ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
КАФЕДРА ІНЖЕНЕРІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

КВАЛІФІКАЦІЙНА РОБОТА

на тему: «Методика адаптивного керування системами розумного будинку в умовах нестабільного енергопостачання на основі штучного інтелекту»

на здобуття освітнього ступеня магістра
зі спеціальності 121 Інженерія програмного забезпечення
освітньо-професійної програми «Інженерія програмного забезпечення»

Кваліфікаційна робота містить результати власних досліджень. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело

_____ Ярослав МУДРИК
(підпис)

Виконав: здобувач вищої освіти група ПДМ-62
Ярослав МУДРИК
Керівник: Оксана ЗОЛОТУХІНА
к.т.н., доцент
Рецензент: _____

Київ 2025

ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ

Навчально-науковий інститут інформаційних технологій

Кафедра Інженерії програмного забезпечення

Ступінь вищої освіти Магістр

Спеціальність 121 Інженерія програмного забезпечення

Освітньо-професійна програма «Інженерія програмного забезпечення»

ЗАТВЕРДЖУЮ
Завідувач кафедри
Інженерії програмного забезпечення
_____ Ірина ЗАМРІЙ
«_____» _____ 2024 р.

ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

Мудрику Ярославу Юрійовичу

1. Тема кваліфікаційної роботи: «Методика адаптивного керування системами розумного будинку в умовах нестабільного енергопостачання на основі штучного інтелекту»

керівник кваліфікаційної роботи Оксана ЗОЛОТУХІНА к.т.н., доцент,

затверджені наказом Державного університету інформаційно-комунікаційних технологій від «15» жовтня 2024 р. № 320.

2. Строк подання кваліфікаційної роботи «17» грудня 2024 р.

3. Вихідні дані до кваліфікаційної роботи: науково-технічна література, параметри, протоколи та опис технологій систем управління розумними будинками, зовнішні API для отримання даних про енергопостачання.

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)
   1. Аналіз систем розумних будинків, їх протоколів та особливостей адаптивного керування розумними будинками.
   2. Аналіз алгоритмів машинного навчання для прогнозування та оптимізації електроспоживання.

3. Розробка методики керування системами розумного будинку в умовах нестабільного енергоспоживанням на основі штучного інтелекту.
4. Програмна реалізація системи та проведення експериментальних досліджень.

5. Перелік графічного матеріалу:
1. Мета, об’єкта та предмет дослідження.
2. Актуальність роботи.
3. Алгоритм прогнозування споживання електроенергії.
4. Алгоритм оптимізації енергоспоживання.
5. Алгоритм аналізу споживання електроенергії.
6. Архітектура застосунку для адаптивного керування системами розумного будинку.
7. Екрані форми – консольний застосунок.
8. Моделювання процесу керування системами розумного будинку в умовах нестабільного енергопостачання.
9. Результати тестування.

6. Дата видачі завдання «16» жовтня 2024 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1	Аналіз науково-технічної літератури предметної галузі управління системами розумного будинку і застосування штучного інтелекту для управління подібними системами.	16.10-22.10.2024	
2	Дослідження технічних можливостей систем розумного будинку та аналіз архітектур розумних будинків.	23.10-30.10.2024	
3	Розробка методики адаптивного керування системами розумного будинку в умовах нестабільного енергопостачання на основі штучного інтелекту.	31.10-07.11.2024	
4	Формування технічних вимог до систем адаптивного управління.	08.11-15.11.2024	
5	Експериментальне дослідження та тестування	16.11-24.11.24	
6	Розробка демонстраційних матеріалів	23.11-27.11.24	
7	Оформлення роботи: вступ, висновки, реферат	28.11-05.12.24	

Здобувач вищої освіти _____ Ярослав МУДРИК
(підпис)

Керівник
кваліфікаційної роботи _____ Оксана ЗОЛОТУХІНА
(підпис)

РЕФЕРАТ

Текстова частина кваліфікаційної роботи на здобуття освітнього ступеня магістра: 95 стор., 4 табл., 21 рис., 21 джерел.

Мета роботи – покращення автономності та ефективності систем розумного будинку шляхом оптимізації енергоспоживання в умовах нестабільного енергопостачання із використанням штучного інтелекту.

Об'єкт дослідження – процес керування системами розумного будинку в умовах нестабільного енергопостачання.

Предмет дослідження – методи, моделі та засоби адаптивного керування системами розумного будинку в умовах нестабільного енергопостачання з використанням штучного інтелекту.

Короткий зміст роботи: У роботі було розроблено методику адаптивного керування системами розумного будинку в умовах нестабільного електропостачання на основі штучного інтелекту. Розробка враховує основні проблеми функціонування таких систем, зокрема, недостатню адаптивність до перебоїв енергопостачання, низьку ефективність енергоспоживання та обмежену інтеграцію з резервними джерелами живлення при відключенні електрики.

Адаптація керування реалізована на основі даних про пристрої в системах розумного будинку та завдяки алгоритмам оптимізації енергоспоживання на основі штучного інтелекту. Для моделювання процесу керування розроблено програмне забезпечення, яке має інтеграцію з API постачальників даних про електроенергію та забезпечує автоматизоване прийняття рішень на основі донавчених моделей штучного інтелекту та підтримку типових сценаріїв керування пристроями розумного будинку. Аналіз результатів моделювання на основі відключень за останні декілька місяців показав ефективність розробленої методики.

КЛЮЧОВІ СЛОВА: РОЗУМНИЙ БУДИНОК, МАШИННЕ НАВЧАННЯ, СТАТИСТИЧНА ІНФОРМАЦІЯ, АДАПТИВНИЙ МЕТОД, TENSORFLOW МОДЕЛЬ.

ABSTRACT

Text part of the master's qualification work: 95 pages, 4 tables, 21 pictures, 21 sources.

The purpose of the work is to improve the autonomy and efficiency of smart home systems by optimizing energy consumption in conditions of unstable energy supply using artificial intelligence.

The object of the study is the process of controlling smart home systems in conditions of unstable energy supply.

The subject of the study is methods, models and tools for adaptive control of smart home systems in conditions of unstable energy supply using artificial intelligence.

Summary of the work: The work developed a methodology for adaptive control of smart home systems in conditions of unstable power supply based on artificial intelligence. The development takes into account the main problems of the functioning of such systems, in particular, insufficient adaptability to power supply interruptions, low energy efficiency and limited integration with backup power sources during power outages.

Control adaptation is implemented based on data about devices in smart home systems and thanks to energy consumption optimization algorithms based on artificial intelligence. To simulate the control process, software has been developed that integrates with the API of electricity data providers and provides automated decision-making based on pre-trained artificial intelligence models and supports typical smart home device control scenarios. Analysis of the simulation results based on outages over the past few months has shown the effectiveness of the developed methodology.

KEYWORDS: SMART HOME, MACHINE LEARNING, STATISTICAL INFORMATION, ADAPTIVE METHOD, TENSORFLOW MODEL.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ 11
ВСТУП 12
1 АНАЛІЗ СИСТЕМ РОЗУМНИХ БУДИНКІВ ТА ЇХ ПРОТОКОЛІВ 15
1.1 Базовий функціонал систем розумного будинку 15
1.2 Існуючі системи розумного будинку 17
1.2.1 Home Assistant 17
1.2.2 SmartThings 18
1.2.3 Tuya Smart 20
1.2.4 Aqara 21
1.2.5 Yeelight 22
1.3 Протоколи 24
1.3.1 WiFi 24
1.3.2 Bluetooth 25
1.3.3 Zigbee 26
1.3.4 Z-Wave 27
1.3.5 Thread 28
1.3.6 Matter 28
1.4 Проблемні аспекти систем розумного будинку 29
1.4 Архітектурні особливості розумних будинків 32
1.4.1 Централізована архітектура 32
1.4.2 Децентралізована архітектура 34
1.4.3 Гібридна архітектура 36
1.4.4 Мережева архітектура 37
1.4.5 Порівняння архітектур 39
1.5 Аналіз інтерфейсів користувача та взаємодії з системами розумного будинку 40

1.5.1 Типи інтерфейсів користувача розумного будинку 41

1.5.2 Зручність та доступність 42

1.5.3 Персоналізація та адаптація 42

1.5.4 Виклики та проблеми у проектуванні інтерфейсів систем розумного будинку 43

1.5.5 Вплив інтерфейсу на досвід користувача 43

1.6 Взаємодія розумних будинків з іншими системами 44

1.6.1 Інтеграція з енергетичними системами 44

1.6.2 Взаємодія розумних будинків з системами безпеки 44

1.6.3 Інтеграція з транспортними системами 45

1.6.4 Інтеграція розумних будинків з системами розумного міста 45

1.6.5 Виклики інтеграції СРБ з іншими системами 46

1.6.6 Переваги інтеграції з іншими системами 46

2 РОЗРОБКА МЕТОДИКИ АДАПТИВНОГО КЕРУВАННЯ СИСТЕМАМИ РОЗУМНОГО БУДИНКУ В УМОВАХ НЕСТАБІЛЬНОГО ЕНЕРГОПОСТАЧАННЯ НА ОСНОВІ ШТУЧНОГО ІНТЕЛЕКТУ 47

2.1 Етапи адаптивного керування системами розумного будинку в умовах нестабільного енергопостачання 47

2.2 Моделювання сценаріїв енергоспоживання 48

2.3 Алгоритми оптимізації енергоспоживання 51

2.4 Інтеграція штучного інтелекту для адаптивного керування 54

2.5 Протоколи зв'язку та їх адаптація для нестабільних умов енергопостачання 56

2.6 Вимоги до реалізації засобів керування системами розумного будинку в умовах нестабільного енергопостачання за розробленою методикою 59

3 ПРАКТИЧНА РЕАЛІЗАЦІЯ ТА ЕКСПЕРИМЕНТАЛЬНІ ДОСЛІДЖЕННЯ. 61

3.1 Програмні засоби реалізації 61

3.1.1 .NET 61

3.1.2 ASP.NET Core 62

3.1.3 ML.NET 63
3.1.4 Супутні бібліотеки 64
3.2 Архітектура системи 65
3.2.1 Консольний застосунок 66
3.2.2 Web API 67
3.3 Навчання штучного інтелекту графікам відключення 69
3.4 Навчання штучного інтелекту читання новин по можливу загрозу 71
3.5 Створення бази відключень електроенергії 75
3.6 Тестування та використання в реальних умовах 78
3.7 Опис розроблених модулів 81
ВИСНОВКИ 84
ПЕРЕЛІК ПОСИЛАНЬ 86
ДОДАТОК А. ДЕМОНСТРАЦІЙНІ МАТЕРІАЛИ 89
ДОДАТОК Б. ЛІСТИНГ ОСНОВНИХ ПРОГРАМНИХ МОДУЛІВ 95

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

СРБ – система розумного будинку

МП – мова програмування

HTML - HyperText Markup Language

ARIMA – AutoRegressive Integrated Moving Average

JWT – JSON Web Token

ВСТУП

Сучасний світ стає все більш технологічно залежним, що зумовлює зростаючу популярність розумних будинків – автоматизованих систем управління житловими приміщеннями, які забезпечують комфорт, безпеку та енергоефективність. За даними роботи [1], впровадження таких технологій дозволяє автоматизувати побутові процеси, знижуючи енерговитрати та підвищуючи рівень комфорту для користувачів.

Однак, як зазначається в дослідженні [2], традиційні системи керування розумним будинком недостатньо адаптовані до умов нестабільного енергопостачання, оскільки вони передбачають наявність стабільного постачання електроенергії. У зв'язку з цим виникає потреба у створенні адаптивної системи, яка здатна гнучко реагувати на зміни в електропостачанні, оптимізуючи споживання енергії на основі аналізу даних та прогнозування. Зокрема, використання штучного інтелекту, згідно з даними роботи [3], є ефективним інструментом для підвищення автономності та надійності розумних будинків, дозволяючи їм функціонувати навіть в умовах обмежених ресурсів.

Об’єкт дослідження – процес керування системами розумного будинку в умовах нестабільного енергопостачання.

Предмет дослідження – методи, моделі та засоби адаптивного керування системами розумного будинку в умовах нестабільного енергопостачання з використанням штучного інтелекту.

Мета роботи – покращення автономності та ефективності систем розумного будинку шляхом оптимізації енергоспоживання в умовах нестабільного енергопостачання із використанням штучного інтелекту.

Для досягнення поставленої мети в роботі передбачено вирішення наступних завдань:

1) провести аналіз систем розумних будинків та особливостей керування ними, в тому числі, в умовах нестабільного енергопостачання;

2) дослідити особливості використання методів та засобів штучного інтелекту при вирішенні задач управління системами розумних будинків;

3) розробити методику адаптивного керування розумними будинками в умовах нестабільного електропостачання з використанням штучного інтелекту;

4) розробити програмну реалізацію методики адаптивного керування розумними будинками в умовах нестабільного електропостачання;

5) провести аналіз результатів впровадження методики адаптивного керування розумними будинками в умовах нестабільного електропостачання.

Методи дослідження – аналіз літературних джерел і сучасних підходів до управління розумними будинками, математичне моделювання процесів енергоспоживання, машинне навчання для прогнозування доступності енергії, а також експериментальне моделювання і тестування розроблених алгоритмів.

Практична значущість результатів – результати дослідження можуть бути використані для створення систем розумного будинку, які адаптуються до нестабільного енергопостачання. Запропонована методика зменшує залежність від безперебійної електрики, оптимізуючи ресурси і забезпечуючи життєво важливі функції. Це підвищує автономність, комфорт і безпеку мешканців, а також підходить для впровадження у різних масштабах, особливо в зонах ризику.

Робота пройшла апробацію на науково-практичних конференціях:

1. Всеукраїнська науково-технічна конференція «Застосування програмного забезпечення в ІКТ».

2. ІІ міжнародна науково-практична конференція «Сучасні аспекти діджиталізації та інформатизації в програмній та комп'ютерній інженерії».

За результатами апробації опубліковано тези доповідей [4] та [5].

Робота складається з вступу, трьох розділів, висновку, списку використаних джерел та додатків. У першому розділі проводиться огляд систем розумних будинків, які наразі існують, їх переваги, недоліки, а також протоколи за рахунок яких ці системи працюють. Другий розділ присвячений аналізу та вимог до систем, які працюють в умовах нестабільного електроспоживання та використовуватимуть штучний інтелект. Третій розділ присвячений опису реалізації та перевірки розробленої методики, а також представленню результатів моделювання.

1 АНАЛІЗ СИСТЕМ РОЗУМНИХ БУДИНКІВ ТА ЇХ ПРОТОКОЛІВ

1.1 Базовий функціонал систем розумного будинку

Створення системи розумного будинку [6] (СРБ) є важливим та цікавим рішенням для захисту та автоматизації процесів всередині житла. СРБ дозволяє зробити помешкання не лише більш комфортним, але й безпечним та енергоефективним, надаючи можливість контролювати та оптимізувати побутові процеси в автоматичному режимі. Завдяки інтеграції різних пристроїв і датчиків, СРБ може виконувати як прості дії, так і складні, з урахуванням різних умов, які задаються користувачем.

Одним із найпростіших прикладів використання системи розумного будинку є сценарій "Надобраніч". Він може бути активований простим натисканням кнопки на смартфоні і передбачає вимкнення більшості електропристроїв або переведення їх у режим низького енергоспоживання. Таким чином, мешканці можуть спокійно піти спати, знаючи, що всі непотрібні пристрої вимкнені і не споживають зайву енергію.

СРБ дозволяє створювати й більш складні сценарії, які можуть працювати на основі багатьох умов. Наприклад, сценарій "Прибирання" може активуватися лише за умови, що в домі нікого немає. Це можна перевірити різними способами, такими як аналіз місцеположення мешканців за допомогою GPS або перевірка стану датчиків руху. Якщо умови виконані, СРБ автоматично запускає робот-пилосос для прибирання всіх приміщень, а також може увімкнути пральну чи посудомийну машину, щоб максимально ефективно використати час, коли приміщення порожнє.

Головною перевагою СРБ є те, що вона надає повне розуміння того, що відбувається всередині будинку, навіть коли мешканці перебувають далеко – на відстані тисячі кілометрів. Завдяки системі можна перевіряти стан датчиків температури як у будинку, так і на вулиці, стежити за рівнем споживання електроенергії кожним із пристроїв, або навіть контролювати вологість у кожній

кімнаті. У той час як більшість звичайних мешканців дізнаються про проблеми типу "потопу" лише після дзвінків від сусідів, СРБ може вчасно попередити про аварію і, за певних умов, автоматично виконати відповідні дії. Наприклад, система може вимкнути живлення пристроїв у постраждалих зонах, перекрити водопостачання, надіслати попередження у вигляді повідомлень у месенджерах як мешканцям, так і сусідам. Це допомагає уникнути значних збитків і забезпечити оперативне реагування на аварійні ситуації.

Ще одна значна перевага СРБ – це інтеграція системи безпеки помешкання. Наприклад, коли хтось із сім'ї їде на відпочинок, є можливість встановити режим "Нікого вдома". У цьому режимі активуються датчики руху, зовнішні і внутрішні камери спостереження, а також інші засоби охорони. Якщо система зареєструє підозрілий рух, буде миттєво надіслано сповіщення на смартфон, а також буде можливість переглядати відео з камер у режимі реального часу. Це дозволяє вчасно реагувати на потенційні загрози і навіть викликати поліцію, якщо це необхідно.

Системи розумного будинку також дозволяють інтегрувати різні інші функції для поліпшення комфорту та безпеки. Є можливість налаштувати автоматичне освітлення, яке вмикається, коли хтось заходить в кімнату, або налаштувати сценарії для конкретних подій, таких як повернення додому. Наприклад, система може заздалегідь увімкнути опалення, щоб мешканці повернулися у вже прогріте приміщення, або увімкнути музику, яку приємно слухати після роботи. Всі ці функції роблять життя мешканців комфортнішим і зручнішим.

СРБ не лише допомагає підвищити безпеку і комфорт житла, але й може значно економити витрати на енергоспоживання. Завдяки можливості стежити за рівнем використання електроенергії кожним приладом і оптимізувати його, мешканці можуть уникнути зайвих витрат. Наприклад, система може самостійно вимикати прилади, які залишилися ввімкненими без необхідності, або переводити їх у режим економії енергії, коли вони не використовуються.

1.2 Існуючі системи розумного будинку

Компаній які розробляють СРБ чимало, проте є компанії які спеціалізуються суто на цьому (серед них Home Assistant [7]), а є компанії, які створюють СРБ до власної інфраструктури (Tapo, SmartThings, Tuya Smart, Aqara, Yeelight).

1.2.1 Home Assistant

Home Assistant – це популярна відкрита платформа для автоматизації та управління розумним будинком, що дозволяє інтегрувати та керувати різноманітними пристроями та сервісами з єдиного інтерфейсу. Home Assistant підтримує локальне управління без потреби в підключенні до хмарних сервісів, що дозволяє користувачам бути впевненими у збереженні своєї конфіденційності. Платформа активно розвивається спільнотою розробників і має гнучку систему налаштування автоматизацій. На рисунку 1.1 можна побачити головний екран.

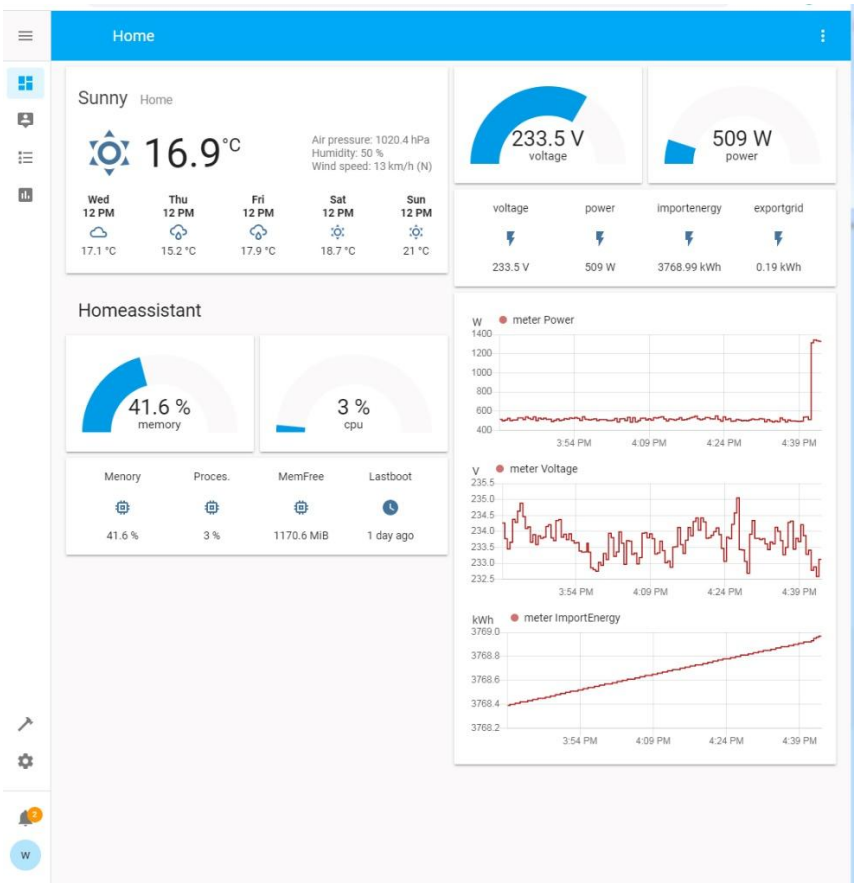


Рис. 1.1. Головна сторінка HomeAssistant

Переваги:

- відкритий код: можливість налаштування та розширення функціоналу відповідно до потреб користувача;
- широка інтеграція: підтримка понад 1000 різних пристроїв та сервісів, таких як Google Home, Amazon Alexa, Zigbee, Z-Wave тощо;
- локальне управління: дані зберігаються локально, що підвищує рівень конфіденційності та безпеки;
- гнучкість налаштувань: можливість створювати складні сценарії та автоматизації, які можуть враховувати різні умови, сенсори та тригери;
- широка спільнота: величезна активна спільнота, яка допомагає вирішувати проблеми та розробляє нові інтеграції;
- інтеграція з іншими системами: підтримка інтеграції з екосистемами Google Home, Amazon Alexa, Apple HomeKit, що дозволяє легко керувати різними пристроями з іншої інфраструктури;
- робота на недорогому обладнанні: платформа може бути встановлена на Raspberry Pi або будь-якому іншому сервері.

Недоліки:

- складність налаштування: для початківців процес встановлення та конфігурації може бути складним;
- вимоги до технічних знань: потрібні базові знання програмування та мережевих технологій.

1.2.2 SmartThings

SmartThings – це платформа від Samsung, яка дозволяє інтегрувати та керувати розумними пристроями з різних категорій, таких як освітлення, системи безпеки, датчики руху, розетки тощо. SmartThings підтримує широкий спектр пристроїв різних виробників та дозволяє створювати сценарії автоматизації для підвищення зручності та безпеки житла. На рисунку 1.2 наведено приклад екранних форм, які відображають обрані пристрої користувача.

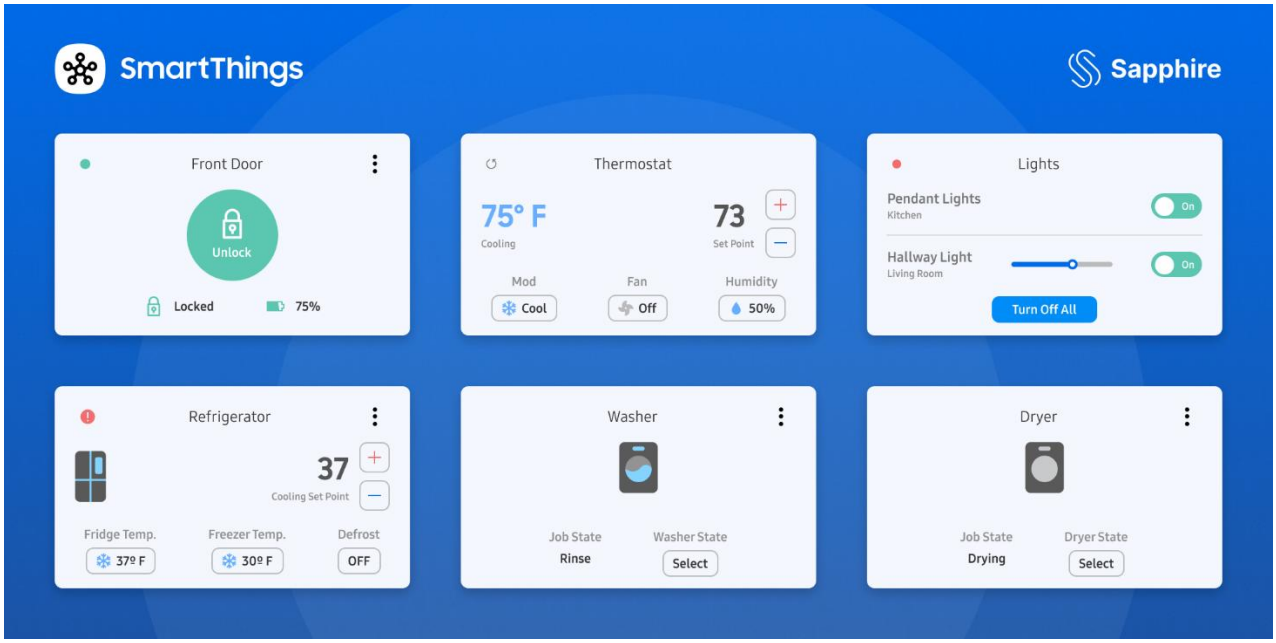


Рис 1.2. Обрані пристрої в SmartThings

Переваги:

- широка сумісність: підтримка багатьох пристроїв і протоколів зв'язку, таких як Zigbee, Z-Wave, Wi-Fi;
- інтуїтивний інтерфейс: зручний додаток для управління та налаштування автоматизацій;
- інтеграція з Home Assistant: можливість інтеграції для розширення функціоналу та локального управління;
- гнучкість автоматизацій: створення складних сценаріїв для різних пристроїв залежно від часу доби, присутності користувачів та інших параметрів;
- підтримка голосових помічників: можливість використання Google Assistant, Amazon Alexa, Bixby для голосового керування;
- велика спільнота: підтримка та активна спільнота, що допомагає у налаштуванні та інтеграції нових пристроїв.

Недоліки:

- залежність від хмарних сервісів: більшість функцій вимагають підключення до інтернету;
- обмежена локальна автоматизація: менше можливостей для локального управління без інтернету.

1.2.3 Tuya Smart

Tuya Smart – це потужна платформа для управління та інтеграції розумних пристроїв різних виробників. Tuya підтримує тисячі пристроїв, що робить її однією з найбільших екосистем розумного будинку. Вона надає виробникам можливість створювати власні пристрої, а користувачам – налаштовувати автоматизацію для будинку. Головний екран зображено на рисунку 1.3.

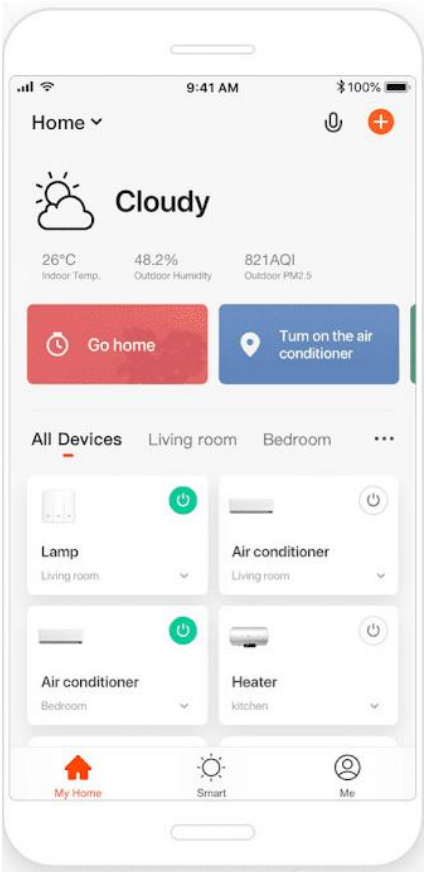


Рис 1.3. Головний екран Tuya Smart

Переваги:

- велика екосистема: підтримка тисяч пристроїв від різних виробників, що дозволяє вибирати з широкого асортименту;
- гнучкість: можливість створення власних автоматизацій та сценаріїв, що враховують різні параметри та умови;
- інтеграція з Home Assistant: підтримка інтеграції для локального управління, що підвищує зручність та швидкість реакції системи;
- доступна ціна: більшість пристроїв Tuya доступні за прийнятною ціною;

- масштабованість: можливість легко розширювати систему, додаючи нові пристрої.

Недоліки:

- залежність від хмарних сервісів: більшість функцій вимагають підключення до інтернету;
- питання безпеки: деякі користувачі висловлюють занепокоєння щодо конфіденційності даних.

1.2.4 Aqara

Aqara – це бренд, що пропонує розумні пристрої для автоматизації будинку, включаючи датчики, камери, розетки, перемикачі та інші. Aqara має сильну інтеграцію з екосистемами, такими як Apple HomeKit, Google Home, Amazon Alexa. Пристрої Aqara відзначаються надійністю та якістю, і часто використовуються в системах безпеки та автоматизації. Список девайсів, які доступні користувачу з головного екрану можна побачити на рисунку 1.4.

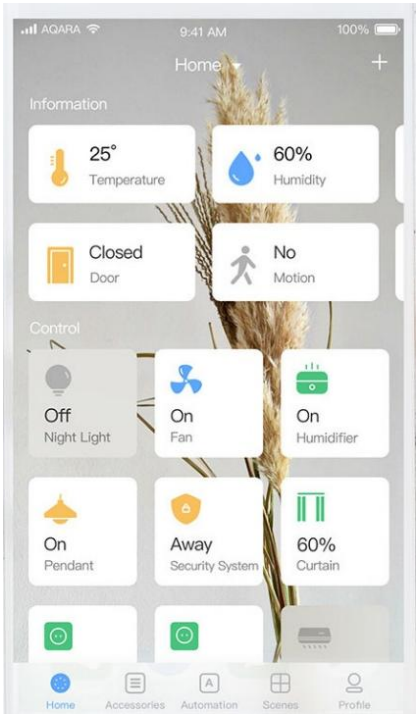


Рис 1.4. Головний інтерфейс Aqara

Переваги:

- сумісність з Home Assistant: підтримка інтеграції для локального управління та автоматизації;
- підтримка Matter: нові продукти підтримують стандарт Matter, що забезпечує кращу сумісність з іншими пристроями;
- широкий асортимент: різноманітність пристроїв для різних потреб – від датчиків до камер та розеток;
- легка інтеграція з Apple HomeKit: сумісність з HomeKit, що робить Aqara ідеальним вибором для користувачів екосистеми Apple;
- висока якість: пристрої вирізняються високою надійністю та довговічністю.

Недоліки:

- залежність від хабу: більшість пристроїв вимагають використання хабу для повної функціональності;
- обмежена підтримка поза екосистемою: деякі пристрої можуть мати обмежену сумісність з іншими платформами.

1.2.5 Yeelight

Yeelight – це бренд, що спеціалізується на розумному освітленні, пропонуючи лампи, стрічки та інші освітлювальні рішення з можливістю дистанційного управління через додаток або інтеграцію з голосовими помічниками. Продукти Yeelight виділяються високою якістю освітлення і підтримкою великої кількості кольорів та режимів. Режим керування конкретною лампою можна побачити на рисунку 1.5.

Переваги:

- простота використання: легке налаштування та управління через додаток Yeelight або голосовими помічниками;
- інтеграція з Home Assistant: підтримка інтеграції для локального управління та створення автоматизацій;

- висока якість освітлення: широкий спектр кольорів та режимів освітлення, можливість регулювання яскравості;
- сумісність з Google Home та Amazon Alexa: легке керування через голосові команди;
- доступна ціна: якісні рішення за доступною ціною, що дозволяє швидко розширювати систему освітлення в домі.

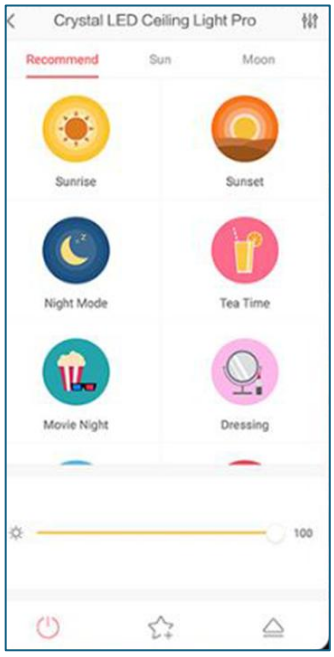


Рис 1.5. Налаштування лампи

Недоліки:

- обмежені функції поза освітленням: Yeelight зосереджена переважно на освітленні, тому можливості інтеграції з іншими пристроями обмежені;
- залежність від хмарних сервісів: деякі функції вимагають підключення до інтернету.

Пристрої, які підтримують роботу з розумними будинком працюють по різним протоколам і по різному взаємодіють між собою. Основна проблема, яка зараз є, полягає в стандартизації і можливості підключення девайсів від інших виробників до однієї СРБ. Тобто пристрої від компанії Aqara будуть погано працювати з СРБ Tuya Smart. Причому в кожній інфраструктурі присутні пристрої

які працюють тільки з якимось конкретним протоколом (Zigbee чи Z-wave) і таким девайсам необхідний хаб, і це погіршує ситуацію.

1.3 Протоколи

Кожен розумний пристрій комунікує з СРБ і ця взаємодія відбувається по різним протоколам. Найбільше популярні серед них наступні: (WiFi, Bluetooth, Zigbee, Z-Wave, Thread, Matter).

1.3.1 WiFi

Протокол бездротового зв'язку, який був розроблений наприкінці 90-х років для передавання даних та потоків інформації через радіоканали. WiFi працює на основі стандарту IEEE 802.11 і став одним з найпоширеніших способів з'єднання пристроїв в домашніх, офісних та громадських мережах. Цей протокол дозволяє передавати великі обсяги даних на високих швидкостях, що забезпечує якісне з'єднання для різних пристроїв, включаючи комп'ютери, смартфони, розумні телевізори та інші елементи розумного будинку.

Кожен пристрій, який підключається до мережі WiFi, отримує свою унікальну IP-адресу і таким чином може функціонувати як частина мережі, з якою можна взаємодіяти не тільки локально, але й ззовні, якщо мережа має доступ до Інтернету. Завдяки цій властивості, пристрої стають своєрідними "хабами", що дозволяє легко інтегрувати їх в системи розумного будинку, забезпечуючи віддалене управління і контроль з будь-якої точки світу.

Переваги WiFi:

- висока пропускна здатність: WiFi забезпечує передачу великих обсягів даних, що є особливо важливим для потокового відео, відеоконференцій, онлайн-ігор та роботи з великими файлами;
- висока швидкість передачі даних: сучасні стандарти WiFi, такі як WiFi 5 (802.11ac) і WiFi 6 (802.11ax), можуть забезпечувати швидкості до кількох гігабіт на секунду;

- широкий діапазон підключень: WiFi здатний підтримувати одночасне підключення багатьох пристроїв, що робить його зручним для використання в домашніх і офісних умовах;
- кожен пристрій є хабом: можливість інтеграції та віддаленого управління кожним пристроєм через IP-адресу дозволяє легко автоматизувати роботу розумних пристроїв і отримувати доступ до них з будь-якого місця.

Недоліки WiFi:

- високе енергоспоживання: пристрої, які використовують WiFi, зазвичай споживають більше енергії порівняно з пристроями, що використовують інші протоколи зв'язку, такі як Zigbee або Z-Wave. Це робить WiFi менш ефективним для деяких малопотужних пристроїв, таких як датчики або автономні пристрої, що працюють від батарей;
- чутливість до перешкод: WiFi мережі можуть страждати від завад через інші радіосигнали, що може погіршувати якість з'єднання, особливо в місцях з великою кількістю інших мереж.

WiFi є ключовим елементом сучасних розумних будинків, забезпечуючи зручний зв'язок між різними пристроями. Однак, через високе енергоспоживання, для деяких елементів автоматизації можуть використовуватись інші технології, які краще підходять для пристроїв з низьким споживанням енергії.

1.3.2 Bluetooth

Bluetooth – це бездротовий протокол, розроблений для короткодистанційного зв'язку між пристроями. Він був розроблений у 1994 році для заміни дротових з'єднань, таких як кабелі для передачі файлів і синхронізації між телефонами та іншими пристроями. Bluetooth часто використовується для створення локальних мереж між смартфонами, навушниками, колонками, а також в пристроях розумного будинку для швидкої передачі даних на близькій відстані.

Переваги:

- низьке енергоспоживання (BLE): технологія Bluetooth Low Energy (BLE) споживає значно менше енергії, що робить її ідеальною для невеликих пристроїв, таких як датчики або розумні замки;
- широка доступність: практично всі сучасні смартфони та пристрої підтримують Bluetooth, що робить цей протокол дуже зручним для підключення;
- простота з'єднання: легке налаштування зв'язку між пристроями, особливо для передачі аудіо або обміну файлами.

Недоліки:

- обмежена дальність дії: Bluetooth має обмежений діапазон, зазвичай до 10-30 метрів (залежно від класу пристрою), що обмежує використання в масштабах всього будинку;
- мале навантаження: Bluetooth не підходить для підтримки багатьох одночасних з'єднань, що зменшує його ефективність у великих мережах розумного будинку;
- Інтерференція: вразливий до перешкод через використання діапазону 2,4 ГГц, що часто переповнений іншими пристроями.

1.3.3 Zigbee

Zigbee – це протокол бездротового зв'язку, призначений для енергоефективного підключення пристроїв у розумних будинках. Використовує сітчасту мережу, що дозволяє кожному пристрою бути вузлом, що підсилює сигнал, забезпечуючи велику надійність і дальність передачі. Zigbee працює в діапазоні 2,4 ГГц і підтримує малі пакети даних для пристроїв, які не потребують великої пропускної здатності.

Переваги:

- низьке енергоспоживання: Zigbee споживає мало енергії, що дозволяє пристроям працювати від батареї роками;
- сітчаста мережа: кожен пристрій виступає як вузол, що розширює мережу, збільшуючи її дальність і надійність;

- широка підтримка: багато виробників використовують Zigbee, що робить його універсальним вибором для автоматизації будинку.

Недоліки:

- складність налаштування: для новачків налаштування Zigbee мережі може бути складним;
- перешкоди: працює в тому ж діапазоні 2,4 ГГц, що і WiFi, що може викликати перешкоди і зменшувати ефективність передачі.

1.3.4 Z-Wave

Z-Wave – це інший протокол бездротового зв'язку, розроблений спеціально для пристроїв розумного будинку. На відміну від Zigbee, Z-Wave працює в діапазоні 800-900 МГц, що забезпечує менше перешкод і кращу проникність через стіни та інші перешкоди. Z-Wave також використовує сітчасту топологію, де кожен пристрій є частиною мережі.

Переваги:

- низьке енергоспоживання: Z-Wave пристрої можуть працювати від батареї протягом тривалого часу, що робить їх ефективними для сенсорів і автономних пристроїв;
- менше перешкод: завдяки використанню частоти 800-900 МГц, Z-Wave менше схильний до перешкод з WiFi та іншими побутовими пристроями;
- надійність: кожен пристрій у мережі підсилює сигнал, збільшуючи надійність та радіус дії.

Недоліки:

- обмежена кількість виробників: Z-Wave є запатентованою технологією, тому менше пристроїв підтримують його порівняно з Zigbee;
- ціна: пристрої на основі Z-Wave часто дорожчі через запатентовану природу технології.

1.3.5 Thread

Thread – це сучасний бездротовий протокол для підключення пристроїв розумного будинку, розроблений спеціально для роботи в сітчастих мережах. Thread був створений, щоб забезпечити більш надійне і безпечне з'єднання, використовуючи малопотужні пристрої. Протокол розроблений консорціумом Thread Group і підтримує відкриті стандарти, що забезпечує сумісність з багатьма виробниками.

Переваги:

- нергоефективність: Thread був розроблений з урахуванням мінімального енергоспоживання, що робить його ідеальним для датчиків та інших автономних пристроїв;
- безпека: вбудовані засоби шифрування забезпечують високий рівень безпеки з'єднання;
- сітчаста мережа: пристрої можуть утворювати сітчасту мережу, що підвищує надійність та збільшує дальність зв'язку;
- сумісність з Matter: Thread підтримує стандарт Matter, що забезпечує кращу сумісність пристроїв між різними виробниками.

Недоліки:

- новизна: протокол ще новий, тому підтримка пристроїв поки що обмежена;
- необхідність Thread Border Router: для роботи з іншими протоколами потрібен спеціальний пристрій – Thread Border Router, що може ускладнювати налаштування.

1.3.6 Matter

Matter – це новий стандарт, який був створений для об'єднання різних екосистем розумного будинку та забезпечення взаємної сумісності пристроїв. Matter був розроблений консорціумом Connectivity Standards Alliance (CSA), до

якого входять великі технологічні компанії, такі як Google, Amazon, Apple, і інші. Основна мета Matter – забезпечити уніфіковане середовище для роботи розумних пристроїв, яке дозволяє легко інтегрувати та управляти пристроями різних виробників.

Переваги:

- взаємна сумісність: Matter забезпечує сумісність між пристроями від різних виробників, спрощуючи інтеграцію та управління розумними пристроями;
- висока безпека: вбудовані засоби шифрування і сучасні стандарти безпеки гарантують захист даних користувачів;
- підтримка провідних компаній: стандарт підтримується великими компаніями, такими як Google, Amazon, Apple, що сприяє швидкому впровадженню та широкій підтримці;
- інтеграція з Thread: Matter працює разом з Thread, що дозволяє використовувати енергоефективні сітчасті мережі для передачі даних.

Недоліки:

- стадія впровадження: стандарт ще перебуває на початковій стадії розвитку, і кількість пристроїв, які його підтримують, поки обмежена;
- залежність від інфраструктури: для використання Matter потрібна інфраструктура, яка підтримує нові стандарти, що може вимагати додаткових витрат на модернізацію існуючих систем.

1.4 Проблемні аспекти систем розумного будинку

У СРБ існує кілька основних проблемних аспектів, які виникають під час побудови мережі та її подальшого використання.

Першою та найголовнішою проблемою є забезпечення безпеки та конфіденційності. Багато СРБ працюють через інтернет і передають великі масиви даних про користувачів. Це створює ризик крадіжки персональних даних,

починаючи від номера телефону, імені та прізвища до точної локації системи, адреси помешкання і навіть плану будинку чи квартири. Оскільки виробники розумних пристроїв намагаються здешевити свою продукцію для збільшення обсягів продажів, це часто призводить до зниження рівня безпеки, що може негативно вплинути на конфіденційність користувачів. У випадку недостатнього захисту каналів передачі даних або некоректного налаштування безпеки існує ймовірність зламу системи зловмисниками.

Друга проблема – енергоефективність. Усі пристрої розумного будинку потребують живлення, що робить підтримку великої кількості підключених пристроїв доволі складним завданням. Оскільки різні виробники використовують різні джерела живлення, від звичайних батарейок CR2032 до прямого підключення в розетку, це може створювати додаткові труднощі для забезпечення безперебійної роботи пристроїв. Хоча розумні пристрої часто дозволяють зменшити загальне енергоспоживання за рахунок автоматизації, їхня велика кількість може призвести до збільшення витрат на енергію та необхідності регулярного обслуговування, зокрема заміни батарейок.

Третя проблема – сумісність розумних пристроїв. У процесі розробки кожен виробник обирає найкращий протокол для забезпечення функціональності своїх пристроїв, однак навіть при використанні однакових протоколів (наприклад, Zigbee) не завжди можливо об'єднати пристрої в одну систему розумного будинку. Наприклад, пристрої від компаній Aqara і Tuya можуть працювати на основі одного протоколу Zigbee, але не всі вони можуть бути підключені до одного хабу через обмеження, пов'язані з екосистемою та вимогами виробників. Більшість екосистем СРБ обмежують можливості підключення пристроїв сторонніх виробників для збереження контролю та надійності роботи системи.

Четвертою проблемою є надійність самих пристроїв або системи загалом. Навіть невеликий збій, наприклад, через проблеми з підключенням або втратою живлення, може призвести до відмови розумних пристроїв. Якщо такі пристрої

використовуються для критичних завдань, як-от вхідний замок, камери спостереження, електронні вимикачі води або регулятори підігріву, їхня відмова може призвести до значних незручностей або навіть створити небезпеку для мешканців будинку. Надійність цих пристроїв є ключовим фактором для збереження як комфорту, так і безпеки.

П'ята проблема стосується доступу до інтернету. Це питання є лише частковою проблемою, оскільки залежить від конкретного виробника розумних пристроїв. Деякі виробники не готові підтримувати хмарну інфраструктуру для користувачів і пропонують лише локальне управління. Це означає, що користувачі зможуть керувати пристроями лише перебуваючи вдома. Як тільки вони покинуть помешкання або будуть використовувати мобільний інтернет, можливості керування зникають. Відсутність інтернет-з'єднання обмежує функціональність системи і може стати значним бар'єром для багатьох користувачів.

Шостою і, можливо, однією з найвагоміших проблем є вартість. Для забезпечення надійної роботи краще обирати якісні пристрої від відомих брендів, які мають досвід в сфері розумного будинку. Однак такі пристрої часто мають високу ціну, що може стати перешкодою для багатьох користувачів. Ціна зростає також залежно від рівня автоматизації: мінімальний набір розумних розеток вже може зробити ваш дім розумним, але для побудови повноцінної інфраструктури, що включає розумні замки, камери спостереження, датчики диму та руху, освітлення, пральні машини, кондиціонери чи навіть терморегулятори для батареї, доведеться витратити значні кошти і може варіюватися від 2к грн до 250к грн якщо мова йде про квартиру, а у випадку приватного будинку може сягати і 500к і 1500кк грн. Це робить автоматизацію помешкання значним фінансовим вкладенням, що не завжди є виправданим для споживачів [8].

Таким чином, хоча системи розумного будинку мають великий потенціал для підвищення рівня комфорту, безпеки та енергоефективності, вони стикаються з кількома суттєвими викликами, що включають безпеку та конфіденційність даних,

енергоефективність, сумісність пристроїв, їхню надійність, необхідність інтернет-з'єднання та вартість обладнання [9]. Вирішення цих проблем може значно поліпшити користувацький досвід та сприяти більш широкому впровадженню розумних технологій у будинках.

1.4 Архітектурні особливості розумних будинків

Архітектура систем розумного будинку є важливою складовою, яка визначає ефективність, масштабованість, надійність і зручність таких систем. Вона визначає, як різні пристрої взаємодіють між собою, як здійснюється керування, які мережеві технології використовуються та наскільки гнучкою є система в адаптації до змін. Існує кілька підходів до побудови архітектури розумних будинків, і вони можуть суттєво відрізнятися залежно від типу мережі, способу управління, вимог до безпеки та масштабів інтеграції. Існує їх багато, але основні архітектурні моделі, що використовуються для побудови розумних будинків можна поділити на: централізовану, децентралізовану, гібридну та мережеву архітектури.

1.4.1 Централізована архітектура

Централізована архітектура є однією з найпоширеніших у системах розумного будинку, де всі пристрої та сенсори з'єднуються з одним головним контролером або хабом, як зображено на рисунку 1.6.

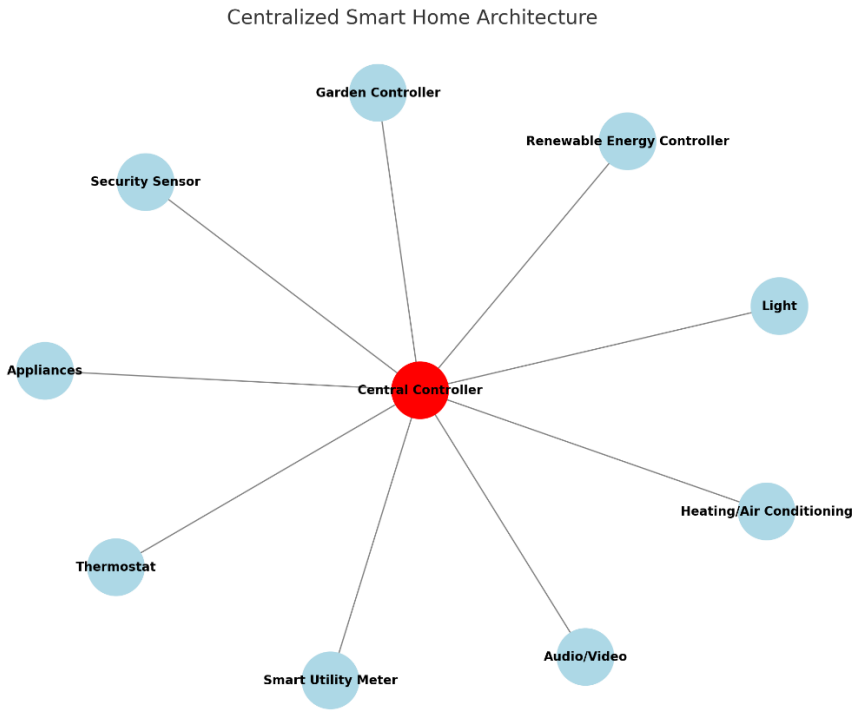


Рис. 1.6. Централізована архітектура

Головний контролер обробляє інформацію, приймає рішення та виконує управління пристроями. До переваг такої архітектури належать спрощене адміністрування і зручність налаштування, оскільки користувач має доступ до всієї системи через єдине горлечко входу.

Також централізована архітектура дозволяє ефективніше здійснювати моніторинг та управління пристроями, оскільки всі дані зберігаються в одному місці. Наприклад, дані від сенсорів температури, датчиків руху та інших пристроїв можуть бути об'єднані в одному контролері, що дозволяє приймати зважені рішення на основі аналізу всієї системи.

Проте централізована архітектура має й свої недоліки. Основним є вразливість до відмови центрального контролера. Якщо головний контролер виходить з ладу, уся система може стати недієздатною. Це створює загрозу безпеці та зниженню комфорту мешканців. Також централізовані системи можуть мати обмеження в масштабованості, оскільки збільшення кількості пристроїв може призвести до перевантаження контролера та зниження продуктивності. Наприклад,

у великому будинку з десятками або навіть сотнями пристроїв є ризик того, що контролер не зможе оперативно обробляти всі сигнали.

Для прикладу, в системі Home Assistant централізована архітектура може бути реалізована через сервер на базі Raspberry Pi [10]. Сервер відповідає за отримання даних від пристроїв через MQTT [11] або Zigbee-протоколи, збереження інформації у базі даних та виконання автоматизацій. У реальному сценарії, якщо температура в приміщенні падає нижче 18°C, система може активувати обігрівач через реле.

Прикладом переваг є легка реалізація автоматизацій через обробку даних в одному центрі, а також проста інтеграція з API таких екосистем, як Google Home чи Amazon Alexa.

1.4.2 Децентралізована архітектура

Децентралізована архітектура є альтернативою централізованій, і полягає в тому, що кожен пристрій або група пристроїв можуть працювати автономно, приймаючи власні рішення. У цій моделі немає єдиного центрального хаба, а взаємодія між пристроями здійснюється безпосередньо або через локальну мережу. Це можна побачити на рисунку 1.7.

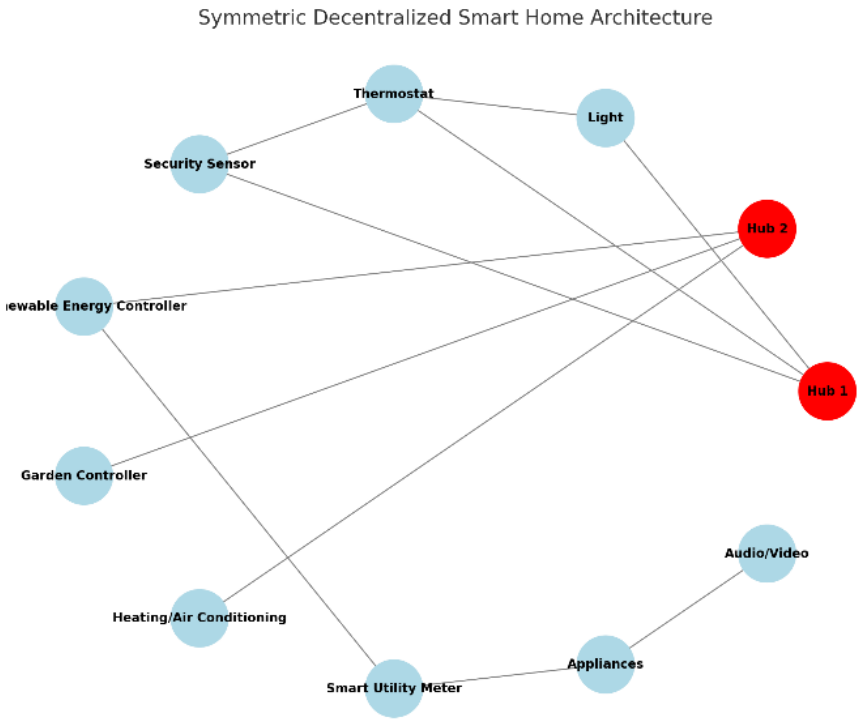


Рис 1.7. Децентралізована архітектура

Децентралізована архітектура є більш стійкою до відмов, оскільки вихід з ладу одного пристрою не впливає на роботу інших компонентів системи. Наприклад, якщо один з освітлювальних контролерів вийде з ладу, інші пристрої, такі як система безпеки або кліматичний контроль, залишаться функціональними. Такий підхід також забезпечує кращу масштабованість, оскільки додавання нових пристроїв не створює навантаження на один головний контролер.

Однак децентралізовані системи можуть бути складнішими в налаштуванні та управлінні, оскільки кожен пристрій потребує індивідуального конфігурування, а взаємодія між ними повинна бути ретельно спланована. Це може вимагати додаткових зусиль для забезпечення сумісності пристроїв різних виробників. Крім того, у таких системах часто виникають проблеми з сумісністю різних пристроїв та протоколів, що може потребувати використання шлюзів для інтеграції різнорідних технологій.

Прикладом використання є налаштування таких мереж за допомогою протоколів Zigbee або Z-Wave, які забезпечують низьке енергоспоживання. Логіка управління при цьому може бути закладена у вбудоване програмне забезпечення пристроїв (firmware).

1.4.3 Гібридна архітектура

Гібридна архітектура поєднує переваги централізованої та децентралізованої моделей, намагаючись мінімізувати їх недоліки. У цій архітектурі система має центральний контролер, але окремі пристрої також мають можливість працювати автономно. Це дозволяє забезпечити баланс між зручністю управління та стійкістю до відмов, приклад наведено на рисунку 1.8.

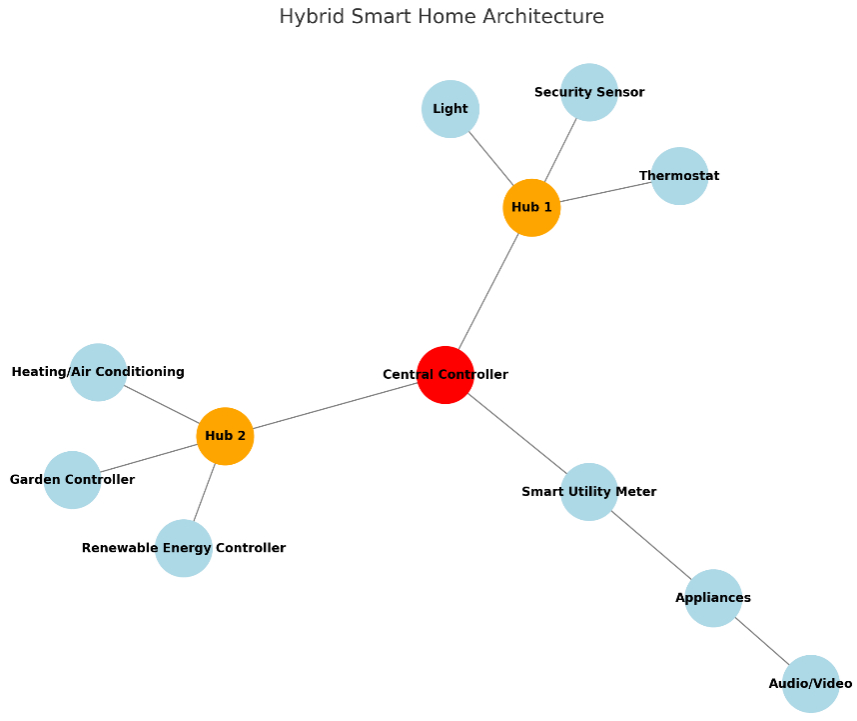


Рис. 1.8. Гібридна архітектура

Гібридна архітектура дозволяє зберегти централізований контроль для основних функцій, таких як моніторинг та загальне управління, водночас надаючи

окремим пристроям можливість самостійного прийняття рішень у разі втрати зв'язку з контролером. Наприклад, система безпеки може залишатися активною та реагувати на загрози навіть при втраті зв'язку з центральним хабом. Це робить систему більш гнучкою та адаптивною до змінних умов, наприклад, під час перебоїв у енергопостачанні.

Гібридні системи також дозволяють оптимізувати використання ресурсів. Наприклад, енергозберігаючі пристрої можуть працювати в автономному режимі, автоматично регулюючи свою роботу залежно від поточних умов. Це допомагає зменшити енергоспоживання і підвищити ефективність системи.

1.4.4 Мережева архітектура

Окрім загальної структурної архітектури, важливу роль у системах розумних будинків відіграє мережева архітектура. Її представлення можна побачити на рисунку 1.9. Зазвичай використовуються різні типи мереж, такі як Wi-Fi, Zigbee, Z-Wave, Bluetooth та інші. Кожна з цих технологій має свої переваги та недоліки залежно від вимог до пропускної здатності, енергоспоживання та дальності дії.

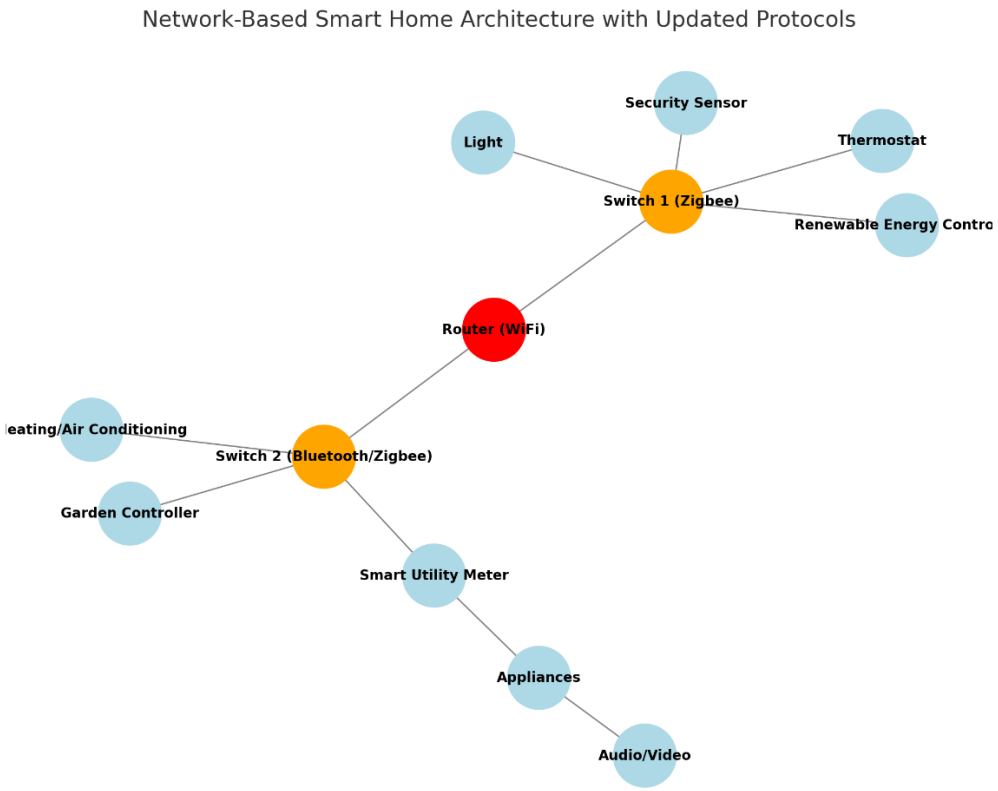


Рис. 1.9. Мережева архітектура

Wi-Fi забезпечує високу швидкість передачі даних, що робить його ідеальним для пристроїв з високими вимогами до пропускної здатності, таких як камери спостереження або мультимедійні пристрої. Однак, він має значне енергоспоживання, що робить його менш придатним для пристроїв, що живляться від батарейок. Наприклад, сенсори на батарейках можуть швидко розряджатися при використанні Wi-Fi.

Zigbee та Z-Wave характеризуються низьким енергоспоживанням і підходять для сенсорів та інших пристроїв, що працюють у режимі економії енергії. Наприклад, датчики руху або температури можуть працювати на одній батареї протягом кількох років. Також ці технології забезпечують надійне з'єднання та мають можливість побудови сітчастих мереж, що дозволяє збільшити дальність дії за рахунок ретрансляції сигналів іншими пристроями.

Bluetooth використовується здебільшого для короткодистанційної взаємодії між пристроями. Він підходить для управління окремими елементами, такими як розумні замки або освітлення. Перевагою Bluetooth є його низька вартість та простота інтеграції, але обмежена дальність дії може бути недоліком у великих будинках або для пристроїв, розташованих на значній відстані один від одного.

1.4.5 Порівняння архітектур

Кожна з розглянутих архітектур має свої унікальні переваги та недоліки, і вибір конкретного підходу залежить від конкретних вимог користувача.

Централізована архітектура підходить для невеликих установок, де важлива зручність управління та простота налаштування. Наприклад, для квартири або невеликого будинку централізована система може бути оптимальною завдяки своїй простоті та єдиній точці управління.

Децентралізована архітектура краще підходить для великих систем, де важлива стійкість до відмов та можливість автономної роботи. Наприклад, у великих приватних будинках або на підприємствах децентралізована система дозволяє зменшити ризик повного відключення в разі виходу з ладу одного елемента.

Гібридна архітектура є оптимальним вибором для забезпечення гнучкості, надійності та масштабованості, особливо в умовах нестабільного енергопостачання. Вона дозволяє поєднати зручність централізованого управління з можливістю автономної роботи окремих компонентів. Це особливо важливо для критичних систем, таких як безпека або управління енергоспоживанням, де необхідно забезпечити безперервну роботу незалежно від стану центрального контролера. Порівняння архітектур наведено в таблиці 1.1.

Таблиця 1.1

Порівняння існуючих архітектур

Тип архітектури	Переваги	Недоліки	Приклади
Централізована	Простота, централізований моніторинг	Уразливість до відмов хабу	Невеликі будинки, Home Assistant
Децентралізована	Автономність, стійкість	Складність налаштування	Великі будинки, Zigbee, Z-Wave
Гібридна	Баланс зручності та автономності	Складність впровадження	Критичні системи, енергозбереження
Мережева	Гнучкість у виборі технологій	Обмеження залежно від протоколу	Wi-Fi, Zigbee, Bluetooth

Архітектура розумного будинку є багатогранною системою, яка потребує комплексного підходу для забезпечення надійності, зручності користування та енергоефективності. Кожна з розглянутих архітектур має свої унікальні особливості, і їх вибір залежить від конкретних вимог, бюджету та очікувань користувачів. В умовах зростання попиту на автоматизацію та інтеграцію різних компонентів розумного будинку гібридні підходи стають все більш популярними, оскільки вони поєднують в собі кращі риси як централізованих, так і децентралізованих систем.

1.5 Аналіз інтерфейсів користувача та взаємодії з системами розумного будинку

Інтерфейси користувача (UI) та досвід взаємодії (UX) з системами розумного будинку є одними з ключових аспектів, що визначають успішність впровадження таких систем. Зручний і зрозумілий інтерфейс дозволяє користувачам ефективно

керувати розумними пристроями, тоді як позитивний досвід взаємодії підвищує задоволеність користувачів і сприяє їх більшій залученості. Різні підходи до проектування інтерфейсів для розумних будинків і їх вплив на зручність користування системою мають важливе значення. Далі буде наведено приклади цих підходів і розглянуто їх особливості.

1.5.1 Типи інтерфейсів користувача розумного будинку

Інтерфейси користувача для систем розумного будинку можуть бути реалізовані різними способами, включаючи мобільні додатки, веб-додатки, голосові асистенти, сенсорні панелі та навіть жести. Кожен з цих типів має свої переваги та недоліки залежно від специфіки використання та потреб користувачів.

Мобільні додатки: мобільні додатки є одним із найпоширеніших інтерфейсів для управління розумним будинком. Вони дозволяють керувати системою з будь-якої точки світу, забезпечуючи високу гнучкість. Мобільні додатки зазвичай мають інтуїтивно зрозумілий інтерфейс, що дозволяє користувачам швидко отримувати доступ до різних функцій, налаштовувати сценарії автоматизації та отримувати сповіщення про стан системи.

Голосові асистенти: голосові асистенти, такі як Samsung Bixby, Amazon Alexa, Google Assistant або Apple Siri, дозволяють користувачам взаємодіяти з СРБ за допомогою голосових команд. Цей тип інтерфейсу особливо зручний, коли користувачі зайняті або не мають доступу до смартфона. Голосове управління забезпечує високий рівень зручності, але може мати обмеження в умовах шуму або для користувачів, які не можуть чітко вимовляти команди.

Сенсорні панелі: сенсорні панелі зазвичай встановлюються на стінах і дозволяють централізовано керувати різними функціями розумного будинку. Вони забезпечують швидкий доступ до основних налаштувань, таких як освітлення, опалення та система безпеки. Перевагою сенсорних панелей є їхня постійна доступність та можливість управління без потреби в мобільному пристрої.

Жести та рухи: інтерфейси, що використовують жести або рухи, є відносно свіжим підходом до управління розумним будинком. Наприклад, можна керувати освітленням або мультимедійними пристроями за допомогою простих жестів. Цей метод є зручним у певних умовах, наприклад, коли руки користувача зайняті, але потребує спеціальних датчиків та може бути менш точним у порівнянні з іншими інтерфейсами.

1.5.2 Зручність та доступність

Зручність використання є важливим фактором для успішного впровадження систем розумного будинку. Інтерфейс повинен бути зрозумілим, інтуїтивним та адаптованим під різні потреби користувачів. Наприклад, простота навігації, логічна організація меню та наявність підказок сприяють кращому розумінню роботи системи.

Доступність інтерфейсу є особливо важливою для людей з обмеженими можливостями. Системи розумного будинку повинні підтримувати адаптивні технології, такі як екранні читачі, масштабування тексту або альтернативні методи введення, щоб забезпечити рівний доступ для всіх користувачів. Наприклад, голосові команди можуть стати незамінними для людей з порушеннями зору, а сенсорні панелі з великими клавішами – для людей з обмеженою моторикою.

1.5.3 Персоналізація та адаптація

Сучасні системи розумного будинку часто підтримують можливість персоналізації інтерфейсу під конкретного користувача. Це може включати налаштування профілів для кожного члена родини, створення індивідуальних сценаріїв автоматизації та адаптацію інтерфейсу до звичок користувача. Наприклад, один користувач може налаштувати систему так, щоб світло автоматично вмикалося при його поверненні додому, тоді як інший може віддавати перевагу управлінню освітленням вручну.

Адаптивність системи є важливою для підвищення комфорту використання. Система повинна мати можливість навчатися з поведінки користувачів і

пропонувати відповідні сценарії автоматизації. Наприклад, якщо система «помітить», що користувач щоранку вмикає кавоварку о 7:00, вона може запропонувати автоматизувати цей процес.

1.5.4 Виклики та проблеми у проектуванні інтерфейсів систем розумного будинку

Проектування інтерфейсів для СРБ має свої виклики. Один з них – це необхідність враховувати різноманітність пристроїв і платформ, з якими система має взаємодіяти. Інтерфейс повинен бути універсальним і підтримувати різні операційні системи, розміри екранів та методи введення.

Іншою проблемою є безпека. Оскільки інтерфейси користувача забезпечують доступ до керування всіма аспектами розумного будинку, вони повинні бути захищеними від несанкціонованого доступу. Це включає такі заходи, як багатофакторна автентифікація, шифрування даних та обмеження доступу для різних користувачів.

1.5.5 Вплив інтерфейсу на досвід користувача

Досвід користувача (UX) є ключовим елементом успішності систем розумного будинку. Гарний UX сприяє тому, що користувачі почуваються комфортно під час взаємодії з системою, не відчуваючи стресу чи складнощів. Інтуїтивний дизайн, швидкий відгук на команди та можливість кастомізації створюють позитивний досвід і підвищують лояльність користувачів до системи.

Наприклад, якщо користувач може легко створити сценарій автоматизації, який вмикає світло, музику і кавоварку одночасно при пробудженні, це додає комфорту і робить систему більш корисною. Водночас, якщо інтерфейс є заплутаним або має складну навігацію, користувачі можуть відчувати роздратування, що може призвести до відмови від використання системи.

Таким чином, інтерфейси користувача та досвід взаємодії з системами розумного будинку є критичними факторами, що визначають їх успішність і прийняття користувачами. Сучасні системи повинні забезпечувати різні способи

взаємодії, враховувати потреби різних користувачів і надавати можливості для персоналізації, щоб створити максимально комфортні умови для використання розумного будинку.

1.6 Взаємодія розумних будинків з іншими системами

Системи розумного будинку є частиною більш широкої екосистеми технологій Інтернету речей (IoT), і їхня здатність взаємодіяти з іншими системами є ключовою для забезпечення комплексного підходу до автоматизації та оптимізації побутових процесів. Взаємодія з іншими системами дозволяє розширити можливості розумного будинку, інтегруючи його з енергетичними системами, безпековими системами, транспортними рішеннями та навіть з іншими будівлями або інфраструктурою розумного міста. Далі будуть розглянуті основні аспекти взаємодії систем розумного будинку з іншими технологічними системами.

1.6.1 Інтеграція з енергетичними системами

Однією з важливих можливостей взаємодії розумного будинку є інтеграція з енергетичними системами. Розумний будинок може оптимізувати енергоспоживання, взаємодіючи з локальними або загальними мережами електропостачання. Наприклад, системи управління енергоспоживанням можуть автоматично змінювати режими роботи побутових приладів залежно від тарифів на електроенергію або в умовах пікових навантажень на енергомережу.

Сонячні панелі та системи зберігання енергії також можуть бути інтегровані в розумний будинок, що дозволяє зменшити залежність від загальної мережі та забезпечити енергетичну незалежність. Завдяки інтеграції з системами моніторингу погоди розумний будинок може прогнозувати рівень вироблення енергії і відповідно коригувати споживання.

1.6.2 Взаємодія розумних будинків з системами безпеки

Системи безпеки є однією з основних складових розумного будинку, і їхня інтеграція з іншими системами забезпечує додатковий рівень захисту та зручності.

Наприклад, розумний будинок може взаємодіяти з локальною системою відеоспостереження, автоматично вмикаючи освітлення при виявленні руху або надсилаючи сповіщення на смартфон власника у разі несанкціонованого доступу.

Також інтеграція з пожежною та газовою сигналізацією дозволяє вчасно реагувати на потенційні небезпеки. У випадку виявлення задимлення або витоку газу система може автоматично вимкнути відповідні прилади, включити вентиляцію і надіслати сповіщення на екстрені служби чи користувачу в залежності від налаштування [12].

1.6.3 Інтеграція з транспортними системами

Розумні будинки можуть взаємодіяти з транспортними системами, що забезпечує додатковий комфорт для користувачів. Наприклад, інтеграція з розумними гаражними воротами дозволяє автоматично відчиняти їх, коли автомобіль власника наближається до будинку. Також розумні будинки можуть отримувати інформацію від навігаційних систем, щоб підготуватися до прибуття власника: вмикати освітлення, регулювати температуру або вмикати певні прилади.

Інтеграція з громадським транспортом або сервісами таксі дозволяє автоматизувати процес замовлення транспорту. Наприклад, система може автоматично викликати таксі в певний час або надати інформацію про розклад громадського транспорту на панель управління розумного будинку.

1.6.4 Інтеграція розумних будинків з системами розумного міста

Розумний будинок є частиною більшої концепції розумного міста, і його взаємодія з міськими інфраструктурними системами відкриває нові можливості для покращення якості життя мешканців. Наприклад, розумні будинки можуть взаємодіяти з системами управління дорожнім рухом, щоб отримувати інформацію про затори або оптимальні маршрути.

Також взаємодія з міськими системами управління енергоспоживанням дозволяє розумним будинкам брати участь у програмах енергозбереження, знижуючи споживання під час пікових навантажень. Інтеграція з міськими службами також може включати отримання сповіщень про планові ремонтні

роботи, відключення води чи електрики, що дозволяє завчасно підготуватися до таких подій [13][14].

1.6.5 Виклики інтеграції СРБ з іншими системами

Інтеграція розумного будинку з іншими системами має свої виклики. Одним із основних є питання сумісності різних протоколів і стандартів. Різні виробники використовують різні стандарти зв'язку, що може ускладнювати процес інтеграції. Для вирішення цієї проблеми все частіше використовуються універсальні шлюзи або стандартизовані протоколи, такі як MQTT або Zigbee, що забезпечують сумісність між різними пристроями.

Іншим викликом є безпека даних. Взаємодія з іншими системами означає передачу великої кількості даних, і необхідно забезпечити, щоб ці дані не потрапили до рук зловмисників. Це вимагає використання надійних методів шифрування, автентифікації та інших засобів кібербезпеки.

1.6.6 Переваги інтеграції з іншими системами

Незважаючи на виклики, інтеграція розумного будинку з іншими системами надає значні переваги. Вона дозволяє створити єдину екосистему, де всі пристрої та системи працюють у синергії, забезпечуючи максимальний комфорт, ефективність та безпеку. Наприклад, взаємодія між системами опалення, вентиляції та кондиціонування повітря (ОВК) і системами моніторингу присутності дозволяє оптимізувати споживання енергії, забезпечуючи комфортну температуру тільки тоді, коли це потрібно.

Інтеграція також підвищує адаптивність системи до змінних умов. Наприклад, під час несподіваного відключення електроенергії розумний будинок може автоматично перемкнутися на резервне живлення та обмежити споживання енергії тільки до критичних приладів, таких як освітлення та система безпеки.

Таким чином, взаємодія систем розумного будинку з іншими технологічними системами є важливим аспектом, що забезпечує їхню комплексність, адаптивність і можливість створення індивідуальних рішень для користувачів. Це дозволяє підвищити ефективність використання ресурсів, забезпечити додатковий рівень безпеки та створити зручніші умови для життя.

2 РОЗРОБКА МЕТОДИКИ АДАПТИВНОГО КЕРУВАННЯ СИСТЕМАМИ РОЗУМНОГО БУДИНКУ В УМОВАХ НЕСТАБІЛЬНОГО ЕНЕРГОПОСТАЧАННЯ НА ОСНОВІ ШТУЧНОГО ІНТЕЛЕКТУ

2.1 Етапи адаптивного керування системами розумного будинку в умовах нестабільного енергопостачання

Опис послідовності дій у методиці відіграє ключову роль, адже саме він забезпечує логічність, точність і стабільність функціонування системи. Для ефективного адаптивного керування розумним будинком в умовах нестабільного енергопостачання методика включає нижченаведені етапи.

Першим кроком є підключення до сторонніх систем розумного будинку (наприклад, SmartThings, Tapo чи Yeelight) та перевірка доступності сервісів енергопостачальників (наприклад, Київ Цифровий, ДТЕК чи Укренерго). Це дозволяє керувати не тільки локальними пристроями, а й інтегрувати пристрої інших екосистем, а також отримувати дані про стан енергомережі безпосередньо від постачальників послуг.

Наступним етапом є аналіз поточної ситуації. Система перевіряє статус усіх підключених пристроїв та отримує актуальні дані енергосистеми. Це створює повну картину доступних ресурсів та технічних можливостей.

Далі виконується моделювання сценаріїв енергоспоживання на основі історичних даних та класифікації пристроїв за їх важливістю, що дозволяє одночасно прогнозувати можливі відключення електроенергії, враховуючи попередні графіки, тривалість відключень і тенденції, а також забезпечувати пріоритетне використання енергоресурсів критичними пристроями, тоді як некритичні пристрої можуть бути переведені в енергоефективний режим або вимкнені, адаптуючи сценарії роботи системи до обмежень у енергопостачанні.

Оперативна інформація збирається та аналізується з різноманітних джерел, включаючи офіційні сервіси оповіщення, дані енергопостачальників про наявність

або відсутність електроенергії, а також новини з телеграм-каналів. Кожен тип повідомлення має унікальну структуру: для телеграму це простий текст, тоді як дані від енергопостачальників подаються у форматі JSON. Залежно від типу повідомлення виконуються різні алгоритми обробки. Наприклад, текстові повідомлення з телеграму аналізуються за допомогою штучного інтелекту, який визначає їхню важливість, якщо у повідомленні виявлено критичну інформацію, автоматично створюється відкладені завдання, що виконуються у визначений час. Подібним чином створюється відкладена задача і під інформацію від енергопостачальників. Такий підхід дозволяє своєчасно виявляти як потенційні загрози для конкретної локації користувача так і стабілізаційні відключення та забезпечувати оперативну реакцію в реальному часі.

На основі зібраної інформації автоматично застосовуються відповідні відкладені задачі чи сценарії. У разі переходу в автономний режим визначається необхідна потужність залежно від доступної, після чого пристрої розподіляються за пріоритетністю. Критичні пристрої залишаються активними, пріоритетні переводяться в економний режим або вимикаються, а некритичні вимикають живлення. В залежності від ситуації для кожного пристрою вибирається найбільш ефективний протокол зв’язку, який забезпечує стабільну роботу за поточних умов.

Ця методика дозволяє інтегрувати різні системи, аналізувати поточний стан, прогнозувати загрози та ефективно керувати енергоресурсами, забезпечуючи стабільність та адаптивність розумного будинку.

2.2 Моделювання сценаріїв енергоспоживання

Одним з головних завдань методики адаптивного керування системами розумного будинку в умовах нестабільного електропостачання є моделювання сценаріїв споживання електроенергії. Це завдання передбачає аналіз поточних потреб системи розумного будинку, визначення потенційних сценаріїв енергоспоживання та їх адаптацію до умов нестабільного електропостачання. У

процесі моделювання враховуються різні фактори, такі як типи пристроїв, їх пріоритетність, критичність для життєдіяльності системи, а також можливість переходу на резервні джерела живлення.Основою для створення сценаріїв є категоризація пристроїв за їх важливістю. Наприклад, критичними пристроями вважаються ті, що забезпечують базові потреби, як-от холодильники, системи опалення, освітлення, пристрої зв'язку та безпеки. Ці пристрої мають найвищий пріоритет у випадку обмеженого доступу до енергоресурсів. Другорядні пристрої, як-от розважальна електроніка або зарядні пристрої для гаджетів, можуть бути вимкнені або переведені у енергоефективний режим.

Процес моделювання сценаріїв також включає врахування можливих перебоїв електропостачання. Для цього проводиться аналіз історичних даних про збої, прогнозування можливих ситуацій та створення відповідних алгоритмів управління. Наприклад, у випадку виявлення перебою система може автоматично переводити низькопріоритетні пристрої у сплячий режим, активувати резервні джерела живлення, такі як батареї EcoFlow Delta 2, і сповіщати користувача про вжиті заходи.

Для ефективного рішення зазначених сценаріїв можна сформулювати математичну модель енергоспоживання пристроїв, яка дозволить прогнозувати та оптимізувати розподіл енергоресурсів у різних умовах (2.1):

$$E(t) = \sum_{i=1}^{n} P_i(t) * T_i(t), \tag{2.1}$$

де $E(t)$ – загальне електроспоживання за час t; $P_i(t)$ – потужність пристрою i у час t; $T_i(t)$ – час роботи пристрою i за період t; n – кількість пристроїв.

В свою чергу маючи інформацію про споживання кожного пристрою розрахувати споживання пристрою в момент t в звичайному режимі можна за формулою (2.2):

$$P_{\text{звичайний}}(t) = U(t) * I(t), \tag{2.2}$$

де *U(t)* – напруга в момент часу *t*, у вольтах (В); *I(t)* – струм в момент часу *t*, у амперах (А).

А розрахунок еко режиму можна звести до (2.3):

$$P_{\text{еко}}(t) = k * U(t) * I(t), \quad (2.3)$$

де *k* – коефіцієнт еко-режиму ($0 < k < 1$), який залежить від пристрою.

Всі ці розрахунки для прогнозування споживання також можна відобразити у вигляді блок-схеми, яку можна побачити на рисунку 2.1.

У цьому випадку коефіцієнт k може відрізнятися залежно від пристрою, адже режим "еко" для кожного з них унікальний: для обігрівача він може становити 0.3 (30%), тоді як для кондиціонера – лише 0.7 (70%).

Маючи рівняння, які допомагають визначити як загальне споживання пристроїв за час *t* так і в конкретний момент тепер можна вивести формулу для розрахунку споживання в момент відключення світла (2.4):

$$P_{\text{відключення}} = \sum_{i \in A} P_{i,\text{еко}} + \sum_{j \in B} P_{j,\text{звичайний}}, \quad (2.4)$$

де $P_{відключення}$ – загальне споживання всіх пристроїв під час відключення світла; *A* – множина пристроїв, які переводяться в еко-режим ($P_{i,еко} = k_i * U_i * I_i$, $0 < k_i < 1$ для $i \in A$); *B* – множина пристроїв, які продовжують працювати у звичайному режимі ($P_{j,звичайний} = U_j * I_j$ для $k \in B$).

Рис. 2.1 Блок-схема процесу прогнозування споживання

2.3 Алгоритми оптимізації енергоспоживання

Оптимізація енергоспоживання є ключовим аспектом для забезпечення автономності та ефективності систем розумного будинку, особливо в умовах нестабільного енергопостачання. Застосування адаптивних алгоритмів дозволяє забезпечити раціональний розподіл ресурсів, знизити енергоспоживання некритичних пристроїв і пріоритизувати роботу важливих систем. Ці алгоритми

базуються на використанні інструментів аналізу для прийняття оптимальних рішень.

Перед початком оптимізації та прогнозувань необхідно розставити пріоритет для кожного конкретного девайсу, це можна зобразити за допомогою такої формули (2.5):

$$Pr_i = w_i * C_i, \quad (2.5)$$

де w_i – вага важливості (наприклад, 3 для критично-важливих систем, 2 для важливих та 1 для некритичних); *Ci* – критичність системи (0-1, де 1 найвища важливість).

Для більш точної оптимізації необхідно навчитись передбачати споживання. Це досягається за рахунок використання історичних даних, які зберігає система. Математично процес прогнозування можна представити наступною формулою (2.6):

$$\hat{y}(t) = f(y(t-1), y(t-2), \dots, y(t-n)), \quad (2.6)$$

де *y^(t)* – прогнозоване споживання енергії в момент часу *t*; *y(t−1),y(t−2),…,y(t−n)* – споживання енергії за попередні n періодів; *f* — функція, що визначає залежність між минулими значеннями та майбутнім прогнозом.

Одним із популярних методів для реалізації функції *f* є модель ARIMA [14] (AutoRegressive Integrated Moving Average), яка базується на лінійних залежностях між попередніми значеннями ряду та залишковими похибками. ARIMA можна представити як (2.7):

$$y_t = \phi_1 y_{t-1} + \phi_2 y_{t-2} + \cdots + \phi_p y_{t-p} + \theta_{1\epsilon t-1} + \theta_{2\epsilon t-2} + \cdots + \theta_q \epsilon_{t-q} + \epsilon_t, \quad (2.7)$$

де y_t – споживання енергії в момент часу t; $\phi_1,\phi_2,\dots,\phi_p$ – коефіцієнти авторегресії; $\theta_1,\theta_2,\dots,\theta_q$ – коефіцієнти ковзного середнього; ϵ_t – залишкова похибка в момент часу t.

Дана модель дає можливість робити прогнозування на основі лінійних трендів і сезонних закономірностей, що робить її ефективною для короткострокового прогнозування.

Тож оптимізація енергоспоживання може бути формалізована через мінімізацію функції витрати (2.8):

$$min\ E(t) = \sum_{i=1}^{n} P_i(t) * T_i(t) + \lambda * \sum_{j=1}^{m} S_j^2, \quad (2.8)$$

де $E(t)$ – загальні енергетичні витрати за час t; $P_i(t)$ – споживана потужність і-го пристрою; $T_i(t)$ – час роботи і-го пристрою; S_j – відхилення від заданих умов комфорту (наприклад, температури); λ – ваговий коефіцієнт, який визначає компроміс між енергоефективністю та комфортом.

Ця формула дозволяє балансувати між зниженням енергоспоживання та збереженням комфорту для мешканців, адаптуючи параметри системи до змін у доступності енергоресурсів.

Згідно опису методики, можна сформувати деяку послідовність дій в блок-схему, яка показана на рисунку 2.2.

Рис. 2.2 Блок-схема оптимізації енергоспоживання

2.4 Інтеграція штучного інтелекту для адаптивного керування

Інтеграція штучного інтелекту (ШІ) у системи розумного будинку є важливим кроком до підвищення ефективності та автономності керування в умовах нестабільного енергопостачання. Завдяки аналітичним можливостям ШІ система отримує здатність адаптувати свою роботу залежно від зовнішніх умов та потреб користувача.

Методика впровадження ШІ базується на зборі даних із різних джерел, таких як сервіси енергетичних операторів (Укренерго, ДТЕК), локальні сенсори та сторонні інформаційні канали, включаючи телеграм-канали. Зібрана інформація аналізується для визначення потенційних загроз або змін у ситуації. Для цього використовуються алгоритми обробки даних, які дозволяють виявляти ключові події та локації користувача, наприклад, загрози, описані повідомленнями типу: "Київ. Крилаті ракети. 10 хв".

Система адаптивного керування розроблена для автоматизації прийняття рішень. Основним підходом є обчислення часу активації сценаріїв, що визначається за формулою (2.9):

$$T_{action} = T_{message} + \Delta T_{analysis}, \tag{2.9}$$

де $T_{message}$ – час отримання повідомлення; $\Delta T_{analysis}$ – затримка на обробку повідомлень і формування рішення.

Для визначення важливості повідомлення можна сформувати таку формулу (2.10):

$$I_{message} = W_c * C + W_t * \frac{1}{T_{response}}, \tag{2.10}$$

де $I_{message}$ – рівень важливості повідомлення; W_c *і* W_t – вагові коефіцієнти для контексту © і часу реакції $(T_{response})$;

Чим вищий рівень важливості ($I_{message}$), тим швидше система має активувати відповідний сценарій, наприклад «Danger».

На основі аналізу даних про стан ресурсів формується оптимальний розподіл енергоспоживання (2.11):

$$Pd_{optimal} = \arg\min_{P}(\sum_{i=1}^{n} C_i * U_i(P)), \quad (2.11)$$

де Pd — набір параметрів для пристроїв; C_i — вартість ресурсу і; $U_i(P)$ — функція корисності для параметра і.

Саму функція корисності можна вивести наступною формулою (2.12):

$$U_i(P) = P * t, \quad (2.12)$$

де P – потужність пристрою; t = час роботи.

Для освітлення функція може враховувати яскравість (B) та комфорт для користувачів (K), то формула буде виглядати так (2.13):

$$U_i(P) = K * B(P), \quad (2.13)$$

де $B(P)$ – функція залежності яскравості від споживання потужності.

Тож ця формула не є універсальною, вона формується під кожну потребу по своєму.

Методика включає можливість адаптувати сценарії залежно від прогнозованих подій. Наприклад, у разі виявлення загрози система автоматично створюю відкладене завдання та згодом запускає сценарій "Danger", який переводить пристрої в енергоефективний режим та підключає резервне живлення, це можна побачити на рисунку 2.3.

Прогнозування можливих ризиків дозволить системі не лише реагувати на події, але й формувати проактивний підхід до управління. Це забезпечує стабільність роботи, ефективне використання ресурсів і високий рівень безпеки мешканців.

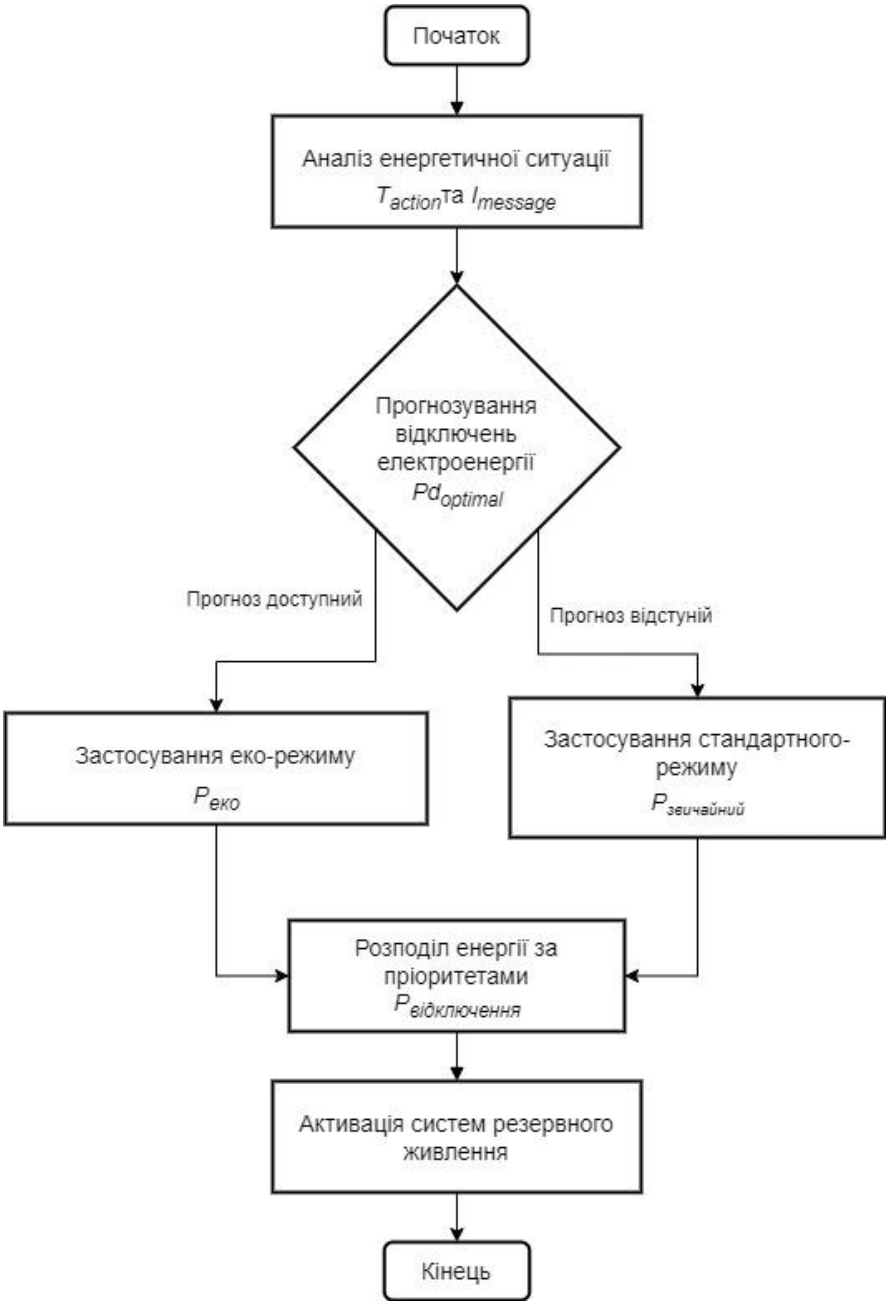


Рис. 2.3 Блок-схема аналізу споживання електроенергії

2.5 Протоколи зв'язку та їх адаптація для нестабільних умов енергопостачання

Взаємодія між пристроями у системах розумного будинку здійснюється за допомогою різних протоколів зв'язку. Серед найпоширеніших протоколів як було описано вище можна виділити WiFi, Zigbee, Z-Wave, MQTT, Bluetooth та інші. Кожен із цих протоколів має свої особливості, переваги та недоліки, які впливають

на їх використання в конкретних умовах. У системах розумного будинку особливо важливо забезпечити стабільну роботу навіть у випадках низької якості зв'язку або нестабільного енергопостачання.

WiFi є одним із найпоширеніших протоколів завдяки високій швидкості передачі даних та широкому охопленню. Він ідеально підходить для пристроїв, що передають великий обсяг інформації, таких як камери спостереження або мультимедійні пристрої. Однак високий рівень енергоспоживання робить його менш ефективним для пристроїв, що живляться від батарейок.

Протоколи Zigbee та Z-Wave, навпаки, розроблені для низькоенергетичних пристроїв, таких як датчики температури, вологості чи руху. Вони використовують сітчасту мережу, яка дозволяє кожному пристрою виступати вузлом, що ретранслює сигнал, забезпечуючи більшу дальність і стабільність зв'язку.

Bluetooth, особливо його версія BLE (Bluetooth Low Energy), забезпечує короткодистанційну взаємодію з низьким енергоспоживанням. Цей протокол зазвичай використовується для розумних замків, освітлення або портативних пристроїв.

MQTT (Message Queuing Telemetry Transport) – це легковаговий протокол передачі даних, що використовує архітектуру "публікація-підписка". Завдяки своїй ефективності він широко застосовується для передачі невеликих обсягів даних навіть за обмежених мережевих ресурсів.

Для забезпечення стабільної роботи системи розумного будинку у складних умовах розроблено механізми адаптації, які дозволяють автоматично змінювати протокол зв'язку залежно від ситуації. Основними факторами, що впливають на вибір протоколу, є такі моменти:

- стан мережі: рівень сигналу поточного протоколу;
- енергоспоживання пристрою: пристрої на батарейках переключаються на протоколи з низьким енергоспоживанням;
- тип передаваних даних: для великого обсягу даних (наприклад, відео) обираються високошвидкісні протоколи.

Тож процес вибору оптимального протоколу можна описати формулою (2.14):

$$Proto = \arg \max_{p \in \{WiFi, Zigbee, Z-Wave, MQTT\}} \left[W_p * C_p * \frac{1}{E_p} \right], \quad (2.14)$$

де *Proto* – обраний протокол; W_p – ваговий коефіцієнт стабільності протоколу; C_p – здатність протоколу забезпечувати зв'язок; E_p – енергоспоживання протоколу.

Як приклад, якщо рівень сигналу WiFi (S_p) падає нижче порогового значення (S_{min}), пристрої автоматично перемикаються на Zigbee через локальний хаб (2.15):

$$Proto = \begin{cases} Zigbee, & S_p < S_{min} \\ WiFi, & \text{інакше} \end{cases}. \quad (2.15)$$

Розглянемо сценарій, коли в будинку відбувається раптове відключення електропостачання. У цьому випадку:

- камери спостереження, що потребують високої пропускної здатності, продовжують працювати через WiFi, живлячись від резервного акумулятора;
- датчики руху і температури автоматично переходять на протокол Zigbee, який забезпечує енергоефективну передачу даних через сітчасту мережу;
- система освітлення у критичних зонах використовує Bluetooth для локального управління, забезпечуючи доступність навіть за відсутності інтернету.

Якщо система виявляє критичне падіння рівня зв'язку у всіх каналах, вона може перейти в автономний режим, використовуючи локальний сервер та протокол MQTT для координації пристроїв.

Адаптація протоколів зв'язку виконує важливу функцію у забезпеченні стабільної роботи систем розумного будинку в умовах змінної якості зв'язку. Вона дозволяє зберігати функціональність навіть тоді, коли сигнал слабкий або нестабільний, забезпечуючи надійний обмін даними між пристроями. Це особливо важливо для таких компонентів системи, як датчики руху, температури або освітлення, які повинні працювати безперебійно незалежно від стану мережі. Кожен протокол має різні можливості до адаптації та роботі в екстремальних умовах, яка можна побачити на табличці 2.1:

Таблиця 2.1

Можливості та проблеми протоколів

Протокол	Адаптаційні можливості	Проблеми
WiFi	Підтримка еко режимів	Високе споживання електрики
Bluetooth	Використання режиму low energy	Обмежена дальність зв’язку
Zigbee	Низьке електроспоживання, підтримка сітчастої мережі	Висока чутливість до перешкод
Z-Wave	Стійкість до перешкод	Висока вартість пристроїв
Thread	Низьке споживання, резервні маршрути	Спец обладнання
Matter	Автономна робота	Мала підтримка серед пристроїв

Водночас ефективне використання протоколів зв'язку та їх адаптація до нестабільних умов відіграють вирішальну роль у забезпеченні надійності, зручності та енергоефективності сучасних систем розумного будинку. Завдяки цьому забезпечується безперебійна робота всіх елементів системи навіть за умов зовнішніх перешкод або нестабільного енергопостачання.

2.6 Вимоги до реалізації засобів керування системами розумного будинку в умовах нестабільного енергопостачання за розробленою методикою

Запропонована методика визначає базові вимоги до засобів керування системами розумного будинку в умовах нестабільного енергопостачання, спрямовані на забезпечення їх ефективності, адаптивності та надійності:

- адаптивність до нестабільного енергопостачання: підсистема керування повинна змінювати пріоритети роботи пристроїв, базуючись на прогнозах відключення електроенергії, та зменшувати енергоспоживання, переходячи на менш енергоємні протоколи;
- підтримка декількох комунікаційних протоколів: пристрої мають підтримувати кілька протоколів, автоматично обираючи найменш енергозатратний залежно від умов;
- інтеграція з джерелами даних про відключення енергії: підсистема інтеграції повинна отримувати інформацію про відключення через API, від загального прогнозу до точних графіків;
- прогнозування відключень на основі штучного інтелекту: використання алгоритмів для аналізу даних, прогнозування відключень і автоматичної оптимізації роботи пристроїв;
- миттєва реакція та гнучкість керування пристроями: забезпечення реакції в реальному часі та адаптивне перемикання режимів роботи після отримання попереджень;
- енергоефективність: оптимізація роботи пристроїв для ефективності в умовах обмеженого енергопостачання і мінімізація споживання через вибір ефективних протоколів;
- безпека та надійність: захист комунікацій від зовнішніх загроз і забезпечення надійності системи в умовах нестабільного енергопостачання;
- простота інтеграції та масштабованість: легка інтеграція нових пристроїв і підтримка нових протоколів, масштабованість для роботи з великою кількістю пристроїв.

3 ПРАКТИЧНА РЕАЛІЗАЦІЯ ТА ЕКСПЕРИМЕНТАЛЬНІ ДОСЛІДЖЕННЯ

3.1 Програмні засоби реалізації

Для реалізації подібних систем та модулів необхідно використовувати високопродуктивні МП, мови які можна запустити на кожній платформі та ОС без необхідності встановлення додаткового програмного забезпечення.

3.1.1 .NET

.NET – це універсальна програмна платформа, створена компанією Microsoft, яка забезпечує середовище для розробки, запуску та підтримки сучасних додатків. Відмінною рисою .NET є її кроссплатформеність, тобто можливість створення додатків, що працюють на різних операційних системах, таких як Windows, Linux та macOS, Android, iOS та Tyzen [15]. Основу платформи становить загальна середа виконання (Common Language Runtime, CLR), яка надає послуги для управління виконанням програм – такі як збірка сміття (автоматичне звільнення пам’яті), безпека, підтримка багатопотоковості, а також динамічна компіляція коду (див. Рис. 2.1).

.NET об’єднує кілька мов програмування, серед яких C#, F#, Visual Basic, J# та багато інших, що дозволяє розробникам вибирати інструменти, які найкраще підходять для конкретного завдання. Крім того, платформа підтримує широкий спектр бібліотек для розробки веб-додатків, мобільних додатків, серверних рішень та навіть ігор. Ці бібліотеки надають готові рішення для роботи з мережею, доступу до баз даних, обробки даних, графічного інтерфейсу користувача тощо.

.NET також славиться своєю модульною архітектурою, що дозволяє розробникам використовувати лише ті компоненти, які необхідні для конкретного проєкту, що знижує навантаження на систему і спрощує підтримку коду. З введенням .NET Core, який зараз об'єднаний у .NET 5 і вище, розробники отримали

можливість використовувати відкритий вихідний код (open-source), що значно збільшило популярність платформи серед міжнародної спільноти розробників.

3.1.2 ASP.NET Core

ASP.NET – це веб-фреймворк, що входить до складу платформи .NET і призначений для створення динамічних веб-додатків і сервісів. Це інструмент, який надає розробникам можливість швидко й ефективно створювати веб-додатки різного рівня складності – від простих сторінок до складних корпоративних систем. ASP.NET підтримує різні архітектурні підходи, що робить його надзвичайно гнучким для вирішення різних завдань.

Однією з найважливіших особливостей ASP.NET є підтримка кількох моделей програмування для розробки веб-додатків. Зокрема, це WebForms, MVC (Model-View-Controller), Web API та Blazor. Кожна з цих моделей має свої переваги та підходить для певних видів проєктів. Web Forms дозволяє швидко створювати додатки за допомогою подібного до Windows Forms підходу, а MVC надає більше контролю над HTML-кодом та структуруванням додатку. Blazor, у свою чергу, дозволяє писати інтерактивні веб-додатки на C# замість JavaScript, що спрощує роботу для тих, хто вже знайомий із мовою C#.

ASP.NET Core, сучасна версія фреймворку, відзначається кросплатформенністю та високою продуктивністю. Вона дозволяє розробникам запускати додатки на різних операційних системах, а також розгортати їх в будь-якому середовищі, включаючи хмарні сервіси, сервери Linux або контейнерні середовища, такі як Docker [16]. Ця версія також підтримує модульність і дозволяє розробникам використовувати лише ті компоненти, які потрібні для конкретного додатка, що значно покращує ефективність і продуктивність.

ASP.NET також має інтеграцію з іншими елементами екосистеми Microsoft, такими як Entity Framework для роботи з базами даних, Azure для хмарних сервісів і IdentityServer для управління автентифікацією та авторизацією. Це робить ASP.NET чудовим вибором для побудови повноцінних рішень, які можуть

масштабуватися та бути безпечними, водночас спрощуючи роботу розробника завдяки використанню зручних і звичних інструментів.

3.1.3 ML.NET

ML.NET – це бібліотека для машинного навчання, яка є частиною платформи .NET і дозволяє розробникам створювати, тренувати та інтегрувати моделі машинного навчання без необхідності переходу до інших екосистем чи мов програмування. Основна ідея ML.NET полягає в забезпеченні розробникам .NET можливість впровадження функціональності машинного навчання у свої додатки, використовуючи знайомі їм інструменти та мови, зокрема C# та F# (див. Рис. 2.3).

ML.NET надає широкий набір можливостей для вирішення різних завдань, таких як класифікація, регресія, кластеризація, аналіз тексту, рекомендаційні системи та виявлення аномалій. Це дозволяє розробляти рішення, які можуть прогнозувати майбутні значення, визначати категорії, виявляти тенденції або аналізувати текстові дані. Однією з ключових переваг є можливість автоматизованого навчання за допомогою функції AutoML, що дозволяє розробникам швидко знаходити найкращі моделі та параметри без глибоких знань у галузі машинного навчання.

ML.NET використовує концепцію "pipelines" для створення моделей машинного навчання, що полегшує структурування та налагодження процесу навчання. Пайплайн складається з різних етапів обробки даних, таких як нормалізація, кодування категоріальних змінних, та алгоритму навчання моделі. Це дозволяє забезпечити гнучкість та зручність у побудові різноманітних робочих процесів.

Для тих, у кого є необхідність використовувати вже готові рішення, ML.NET також інтегрується з ONNX [17] (Open Neural Network Exchange), що дає можливість використовувати попередньо натреновані моделі, створені за допомогою інших фреймворків, таких як TensorFlow [18] або PyTorch [19]. Це

значно розширює можливості платформи, дозволяючи використовувати складні глибокі нейронні мережі без необхідності їх створення з самого початку.

ML.NET добре інтегрується з іншими інструментами та екосистемами .NET, що дозволяє легко вбудовувати рішення на основі машинного навчання в існуючі додатки. Завдяки цьому розробники можуть швидко створювати інтелектуальні сервіси, наприклад, рекомендаційні системи для онлайн-магазинів, прогнозні моделі для бізнес-аналітики або системи аналізу користувацької поведінки.

Таким чином, ML.NET пропонує зручний спосіб використання машинного навчання у додатках, що розробляються на .NET, роблячи технології штучного інтелекту доступнішими для розробників, які не мають високої кваліфікації в цій галузі.

3.1.4 Супутні бібліотеки

Для забезпечення взаємодії з різними пристроями через різні протоколи необхідно налаштувати ефективну систему комунікації. Це завдання можна вирішити двома шляхами: розробити власні бібліотеки або скористатися вже існуючими готовими рішеннями. Розробка з нуля дає повний контроль над реалізацією, але може супроводжуватись значними труднощами, такими як виявлення помилок, необхідність глибокого тестування та відсутність підтримки. Враховуючи ризики та обмежений час, було прийнято рішення інтегрувати перевірені наявні бібліотеки.

Для інтеграції з пристроями, що використовують Zigbee, було обрано бібліотеку ZigbeeNET. Ця бібліотека забезпечує повну реалізацію специфікацій Zigbee, включаючи підтримку кластерів, управління мережевою топологією, а також розширену підтримку бекапів та відновлення даних. Вона дозволяє забезпечити стабільність роботи навіть після перезавантаження, підтримує роботу з кількома десятками пристроїв одночасно із можливістю класифікації та інтегрує пристрої від різних виробників.

Для роботи з пристроями, що підтримують протокол Z-Wave, використовується бібліотека zwave-lib-dotnet. Вона забезпечує зручний інтерфейс для управління пристроями та дозволяє контролювати підключені пристрої через API з мінімальною затримкою. Ця бібліотека також об'єднує пристрої від різних виробників під один хаб, що полегшує їх інтеграцію, і має детальну документацію, яка допомагає швидко впровадити рішення у проєкт.

Для роботи з сучасними протоколами, такими як WiFi, Bluetooth, Thread та Matter, було обрано бібліотеку dotnet-matter. Вона пропонує універсальний API для управління пристроями, забезпечуючи високий рівень стабільності та безпеки. Бібліотека реалізує специфікації Matter, що є стандартом для взаємодії IoT-пристроїв, дозволяє створювати налаштовувані сценарії для управління та синхронізації пристроїв, а також забезпечує стабільність мережі завдяки функціям відновлення зв'язку та мінімізації переривань.

3.2 Архітектура системи

Загальний вигляд архітектури системи представлений на рисунку 3.1. Схема ілюструє основні компоненти системи керування розумним будинком в умовах нестабільного енергопостачання та їх взаємодію.

Далі наведено опис основних компонент системи та порядок їх функціонування і взаємодії.

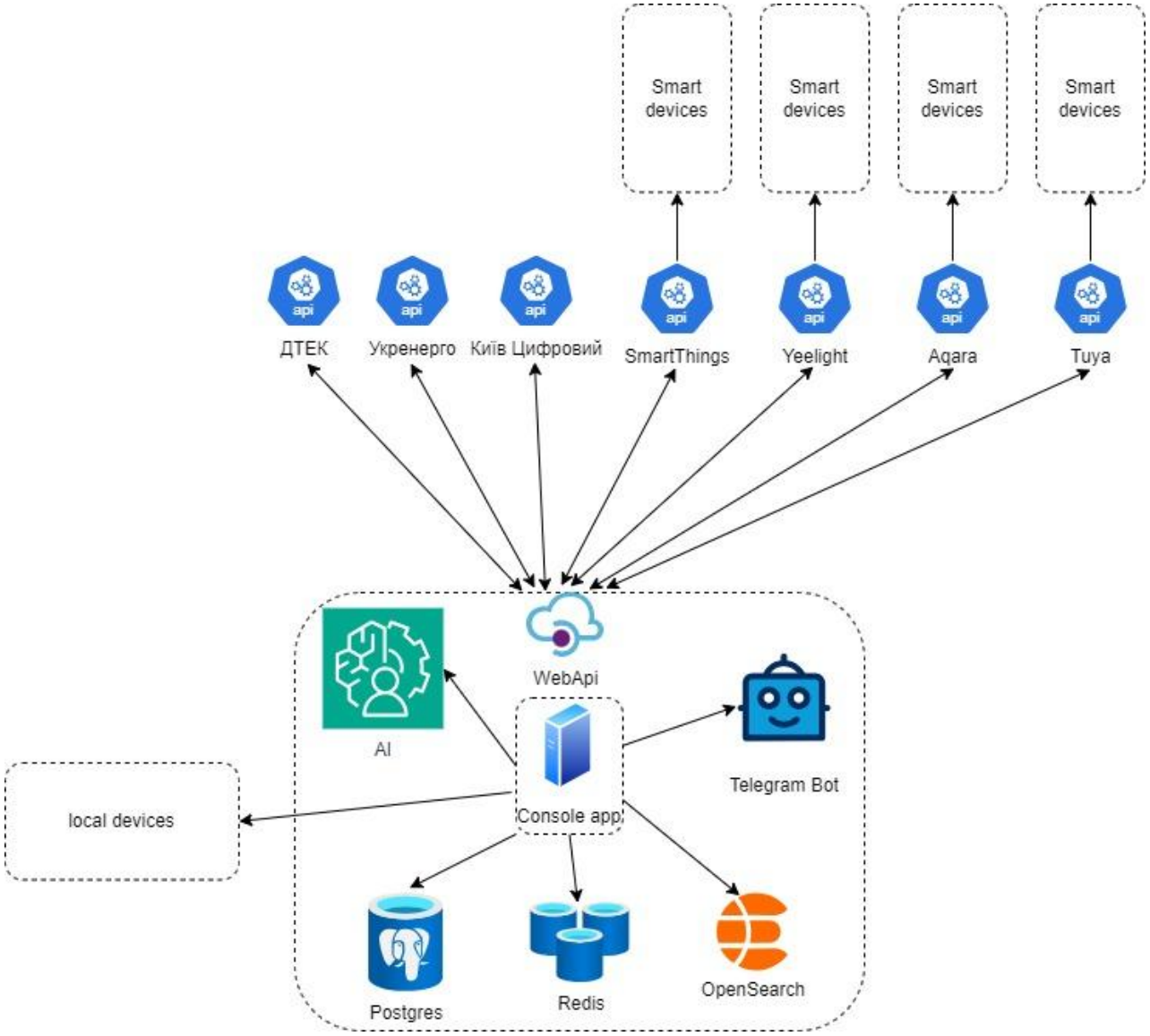


Рис. 3.1. Архітектура системи керування розумним будинком в умовах нестабільного енергопостачання

3.2.1 Консольний застосунок

Консольний застосунок (Core) виступає ядром системи керування розумним будинком, яке працює постійно та забезпечує взаємодію з різними пристроями в домі, він виконує кілька ключових завдань:

Застосунок повинен забезпечувати підключення до пристроїв за допомогою різних протоколів, таких як WiFi, Bluetooth, Z-wave, Zigbee та Matter. Кожен із цих протоколів має свої особливості, тому для їх підтримки необхідно використовувати спеціальні бібліотеки або SDK. Наприклад, Azure IoT SDK може

слугувати для роботи з WiFi-пристроями, тоді як спеціалізовані бібліотеки забезпечують надійне підключення до мереж Z-wave і Zigbee.

Реалізація функції автоматичного виявлення пристроїв дозволить системі самостійно знаходити нові пристрої в мережі та підключатися до них, що робить систему більш гнучкою та простою в користуванні.

Консольний застосунок виконує функції хаба, об'єднуючи всі пристрої системи. Він отримує інформацію про їхній стан і може надсилати команди для керування ними.

Використання патерну "спостерігач" (Observer) дозволяє забезпечити постійне оновлення стану пристроїв. Наприклад, якщо якийсь з пристроїв змінює свій стан, система одразу буде про це проінформована і зможе відповідно реагувати.

Для обробки даних від пристроїв у реальному часі консольний застосунок може використовувати чергу повідомлень, наприклад RabbitMQ [20] або MQTT, для передачі інформації від пристроїв до системи.

Також можна додати локальну базу даних для тимчасового зберігання даних, що зменшить навантаження на основну базу і дозволить зберігати інформацію в разі перебоїв із зв'язком.

Система повинна забезпечувати оперативну передачу інформації про відключення електроенергії до модуля штучного інтелекту, що дозволить використовувати ці дані для подальшого навчання і підвищення адаптивності в умовах нестабільного енергопостачання. Використання стрімінгової передачі даних до модуля AI забезпечить безперервне навчання і вдосконалення моделей прогнозування.

3.2.2 Web API

Web API виступає інтерфейсом для взаємодії з іншими системами та сервісами, що дозволяє отримувати інформацію та оновлювати стан системи.

Web API забезпечує прийом інформації від зовнішніх сервісів, таких як прогнози погоди, дані про стан енергопостачання, повідомлення від "Укренерго"

про можливі відключення, а також локальні застосунки, які попереджають про конкретні терміни відключення електроенергії з точністю до години чи 15 хвилин. Це дозволяє системі враховувати зовнішні умови та покращувати свою адаптивність.

Дані, отримані через API, можуть бути використанні для прийняття рішень в реальному часі, наприклад, у випадку очікуваного відключення електрики система може заздалегідь підготуватися до роботи в автономному режимі створивши «роботу» (задачу) на конкретну годину, яка в свою чергу автоматично запуститься.

Для зберігання інформації про стан пристроїв, історію відключень електроенергії, налаштування користувачів та інші дані використовується реляційна база даних PostgreSQL.

Використання пошукового ядра OpenSearch [21], буде корисним кроком для зберігання неструктурованих даних, таких як журнали подій або оперативна та найголовніше актуальна інформація про стан енергосистеми, які можуть мати різні формати.

Для захисту системи реалізовано автентифікацію та авторизацію користувачів. Використання JWT-токенів забезпечує захищений доступ до API та запобігає несанкціонованим запитам.

Обробка та використання даних виконується на наступних етапах:

Дані від пристроїв: Консольний застосунок отримує інформацію про стан пристроїв, обробляє її та зберігає у базі даних. Це дозволяє системі завжди мати актуальну інформацію про стан будинку.

Зовнішні дані: Web API отримує дані від зовнішніх джерел і зберігає їх у базі даних для подальшого використання. Ці дані можуть бути використані для прийняття рішень у режимі реального часу або для адаптації системи до зовнішніх умов.

Навчання AI: Дані про відключення світла або інші важливі події зберігаються у базі даних та передаються до модуля AI для навчання. Це дозволяє

моделі штучного інтелекту вчитися на основі реальних даних та покращувати свої прогнози і реакції на події.

Технологічні інструменти, які були використані під час розробки:

- .NET Core для створення консольного застосунку та Web API;
- Entity Framework Core для роботи з базою даних, забезпечуючи легкість інтеграції та можливість використання LINQ для запитів;
- Azure IoT SDK для роботи з пристроями, підключеними через WiFi, а також спеціальні бібліотеки для інших протоколів (наприклад, Z-wave чи Zigbee);
- MQTT для обробки повідомлень в реальному часі, що забезпечить стабільну передачу даних від пристроїв у систему;
- TensorFlow.NET для інтеграції моделі штучного інтелекту, що будуть керувати системою на основі отриманих даних.

3.3 Навчання штучного інтелекту графікам відключення

Для прогнозування графіків відключень електроенергії, яке базується на використанні технології ML.NET, а враховуючи інтеграцію з WebApi та Core у вигляді консольного застосунку, цей підхід забезпечує високу інтеграцію системи з існуючими компонентами розумного будинку та ефективну адаптацію до умов нестабільного енергопостачання і відбувається він за наступним алгоритмом:

Найголовнішим пунктом є збір і підготовка даних, тому для ефективного навчання штучного інтелекту були визначені ключові джерела даних, а саме:

- графіки планових відключень, отримані через API від постачальника електроенергії. Ці дані містять часові рамки відключень для конкретних регіонів;
- історичні дані про відключення, які зберігаються в базі даних системи. Ці дані формуються на основі попередніх запитів до API та реальних подій;
- оперативні попередження про аварійні відключення, які надходять через API.

Перед завантаженням у модель, ці дані попередньо обробляються в консольному застосунку, який:

- перевіряє дані на повноту, цілісність і відсутність аномалій;
- виконує нормалізацію часових відміток і узгодження з іншими даними, наприклад, погодними умовами, піковими періодами споживання тощо;
- формує вибірку для навчання у вигляді набору за день, тиждень, квартал чи рік;
- навчання моделі за допомогою ML.NET.

Для навчання обрано платформу ML.NET, яка дозволяє використовувати алгоритми машинного навчання для задач прогнозування, тож є основні етапи:

- для прогнозування графіків відключень обирається алгоритм Time Series Forecasting;
- виконується “розбиття даних” на тренувальний набір, який використовується для побудови моделі та тестовий набір, який в свою чергу використовується для перевірки точності моделі;
- консольний застосунок завантажує оброблені дані в модель ML.NET, налаштовує її параметри та виконує навчання;
- після навчання модель проходить валідацію на тестових даних. В даному випадку використовується метрика для оцінки помилок під назвою RMSE (Root Mean Squared Error).

Готова модель інтегрується у серверну частину системи, а саме у консольний застосунок, який регулярно оновлює модель на нових даних та виконує прогнозування про нові відключення та WebApi, який надає зовнішнім сервісам доступ до прогнозів чи отримує нові дані про відключення через інтеграцію з API постачальника електроенергії.

Як приклад прогнозування, можна застосувати модель ARIMA для оцінки споживання енергії на неділю, використовуючи дані з понеділка по суботу:

y_{t-1} = 120 кВт/год, y_{t-2} = 125 кВт/год, y_{t-3} = 130 кВт/год, y_{t-4} = 128 кВт/год,

y_{t-5} = 127 кВт/год, y_{t-6} = 124 кВт/год

Коефіцієнти ARIMA:

$\phi_1 = 0.5$, $\phi_2 = 0.3$, $\phi_3 = 0.2$ (авторегресія)

$\theta_1 = 0.4$, $\theta_2 = 0.2$ (ковзне середнє)

Залишкова похибка:

$\epsilon_{t-1} = -1$, $\epsilon_{t-2} = 2$,

Використовуючи формулу методом ARIMA отримуємо:

$$y_t = (0.5 * 120) + (0.3 * 125) + (0.2 * 130) + (0.4 * -1) + (0.2 * 2) + 0$$

З розрахунків за 3 дні та використанням похібки на неділю маємо наступне споживання:

$$y_t = 123.5 \text{ кВт/год}$$

Також система передбачає регулярне навчання моделі для адаптації до змінних умов:

- актуалізація даних через нові графіки відключень, які додаються в базу даних і враховуються для навчання;
- у разі непередбачених відключень система аналізує їх причини та включає ці дані в навчальний процес.

Після закінчення інтеграції моделі, вона дозволяє прогнозувати майбутні відключення з точністю до кількох хвилин, забезпечуючи користувачів завчасним попередженням, а також оптимізувати роботу розумного будинку заздалегідь вимикнувши або перемкнувши в економний режим непотрібні пристрої та змінити тип живлення перейшовши на альтернативні джерела, такі як батареї чи генератори.

3.4 Навчання штучного інтелекту читання новин по можливу загрозу

Реалізація можливості аналізу новинних повідомлень про загрози енергетичній інфраструктурі є ключовою складовою системи адаптивного керування розумним будинком в наш час, а тим паче в Україні. Це дозволяє вчасно реагувати на потенційні ризики, зокрема ракетні атаки чи інші надзвичайні ситуації, та забезпечувати безперебійне функціонування важливих систем у будинку.

Збір даних про потенційні загрози може відбуватися одночасно з декількох джерел даних, а саме телеграм каналів офіційних установ чи місцевих адміністрацій, новинних каналів, які оперативно повідомляють про загрози та за потреби можуть деталізувати. Агрегатори місцевих даних чи сервіси оповіщень, до них можна віднести сервіс «Тривога», яка повідомляє про тип тривоги в конкретному регіоні чи «Київ Цифровий», який надає дані по місту разом з даними про графікі відключень чи прямими попередженнями про вимкнення\увімкнення світла за конкретною адресою. Також можна згадати публічні API від «Укренерго» та компанії «ДТЕК», які можна використовувати як альтернативу «Київ Цифровому».

Інтеграція з Telegram реалізується через бібліотеку під назвою WTelegramClient, яка забезпечують отримання повідомлень у реальному часі. Консольний застосунок регулярно аналізує нові повідомлення, шукаючи ключові слова чи фрази, які вказують на загрозу (наприклад, "ракета", "атака на ТЕС", "знеструмлення регіону", "Київ, ракета на підльоті. 2 хв").

Обробка текстових повідомлень здійснюється етапами, а саме – попередня обробка включає в себе виділення ключових слів і фраз (за допомогою бібліотеки ML.NET Text Classification), фільтрацію зайвих даних, які наприклад не включають інформацію про регіон користувача чи якусь рекламу. Класифікація повідомлень дає визначити, чи є повідомлення важливим для користувача. До класифікації входять дані про тип загрози (ракетна/дронова атака, аварійне відключення, технічні несправності) та локація (віддалені регіони чи близьке до користувача місце).

Для аналізу загроз враховується місцезнаходження користувача, яке можна отримати через базу даних або інтеграцію з GPS-пристроями. Якщо загроза стосується найближчого регіону, повідомлення отримує вищий пріоритет.

Також важливим пунктом є прогнозування впливу загроз, яким якраз може займатися штучний інтелект на основі отриманих даних, а саме небезпека яка аналізується на базі технічних характеристик ракет, наприклад, дальність, швидкість та потенційна шкода. Ці дані допомагають оцінити ймовірність збиттям

ракети ППО або влучання в критичну інфраструктуру. Дані про температуру є важливим аспектом особливо враховуючи аномальну спеку влітку чи аномальну зиму, яка присутня останні 2 роки, таким чином виконується деякі сценарії:

- при похолоданні пріоритет надається підтримці роботи системи опалення;
- у спеку система зберігає роботу охолоджувальних пристроїв.

Тож, враховуючи описаний функціонал СРБ може виконувати дії з пристроями розумного будинку:

- перехід із центрального енергопостачання на акумулятори чи генератори;
- пріоритизація роботи важливих пристроїв, таких як опалення, холодильники чи системи освітлення;
- вимкнення другорядних пристроїв;
- переключення системи на режим економії;
- завчасне сповіщення про очікувану загрозу через телеграм в чат сім'ї;
- інструкції щодо дій у надзвичайній ситуації (за необхідністю).

Для адаптації до змінних умов штучний інтелект проходить регулярне навчання, воно полягає у поповненні баз даних про нові типи загроз чи сценарії, які додаються до системи для вдосконалення аналізу (це може бути використання нових ракет «арєшнік», проти яких українська ППО має невеликий шанс на перехоплення).

Прикладом навчання моделі є зібрана статистика за 2 останніх місяці її роботи, представлена в таблиці 3.1.

Таблиця 3.1

Точність оцінювання загрозі моделлю ШІ

Тиждень	Точність розпізнавання загрози	Прогнозовані/Реальні вимкнення
1	50.0%	1/2
2	54.5%	6/11
3	60.0%	6/10
4	66.0%	6/9
5	33.1%	3/9
6	86.9%	8/10
7	90.0%	9/10
8	95.0%	9.5/10

Протягом кількох тижнів результати роботи покращувалися, але на початку п’ятого тижня відсоток розпізнавання різко знизився, а згодом знову виріс. Аналіз логів та повідомлень показав, що під час однієї з ракетних атак перша хвиля ракет летіла у напрямку Києва. Штучний інтелект розпізнав це як загрозу і створив відкладене завдання, яке перевело пристрої на резервний режим живлення. Однак курс ракет змінився, а завдання вже було виконано, що призвело до зниження точності визначення загрози.

Згодом точність значно зросла, оскільки провайдери інформування почали надавати більш точний час планових відключень, що покращило статистику. Це підтверджує, що більша кількість даних дозволяє легше прогнозувати можливі відключення.

У результаті навчання читанням новин була покращена можливість користуванням СРБ, зокрема, система забезпечує раннє попередження про можливі знеструмлення, що дозволяє користувачам підготуватися за кілька хвилин або годин до виникнення проблеми. Також дає змогу адаптуватися до погодних умов, оптимізуючи використання енергії для уникнення перегріву чи переохолодження

приміщення. Крім того, в разі загроз, система може автоматично переходить у безпечний режим роботи, що забезпечує захист без участі користувача.

3.5 Створення бази відключень електроенергії

Розробка системи для обліку, аналізу та швидкого доступу до інформації про відключення електроенергії вимагає комплексного підходу до організації бази даних. У даному випадку для ефективного вирішення поставлених завдань використовуються три основні технології зберігання даних: PostgreSQL, OpenSearch та Redis. Кожна з них має свої унікальні можливості, що дозволяють оптимально реалізувати різні аспекти системи.

Історичні та актуальні дані стосовно відключень зберігаються в таблицях PostgreSQL за такою схемою, як зображено на рисунку 3.2:

- Outages:
  - id (унікальний ідентифікатор).
  - location_id (локація користувача, де відбулося відключення).
  - start_time і end_time (час початку та завершення відключення).
  - type (планове, аварійне).
  - cause (причина, наприклад, атака, ремонт, аварія).
- Locations:
  - id (унікальний ідентифікатор).
  - name (назва локації користувача)
  - region (назва області/регіону)

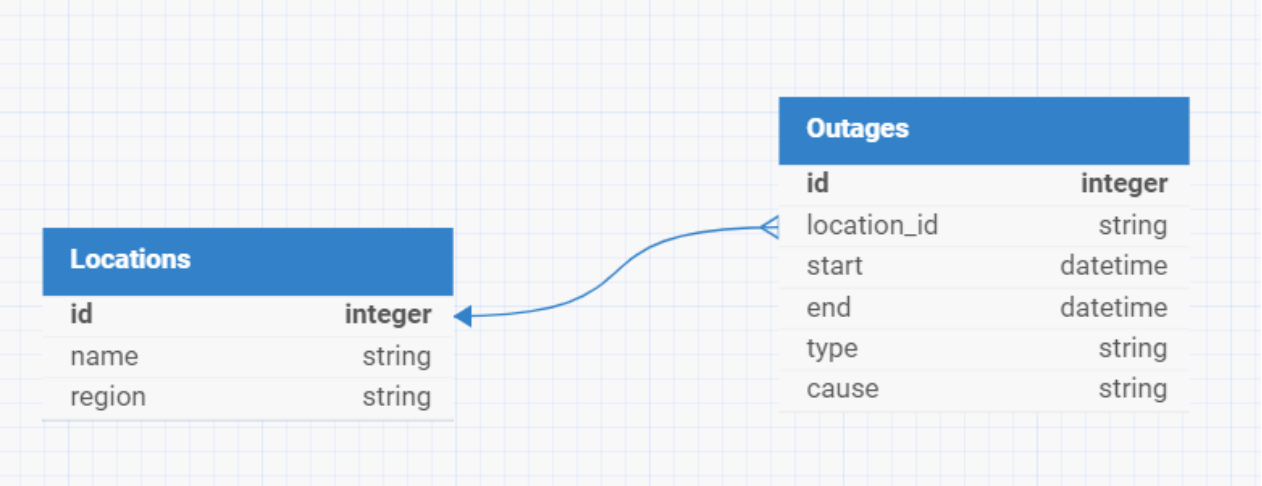


Рис. 3.2 Схема бази даних Postgres

Всі сповіщення пов’язані з телеграм-каналами про попередження небезпеки, API постачальників (таких як Київ Цифровий, ДТЕК чи Укренерго) зберігаються в OpenSearch в такій побудові індексу, як на рисунку 3.3, де:

- message_id: унікальний ідентифікатор повідомлення.
- timestamp: час надходження даних.
- source: джерело інформації (Telegram, Київ Цифровий, Укренерго).
- content: текст повідомлення про відключення.
- tags: ключові слова або категорії (наприклад, "атака", "ремонт", "аварія").

Останній компонент як сховище даних буде виступати Redis, який використовується для оптимізації роботи системи, коли потрібні миттєві відповіді.

В цьому випадку буде зберігати найбільш запрошувані дані, а саме current_outages (список активних відключень по локації), alerts (термінові попередження) вони зберігаються протягом доби, temperature_forecast (дані про погодні умови в локації) беруться дані на добу вперед.

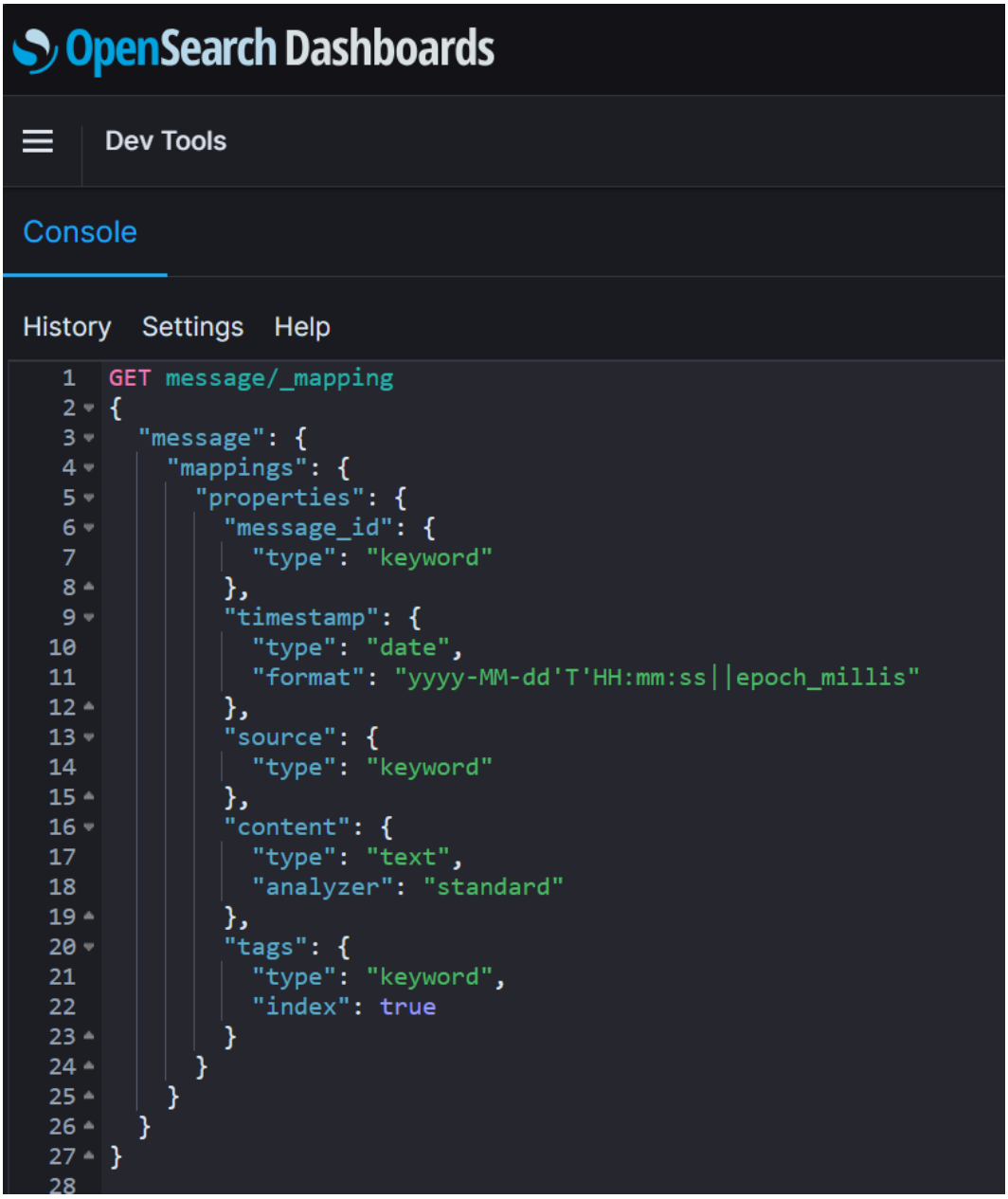


Рис. 3.3 Схема індексу «message»

Комбінування PostgreSQL, OpenSearch та Redis дає змогу забезпечити високу ефективність системи, зокрема:

- PostgreSQL забезпечує цілісність історичних даних;
- OpenSearch виконує аналітику великих потоків інформації в реальному часі;
- Redis надає швидкий доступ до актуальних даних.

Ця трикомпонентна архітектура гарантує стабільну роботу системи навіть за високих навантажень і забезпечує швидке реагування на критичні події.

3.6 Тестування та використання в реальних умовах

Для перевірки працездатності розробленої методики було здійснено запуск спроектованої системи в реальних умовах, із застосуванням набору деяких тестових даних, що охоплюють усі основні функціональні можливості. Роботу самого застосунку можна побачити на рисунку 3.4.

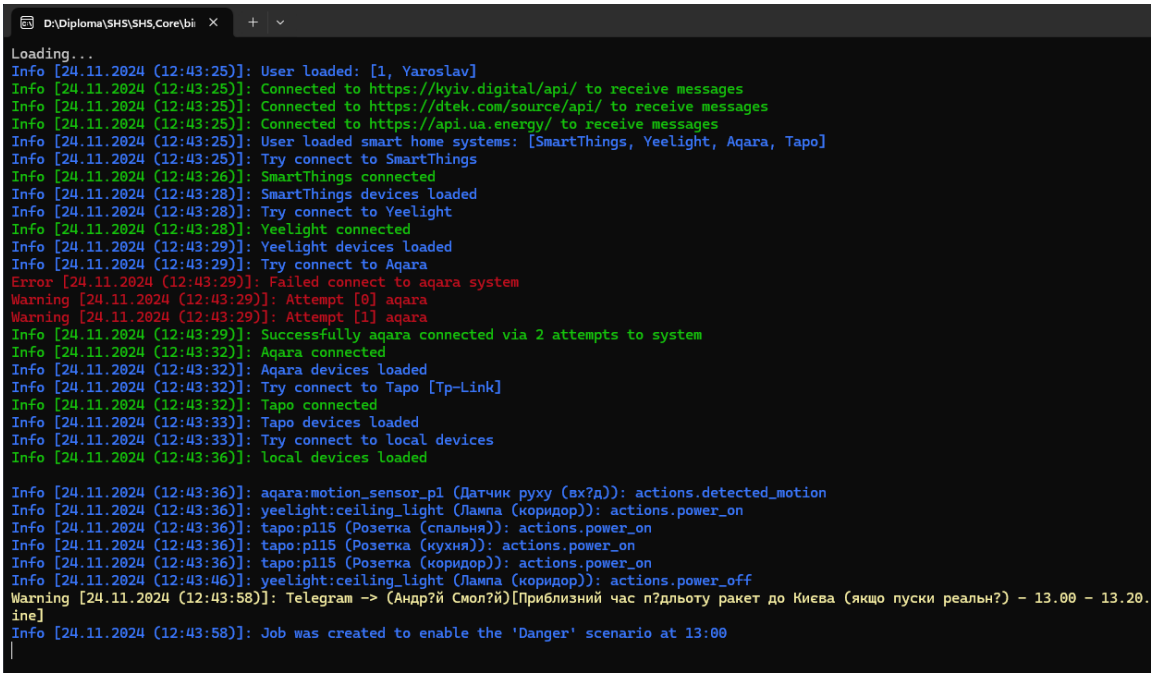


Рис. 3.4 Консольний інтерфейс системи

Як видно з проведених тестів, консольний застосунок успішно запустився та коректно завантажив профіль користувача, демонструючи стабільну роботу на початковому етапі. Одразу після цього було активовано Web API, яке успішно підключилося до API постачальників даних про електроенергію. Наступним кроком система автоматично визначила платформи, на яких базуються локальні пристрої, встановила з'єднання з кожним із них та перейшла в режим очікування подій, що можуть надходити з різних джерел, зокрема від «Київ Цифровий» через Telegram або НЕК «Укренерго». Така інтеграція дозволяє системі взаємодіяти з різноманітними платформами, об'єднуючи функціонал та надаючи можливість управління пристроями через штучний інтелект.

Під час тестування було зафіксовано повідомлення про потенційну небезпеку для Києва. Штучний інтелект, проаналізувавши отриману інформацію, створив завдання (job), що автоматично активувало сценарій «Danger». Цей сценарій, у свою чергу, ініціював виконання окремих ліній задач для кожної системи розумного будинку.

Для перевірки роботи сценарію «Danger» було змінено системний час, встановивши годинник на 12:59, щоб імітувати момент спрацювання події. Це дозволяє проаналізувати, як система реагує на критичні ситуації та чи правильно активуються відповідні процеси в межах заданого сценарію, результат можна побачити на рисунку 3.5.

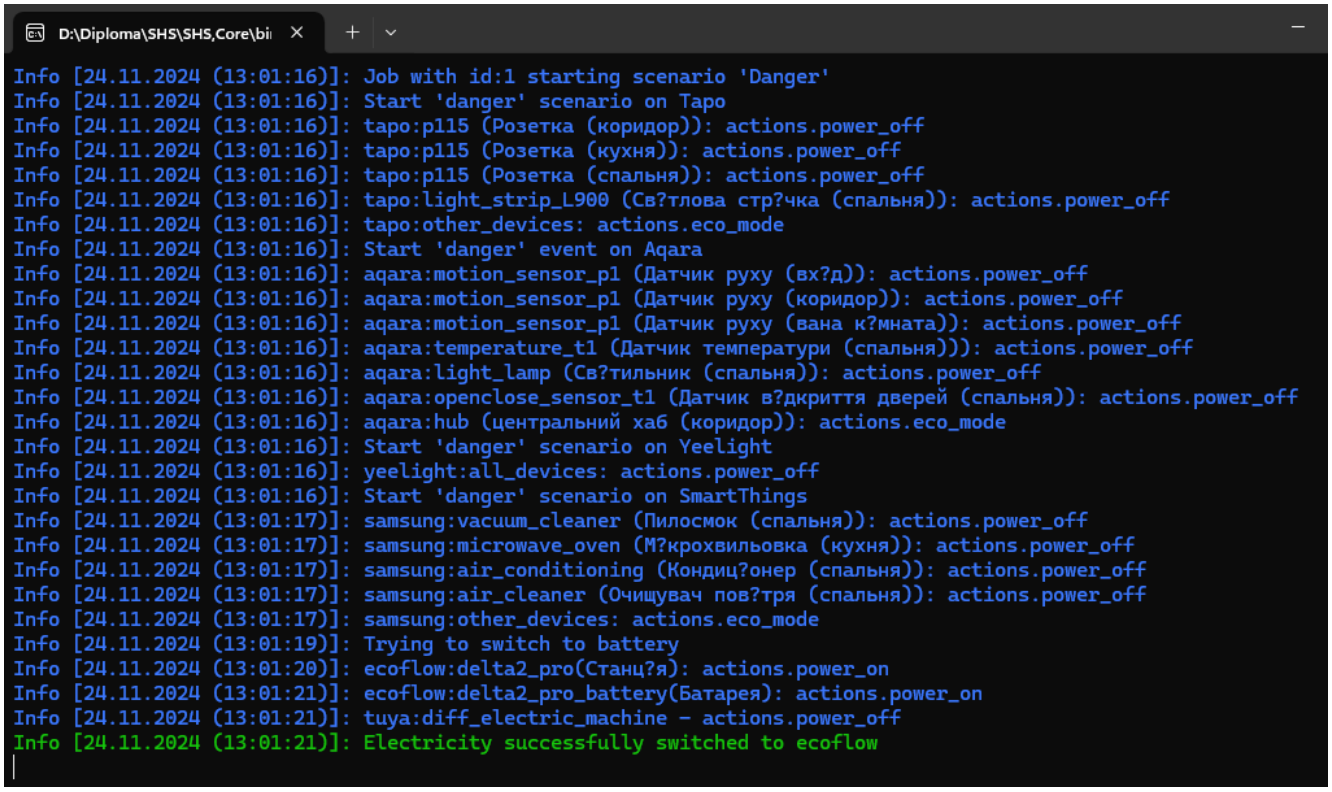


Рис. 3.5 Логи про виконання сценарію «Danger»

Як видно, сценарій «Danger» успішно активував задачі, які почали поступово відключати пристрої в кожній з інтегрованих екосистем. Після завершення вимкнення всіх необов’язкових пристроїв та переведення частини з них у режим енергозбереження (Eco), система автоматично деактивувала основний

диференційний автомат і підключила батареї EcoFlow. У результаті квартира успішно перейшла в режим «Небезпека», в якому забезпечується функціонування лише найнеобхідніших приладів для мінімального споживання енергії та підтримки критичних функцій.

Для більшого розуміння роботи цього кейсу можна подивитися на діаграму послідовності, яка зображена на рисунку 3.6.

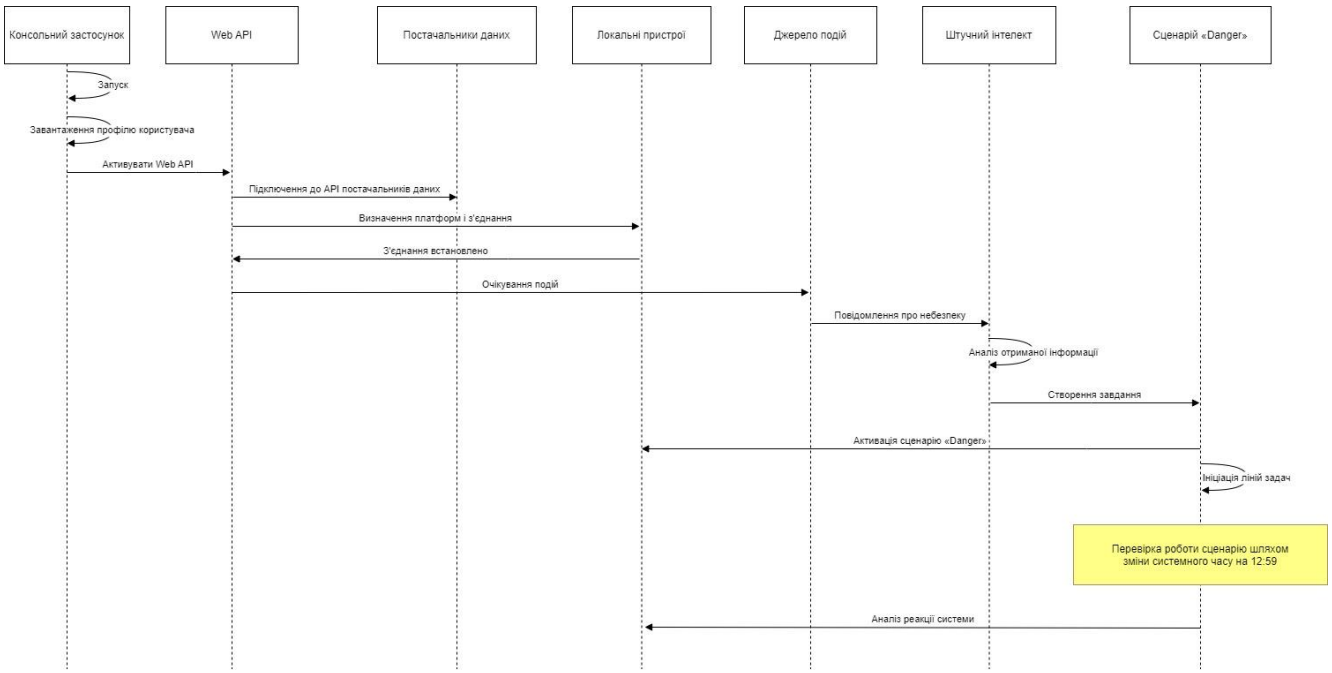


Рис. 3.6. Діаграма послідовності сценарію Danger

Під час тестування було отримано результати переходу в режим “Danger”, які демонструють ефективність запропонованої методики (рис.3.7). Як видно з графіку, споживання електроенергії без використання методики значно перевищує споживання з її застосуванням. Особливо помітна різниця у пікові моменти, такі як 15:00:20, де впровадження методики дозволило суттєво зменшити споживання кіловат. Варто зазначити, що стрибок до 4 кВт без методики стався через те, що була необхідність вручну вмикати всі пристрої, а вони, як відомо, на старті можуть споживаті набагато більше ніж при звичайному оежимі рооти і згодом переводити їх у еко-режим. Натомість запропонована методика автоматизує цей процес,

самостійно оптимізуючи роботу пристроїв та знижуючи споживання електроенергії, що робить її більш ефективною та зручною для користувачів.

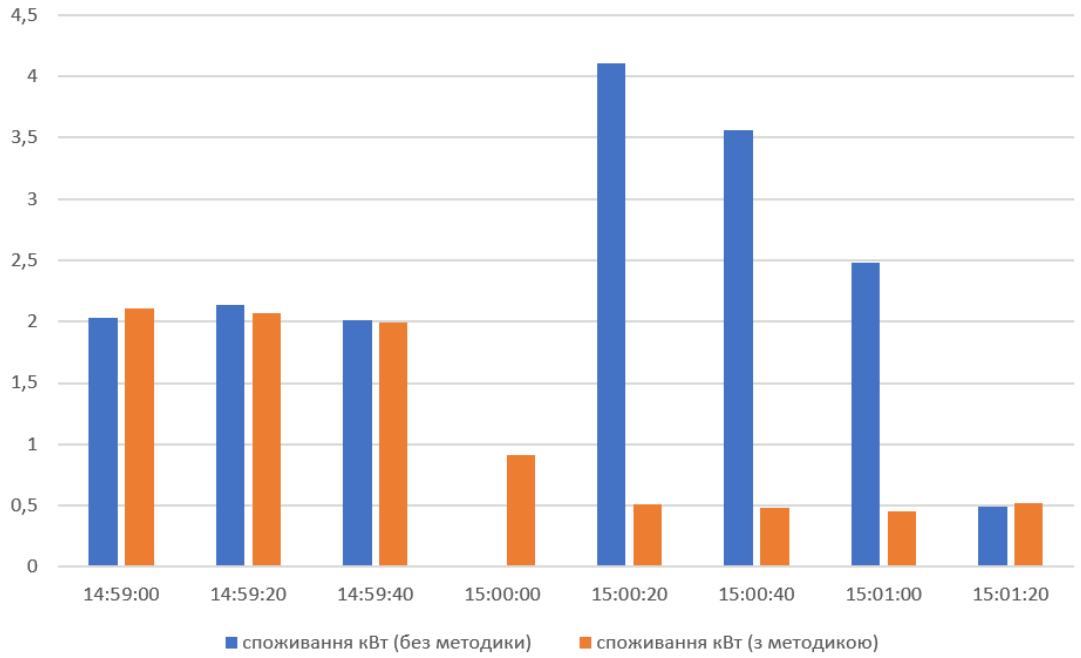


Рис. 4.7. Споживання електрики в момент часу

3.7 Опис розроблених модулів

Розроблена система побудована на основі модульної архітектури, яка забезпечує гнучкість, масштабованість та чіткий розподіл функціональності між окремими компонентами. Кожен модуль виконує певну роль у роботі системи, інтегруючи різні технології для досягнення поставлених цілей, її вигляд можна побачити на зображенні 3.8 та 3.9.

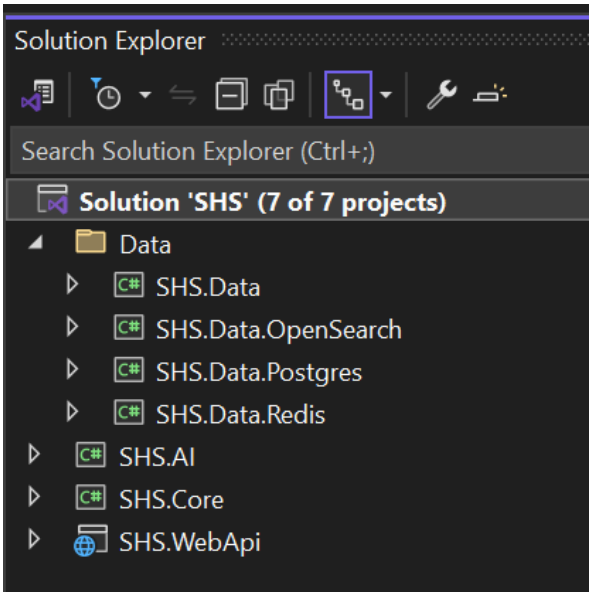


Рис. 3.8 Solution у Visual Studio

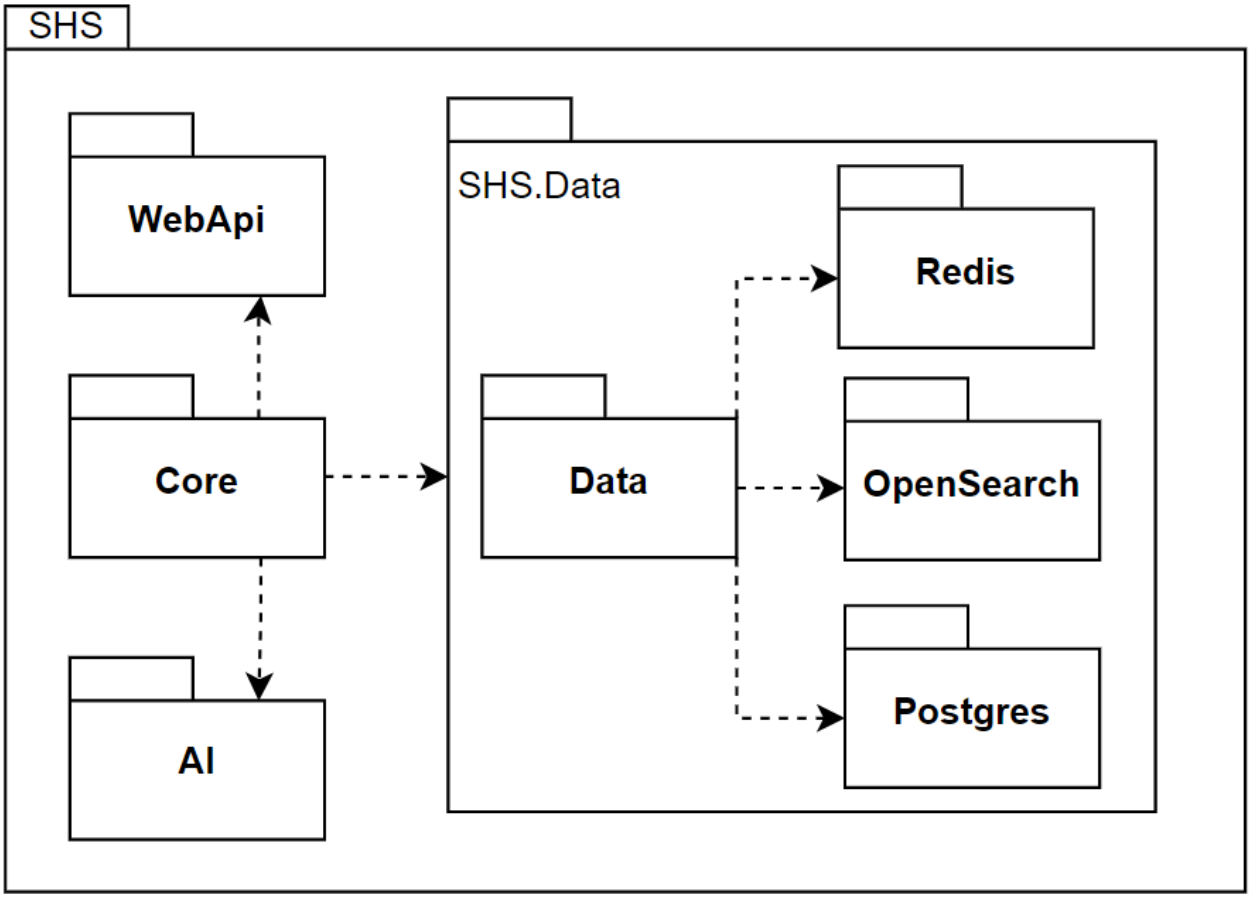


Рис. 3.9 Структура застосунку

Основу системи складає модуль SHS.Data, який відповідає за роботу з даними та виступає центральним ядром для взаємодії з різними джерелами інформації. Цей

модуль забезпечує спільний функціонал для обробки, доступу та управління даними, слугуючи базою для спеціалізованих проектів. Зокрема, SHS.Data.OpenSearch займається інтеграцією з платформою OpenSearch, яка використовується для швидкого пошуку та аналізу даних. Модуль SHS.Data.Postgres реалізує взаємодію з базою даних PostgreSQL, забезпечуючи зберігання структурованої інформації, а SHS.Data.Redis підтримує інтеграцію з Redis для кешування та швидкодоступних даних.

Модуль SHS.AI відповідає за функціонування компонентів штучного інтелекту, які є центральними для адаптивної роботи системи. Він включає механізми донавчання моделей, аналізу даних та автоматизованого прийняття рішень на основі отриманих результатів. Цей компонент також забезпечує створення сценаріїв автоматизації, що дозволяє ефективно керувати ресурсами та реагувати на різноманітні події, включаючи критичні ситуації.

Центральну роль у роботі системи виконує модуль SHS.Core. У ньому реалізована вся бізнес-логіка, яка відповідає за координацію роботи інших модулів. Саме тут відбувається обробка основних сценаріїв, аналіз подій і взаємодія між компонентами системи. Цей модуль є ядром, яке об'єднує всю функціональність і забезпечує коректну роботу всієї архітектури.

Фасадом системи виступає модуль SHS.WebApi, який забезпечує взаємодію з зовнішніми сервісами та користувачами. У ньому реалізовано механізми прийому вхідних даних, виконання викликів до API сторонніх сервісів і обробки запитів, що надходять до системи. Завдяки цьому модулю система інтегрується з платформами, що надають дані про енергопостачання та інші зовнішні події.

Кожен із розроблених модулів відіграє ключову роль у забезпеченні ефективної, адаптивної та надійної роботи системи. Завдяки чітко визначеній структурі та розподілу обов’язків між компонентами, система є легкою для масштабування, супроводу та адаптації до нових вимог. Вона здатна не лише працювати в умовах нестабільного енергопостачання, але й забезпечувати інтеграцію з різноманітними платформами, надаючи користувачам широкий спектр функціональності для управління розумним будинком.

ВИСНОВКИ

У цій роботі проведено поглиблений аналіз існуючих систем розумних будинків та підходів до їхнього керування в умовах сучасних викликів. Особлива увага приділяється виявленню ключових проблем, які виникають під час функціонування таких систем за нестабільного електропостачання. До основних труднощів віднесено недостатню адаптивність до перебоїв електроенергії, низьку ефективність енергоспоживання, обмежені можливості інтеграції з резервними джерелами живлення, а також брак ефективних механізмів пріоритетного управління критично важливими пристроями. Ці аспекти значно знижують надійність і зручність використання розумних будинків у нестабільних умовах.

На основі проведеного аналізу було розроблено вдосконалену методику керування розумними будинками, яка спрямована на підвищення стабільності та ефективності їх роботи в умовах частих і непередбачуваних перебоїв електропостачання. Запропонована методика ґрунтується на впровадженні адаптивного підходу до управління енергоспоживанням, що передбачає пріоритизацію роботи критичних пристроїв, автоматичне перемикання на резервні джерела живлення та інтеграцію сучасних алгоритмів оптимізації енергоспоживання з використанням можливостей штучного інтелекту.

Розроблено програмну реалізацію запропонованої методики, яка охоплює широкий функціонал, включаючи моніторинг стану електромережі, контроль за рівнем споживання електроенергії, а також управління підключеними пристроями через такі протоколи, як Wi-Fi, Bluetooth, Zigbee, Z-Wave і Matter. Крім цього, система забезпечує автоматичне інформування користувача про поточний стан енергоспоживання, наявність перебоїв у електропостачанні та активацію резервного живлення. Така інтеграція дозволяє не лише підвищити ефективність управління системою, але й зробити її використання максимально зручним для кінцевого користувача.

Експериментальні результати тестування довели ефективність впровадженої методики: середній час реакції системи на перебої електропостачання зменшився з

1 хвилини 40 секунд до 20 секунд, що демонструє прискорення реакції на 80%. Порівняння роботи системи з використанням запропонованого підходу та без нього продемонструвало суттєві переваги методики. Це включає забезпечення стабільної роботи розумного будинку навіть за умов непередбачуваних перебоїв енергопостачання, що робить запропоноване рішення практичним і надійним для впровадження у реальних умовах.Отримані результати можуть бути використані при розробці сучасних систем розумних будинків у реальних умовах нестабільного енергопостачання.

Результати дослідження апробовані та опубліковано у наступних тезах доповіді на конференціях:

1. Золотухіна О.А., Мудрик Я.Ю. Покращення отримання даних через Redis з можливістю фільтрації та сортування. // Всеукраїнська науково-технічна конференція «Застосування програмного забезпечення в ІКТ». – Київ: ДУІКТ, 2024. – С.188-189.

2. Золотухіна О.А., Мудрик Я.Ю. Застосування штучного інтелекту для оптимізації енергоспоживання в розумних будинках при нестабільному енергопостачанні. // ІІ міжнародна науково-практична конференція «Сучасні аспекти діджиталізації та інформатизації в програмній та комп'ютерній інженерії». – Київ: ДУІКТ, 2024. – подано до друку.

ПЕРЕЛІК ПОСИЛАНЬ

1. Розумний будинок: комфорт та ефективність в єдиній системі [Електронний ресурс] // Vodafone. – 2023. – Режим доступу до ресурсу: https://www.vodafone.ua/shop/ua/blog/rozumnij-budinok-komfort-ta-efektivnist-v-edinij-sistemi.html.

2. Облаштовуємо «розумний» і енергонезалежний будинок [Електронний ресурс] // dou.ua. – 2024. – Режим доступу до ресурсу: https://dou.ua/lenta/articles/smart-home-tips-and-best-practices/.

3. Штучний інтелект – це наступний крок для розумних будинків [Електронний ресурс] // unite.ai. – 2022. – Режим доступу до ресурсу: https://www.unite.ai/uk/artificial-intelligence-is-the-next-step-for-smart-homes/.

4. Золотухіна О.А., Мудрик Я.Ю. Покращення отримання даних через Redis з можливістю фільтрації та сортування. // Всеукраїнська науково-технічна конференція «Застосування програмного забезпечення в ІКТ». – Київ: ДУІКТ, 2024. – С.188-189.

5. Золотухіна О.А., Мудрик Я.Ю. Застосування штучного інтелекту для оптимізації енергоспоживання в розумних будинках при нестабільному енергопостачанні. // ІІ міжнародна науково-практична конференція «Сучасні аспекти діджиталізації та інформатизації в програмній та комп'ютерній інженерії». – Київ: ДУІКТ, 2024. – ???.

6. Що таке «розумний будинок» і навіщо він потрібен? [Електронний ресурс] // stylus. – 2023. – Режим доступу до ресурсу: https://blog.stylus.ua/uk/528.html.

7. Home Assistant 101. Посібник для початківців [Електронний ресурс] // dou.ua. – 2022. – Режим доступу до ресурсу: https://dou.ua/forums/topic/38947/.

8. Жовнір Ю., Буров Є. Еволюція архітектурних рішень для розумних будинків // Computer Systems and Information Technologies. — 2024. — № 3. — С. 74–85.

9. How Expensive Are Smart Homes? [Електронний ресурс] // todayshomeowner.com. – 2024. – Режим доступу до ресурсу: https://todayshomeowner.com/smart-home/.

10. Фейрхед, Г. Introducing the Raspberry Pi Pico [Електронний ресурс] // Exploring Raspberry Pi: Interfacing to the Real World with Embedded Linux. — Springer, 2021. — Режим доступу до ресурсу: https://link.springer.com/chapter/10.1007/978-1-4842-8135-2_1.

11. The Use of MQTT in M2M and IoT Systems: A Survey // IEEE Xplore. — 2019. — URL: https://ieeexplore.ieee.org/document/9247996.

12. Системи безпеки розумного будинку [Електронний ресурс] // exposervice-p.com.ua. – 2023. – Режим доступу до ресурсу: https://exposervice-p.com.ua/sistemi-bezpeki-rozumnogo-budinku/.

13. Розумні технології покращують транспорт у містах [Електронний ресурс] // bezpeka-shop.com. – 2020. – Режим доступу до ресурсу: https://www.bezpeka-shop.com/ua/blog/obzor/umnye-tekhnologii-uluchshayut-transport-v-gorodakh/?srsltid=AfmBOorf0J41Xs-j6EbcNe2lyJfwKISCe8YRGapkC900wZQ7h0AOiPti.

14. How Smart Homes will be integrated into Smart Cities [Електронний ресурс] // Smart Building. – 2024. – Режим доступу до ресурсу: https://www.thesmartcityjournal.com/en/smart-building/how-smart-homes-will-be-integrated-into-smart-cities.

15. Tizen OS: Features, Pros and Cons [Електронний ресурс] // Techopedia. — Режим доступу до ресурсу: https://www.techopedia.com/definition/30250/tizen-os.

16. Turnbull, J. The Docker Book: Containerization is the new virtualization. — James Turnbull, 2014. — 348 p.

17. Raschka, S., Mirjalili, V. Python Machine Learning: Machine Learning and Deep Learning with Python, scikit-learn, and TensorFlow 2. 3rd ed. — Packt Publishing, 2019. — 770 p.

18. Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., Devin, M., Ghemawat, S., Irving, G., Isard, M., Kudlur, M., Levenberg, J., Monga, R., Moore, S.,

Murray, D. G., Steiner, B., Tucker, P., Vasudevan, V., Warden, P., Wicke, M., Yu, Y., Zheng, X. TensorFlow: A System for Large-Scale Machine Learning. In *Proceedings of the 12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16)*, 2016, pp. 265–283. URL: https://www.usenix.org/system/files/conference/osdi16/osdi16-abadi.pdf.

19. Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., Chintala, S. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In *Advances in Neural Information Processing Systems 32 (NeurIPS 2019)*, 2019, pp. 8024–8035. URL: https://papers.nips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.

20. Roy, G. M. RabbitMQ in Depth. — Manning Publications, 2017. — 320 p.

21. IBM Documentation. Використання OpenSearch для доступу до індексованого контенту [Електронний ресурс]. — Режим доступу до ресурсу: https://www.ibm.com/docs/uk/product-master/12.0.0?topic=search-using-opensearch-access-indexed-content.

ДОДАТОК А. ДЕМОНСТРАЦІЙНІ МАТЕРІАЛИ

ДЕРЖАВНИЙ УНІВЕРСИТЕТ ІНФОРМАЦІЙНО - КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ

НАВЧАЛЬНО–НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

КАФЕДРА ІНЖЕНЕРІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Магістерська робота

МЕТОДИКА АДАПТИВНОГО КЕРУВАННЯ СИСТЕМАМИ РОЗУМНОГО БУДИНКУ В УМОВАХ НЕСТАБІЛЬНОГО ЕНЕРГОПОСТАЧАННЯ НА ОСНОВІ ШТУЧНОГО ІНТЕЛЕКТУ

Виконав: студент групи ПДМ-62 Мудрик Ярослав Юрійович

Керівник: к.т.н., доц., доцент кафедри ІПЗ Золотухіна Оксана Анатоліївна

Київ - 2025

МЕТА, ОБ'ЄКТА ТА ПРЕДМЕТ ДОСЛІДЖЕННЯ

Мета роботи: покращення автономності та ефективності систем розумного будинку шляхом оптимізації енергоспоживання в умовах нестабільного енергопостачання із використанням штучного інтелекту.

Об'єкт дослідження: процес керування системами розумного будинку в умовах нестабільного енергопостачання.

Предмет дослідження: методи, моделі та засоби адаптивного керування системами розумного будинку в умовах нестабільного енергопостачання з використанням штучного інтелекту.

АКТУАЛЬНІСТЬ РОБОТИ

Існуючі проблеми керування системами розумного будинку в умовах нестабільногоелектропостачання:

- залежність централізованих алгоритмів від контролера та складність синхронізації в децентралізованих системах, що ускладнює адаптацію до нестабільного енергопостачання;
- слабка адаптивність традиційних алгоритмів до зовнішніх умов, таких як раптові відключення електроенергії;
- відсутність інтеграції моделей штучного інтелекту для прогнозування, адаптації та оптимізації енергоспоживання.

Запропонованерішення:

- використання штучного інтелекту для адаптивного керування системами розумногобудинку;
- оптимізація функцій життєзабезпечення в умовах нестабільного енергопостачання.

3

АЛГОРИТМ ПРОГНОЗУВАННЯ СПОЖИВАННЯ ЕЛЕКТРОЕНЕРГІЇ

$$E(t) = \sum_{i=1}^{n} P_i(t) * T_i(t),$$

де $E(t)$ – загальне електроспоживання за час t;

$P_i(t)$ – потужність пристрою i у час t;

$T_i(t)$ – час роботи пристрою i за період t;

n – кількість пристроїв.

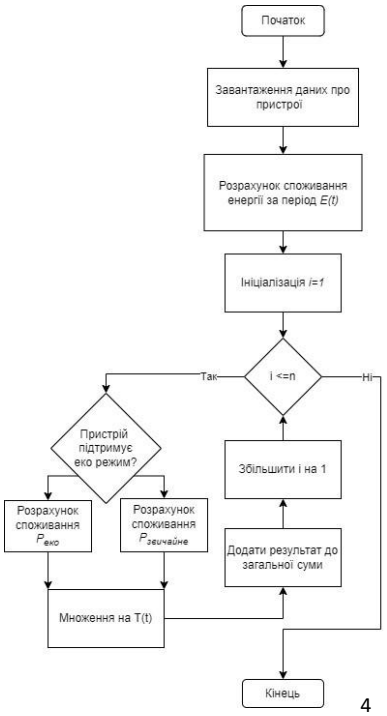
$$P_{\text{звичайний}}(t) = U(t) * I(t)$$

$$P_{\text{еко}}(t) = k * U(t) * I(t)$$

де $U(t)$ – напруга в момент часу t, у вольтах (В);

$I(t)$ – струм в момент часу t, у амперах (А);

k – коефіцієнт еко-режиму ($0 < k < 1$)



4

АЛГОРИТМ ОПТИМІЗАЦІЇ ЕНЕРГОСПОЖИВАННЯ

$$Pr_i = w_i * C_i$$

де w_i – вага важливості (наприклад, 3 для критично-важливих систем, 2 для важливих та 1 для некритичних);

Ci – критичність системи (0-1, де 1 найвища важливість).

$$min\ E(t) = \sum_{i=1}^{n} P_i(t) * T_i(t) + \lambda * \sum_{j=1}^{m} S_j^2$$

де $E(t)$ – загальні енергетичні витрати за час t;

$P_i(t)$ – потужність пристрою i у час t;

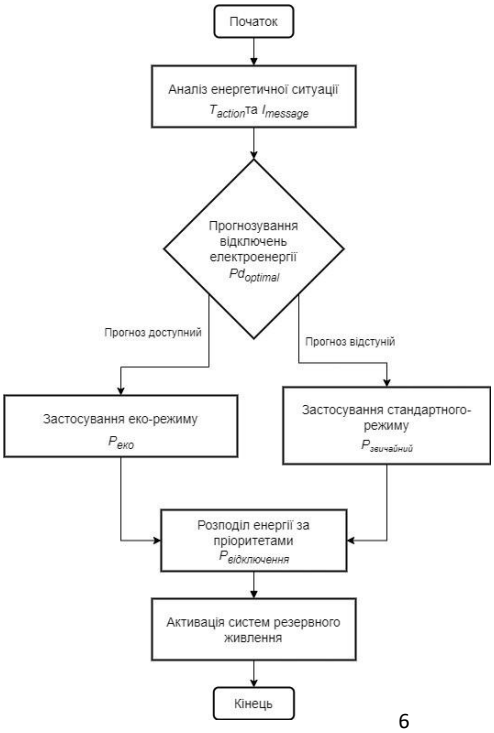
$T_i(t)$ – час роботи і-го пристрою;

S_j – відхилення від заданих умов комфорту;

λ – ваговий коефіцієнт, який визначає компроміс між енергоефективністю та комфортом.

5

АЛГОРИТМ АНАЛІЗУ СПОЖИВАННЯ ЕЛЕКТРОЕНЕРГІЇ



$$T_{action} = T_{message} + \Delta T_{analysis}$$

де $T_{message}$ – час отримання повідомлення;

$\Delta T_{analysis}$ – затримка на обробку повідомлень і формування рішення.

$$Pd_{optimal} = \arg\min_{P}(\sum_{i=1}^{n} C_i * U_i(P))$$

де Pd — набір параметрів для пристроїв;

C_i — вартість ресурсу і;

$U_i(P)$ — функція корисності

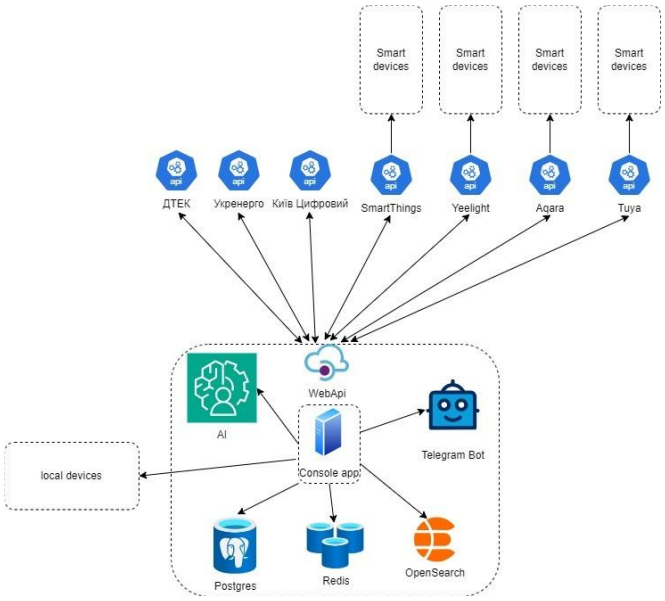
$$U_i(P) = P * t$$

де P – потужність пристрою;

t = час роботи

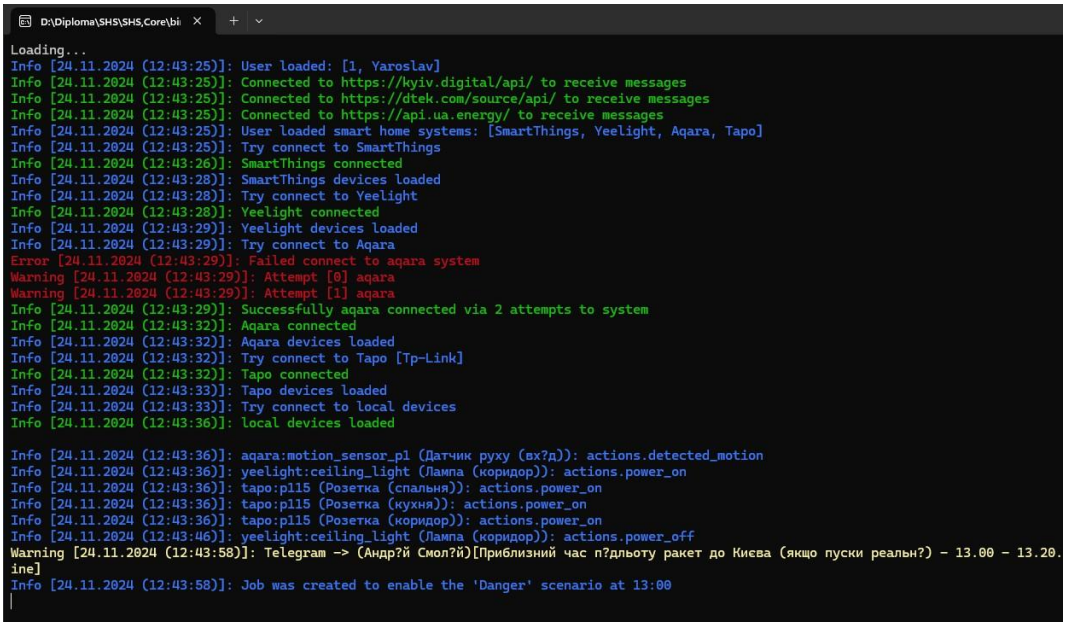
6

АРХІТЕКТУРА ЗАСТОСУНКУ ДЛЯ АДАПТИВНОГО КЕРУВАННЯ СИСТЕМАМИ РОЗУМНОГО БУДИНКУ



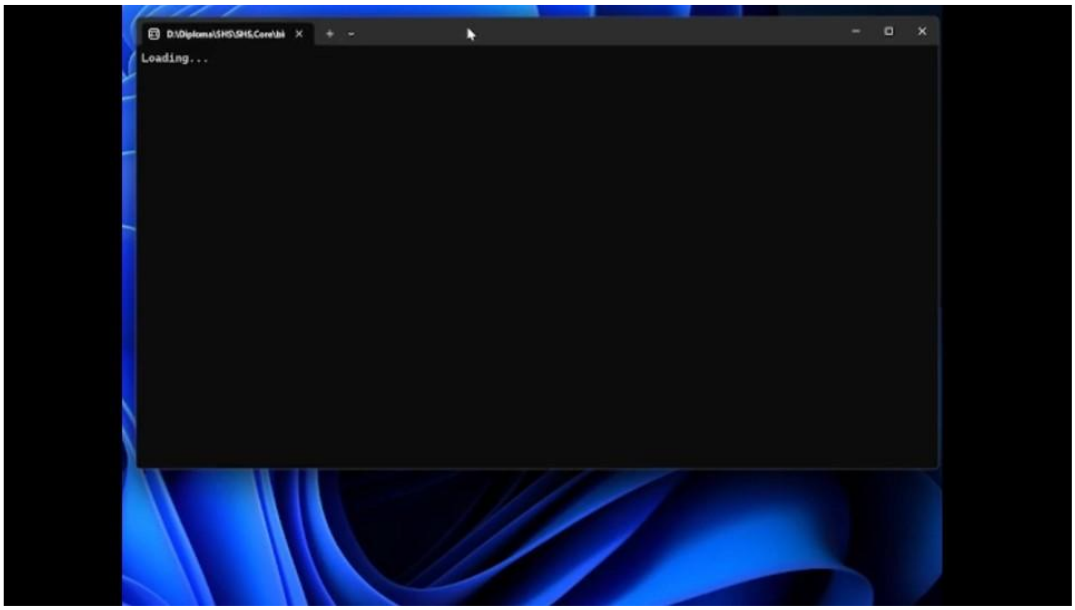
7

ЕКРАНІ ФОРМИ – КОНСОЛЬНИЙ ЗАСТОСУНОК



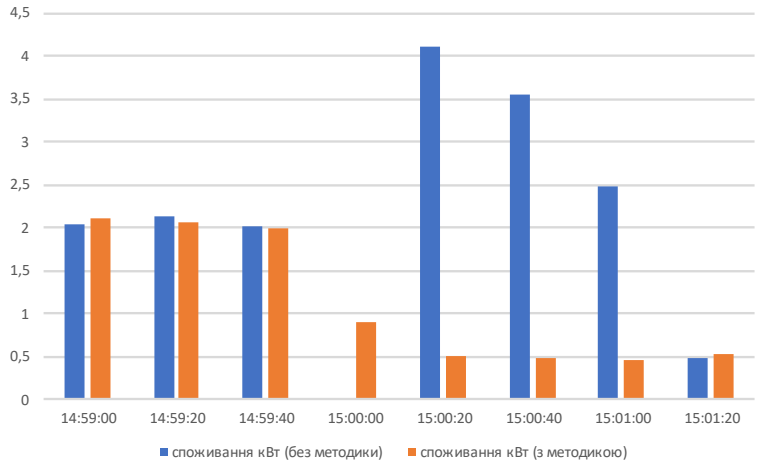
8

МОДЕЛЮВАННЯ ПРОЦЕСУ КЕРУВАННЯ СИСТЕМАМИ РОЗУМНОГО БУДИНКУ В УМОВАХ НЕСТАБІЛЬНОГО ЕНЕРГОПОСТАЧАННЯ



9

РЕЗУЛЬТАТИ ТЕСТУВАННЯ



10

ВИСНОВКИ

1. Проведено аналіз існуючих систем розумних будинків та підходів до їх керування. Виявлено основні проблеми функціонування таких систем в умовах нестабільного електропостачання, серед яких недостатня адаптивність до перебоїв електроенергії, низька ефективність енергоспоживання та обмежена інтеграція з резервними джерелами живлення.
2. На основі проведеного аналізу розроблено методику керування розумними будинками, яка спрямована на забезпечення стабільності роботи в умовах частих перебоїв електропостачання. Запропонована методика передбачає використання адаптивного підходу до управління споживанням енергії, пріоритизацію критичних пристроїв, автоматичне перемикання на резервні джерела живлення та інтеграцію алгоритмів оптимізації енергоспоживання з використанням штучного інтелекту.
3. Розроблено програмну реалізацію методики, яка включає можливість моніторингу стану мережі, контролю за споживанням електроенергії та управління пристроями за допомогою протоколів Wi-Fi, Bluetooth, Zigbee, Z-Wave та Matter. Система також забезпечує автоматичне сповіщення користувача про стан енергоспоживання та використання резервного живлення.
4. Результати тестування показали, що застосування розробленої методики дозволяє знизити середню затримку реакції системи на перебої електропостачання з 1 хвилини 40 секунд до 20 секунд, або до 80% більш ефективно. Порівняння роботи системи з використанням методики та без неї продемонструвало значну перевагу запропонованого підходу в умовах непередбачуваних перебоїв енергопостачання.

ПУБЛІКАЦІЇ ТА АПРОБАЦІЯ РОБОТИ

Тези доповідей:

1. Золотухіна О.А., Мудрик Я.Ю. Покращення отримання даних через Redis з можливістю фільтрації та сортування. // Всеукраїнська науково-технічна конференція «Застосування програмного забезпечення в ІКТ». – Київ: ДУІКТ, 2024. – С.188-189.
2. Золотухіна О.А., Мудрик Я.Ю. Застосування штучного інтелекту для оптимізації енергоспоживання в розумних будинках при нестабільному енергопостачанні. // ІІ міжнародна науково-практична конференція «Сучасні аспекти діджиталізації та інформатизації в програмній та комп'ютерній інженерії». – Київ: ДУІКТ, 2024. – подано до друку.

ДОДАТОК Б. ЛІСТИНГ ОСНОВНИХ ПРОГРАМНИХ МОДУЛІВ

```
// модуль старту
    static async Task Main(string[] args)
    {
        var handler = new EventHandler();
        var shs = new SmartHomeSystem();
        shs.LoadSystem();
        var connected = shs.ConnectToSmartHomeVendors();

        if (connected)
        {
            shs.StartReceiveEvents(@event =>
            {
                if (@event.Source == "telegram")
                {
                    Logger.LogWarning("Telegram -> ({@event.Telegram.AuthorName}) [{@event.Telegram.Message]");
                }
                if (@event.Source == "devicesActions")
                {
                    Logger.LogInformation("@event.Device.Manafacture:@event.Device.Model (@event.Device.CustomName): @event.Action");
                }
                if (@event.Source == "backgroudJob")
                {
                    Logger.LogInformation("@event.Job.Id:@event.Job.Title - @event.Job.Time (@event.Job.StartScenarios)");
                }
                handler.HandleEvent(@event);
            });
        }
    }

//модуль SmartHomeSystem
public class SmartHomeSystem
{
    public void LoadSystem()
    {
        var energyProviderResult = _energyProviderService.ConnectToProviders();
        var devices = LoadAllDevicesInfo();

        var comfortWeight = GetComfortWeight(energyProviderResult);

        CalculationUsageEnergy(devices);
        OptimiseUsageEnergyIfNeeded(devices, comfortWeight);
    }

    private void CalculationUsageEnergy(List<Device> devices)
    {
        var totalEnergyConsumption = _energyCalculationService.CalculateTotalEnergyConsumption(devices);
        var outageConsumptions = _energyCalculationService.CalculateOutageConsumption(GetEcoDevices(devices), GetNormalDevices(devices));
    }

    private void OptimiseUsageEnergyIfNeeded(List<Device> devices, double comfortWeight)
    {
        if (comfortWeight > 0 && comfortWeight < 0.5)

            _energyCalculationService.OptimizeEnergyConsumption(devices, comfortWeight);
    }
    //other code
}

// модуль для розрахунків споживання
public class EnergyCalculationService
{
    public double CalculateTotalEnergyConsumption(List<Device> devices)
    {
        double total = 0;

        foreach (var device in devices)
        {
            double normalConsumption = CalculateNormalConsumption(device.Voltage, device.Current);
            total += normalConsumption * device.TimeActive;
        }

        return total;
    }

    public double CalculateNormalConsumption(double voltage, double current)
    {
        return voltage * current;
    }
```

```
        public double CalculateEcoConsumption(double voltage, double current, double ecoCoefficient)
        {
            return ecoCoefficient * voltage * current;
        }

        public double CalculateOutageConsumption(List<Device> ecoDevices, List<Device> normalDevices)
        {
            double outageConsumption = 0;

            foreach (var device in ecoDevices)
            {
                outageConsumption += CalculateEcoConsumption(device.Voltage, device.Current, device.EcoModeCoefficient);
            }

            foreach (var device in normalDevices)
            {
                outageConsumption += CalculateNormalConsumption(device.Voltage, device.Current);
            }

            return outageConsumption;
        }

        public double OptimizeEnergyConsumption(List<Device> devices, double comfortDeviationWeight)
        {
            double totalEnergyCost = 0;

            foreach (var device in devices)
            {
                double energyCost = device.Power * device.TimeActive;
                double comfortPenalty = comfortDeviationWeight * device.ComfortDeviation;
                totalEnergyCost += energyCost + comfortPenalty;
            }

            return totalEnergyCost;
        }
    }
    //модуль машинного навчання
    public class MLModelTrainer
    {
        private readonly MLContext _mlContext;
        private readonly string _modelPath;

        public MLModelTrainer(string modelPath)
        {
            _mlContext = new MLContext();
            _modelPath = modelPath;
        }

        public void TrainModel(string dataPath)
        {
            IDataView data = _mlContext.Data.LoadFromTextFile<OutageData>(
                dataPath,
                separatorChar: ',',
                hasHeader: true
            );
            var trainTestData = _mlContext.Data.TrainTestSplit(data, testFraction: 0.2);
            var pipeline = _mlContext.Transforms.Concatenate("Features", "RegionId", "Temperature")

.Append(_mlContext.Regression.Trainers.Sdca(labelColumnName: "Duration", featureColumnName: "Features"));
            var model = pipeline.Fit(trainTestData.TrainSet);
            var predictions = model.Transform(trainTestData.TestSet);
            var metrics = _mlContext.Regression.Evaluate(predictions, labelColumnName: "Duration");

            Console.WriteLine("RMSE: " + metrics.RootMeanSquaredError);
            Console.WriteLine("R^2: " + metrics.RSquared);

            _mlContext.Model.Save(model, data.Schema, _modelPath);
        }

        public float PredictNewDuration(float regionId, float temperature)
        {
            ITransformer model = _mlContext.Model.Load(_modelPath, out _);

            var predictionEngine = _mlContext.Model.CreatePredictionEngine<OutageData, OutagePrediction>(model);

            var prediction = predictionEngine.Predict(new OutageData
            {
                RegionId = regionId,
                Temperature = temperature
            });

            return prediction.PredictedDuration;
        }

        public void RetrainModel(string newDataPath)
        {
            TrainModel(newDataPath);
        }
    }
```