стабільного зв'язку та оптимізації функціоналу розумного дому. Мінімізація кількості шлюзів IoT в розумному домі має стратегічне значення для оптимізації витрат та підвищення ефективності мережі. Застосування

відповідних алгоритмів та моделей, таких як класичні методи кластеризації або алгоритми оптимізації рою частинок (PSO), дозволяє знайти оптимальне розташування шлюзів для забезпечення ефективного покриття та зниження споживання енергії. Алгоритм, що використовується для мінімізації кількості шлюзів IoT, базується на принципах оптимізації рою частинок (PSO). Цей метод дозволяє ефективно розташовувати шлюзи з урахуванням факторів, таких як відстань передачі даних, покриття мережі та енергоефективність. Використання PSO сприяє знаходженню глобальної оптимізації та поліпшує результати порівняно з іншими методами.

Стартап, що пропонує екосистему розумного дому, може відзначитися конкурентоздатністю на ринку. Здійснений SWOT-аналіз дозволив виявити як сильні, так і слабкі сторони нашого проекту. Виявлено значні можливості для росту, такі як збільшення попиту на розумні системи, простота використання та інноваційні можливості. Однак існують також загрози, пов'язані з конкурентним середовищем та технологічними викликами. Враховані особливості конкурентного поля та проведено аналіз пропозицій на ринку, що дозволило визначити ключові конкурентні переваги стартапу, такі як інтуїтивний інтерфейс, ефективне використання ресурсів, інноваційні рішення, надійність та оперативна підтримка.

Для успішного впровадження проекту необхідно зосередитися на вимогах цільової аудиторії та розробити стратегію розвитку, спрямовану на задоволення їхніх потреб. Ключовим є будівництво позиції проекту навколо таких асоціацій, як зручність, економічність, інноваційність та надійність. Аналіз альтернативних шляхів введення на ринок допоможе визначити оптимальні стратегії проникнення, враховуючи готовність споживачів та інтенсивність конкуренції у різних сегментах. Загалом, стартап має потенціал стати відомим гравцем у сфері розумного дому при належній стратегії розвитку, надійному технічному рішенні та ефективному маркетинговому підході.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. LeCun Y., Bengio Y., Hinton G. (2015) Deep learning. Nature 521:436–444
2. Hinton G.E., Srivastava N., Krizhevsky A., Sutskever I., Salakhutdinov R.R. (2012) Improving neural networks by preventing co-adaptation of feature detectors. arXiv. URL: <https://arxiv.org/pdf/1207.0580.pdf>
3. Zhong Z., Zheng L., Kang G., Li S., Yang Y. (2017) Random erasing data augmentation. arXiv. URL: <https://arxiv.org/pdf/1708.04896.pdf>
4. Кластеризація методом к-середніх. Вікіпедія. URL: https://uk.wikipedia.org/wiki/Кластеризація_методом_к-середніх.
5. Особливості Zabbix. URL: <http://www.zabbix.com/features.php>.
6. Проектування пакетів. URL: <http://www.networkworld.com/article/2338253/infrastructure-management>.
7. Zabbix True Open Source. URL: http://www.zabbix.com/true_open_source.php.
8. Автоматичне виявлення на рівні управління мережею та послугами. <http://ieeexplore.ieee.org/xpl/articleDetails.jsp?reload=true&arnumber=1194192&abstractAccess=no&userType=inst>
9. Zabbix автоматичне виявлення. URL: http://www.zabbix.com/auto_discovery.php
- 10.10. Додавання точок мережі Zabbix 5.0. URL: <https://it-school.pw/dobavlenie-uzlov-seti-v-zabbix-5-0-lts>
11. Машинне навчання: комп'ютерний практикум з дисципліни «Машинне навчання» [Електронний ресурс]: навч. посіб. для студ. спеціальності 121 «Інженерія програмного забезпечення» (освітня програма «Інженерія програмного забезпечення мультимедійних та інформаційно-пошукових систем»)/ Л.М. Олещенко; КПІ ім. Ігоря Сікорського. – Електронні текстові дані. – Київ: КПІ ім. Ігоря Сікорського, 2022. – 92 с

12. Технології інтернету речей в електроніці: Комп'ютерний практикум [Електронний ресурс]: навч. посіб. для студ. спеціальності 171 «Електроніка», спеціалізації «Електронні компоненти і системи» / КПІ ім. Ігоря Сікорського; уклад.: Ю. С. Ямненко, Ю. В. Хохлов. – Електронні текстові данні (1 файл: X,XX Мбайт). – Київ : КПІ ім. Ігоря Сікорського, 2018. – 76 с.
13. OpenHAB «A vendor and technology agnostic open source automation software for your home». URL: <http://www.homeautomationforgeeks.com/project/openhab.shtml>
14. OpenHab. URL: <https://www.openhab.org/>
15. Менахем Домб. Системи розумного дому, заснованого на основі інтернет речей. URL: <https://www.intechopen.com/books/internet-of-things-iot-for-automated-and-smart-applications/smart-home-systems-based-on-internet-of-things>
16. Паньків В. Г. Український ринок систем автоматизації та диспетчеризації. Мережі та бізнес системи, 2011. №3. С. 58–62.
17. Баранов В. Н. Застосування Мікроконтролерів AVR: схеми, алгоритми, програми. Додека-XXI, 2006р. 231 с.
18. Огляд бездротової системи безпеки Ajax: центральний хаб та універсальні датчики. URL: <https://www.ixbt.com/home/ajax-wireless-security-review.html>
19. Огляд готових рішень систем «Розумний дім». URL: <https://sprut.ai/client/article/1544>
20. Бездротові мережі URL: http://old.ci.ru/inform11_04/p_24.htm
21. Горбенко І. Д., Замула О. А. Інформаційні технології. Оцінка показників захищеності сучасних бездротових систем зв'язку широкопasmового доступу на основі врахування особливостей технологій OFDM. ХНУ. 2012. 67-75 с
22. Безпроводна передача сигналів. Безпроводні локальні мережі. URL: <https://infotel.ua/bezprovidni-wifi-merezhi>

- 23.Science Direct. Спектр поширення прямої послідовності. URL:
[sciencedirect.com/topics/engineering/direct-sequence-spread-spectrum](https://www.sciencedirect.com/topics/engineering/direct-sequence-spread-spectrum)
- 24.Підручник з технології розповсюдження спектру. EE Times. URL:
<https://www.eetimes.com/tutorial-on-spread-spectrum-technology/>
- 25.Канали ZigBee та Wi-Fi. Співіснування ZigBee та WiFi Networks. URL:
<https://support.metageek.com/hc/en-us/articles/203845040-ZigBee-and-WiFiCoexistence>
- 26.Міні SMD радіомодуль. URL: <https://diylab.com.ua/p81695762-minismdradiomodul.html>
- 27.M.Sangeetha, C.Udhayanila, G.Gayathri, N.Rakshana. Smart Home Control System by Internet of Things Based on WIFI Module. International Journal of Innovative Research in Computer and Communication Engineering. Vol. 5, Issue 3, March 2017.
- 28.Domb, Menachem. «Smart Home Systems Based on Internet of Things» IoT and Smart Home Automation. IntechOpen. URL:
<https://www.intechopen.com/books/internetof-things-iot-for-automated-and-smart-applications/smart-home-systems-basedon-internetof-things>
- 29.E.M. Mirkes (2011) «K-means and K-medoids applet»
- 30.Erich Schubert, Jörg Sander, Martin Ester, Hans Peter Kriegel, Xiaowei Xu. (2017) «DBSCAN: Why and How You Should (Still) Use DBSCAN» // ACM Trans. Database Syst.
- 31.Hastie Trevor, Tibshirani Robert, Friedman Jerome (2001) «The EM algorithm» — New York: Springer. — С. 236—24
- 32.Yinfei Yang, Daniel Cer, Amin Ahmad (2019) «Multilingual Universal Sentence Encoder for Semantic Retrieval»
- 33.Kassambara A. Cluster Validation Statistics: Must Know Methods. CLUSTER VALIDATION ESSENTIALS. DATA NOVA. URL:
<https://www.datanovia.com/en/lessons/cluster-validation-statisticsmust-know-methods/#internal-measures-for-cluster-validation>

- 34.Kumar, M.; Kumar, S.; Kashyap, P.K.; Aggarwal, G.; Rathore, R.S.; Kaiwartya, O.; Lloret, J. Green Communication in Internet of Things: A Hybrid Bio-Inspired Intelligent Approach. *Sensors* 2022
- 35.Karthikeya, S.A.; Vijeth, J.; Murthy, C.S.R. Leveraging solution-specific gateways for cost-effective and fault-tolerant IoT networking. In *Proceedings of the 2016 IEEE Wireless Communications and Networking Conference, Doha, Qatar, 3–6 April 2016*; IEEE: Piscataway, NJ, USA, 2016; pp. 1–6
- 36.Lin, P.C. Optimal smart gateway deployment for the Internet of Things in smart home environments. In *Proceedings of the 2015 IEEE 4th Global Conference on Consumer Electronics (GCCE), Osaka, Japan, 27–30 October 2015*; IEEE: Piscataway, NJ, USA, 2015; pp. 273–274
- 37.Jinaporn, N.; Saengudomlert, P. Impact of Gateway Placement and Energy Consumption for Data Processing on Lifetime of IoT Networks. In *Proceedings of the 2021 18th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON), Chiang Mai, Thailand, 19–22 May 2021*; IEEE: Piscataway, NJ, USA, 2021; pp. 90–93
- 38.Xu, W.; Kim, J.Y.; Huang, W.; Kanhere, S.S.; Jha, S.K.; Hu, W. Measurement, Characterization, and Modeling of LoRa Technology in Multifloor Buildings. *IEEE Internet Things J.* 2020, 7, 298–310
- 39.Capone, A.; Cesana, M.; Donno, D.D.; Filippini, I. Deploying multiple interconnected gateways in heterogeneous wireless sensor networks: An optimization approach. *Comput. Commun.* 2010, 33, 1151–1161
- 40.Ali, H.M.; Liu, J.; Bukhari, S.A.C.; Rauf, H.T. Planning a secure and reliable IoT-enabled FOG-assisted computing infrastructure for healthcare. *Clust. Comput.* 2022, 25, 2143–2161
- 41.Dang, L.M.; Piran, M.J.; Han, D.; Min, K.; Moon, H. A survey on internet of things and cloud computing for healthcare. *Electronics* 2019, 8, 768

42. Almarzoqi, S.A.; Yahya, A.; Matar, Z.; Gomaa, I. Re-Learning EXP3 Multi-Armed Bandit Algorithm for Enhancing the Massive IoT-LoRaWAN Network Performance. *Sensors* 2022
43. MESQUITA, E.d.M. Estudo Comparativo de Metaheurísticas Aplicadas ao Controle Preditivo Baseado em Modelo. Master's Thesis, Universidade de Brasília, Brasília, Brazil, 2018
44. Butler, E. Milton Friedman; Leya: Alfragide, Portugal, 2012
45. Thaljaoui, A.; Val, T.; Nasri, N.; Brulin, D. BLE localization using RSSI measurements and iRingLA. In Proceedings of the 2015 IEEE International Conference on Industrial Technology (ICIT), Seville, Spain, 17–19 March 2015; IEEE: Piscataway, NJ, USA, 2015; pp. 2178–2183
46. MacQueen, J. Some methods for classification and analysis of multivariate observations. In Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, Oakland, CA, USA, 1 January 1967; Volume 1, pp. 281–297
47. Petroni, A.; Biagi, M. Interference Mitigation and Decoding Through Gateway Diversity in LoRaWAN. *IEEE Trans. Wirel. Commun.* 2022
48. Planning and Optimization of Software-Defined and Virtualized IoT Gateway Deployment for Smart Campuses / D. Ferreira та ін. *Sensors*. 2022. Т. 22, № 13. С. 4710. URL: <https://doi.org/10.3390/s22134710>.
49. Efficient Gateways Placement for Internet of Things with QoS Constraints / I. Gravalos та ін. *GLOBECOM 2016 - 2016 IEEE Global Communications Conference*, м. Washington, DC, USA, 4–8 груд. 2016 р. 2016. URL: <https://doi.org/10.1109/glocom.2016.7841780>.
50. Екосистема розумного дому. URL: <https://slavinskiy18.wixsite.com/smart-home>
51. OpenHAB vs. Home Assistant. URL: <https://www.wundertech.net/openhab-vs-home-assistant/>
52. Getting Started - Home Assistant. URL: <https://home-assistant.io/getting-started/>