ДЕРЖАВНИЙ УНІВЕРСИТЕТ ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
КАФЕДРА ІНФОРМАЦІЙНИХ СИСТЕМ ТА ТЕХНОЛОГІЙ

КВАЛІФІКАЦІЙНА РОБОТА

на тему:
«МОДЕЛЬ УПРАВЛІННЯ «РОЗУМНИМ БУДИНКОМ» НА ОСНОВІ АНАЛІТИКИ КОРИСТУВАЦЬКОЇ ПОВЕДІНКИ»

на здобуття освітнього ступеня бакалавр
за спеціальності 126 Інформаційні системи та технології
(код,найменування спеціальності)
освітньо-професійної програми Інформаційні системи та технології
(назва)

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

_____ Іван ЧЕРНЄВ
(підпис) *(ім'я, ПРІЗВИЩЕ здобувача)*

Виконав: здобувач вищої освіти гр. ІСД-41
Іван ЧЕРНЄВ
(ім'я, ПРІЗВИЩЕ)

Керівник: Віктор Сагайдак
к.т.н.,доцент *(ім'я, ПРІЗВИЩЕ)*

Рецензент: _____
(ім'я, ПРІЗВИЩЕ)

Київ 2026

ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ

Навчально-науковий інститут інформаційних технологій

Кафедра Інформаційних систем та технологій

Ступінь вищої освіти Бакалавр

Спеціальність Інформаційні системи та технології

Освітньо-професійна програма Інформаційні системи та технології

ЗАТВЕРДЖУЮ
Завідувач кафедрою ІСТ

_____ Каміла СТОРЧАК
«_____» _____ 2026 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

Чернєв Іван Юрійович
(прізвище, ім'я, по батькові здобувача)

1. Тема кваліфікаційної роботи: Модель управління «розумним будинком» на основі аналітики користувацької поведінки.

керівник кваліфікаційної роботи Віктор САГАЙДАК доктор філософії, доцент
(Ім'я, ПРІЗВИЩЕ науковий ступінь, вчене звання)
затверджені наказом Державного університету інформаційно-комунікаційних технологій від «20» 02.2026р. №51

2. Строк подання кваліфікаційної роботи «01» червня 2026 р.

3. Вихідні дані до кваліфікаційної роботи: науково-технічна література з питань Інтернету речей, кіберфізичних систем, систем «розумного будинку» та предиктивної аналітики; технічна документація.

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

Аналіз сучасного стану та методів управління в системах «розумний будинок».
Проєктування моделі управління на основі предиктивної аналітики користувацької поведінки.

Розробка, моделювання та оцінка програмно-апаратної реалізації IoT-системи «розумного будинку».

5. Перелік графічного матеріалу: *презентація*

1. Теоретичні засади систем «розумного будинку».
2. Архітектура IoT-системи та апаратно-програмні складові.
3. Модель предиктивного управління на основі аналітики поведінки користувача.
4. Результати моделювання та оцінка ефективності системи.

6. Дата видачі завдання «01» червня 2026 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1	Аналіз літератури з IoT, Smart Home та предиктивної аналітики	02.02–16.02.2026	
2	Дослідження сучасних систем «розумного будинку» та існуючих аналогів	17.02–02.03.2026	
3	Аналіз технологій збору, передачі та захисту даних в IoT-системах	03.03–16.03.2026	
4	Проєктування моделі управління системою Smart Home	17.03–31.03.2026	
5	Обґрунтування вибору програмно-апаратних засобів	01.04–14.04.2026	
6	Розробка алгоритму предиктивного управління	15.04–30.04.2026	
7	Оформлення роботи: вступ, висновки, реферат	01.05–15.05.2026	
8	Розробка демонстраційних матеріалів	16.05–01.06.2026	

Здобувач вищої освіти _____ Іван ЧЕРНЄВ
(підпис) *(Ім'я, ПРІЗВИЩЕ)*

Керівник
кваліфікаційної роботи _____ Віктор САГАЙДАК
(підпис) *(Ім'я, ПРІЗВИЩЕ)*

РЕФЕРАТ

Текстова частина кваліфікаційної роботи на здобуття освітнього ступеня магістра: 57 стор., 35 рис., 35 джерел.

Мета роботи – розробка інтелектуальної моделі управління системою «розумного будинку», здатної адаптуватися до поведінкових патернів користувача на основі аналізу зібраних даних.

Об’єкт дослідження – процеси автоматизованого управління програмно-технічними засобами в кіберфізичних системах типу «розумний будинок».

Предмет дослідження – моделі, алгоритми та програмно-апаратні засоби аналізу користувацької поведінки для формування адаптивних сценаріїв керування компонентами системи «розумного будинку».

Короткий зміст роботи:

У роботі досліджено сучасні підходи до побудови систем «розумного будинку», проаналізовано архітектуру IoT-рішень, методи збору та обробки даних від сенсорів, а також питання кібербезпеки при передаванні інформації. Обґрунтовано використання технологій ZigBee, MQTT, TLS, Home Assistant, Docker та InfluxDB для побудови локальної програмно-апаратної платформи. Розроблено трирівневу модель управління Smart Home на основі предиктивної аналітики, запропоновано алгоритм прийняття рішень для адаптивного керування та виконано моделювання поведінкових патернів користувача із застосуванням аналізу часових рядів і ймовірнісних методів.

КЛЮЧОВІ СЛОВА: ІНТЕРНЕТ РЕЧЕЙ, РОЗУМНИЙ БУДИНОК, SMART HOME, КІБЕРФІЗИЧНІ СИСТЕМИ, ПРЕДИКТИВНА АНАЛІТИКА, КОРИСТУВАЦЬКА ПОВЕДІНКА, ZIGBEE, MQTT, TLS, HOME ASSISTANT, IOT-СИСТЕМИ.

ABSTRACT

The text part of the qualifying work for obtaining a master's degree: 57 pages, 35 figures, 35 sources.

Purpose – to develop an intelligent control model for a smart home system capable of adapting to user behavior patterns based on the analysis of collected data.

The object of research is the processes of automated control of software and hardware components in cyber-physical smart home systems.

The subject of research is models, algorithms, and software-hardware tools for analyzing user behavior in order to form adaptive control scenarios for smart home system components.

Summary of work:

The work examines modern approaches to the development of smart home systems, analyzes the architecture of IoT solutions, methods of collecting and processing sensor data, and cybersecurity issues related to data transmission. The use of ZigBee, MQTT, TLS, Home Assistant, Docker, and InfluxDB technologies for building a local software and hardware platform is substantiated. A three-level Smart Home control model based on predictive analytics is developed, a decision-making algorithm for adaptive control is proposed, and user behavior patterns are modeled using time series analysis and probabilistic methods.

KEYWORDS: INTERNET OF THINGS, SMART HOME, CYBER-PHYSICAL SYSTEMS, PREDICTIVE ANALYTICS, USER BEHAVIOR, ZIGBEE, MQTT, TLS, HOME ASSISTANT, IOT SYSTEMS.

ЗМІСТ

ВСТУП

Сучасний розвиток Інтернету речей (IoT) та кіберфізичних систем суттєво впливає на повсякденне життя людини, зокрема через широке впровадження технологій «розумного будинку», зумовлює актуальність досліджень у сфері інтелектуального управління такими системами. Незважаючи на значний прогрес, більшість існуючих рішень Smart Home функціонують на основі жорстко заданих сценаріїв або потребують постійного ручного налаштування, що обмежує їх гнучкість та ефективність. Створення моделей, здатних до проактивного управління на основі аналізу поведінки користувача, є необхідним кроком для підвищення енергоефективності та комфорту сучасних осель. Актуальність роботи підтверджується тим, що існуючі комерційні екосистеми (наприклад, Google Home, Amazon Alexa, Apple HomeKit) та відкриті платформи (Home Assistant) здебільшого реалізують реактивну автоматизацію на основі заздалегідь визначених сценаріїв. Проте, вони не забезпечують повноцінного самонавчання та проактивного управління системою на основі аналізу користувацької поведінки в реальному часі. Обґрунтування впровадження таких адаптивних моделей є предметом даного дослідження.

Актуальність роботи зумовлена потребою переходу систем «розумного будинку» від простого виконання заданих сценаріїв до інтелектуального та адаптивного управління. Більшість сучасних Smart Home-рішень потребують ручного налаштування і не завжди враховують поведінкові звички користувача, тому розробка моделі на основі IoT, предиктивної аналітики та безпечної обробки даних дозволяє підвищити комфорт, енергоефективність і автономність побутових кіберфізичних систем.

Аналіз існуючих підходів до управління «розумними будинками» свідчить, що більшість систем базуються на попередньо визначених алгоритмах і недостатньо враховують динаміку поведінки користувачів. Існуюча наукова база потребує доопрацювання в частині створення адаптивних моделей, що здатні самостійно інтерпретувати дії мешканців та захищати їхні персональні дані.

Інформаційною базою дослідження стали сучасні наукові публікації у сфері IoT, кіберфізичних систем та машинного навчання, а також технічна документація та стандарти передачі даних у розподілених системах.

Об’єктом дослідження є процеси автоматизованого управління програмно-технічними засобами в кіберфізичних системах типу «розумний будинок».

Предметом дослідження є моделі та алгоритми аналізу даних про поведінку користувача, що забезпечують формування адаптивних сценаріїв керування компонентами системи.

Метою роботи є розробка інтелектуальної моделі управління підсистемами «розумного будинку», здатної адаптуватися до поведінкових патернів користувача на основі аналізу зібраних даних.

Завдання дослідження:

- проаналізувати сучасні підходи та архітектурні особливості систем Smart Home;
- дослідити методи збору та попередньої обробки даних від сенсорів і виконавчих пристроїв;
- обґрунтувати вибір програмно-апаратних засобів та протоколів зв’язку (MQTT/TLS) для реалізації моделі;
- розробити трирівневу модель аналізу поведінки користувача;
- створити алгоритм прийняття рішень для адаптивного та проактивного управління;
- провести моделювання роботи системи та оцінити її енергоефективність.

У роботі використано такі методи дослідження, як методи системного аналізу, теорії алгоритмів, математичної статистики для обробки часових рядів, а також методи машинного навчання для прогнозування поведінкових сценаріїв [32].

Наукова новизна отриманих результатів полягає в удосконаленні моделі управління «розумним будинком» шляхом інтеграції предиктивної аналітики та механізмів безпечної передачі даних, що забезпечує перехід від реактивного до проактивного керування.

Практичне значення роботи полягає у можливості застосування розробленої моделі та програмних модулів під час проєктування IoT-контролерів, що дозволяє знизити енергоспоживання на 15–20% та забезпечити автоматизацію побутових процесів без постійного втручання користувача.

РОЗДІЛ 1. АНАЛІЗ СУЧАСНОГО СТАНУ ТА МЕТОДІВ УПРАВЛІННЯ В СИСТЕМАХ «РОЗУМНИЙ БУДИНОК»

1.1 Огляд архітектури та функціональних можливостей систем «розумного будинку»

Сучасний етап технологічного розвитку характеризується переходом від автономних пристроїв до цілісних кіберфізичних систем (КФС), де програмні алгоритми інтегровані з фізичними процесами у реальному часі. У побутовому секторі ця концепція реалізується через системи «розумного будинку», які базуються на технологіях Інтернету речей (IoT). Такі системи об’єднують програмно-технічні засоби, стаціонарні та мобільні пристрої, а також інтерфейси та протоколи їх взаємодії в єдину мережеву інфраструктуру.

В основі «розумного будинку» лежить ідея створення інтелектуального середовища, де кожен об'єкт – від простого сенсора до складного побутового приладу – має здатність передавати дані мережею без безпосередньої участі людини. На відміну від традиційної автоматизації, кіберфізичний підхід передбачає не лише виконання сценаріїв, а й глибоке опрацювання інформації за допомогою математичних моделей для адаптації системи до потреб мешканців.

Аналізуючи архітектурну побудову сучасних IoT-рішень, доцільно спиратися на ієрархічну модель, яка включає етапи проміжної обробки даних. Аналізуючи архітектурну побудову сучасних IoT-рішень, доцільно спиратися на ієрархічну модель, яка включає етапи проміжної обробки даних. Згідно з сучасними підходами, зокрема за методологією AltexSoft, класична структура трансформується у багаторівневу систему [1], що представлена на рис. 1.1:

Рис. 1.1 – Ієрархічна архітектура системи «розумного будинку» з елементами аналітики

Щоб краще зрозуміти, як працюють сучасні системи Smart Home, варто розглянути їхню архітектуру як послідовність взаємопов’язаних рівнів. Кожен із них виконує свою роль, але разом вони формують цілісну та ефективну екосистему.

Починається все з рівня речей (Things / Perception Layer). Це фізичні об'єкти, оснащені датчиками, які збирають дані про активність, енергоспоживання та мікроклімат. Саме тут реальні фізичні явища перетворюються на цифрові сигнали, з якими система може працювати далі.

Далі інформація переходить на рівень збору та попередньої обробки (Data Acquisition). Тут дані агрегуються, структуруються та готуються до подальшого аналізу, що є особливо важливо для систем та працюють із часовими рядами, де точність і впорядкованість інформації мають вирішальне значення.

Наступний крок – рівень межових обчислень (Edge/Fog Computing). У цьому шарі виконується первинна аналітика безпосередньо поблизу джерела даних. Такий підхід дозволяє значно зменшити затримки у роботі системи та забезпечити її автономність навіть у разі втрати зв’язку з хмарою.

Після цього дані передаються через мережевий рівень. Він охоплює як локальні, так і глобальні комп’ютерні мережі, включаючи Інтернет і відповідні протоколи

взаємодії. Основне завдання цього рівня – гарантувати надійну та безпечну передачу інформації між усіма компонентами системи.

Завершальним етапом є прикладний рівень (Cloud Analytics / управління). Саме тут зосереджені основні обчислювальні ресурси: за допомогою методів машинного навчання система аналізує накопичені дані, формує довгострокові прогнози та створює моделі поведінки користувача, що дозволяє зробити управління максимально адаптивним.

Сучасні функціональні можливості системи «розумного будинку» забезпечують виконання широкого спектру задач:

- адаптивне управління дає можливість автоматичного налаштування параметрів середовища на основі виявлених закономірностей;
- енергоефективність базується на оптимізації споживання ресурсів завдяки використанню енергоефективних технологій обчислень;
- безпека та конфіденційність дає захист персональних даних та моніторинг стану об’єкта за допомогою безпечних та автономних методів опрацювання інформації.

Таким чином, архітектура «розумного будинку» є складною системою, ефективність якої залежить від інтеграції сучасних методів аналітики та здатності системи до самостійної адаптації під мінливу поведінку користувача.

1.2 Порівняльний аналіз методів управління: від статичних сценаріїв до інтелектуальних моделей

Ефективність функціонування системи «розумного будинку» безпосередньо залежить від обраного методу управління її компонентами. Історично розвиток таких систем пройшов шлях від простої автоматизації окремих процесів до створення складних інтелектуальних екосистем. Для обґрунтування необхідності впровадження моделі на основі аналітики користувацької поведінки, необхідно провести порівняльний аналіз існуючих підходів до управління.

Традиційні підходи до автоматизації в системах Smart Home базуються на двох основних принципах: ручному керуванні через інтерфейси користувача та використанні жорстко заданих алгоритмів (сценаріїв).

1. Ручне керування передбачає безпосередню взаємодію користувача із системою через мобільні додатки, вебінтерфейси або настінні панелі. Такий підхід забезпечує повний контроль, але має суттєві недоліки:
   - Високе навантаження на користувача, тобто це необхідність постійного втручання для налаштування параметрів (освітлення, температури), що суперечить концепції «розумного» середовища.
   - Відсутність проактивності через те, як система реагує лише на команди, не передбачаючи потреби мешканців.
2. Жорстко задані алгоритми (сценарії) базуються на логіці «If-This-Then-That» (IFTTT) [29]. Наприклад: «Якщо спрацював датчик руху, увімкнути світло». Незважаючи на ширше впровадження (що підвищує життєвий цикл програмно-технічних засобів), цей підхід також має недоліки:
   - Обмежена гнучкість – сценарії не адаптуються до змінних умов та не враховують індивідуальні вподобання користувачів.
   - Складність налаштування – створення складних взаємозв’язків між пристроями потребує від користувача технічних навичок та часу.
   - Відсутність навчання – система не аналізує історичні дані та не вдосконалює свою роботу.

Проведений аналіз стає ще більш переконливим, якщо звернутися до огляду найпоширеніших комерційних і відкритих рішень у сфері Smart Home. Саме вони на практиці демонструють, як реалізуються сучасні підходи до автоматизації.

Одними з найвідоміших є Google Home та Amazon Alexa. Дані системи працюють на основі хмарних технологій і розпізнавання мовлення [27], що дозволяє легко об’єднувати різноманітні програмно-технічні засоби в єдину екосистему. Їх сильна сторона – висока сумісність пристроїв і зручність у використанні. Водночас є й обмеження: автоматизація тут здебільшого залежить від

ручного налаштування сценаріїв (IFTTT), тобто система реагує на події, а не передбачає їх.

Інший підхід пропонує Apple HomeKit, де акцент зроблено на безпеці та конфіденційності [28]. Завдяки локальній обробці даних і наскрізному шифруванню (Security by Design) користувач отримує стабільну роботу навіть без доступу до інтернету. Проте така модель має свою ціну – перелік сумісних пристроїв є обмеженим, а алгоритми управління залишаються доволі жорстко визначеними.

Окремо варто звернути увагу на Home Assistant – відкрите рішення, яке встановлюється локально [13], наприклад, на Raspberry Pi. Воно приваблює максимальною гнучкістю, підтримкою великої кількості протоколів і пристроїв, а також повним контролем над даними. Однак така свобода потребує відповідного рівня підготовки, тобто налаштування вимагає технічних навичок, а готові моделі адаптивного управління за замовчуванням відсутні.

Проведений аналіз існуючих аналогів («Google Home», «Amazon Alexa», «Apple HomeKit» та «Home Assistant») підтверджується оглядом найбільш поширених комерційних та відкритих рішень Smart Home. Спільними недоліками є відсутність самонавчання, залежність від ручних правил та низька адаптивність.

Узагальнена схема еволюції методів управління в системах «розумного будинку»: від ручного до предиктивного, представлена на рис. 1.2:

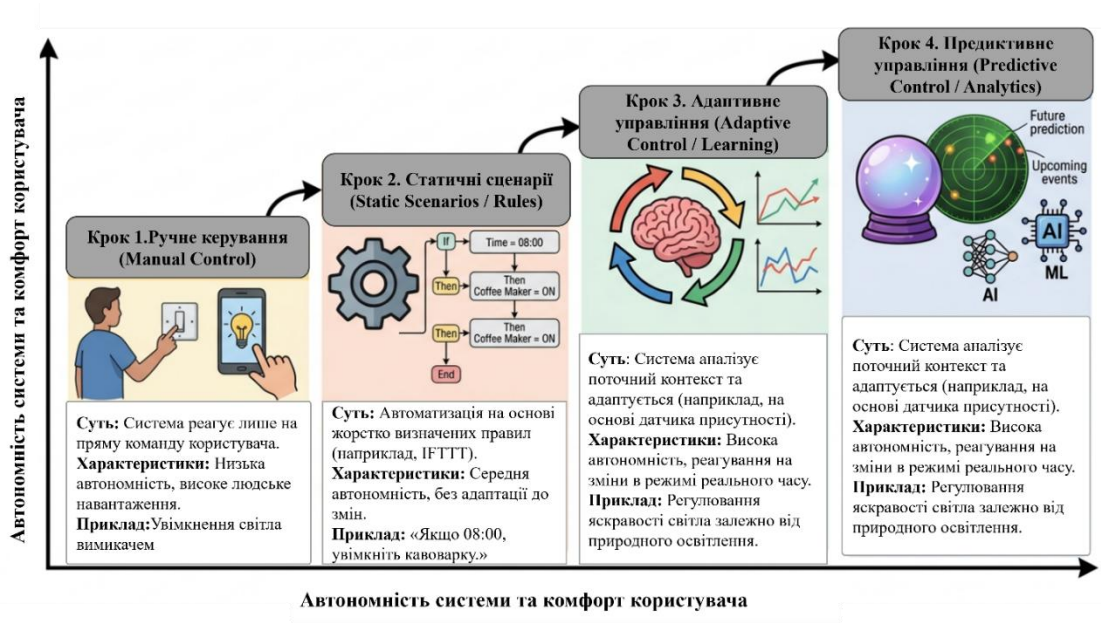


Рис. 1.2 – Еволюція методів управління в системах «розумного будинку»: від ручного до предиктивного

На наведеній схемі (див. рис. 1.2) візуалізовано етапи технологічного розвитку та перехід до предиктивного управління. Кожна сходинка демонструє зростання рівня автономності системи та комфорту користувача при впровадженні методів аналітики великих масивів даних. Діаграма наочно демонструє, як саме предиктивне управління нейтралізує недоліки застарілих «жорстких сценаріїв», пропонуючи проактивний підхід, де система передбачає потреби мешканців ще до того, як вони будуть висловлені, і дозволяє реалізувати підхід «Security by Design» та виконати вимоги щодо конфіденційності при опрацюванні персональних даних.

Перехід від статичних сценаріїв до інтелектуальних моделей управління є логічним етапом розвитку IoT-систем. Інтелектуальні моделі, на відміну від жорстких алгоритмів, базуються на методах математичного моделювання обчислювальних процесів та аналітиці великих масивів даних.

Застосування методів аналізу користувацької поведінки відкриває для системи значно ширші можливості, ніж може здатися на перший погляд. Йдеться не просто про збір даних, а про їх осмислене використання для підвищення комфорту та ефективності.

Насамперед система вміє виявляти патерни. Вона аналізує накопичені історичні дані й поступово «розуміє» звички користувача – наприклад, коли люди зазвичай бувають вдома або яку температуру вважають комфортною у різний час доби.

На основі цього з’являється ще одна важлива можливість – прогнозування потреб. Система не просто реагує, а випереджає події: вона може передбачити дії мешканців і заздалегідь налаштувати параметри середовища так, щоб вам було максимально зручно.

У підсумку це природно переходить до оптимізації ресурсів. Наприклад, коли вдома нікого немає, система автоматично знижує опалення або інші витрати енергії.

Такий підхід не лише економить кошти, а й сприяє впровадженню більш енергоефективних технологій у повсякденне життя.

Проведення порівняльного аналізу вказує на те, що інтелектуальні моделі на основі аналітики поведінки мають значні переваги над традиційними підходами в частині гнучкості, проактивності та енергоефективності. Водночас, їх впровадження потребує використання складніших програмно-апаратних засобів та врахування стандартів безпеки та конфіденційності при опрацюванні персональних даних.

Таким чином, перехід від статичних сценаріїв до інтелектуальних моделей є необхідним кроком для створення справді адаптивних систем «розумного будинку», які здатні підвищити комфорт та енергоефективність житлового простору.

1.3 Огляд технологій збору та обробки даних про поведінку користувачів

Ефективність предиктивної моделі управління «розумним будинком» критично залежить від якості, повноти та частоти отримання даних про активність користувачів. Процес збору даних у таких системах реалізується через розгалужену мережу IoT-сенсорів, що фіксують параметри середовища (температура, освітленість) та дії мешканців (відкриття дверей, рух у кімнатах, використання електроприладів).

Основним викликом при побудові інтелектуальної системи є вибір протоколу зв'язку, який би забезпечував стабільну передачу даних від великої кількості кінцевих пристроїв при мінімальному енергоспоживанні. На основі аналізу сучасних бездротових технологій, найбільш адаптованим для збору поведінкових даних є протокол ZigBee (стандарт IEEE 802.15.4) [3].

На відміну від Wi-Fi, що характеризується високим енергоспоживанням та топологією «зірка», ZigBee базується на Mesh-топології (рис. 1.3) [5], що дозволяє кожному пристрою з постійним живленням (роутеру) виступати ретранслятором сигналу, що гарантує цілісність даних у великих приміщеннях. Для навчання предиктивних алгоритмів це має вирішальне значення, оскільки втрата навіть

невеликого пакету даних про переміщення користувача може призвести до помилкового прогнозу моделі.

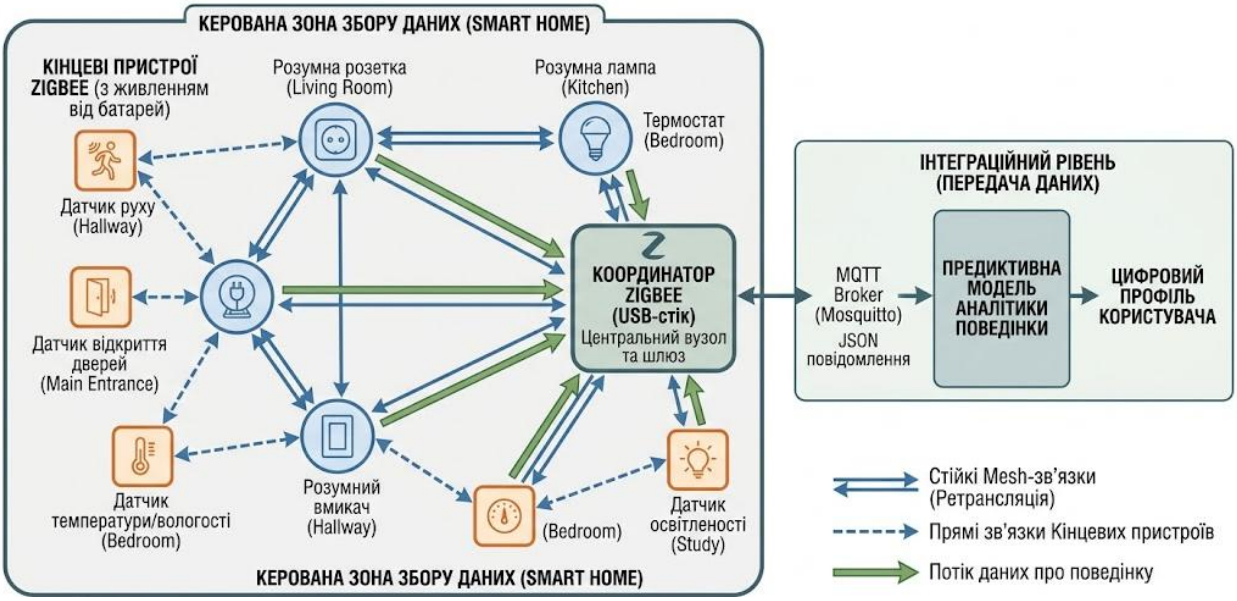


Рис. 1.3 – Схема Mesh-топології мережі ZigBee: надійний збір даних про поведінку користувача

Процес обробки зібраних даних у запропонованій моделі проходить через три ключові рівні:

1. Рівень сприйняття (Perception Layer). Кінцеві ZigBee-сенсори фіксують подію (наприклад, спрацювання датчика присутності).
2. Інтеграційний рівень (Integration Layer). Дані через ZigBee-координатор потрапляють до програмного шлюзу (наприклад, Zigbee2MQTT) [11]. Тут відбувається конвертація специфічних стеків протоколу у легковагий формат повідомлень JSON.
3. Рівень брокера (Transport Layer). Використання протоколу MQTT дозволяє передавати дані за моделлю «видавець-передплатник» (рис. 1.4). Предиктивний модуль виступає передплатником на певні топіки, отримуючи дані миттєво після їх появи в мережі [7].

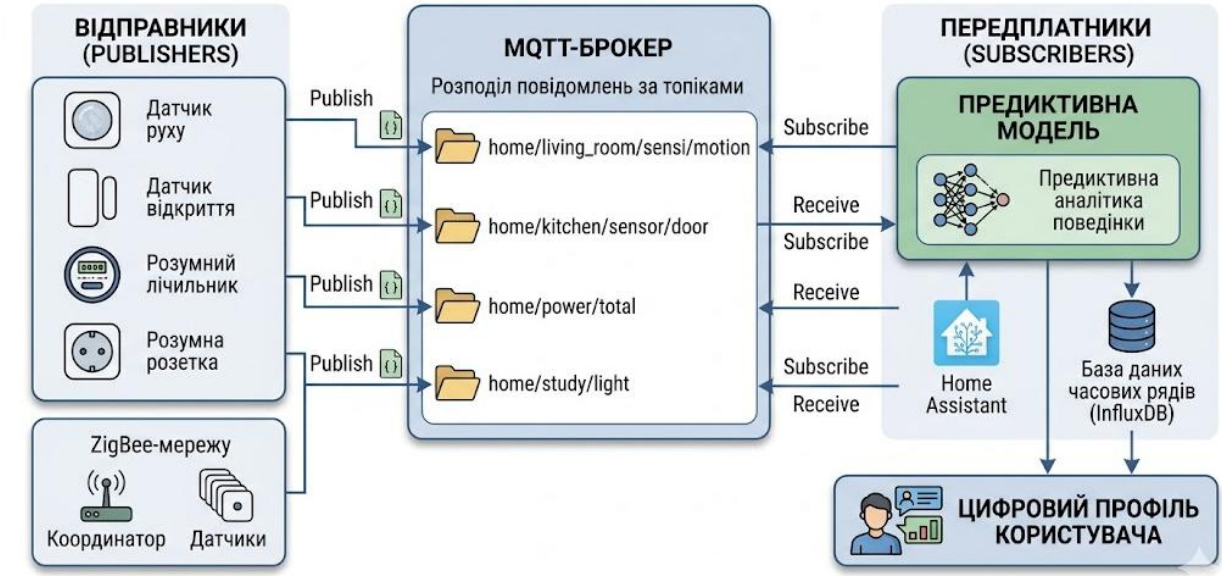


Рис. 1.4 – Схема передачі даних за протоколом MQTT (Publish/Subscribe)

Основними джерелами інформації для аналізу поведінки користувача в межах даної моделі визначено:

- Сенсори активності та присутності, тобто PIR-датчики руху та датчики відкриття (геркони). Опрацювання інформації від мережі таких датчиків дає змогу будувати теплові карти активності та визначати траєкторію переміщення.
- Лічильники енергоспоживання (Smart Meters) через використання методів математичного моделювання дозволяє реалізувати метод NILM (Non-Intrusive Load Monitoring) для ідентифікації працюючих приладів за загальним профілем навантаження.
- Інтелектуальні пристрої (Smart Appliances) через побутову техніку з IoT-інтерфейсами, що генерує масив даних про режими роботи та індивідуальні вподобання мешканців щодо мікроклімату.

Зібрані дані характеризуються високою гетерогенністю та зашумленістю, що потребує етапу попередньої обробки. Використання відкритого стеку Zigbee2MQTT разом із брокером повідомлень дозволяє уніфікувати дані від датчиків різних виробників (Xiaomi, Aqara, Sonoff) в єдиний інформаційний потік.

Узагальнена схема збору та обробки даних [10], що використовуються для формування цифрового профілю користувача, представлена на рис. 1.5.

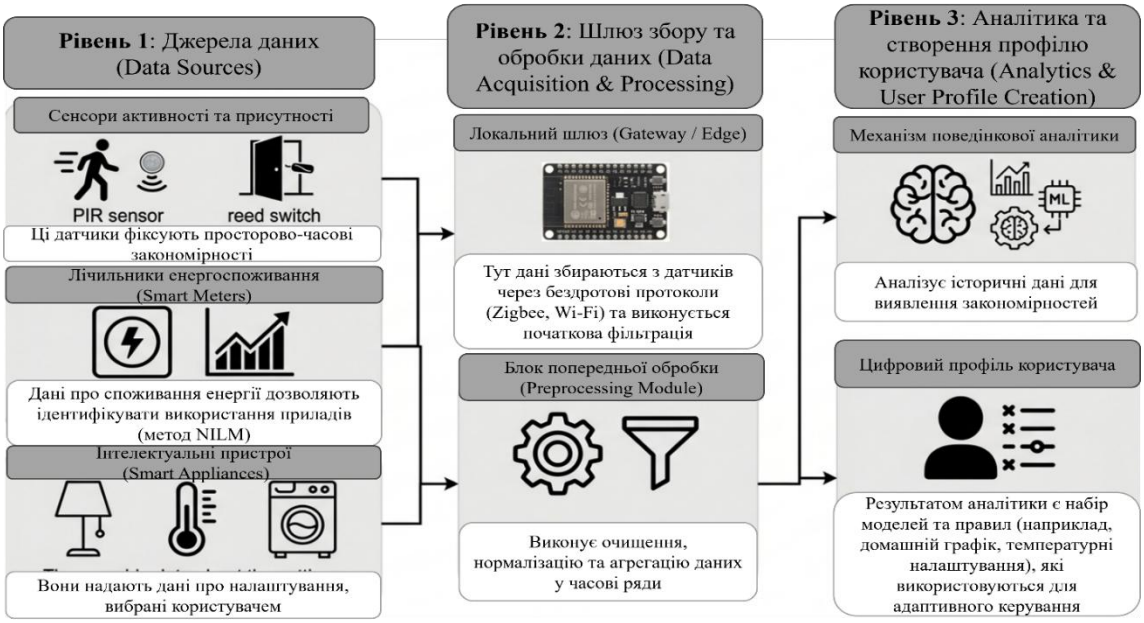


Рис. 1.5 – Схема збору та обробки даних про поведінку користувача: від сенсора до цифрового профілю

На наведеній схемі візуалізовано потік інформації, що складається з трьох ключових етапів. На першому етапі (Data Sources) відбувається збір даних від фізичних пристроїв. На другому (Data Acquisition & Processing) дані агрегуються шлюзом, де проходять фільтрацію та нормалізацію. На третьому (Analytics & User Profile Creation) оброблені часові ряди аналізуються модулем поведінкової аналітики для формування предиктивних моделей, що складають цифровий профіль користувача.

Проведений аналіз показує, що комплексна інтеграція ZigBee-мережі та архітектури MQTT створює надійну інформаційну базу для розробки адаптивних моделей управління «розумним будинком».

1.4 Аналіз загроз кібербезпеці та протоколів захисту інформації в IoT-мережах

Впровадження моделей управління «розумним будинком» на основі аналітики поведінки створює значні ризики для конфіденційності. Системи IoT генерують та передають великі масиви чутливих даних (графіки перебування вдома, активність приладів, геопозиція), компрометація яких може призвести до

втручання в приватне життя. Специфіка таких мереж (обмежені ресурси пристроїв та бездротові канали) робить їх вразливими до широкого спектру кіберзагроз.

Основні загрози за рівнями архітектури IoT включають:

Рівень перцепції (датчики) – фізичне втручання та «клонування» пристроїв для введення неправдивих даних про активність.

Мережевий рівень (передача) – перехоплення даних (Eavesdropping), атаки «людина посередині» (MitM) та DoS-атаки на шлюз.

Прикладний рівень (аналітика) – ін’єкція даних для маніпуляції алгоритмами аналізу поведінки та крадіжка цифрового профілю користувача.

Для нейтралізації цих загроз у роботі запропоновано багаторівневу модель захисту, що базується на комбінації стандартів ZigBee та TLS.

Захист на рівні локальної мережі (ZigBee Security). Оскільки збір даних у системі базується на протоколі ZigBee, безпека забезпечується на рівні стандарту IEEE 802.15.4. Використовується симетричне шифрування AES-128 для захисту даних між датчиками та координатором. Для запобігання несанкціонованому приєднанню пристроїв у системі реалізовано режим обмеженого часу спарювання (Joining time), що мінімізує ризик фізичного злому мережі [4].

Захист рівня передачі даних (MQTT + TLS).

Для безпечної взаємодії між локальним шлюзом та аналітичним модулем обґрунтовано використання зв'язки протоколів MQTT v5.0 та TLS v1.3 (рис. 1.6) [12].

Використання TLS (Transport Layer Security) поверх MQTT дозволяє забезпечити:

- конфіденційність через шифрування всього трафіку (даних від сенсорів та команд управління) за допомогою стійких алгоритмів (AES-256);
- цілісність через гарантію того, що дані про поведінку не були змінені під час передачі до бази даних;
- автентифікація використання цифрових сертифікатів (X.509) для перевірки справжності як шлюзу, так і клієнтів аналітики.

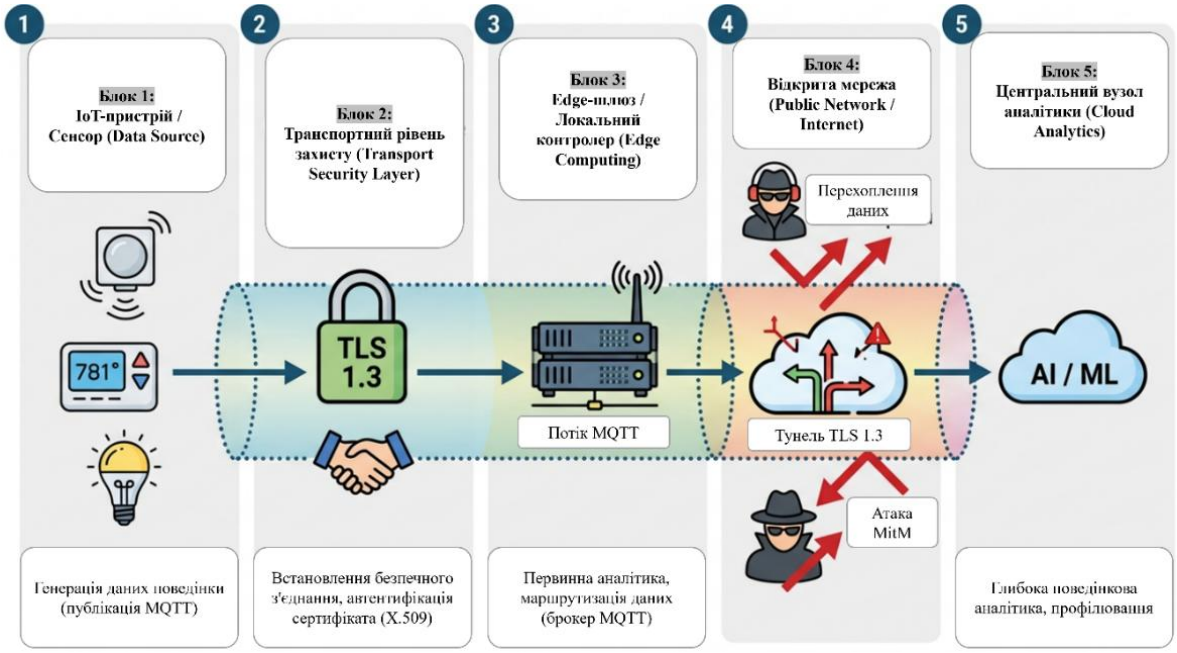


Рис. 1.6 – Схема забезпечення кібербезпеки при передачі даних про поведінку користувача

На наведеній схемі візуалізовано багаторівневий захист від шифрування на сенсорному рівні до створення захищеного тунелю TLS між Edge-шлюзом та хмарною/серверною частиною. Вибір TLS 1.3 є пріоритетним, оскільки він забезпечує швидше встановлення з’єднання (Zero Round-Trip Time) та відмову від застарілих вразливих алгоритмів.

Таким чином, використання стеку ZigBee (AES-128) + MQTT v5.0 + TLS v1.3 дозволяє реалізувати підхід «Security by Design», гарантуючи захист цифрового профілю користувача на всіх етапах його формування.

РОЗДІЛ 2. ПРОЄКТУВАННЯ МОДЕЛІ УПРАВЛІННЯ НА ОСНОВІ ПРЕДИКТИВНОЇ АНАЛІТИКИ

2.1 Розробка структурної схеми трирівневої моделі управління

На основі проведеного в першому розділі аналізу існуючих екосистем та виявлених вразливостей IoT-мереж, у даному підрозділі розроблено структурну схему інтелектуальної моделі управління. Запропоноване рішення базується на трирівневій ієрархічній архітектурі, що дозволяє відокремити фізичний збір гетерогенних даних від процесів інтелектуальної обробки та забезпечення кібербезпеки.

Декомпозиція системи на окремі функціональні шари дозволяє реалізувати принцип незалежності програмних модулів та забезпечити високу масштабованість. Загальна структурна схема моделі представлена на рис. 2.1.

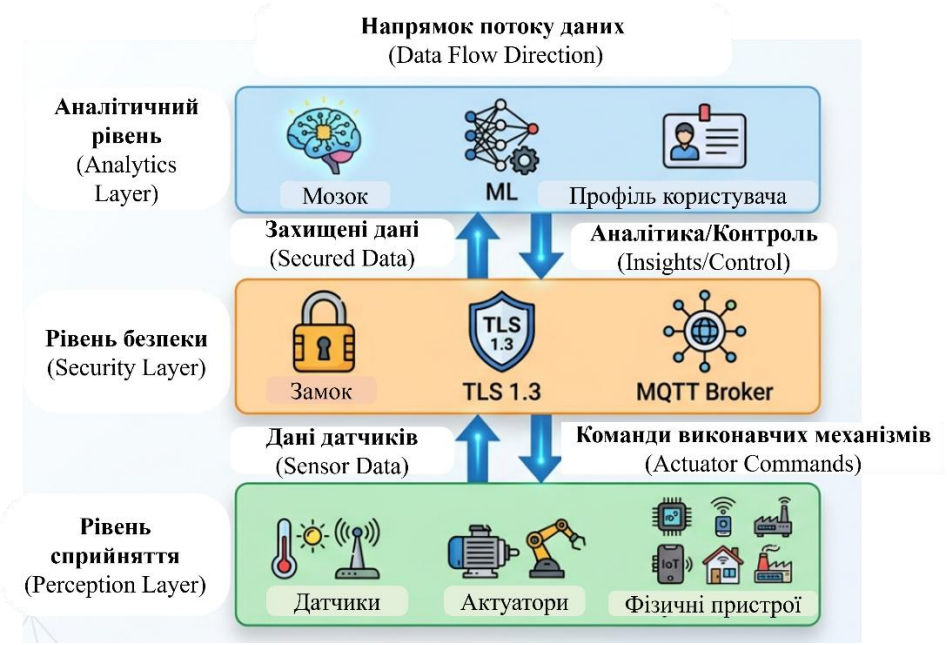


Рис. 2.1 — Структурна схема трирівневої моделі управління Smart Home на основі аналітики поведінки

Логічним центром системи є шар аналітики, розгорнутий на локальному Edge-шлюзі. Така архітектура дозволяє уникнути залежності від хмарних сервісів

(підхід Local Push), що критично для конфіденційності поведінкових даних. Згідно з розробленою моделлю, взаємодія між компонентами реалізується через наступні рівні:

1. Шар перцепції (Perception Layer)изабезпечує безпосередню взаємодію з фізичним середовищем об'єкта.тМережа гетерогенних IoT-сенсорів (руху, освітленості, відкриття дверей, розумних лічильників) та виконавчі пристрої (актуатори). Використовується ZigBee-координатор (наприклад, на базі чіпа CC2531), інтегрований у Edge-сервер. Це дозволяє об’єднувати пристрої різних виробників у єдину Mesh-мережу. Збір первинних даних про активність мешканців та стан середовища. Дані на цьому рівні є неструктурованими та передаються на наступний етап у форматі JSON-повідомлень.

2. Шар захисту та транспорту (Security & Transport Layer) виконує роль «захисного фільтра», забезпечуючи цілісність та конфіденційність обміну інформацією між фізичними пристроями та логічним ядром. Брокер повідомлень (Mosquitto MQTT), механізми шифрування TLS 1.3 та цифрові сертифікати X.509. Реалізація підходу «Security by Design» [12]. Шар створює захищений тунель, запобігаючи перехопленню даних та MitM-атакам. Для безпечного віддаленого доступу пропонується інтеграція сервісу Duck DNS та отримання SSL-сертифікатів через Let's Encrypt, що дозволяє користувачеві керувати системою через захищене HTTPS-з’єднання [9].

3. Шар аналітики та управління (Analytics & Control Layer), де відбувається інтерпретація зібраної інформації та прийняття проактивних рішень. Модуль предиктивної аналітики (AI/ML), база даних часових рядів (InfluxDB) та рушій автоматизації (Home Assistant Core). Для підвищення відмовостійкості, ізоляції компонентів та спрощення процесу оновлення системи, всі ключові сервіси аналітичного шару розгорнуто за допомогою технології контейнеризації.

Детальна архітектура програмного забезпечення на локальному Edge-шлюзі представлена на рис. 2.2.

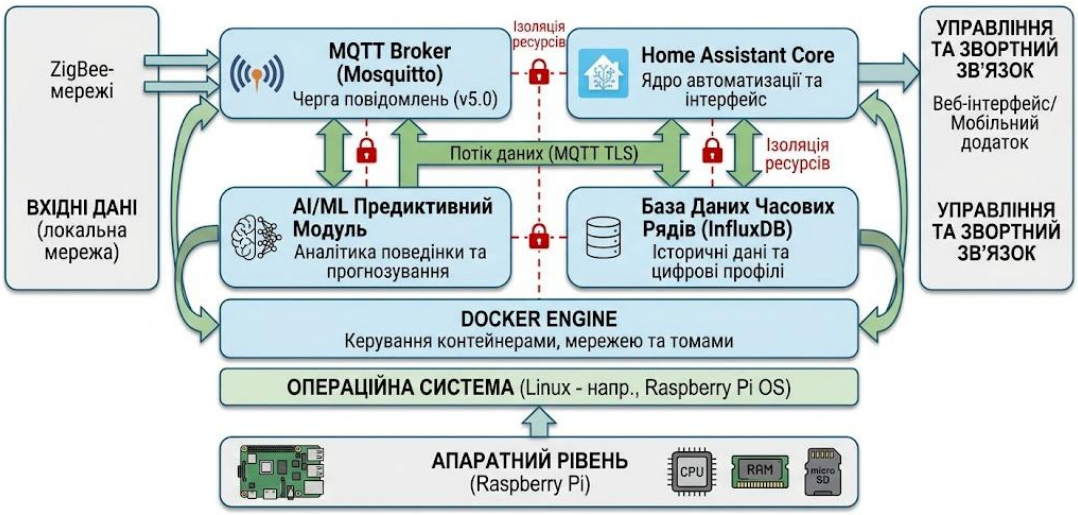


Рис. 2.2 – Схема контейнеризації сервісів на Edge-шлюзі (Docker)

На наведеній схемі візуалізовано (див. рис. 2.2) використання Docker Engine дозволяє запустити MQTT-брокер, базу даних InfluxDB та ядро Home Assistant в ізольованих контейнерах і гарантує, що ресурси апаратного забезпечення (Raspberry Pi) розподіляються ефективно [18], а критичні дані предиктивного модуля захищені від конфліктів між програмними залежностями. Такий підхід забезпечує стабільну роботу аналітичного шару навіть при одночасній обробці великої кількості запитів від гетерогенної мережі сенсорів[19].

Виявлення поведінкових патернів, формування цифрового профілю користувача та генерація команд управління. На відміну від статичних сценаріїв, цей шар працює проактивно, наприклад, система ініціює прогрів приміщення на основі прогнозованого часу повернення власника додому.

Взаємодія в моделі відбувається за циклічним принципом (рис. 2.1). Дані від шару перцепції проходять крізь шар захисту, де шифруються та автентифікуються. Після обробки в модулі аналітики, що працює в ізольованому контейнері, формується прогноз, і команда управління передається у зворотному напрямку до виконавчих пристроїв. Такий підхід забезпечує безперервне самонавчання системи та мінімізує втручання користувача в процеси управління.

2.2 Математичне моделювання поведінкових патернів користувача

Реалізація предиктивного управління потребує створення математичних моделей, здатних інтерпретувати дані від сенсорів та виявляти приховані закономірності у діях мешканців. Основна складність моделювання в системах Smart Home полягає у високій гетерогенності та асинхронності надходження інформації. Для вирішення цієї задачі обґрунтовано використання комбінації методів математичної статистики, аналізу часових рядів (ARIMA) та ймовірнісного моделювання.

Первинні дані, що надходять від шару перцепції через захищений TLS-тунель, розглядаються як множина гетерогенних часових рядів T_i:

$$T_i = \{(t_1, x_1), (t_2, x_2), ..., (t_k, x_k)\}$$

де t_k – момент часу вимірювання, x_k – значення фізичного параметра (для бінарних сенсорів $x_k \in \{0,1\}$, для аналогових $x_k \in R$).

Процес математичного моделювання поведінкових патернів представлено на рис. 2.3.

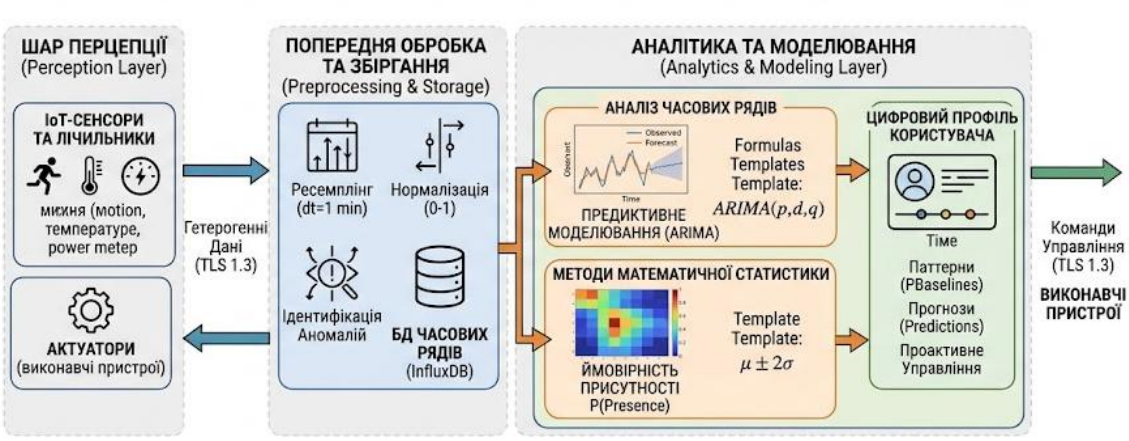


Рис. 2.3 – Візуалізація процесу математичного моделювання поведінкових патернів

З огляду на специфіку роботи PIR-сенсорів та ймовірність виникнення шумів, критично важливим є перетворення вхідних потоків у структурований вигляд. Дана задача вирішується на наступному рівні:

1. Етап попередньої обробки (Preprocessing Step). Оскільки дані генеруються по події, вони є нерівновіддаленими. На цьому етапі виконується:

– ресемплінг (Resampling), тобто приведення рядів до єдиного кроку Δt (наприклад, 1 хв).

– нормалізація, тобто масштабування значень для підвищення стабільності алгоритмів.
– фільтрація аномалій, тобто відсікання помилкових спрацювань PIR-датчиків.

2. Етап аналізу та моделювання (Analytics & Modeling Layer), де реалізовано поєднання двох підходів:

Методи математичної статистики використовуються для побудови ймовірнісних теплових карт присутності (рис. 2.4). Для кожної кімнати R_j та інтервалу доби H_m розраховується ймовірність перебування:

$$P(R_j|H_m) = \frac{1}{D}\sum_{d=1}^{D} I(R_{j,d,m})$$

де D – кількість днів спостереження, I – індикаторна функція присутності, (1– якщо зафіксовано рух, 0 – інакше), R_j – кімната (наприклад, кухня, спальня), H_m– година доби, $I\ (R_j, H_m, d)$ – індикатор (1 – якщо зафіксовано рух, 0 – якщо рух відсутній).

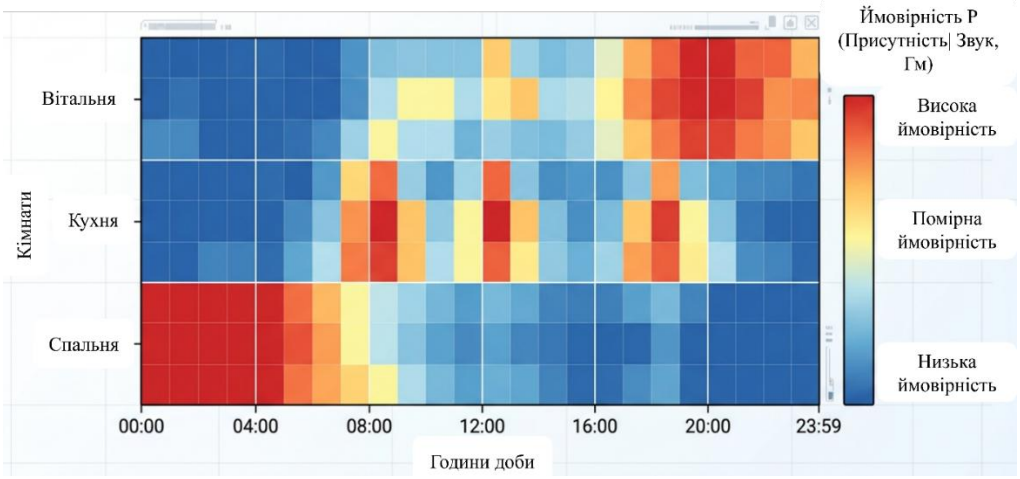


Рис. 2.4 – Ймовірнісна теплова карта присутності користувача

Аналіз часових рядів модель ARIMA(p,d,q) використовується для короткострокового прогнозування ймовірності подій (рис. 2.5), що дозволяє системі, наприклад, ініціювати прогрів кавоварки або ввімкнення опалення за 10–15 хвилин до звичного моменту активності користувача [23].

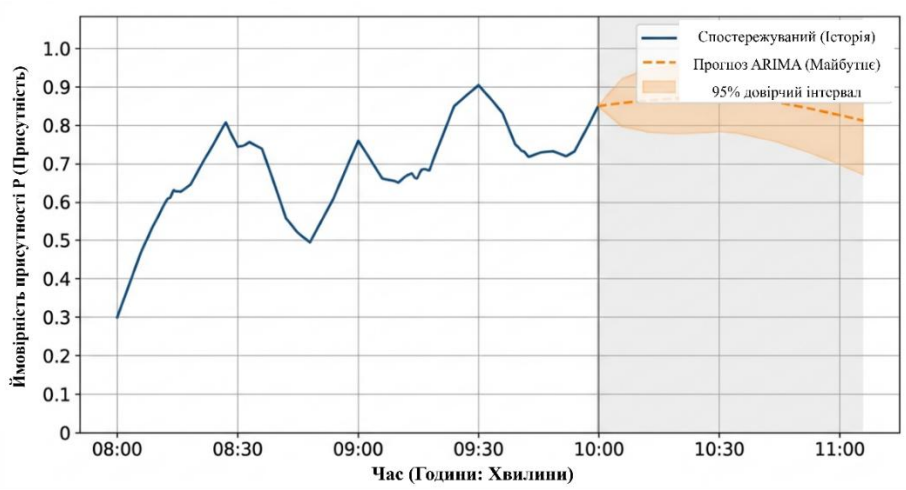


Рис. 2.5 – Візуалізація результатів прогнозування ARIMA для ймовірності присутності

3. Етап формування цифрового профілю (User Profile Creation), де результати статистичного аналізу (теплові карти) та короткострокові прогнози (ARIMA) об’єднуються в Цифровий профіль користувача.

Для моделювання переходів між станами (наприклад, «Кухня» →«Вітальня») додатково застосовуються ланцюги Маркова, що дозволяє розрахувати матрицю ймовірностей переходів, де кожне значення p_{ij} визначає шанс того, що користувач перейде в кімнату j після перебування в кімнаті i.

Комбінація цих методів забезпечує перехід від реактивного управління до проактивного, підвищуючи комфорт та енергоефективність системи при повному забезпеченні конфіденційності даних на локальному Edge-шлюзі.

2.3 Обґрунтування вибору програмно-апаратних засобів та протоколів взаємодії компонентів

Реалізація трирівневої моделі управління, що базується на предиктивній аналітиці, потребує обґрунтованого вибору засобів, здатних забезпечити збір, захищену передачу та високопродуктивне опрацювання масивів даних у реальному часі. Вибір компонентів здійснювався за критеріями сумісності, енергоефективності та відповідності підходу Security by Design.

1. Обґрунтування вибору програмної платформи (Analytics Layer) для реалізації шару аналітики та управління (Edge Gateway) обрано платформу Home Assistant (HA), розгорнуту в Docker-контейнерах на базі HAOS (Home Assistant Operating System). Даний вибір базується на наступних перевагах:

- Локальна обробка, де дані не передаються у хмарні сховища, що забезпечує мінімальні затримки та повну конфіденційність цифрового профілю користувача.
- Контейнеризація (Docker) дозволяє ізолювати AI/ML-модулі, MQTT-брокер та базу даних InfluxDB, що підвищує відмовостійкість системи (див. рис. 2.2).
- Висока інтегрованість через підтримку понад 2000 інтеграцій дозволяє об’єднувати гетерогенні IoT-пристрої від різних виробників у єдину екосистему.

2. Апаратне забезпечення Edge-шлюзу для стабільного функціонування предиктивного модуля та обробки часових рядів (див. рис. 2.5) обґрунтовано вибір одноплатного ПК Raspberry Pi 4 Model B (4GB RAM). Дана апаратна база забезпечує оптимальний баланс між продуктивністю чотирьохядерного процесора, низьким енергоспоживанням та можливістю підключення Zigbee-координаторів через USB-інтерфейс.

3. Вибір протоколів взаємодії та шлюзів (Integration Level) для подолання проблеми гетерогенності пристроїв та перетворення сирих даних у формат, придатний для аналітики, обрано програмний шлюз Zigbee2MQTT.

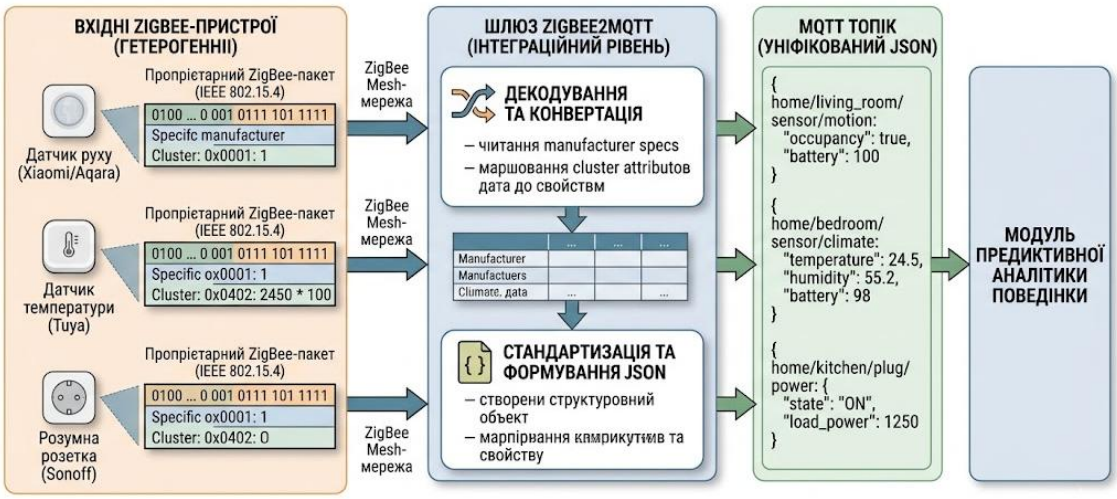


Рис. 2.6 – Схема перетворення пропрієтарних ZigBee-пакетів у JSON-повідомлення через шлюз Zigbee2MQTT

Як показано на рис. 2.6, використання Zigbee2MQTT дозволяє уніфікувати дані від датчиків різних виробників (Xiaomi, Tuya, Sonoff). Координатор отримує радіосигнал, а програмний шлюз публікує його в топік MQTT. Це критично для моделі аналітики, оскільки AI-модуль очікує дані в єдиному форматі JSON для подальшого аналізу.

Для захисту транспортного шару обрано наступні стандарти:

- MQTT v5.0 є основний транспортний протокол для шару перцепції, що характеризується асинхронністю та підтримкою властивостей користувача (User Properties).
- TLS v1.3обов'язкове шифрування всього MQTT-трафіку для нейтралізації загроз перехоплення та MitM-атак.
- HTTPS та Duck DNS є реалізація безпечного віддаленого доступу через динамічне оновлення доменної адреси та використання SSL-сертифікатів Let’s Encrypt, що дозволяє шифрувати веб-інтерфейс управління.

4. Вибір пристроїв шару перцепції (Perception Layer) Для збору гетерогенних даних обрано мережу сенсорів, що працюють за протоколом Zigbee 3.0, та універсальний координатор (наприклад, ConBee II або CC2531), що дозволяє створити енергоефективну Mesh-мережу з наступним складом обладнання:

- PIR-датчики руху та присутності фіксують просторово-часову активність для побудови теплових карт.
- Сенсори температури та вологості збирають дані про параметри мікроклімату для адаптивного управління.
- Розумні лічильники та розетки (Smart Meters) забезпечують збір часових рядів енергоспоживання для моделей короткострокового прогнозування.
- Актуатори (розумні лампи, термоголовки) виконують проактивні команди, сформовані системою на основі передбачених потреб користувача.

Таким чином, обраний програмно-апаратний стек утворює замкнену, безпечну та продуктивну платформу, яка є технічним підґрунтям для реалізації інтелектуальних алгоритмів управління.

Для повного розуміння механізмів взаємодії компонентів необхідно розглянути стек протоколів, що забезпечує функціонування системи на різних рівнях моделі OSI, що дозволяє чітко розмежувати зони відповідальності між фізичним обладнанням (ZigBee) та програмними аналітичними модулями (MQTT/Application).

Рис. 2.7 – Стек протоколів ZigBee в контексті ієрархічної архітектури управління

Як показано на рис. 2.7, нижні рівні (Physical та MAC) базуються на стандарті IEEE 802.15.4, що забезпечує низьку латентність та енергоефективність, необхідну для постійного збору поведінкових даних. Мережевий рівень відповідає за Mesh-

маршрутизацію, тоді як рівень застосунків (Application Layer) через шлюз Zigbee2MQTT інтегрується з протоколом MQTT [32]. Такий підхід гарантує, що ваша математична модель (ARIMA, ланцюги Маркова) отримує чисті, стандартизовані дані, незалежно від того, яким фізичним пристроєм вони були згенеровані.

2.4 Розробка алгоритму прийняття предиктивних рішень для адаптивного управління

У даному підрозділі розроблено алгоритм проактивного управління, який реалізує концепцію Cyber-Physical-Social Systems, поєднуючи AI/ML-моделі поведінки мешканців з принципами Security by Design. Запропонований алгоритм дозволяє перейти від реактивної автоматизації до адаптивного управління, що здатне до самонавчання на основі зворотного зв'язку.

Загальна логіка роботи алгоритму представлена на рис. 2.8. Він функціонує в режимі реального часу і включає наступні критичні етапи:

1. Етап ініціалізації та сприйняття (Perception Step) на основі протоколів, описаних у підрозділі 2.3, відбувається ініціалізація Zigbee-мережі. Як і в базових архітектурах IoT-управління (див. рис 2.6), кожен новий пристрій проходить процедуру автентифікації. Дані про активність мешканців шифруються TLS 1.3 і через MQTT-брокер надходять до Edge-шлюзу [20].

2. Попередня обробка та аналітика (Edge Computing Layer) через шлюз на базі Home Assistant виконує ресемплінг даних із кроком $\Delta t = 1$ хв та виявлення аномалій (правило $\mu \pm 2\sigma$, μ – середнє значення вибірки, σ – середньоквадратичне відхилення.). Очищені дані зберігаються в InfluxDB, створюючи базу для навчання AI/ML-модуля [24].

3. Прогнозування та прийняття рішень (Predictive Logic) є центральний блок алгоритму, де на основі цифрового профілю користувача (теплових карт присутності та моделей ARIMA) формується короткостроковий прогноз. Основна логіка прийняття рішення формалізується через перевірку порогових ймовірностей:

- Сценарій «Комфорт» реалізується таким чином, якщо P(Presence | R_j, H_m)>0,8, система ініціює проактивне ввімкнення кліматичної техніки за 20 хвилин до очікуваної появи користувача.
- Сценарій «Енергоефективність» реалізується таким чином, якщо P(Presence)<0,2, актуатори переводяться в режим енергозбереження ще до фактичного виходу користувача з зони [33].

Узагальнена логіка роботи системи предиктивного управління представлена у вигляді алгоритму, що складається з послідовних етапів (рис. 2.8).

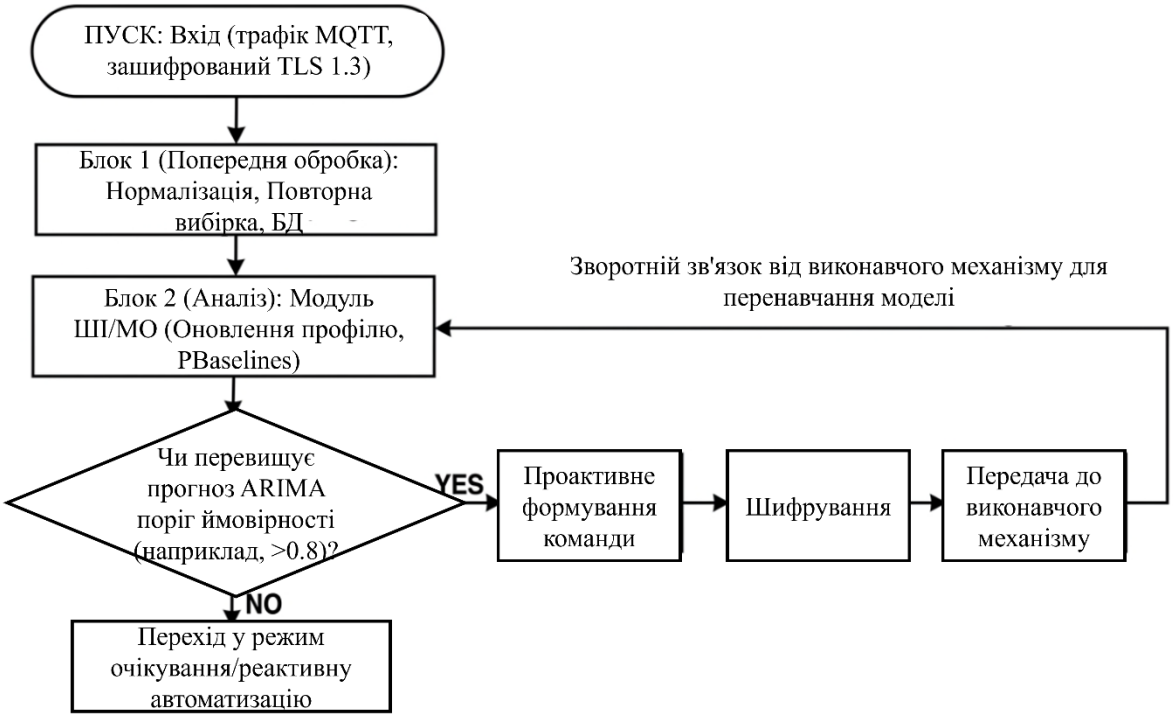


Рис. 2.8 – Блок-схема алгоритму прийняття предиктивних рішень Smart Home

Аналіз наведеної блок-схеми (рис. 2.8) наочно демонструє інтеграцію математичного та алгоритмічного апарату. Видно послідовність дій: від захищеного збору гетерогенних даних на рівні перцепції, їх аналізу та прогнозування ARIMA на рівні аналітики, до прийняття проактивного рішення та його зашифрованої передачі TLS 1.3 на виконавчий пристрій.

Особливістю алгоритму є наявність замкненого циклу зв'язку. Якщо мешканець скасовує автоматичну дію (наприклад, вимикає світло, увімкнене проактивно), сигнал «Зворотній зв'язок» надходить до ШІ-модуля для негайного донавчання моделі та підвищення точності прогнозів у майбутньому.

РОЗДІЛ 3. РЕАЛІЗАЦІЯ ТА ДОСЛІДЖЕННЯ ЕФЕКТИВНОСТІ РОЗРОБЛЕНОЇ МОДЕЛІ

3.1 Підготовка та розгортання власної IoT-платформи для управління розумним будинком

У межах даного дослідження використовується сервер, реалізований на базі міні-ПК, на якому інстальовано операційну систему Debian. На цьому сервері розгорнуто середовище віртуалізації Proxmox, що виконує функції гіпервізора та забезпечує можливість створення і керування віртуальними машинами та контейнерами. Один із таких контейнерів задіяний для інсталяції IoT-платформи Home Assistant.

Платформа Home Assistant забезпечує централізоване керування широким спектром датчиків і пристроїв розумного будинку незалежно від їх виробника або використовуваного протоколу зв’язку.

Home Assistant являє собою універсальне програмне рішення з відкритим кодом, призначене для інтеграції, моніторингу та автоматизації пристроїв інтернету речей. Вона функціонує локально, не потребує підключення до хмарних сервісів або постійного доступу до мережі Інтернет, і підтримує велику кількість протоколів зв’язку, зокрема: Wi-Fi, Zigbee, Z-Wave, Bluetooth, Bluetooth Mesh, інфрачервоний канал (IR) тощо.

Основні характеристики та переваги Home Assistant:

- Універсальність та інтеграційні можливості. Платформа підтримує велику кількість пристроїв різних виробників і стандартів зв’язку (зокрема Zigbee, Wi-Fi, Z-Wave, Bluetooth тощо), що забезпечує їх об’єднання в єдину систему керування.
- Зручність конфігурації. Наявність інтуїтивного вебінтерфейсу та готових інтеграцій значно спрощує процес підключення, налаштування та адміністрування пристроїв.

– Масштабованість і гнучкість. Система дозволяє легко додавати нові пристрої, сценарії автоматизації та кастомні компоненти без суттєвих технічних ускладнень.
– Активна спільнота та підтримка. Велика кількість користувачів і розробників забезпечує регулярні оновлення, документацію та готові рішення для типових задач.

Приклади сценаріїв використання:

– Система безпеки: інтеграція відеокамер, охоронної сигналізації, датчиків диму, відкриття дверей/вікон, руху та вібрації.
– Освітлення: управління яскравістю, кольоровою гамою, температурою світла, а також реалізація ефектів типу ambilight.
– Побутова техніка: контроль роботи кондиціонерів, роботів-пилососів, телевізорів, електрочайників та іншої техніки.
– Електропостачання: керування розумними розетками, вимикачами, реле, а також моніторинг споживання електроенергії через лічильники.
– Моніторинг параметрів середовища: відстеження температури, вологості, атмосферного тиску, рівня CO_2, а також виявлення протікань води.

Встановлення та мінімальні системні вимоги для розгортання Home Assistant на одноплатному комп’ютері (наприклад, Raspberry Pi) доцільно враховувати такі базові параметри:

– Оперативна пам’ять від 2 ГБ (рекомендовано 4 ГБ і більше) забезпечує стабільну роботу системи при одночасному обслуговуванні значної кількості пристроїв та автоматизацій.
– Накопичувач від 32 ГБ (рекомендовано SSD) необхідний для зберігання історичних даних, логів, резервних копій та конфігурацій; використання SSD підвищує надійність і швидкодію системи порівняно з microSD.
– Стабільне мережеве підключення бажано використання Ethernet для зменшення затримок і підвищення надійності взаємодії між компонентами системи.

– Актуальна версія програмного забезпечення регулярне оновлення Home Assistant забезпечує підтримку нових інтеграцій, покращення безпеки та продуктивності.

Застосування протоколів Zigbee та Wi-Fi у дослідженні:

– У межах дослідження протокол Zigbee використовується для підключення різноманітних пристроїв розумного будинку до центрального координатора через бездротову mesh-мережу. Його ключовими перевагами є низьке енергоспоживання, стабільність роботи та можливість масштабування мережі за рахунок великої кількості вузлів.
– Завдяки широкому поширенню та простоті конфігурації, Wi-Fi активно застосовується для підключення пристроїв розумного будинку. Цей протокол забезпечує високу пропускну здатність, що особливо важливо для пристроїв із великим обсягом переданих даних (наприклад, камер відеоспостереження).

Застосування цих двох протоколів дозволяє виконати порівняльний аналіз їх технічних характеристик, енергоефективності та доцільності використання в реальних умовах функціонування системи розумного будинку.

3.2. Логіка управління розумним будинком

Логіка організації управління розумним будинком на базі протоколу Zigbee у середовищі Home Assistant передбачає інтеграцію різноманітних пристроїв, зокрема датчиків, реле, вимикачів, термостатів та інших компонентів, що підтримують даний стандарт бездротового зв’язку.

Основні етапи реалізації логіки управління через Zigbee в Home Assistant:

1. Інтеграція з Zigbee-координатором через підключення координатора Zigbee до системи. Зазвичай це реалізується через USB-адаптер (наприклад, на базі чіпів CC2652, ConBee II або аналогічних), який виконує функцію центрального вузла мережі.
2. Підключення пристроїв до мережі після ініціалізації координатора здійснюється додавання кінцевих пристроїв. Home Assistant підтримує

автоматичне виявлення та підключення через інтеграції, такі як Zigbee2MQTT, ZHA (Zigbee Home Automation) або інші сумісні рішення.

3. Керування пристроями та моніторинг через інтегровані пристрої стають доступними в інтерфейсі Home Assistant, що дозволяє виконувати їх керування (увімкнення/вимкнення, налаштування параметрів) та здійснювати постійний моніторинг стану.
4. Реалізація автоматизованих сценаріїв, де платформа надає можливість створення складних сценаріїв автоматизації на основі подій, станів та умов. Наприклад, автоматичне ввімкнення освітлення при виявленні руху або адаптивне регулювання температури відповідно до показників сенсорів.
5. Інтеграція з іншими технологіями, де Home Assistant підтримує об’єднання пристроїв, що працюють за різними протоколами (Zigbee, Wi-Fi, Z-Wave тощо), що дозволяє реалізувати єдину централізовану систему управління.

Структурна схема розумного будинку базується на багаторівневій архітектурі, яка включає рівень пристроїв (датчики та виконавчі механізми), рівень передачі даних (мережеві протоколи Zigbee, Wi-Fi) та рівень обробки й управління (сервер із Home Assistant).

На нижньому рівні функціонують сенсори та виконавчі пристрої, які здійснюють збір даних і виконання команд. Середній рівень відповідає за передачу даних через бездротові мережі. Верхній рівень представлений серверною частиною, де реалізовано обробку даних, логіку автоматизації та інтерфейс користувача.

Така архітектура забезпечує гнучкість, масштабованість та надійність системи, а також дозволяє ефективно інтегрувати нові пристрої та сервіси без суттєвих змін у вже існуючій інфраструктурі.

3.3. Схема побудови розумного будинку

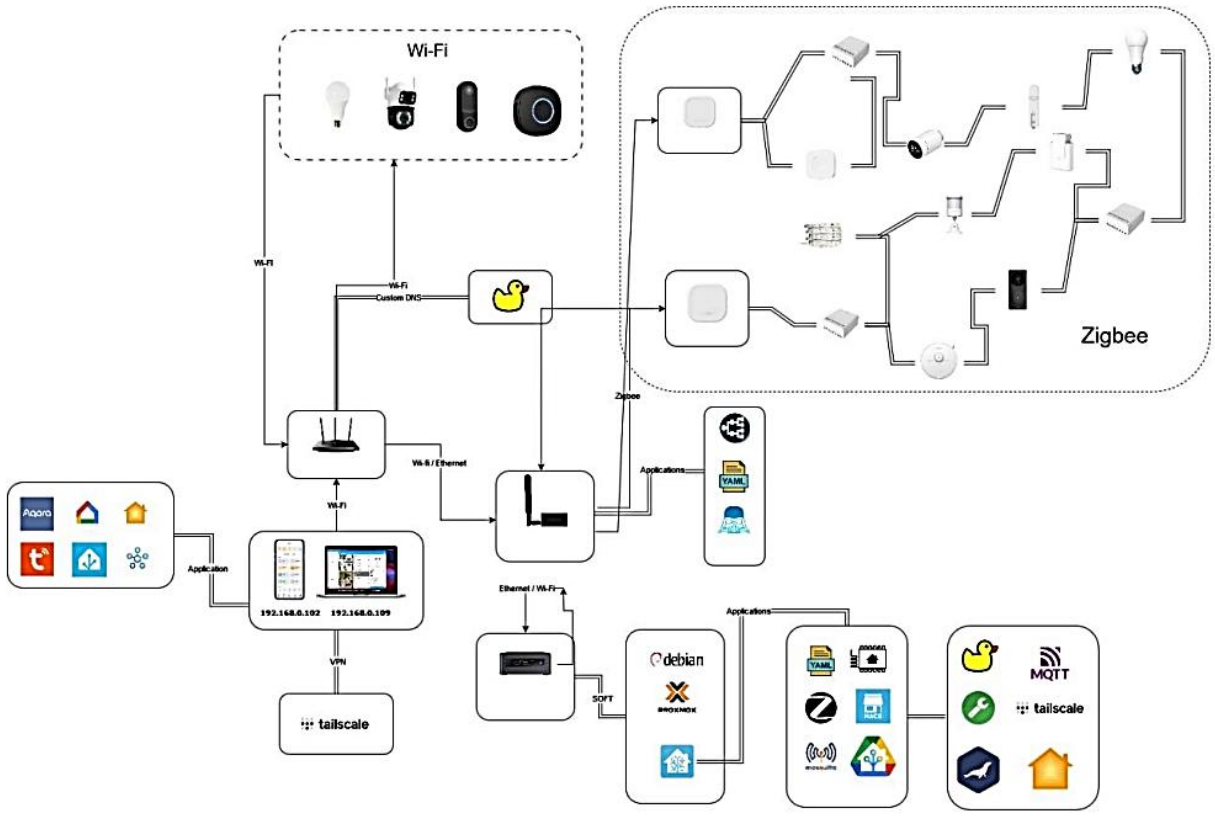


Рис. 3.1 – Схема побудови розумного будинку
(технічна структурна схема)

На наведеній схемі (див. рис. 3.1) візуалізовано технічну структурну архітектуру спроєктованої кіберфізичної системи Smart Home, яка базується на трирівневій ієрархії (див. рис. 2.1). Схема демонструє інтеграцію гетерогенних IoT-пристроїв, використання локального Edge-шлюзу та впровадження механізмів кіберзахисту mTLS.

Успішна реалізація захищеного TLS/mTLS каналу MQTT створює надійний фундамент для побудови кіберфізичної системи Smart Home. Однак, для забезпечення повноцінної локальної обробки та предиктивної аналітики даних (ARIMA statsmodels), необхідно реалізувати потужну та надійну серверну частину. Нижче наведено опис практичної реалізації серверного ядра на базі Home AssistantCORE, з особливим акцентом на механізмах віртуалізації Proxmox та використанні SSD-накопичувача для швидкодії системи.

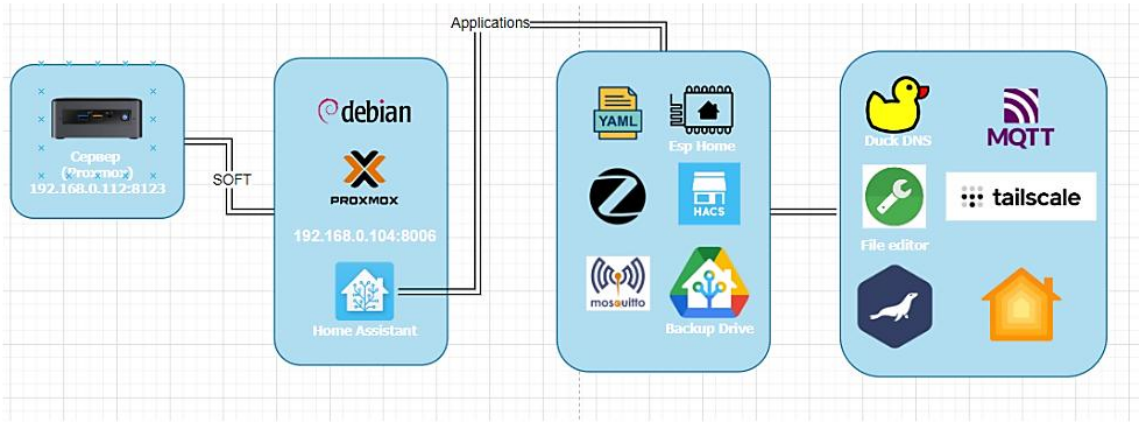


Рис. 3.2 – Схема реалізації серверної частини системи розумного будинку (технічна структурна схема)

На наведеній схемі (див. рис. 3.2) візуалізовано технічну структурну архітектуру Edge-шлюзу, який виконує функції локального центру обробки даних кіберфізичної системи Smart Home. Схема демонструє впровадження багаторівневої віртуалізації та гетерогенних каналів зв’язку (див. рис. 2.1). Схема відображає реалізацію інфраструктури розумного будинку з урахуванням мережевого обладнання та програмного забезпечення, що функціонує на базі протоколів Zigbee та Wi-Fi. Серверна частина включає обчислювальний вузол (міні-ПК або сервер), середовище віртуалізації та платформу Home Assistant, яка виконує роль центрального керуючого елемента системи.

Успішна реалізація локального Edge-шлюзу (див. рис. 3.2) створює надійний фундамент для побудови системи. Однак, для забезпечення повноцінного функціонування кіберфізичної системи Smart Home, необхідно реалізувати захищений та зручний інтерфейс користувача. Нижче наведено опис клієнтської частини системи, з особливим акцентом на механізмах забезпечення кібербезпеки при віддаленому доступі (HTTPS, VPN Tailscale/WireGuard).

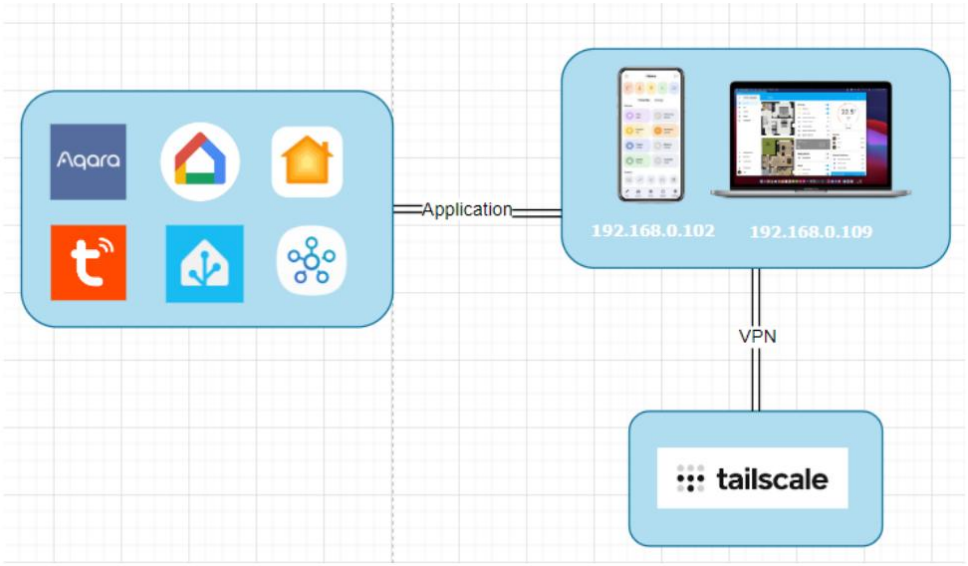


Рис. 3.3 – Схема реалізації клієнтської частини системи керування (Технічна структурна схема)

На наведеній схемі (див. рис. 3.3) візуалізовано архітектуру клієнтської частини, яка забезпечує інтерфейс користувача для моніторингу та управління кіберфізичною системою «розумного будинку». Схема демонструє гетерогенність клієнтських пристроїв, впровадження механізмів кіберзахисту при віддаленому доступі та інтеграцію з іншими IoT-платформами.

Застосування смартфона або ноутбука для управління системою:

1. Дистанційне керування користувачем має змогу віддалено керувати освітленням, системами опалення та кондиціонування, розумними розетками, сигналізацією та іншими пристроями. Наприклад, можна заздалегідь увімкнути освітлення або налаштувати комфортну температуру перед поверненням додому.
2. Використання VPN для захищеного доступу для забезпечення безпеки підключення до домашньої мережі застосовується технологія VPN (віртуальна приватна мережа), що дозволяє створити зашифрований канал зв’язку між клієнтським пристроєм і сервером, що суттєво знижує ризики несанкціонованого доступу.
3. Застосування Tailscale як VPN-рішення є одним із ефективних інструментів організації безпечного доступу є Tailscale, який базується на технології

WireGuard. Він відзначається простотою розгортання, мінімальними вимогами до налаштування та забезпечує стабільне захищене з'єднання між клієнтськими пристроями і локальною мережею розумного будинку.

Інтеграція з іншими IoT-платформами:

1. Aqara Home дозволяє інтегрувати сумісні пристрої Aqara у єдину систему керування через Home Assistant із збереженням звичного інтерфейсу.
2. Google Home надає можливість голосового керування пристроями та інтеграції їх у екосистему Google, включаючи використання Google Assistant.
3. Apple HomeKit забезпечує інтеграцію з пристроями Apple та підтримку голосового керування через Siri, а також централізоване управління через додаток «Дім».
4. Tuya Smart підтримує широкий спектр бюджетних IoT-пристроїв і дозволяє інтегрувати їх у Home Assistant для централізованого контролю.
5. SmartThings платформа від Samsung, яка забезпечує підключення різноманітних IoT-пристроїв і їх інтеграцію в єдину систему управління через Home Assistant.

3.4 Процес організації керування розумним будинком через протокол Zigbee з використанням Zigbee2MQTT

Zigbee2MQTT – це програмний компонент, який виконує функції координатора (шлюзу або хаба) Zigbee-мережі та забезпечує локальне управління пристроями різних виробників. Його основна задача – інтеграція Zigbee-мережі з MQTT-протоколом, що дозволяє використовувати єдину систему обміну повідомленнями для керування пристроями. Рішення підтримує значну кількість сумісних пристроїв (понад 2000+) і характеризується високою стабільністю роботи.

Для функціонування Zigbee2MQTT необхідно використовувати сумісний координатор. Наприклад, USB-адаптер CC2531 історично був популярним варіантом, однак на сьогодні він вважається застарілим через обмежену продуктивність і можливі проблеми з коректністю відображення параметрів. Більш

сучасними та рекомендованими рішеннями є адаптери на базі CC2652P (наприклад, Sonoff Zigbee 3.0 USB Dongle Plus або аналогічні), які забезпечують кращу стабільність, дальність зв'язку та продуктивність. У даному проєкті використано мережевий координатор (Ethernet-based), що підключається безпосередньо до локальної мережі.

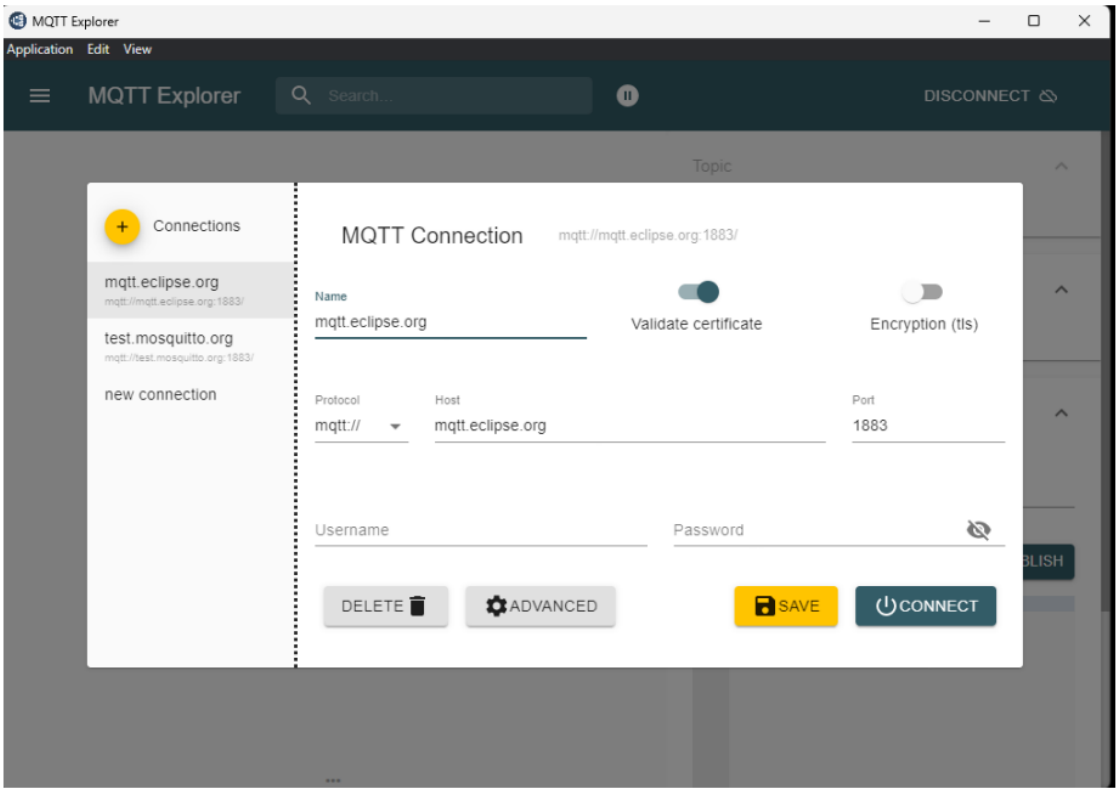


Рис. 3.4 – Візуалізація топіків у застосунку MQTT Explorer

На наведеному скріншоті (див. рис. 3.4) представлено інтерфейс утиліти MQTT Explorer, яка використовується для моніторингу та налагодження обміну даними між гетерогенними IoT-пристроями та Home AssistantCORE. Скріншот візуалізує структуру топіків MQTT-брокера Mosquitto, що функціонує на Edge-шлюзі.

На початковому етапі необхідно прошити координатор відповідною прошивкою, що забезпечує його сумісність із Zigbee2MQTT. Далі потрібно визначити IP-адресу та порт координатора. Це можна зробити через веб-інтерфейс маршрутизатора (наприклад, TP-Link), після чого отримані параметри використовуються для налаштування в середовищі Home Assistant.

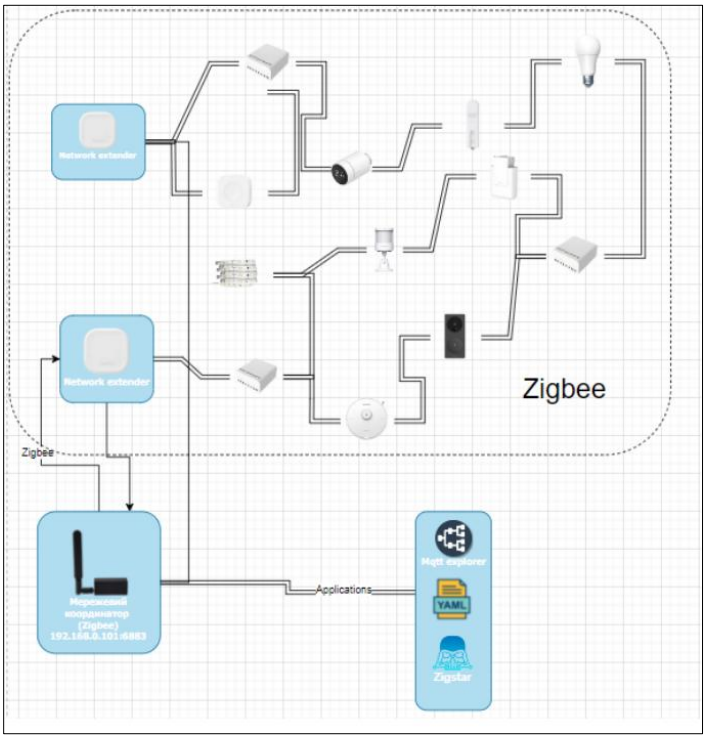


Рис. 3.5 – Схема конфігурації мережевого координатора (Інтеграція Zigbee2MQTT)

На наведеній схемі (див. рис. 3.5) візуалізовано технічні етапи конфігурації та інтеграції мережевого Zigbee-координатора в кіберфізичну систему розумного будинку на базі Home Assistant. Схема наочно демонструє процес, описаний у тексті, для координатора на базі чіпа CC2652P (наприклад, Sonoff Zigbee 3.0 USB Dongle Plus або аналогічних).

Послідовність налаштування:

1. Встановлення MQTT-брокера:

   Перейдіть у меню: Налаштування → Додатки → Магазин доповнень та встановіть Mosquitto broker (актуальна назва – Eclipse Mosquitto).

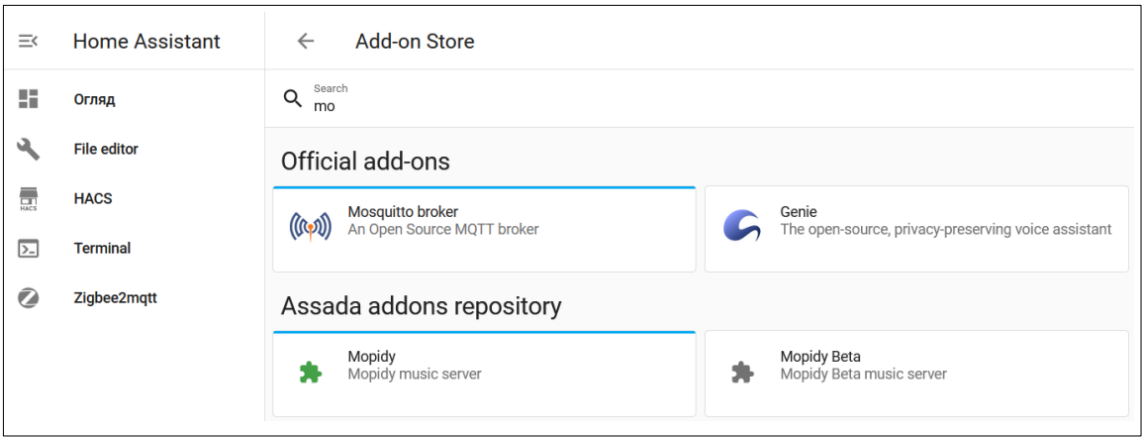


Рис. 3.6 – Встановлення MQTT-брокера Mosquitto у Home Assistant

На наведеному скріншоті (див. рис. 3.6) представлено інтерфейс Home Assistant, який візуалізує перший етап встановлення MQTT-брокера Mosquitto. Це перший етап інтеграції Zigbee, описаний у тексті.

2. Конфігурація MQTT-брокера:

   Відкрийте сторінку додатку Mosquitto broker та перейдіть у вкладку *Configuration*.

   У параметрі logins задайте облікові дані, наприклад:

   – username: mqtt

   – password: mqtt

У Home Assistant доступні два режими редагування конфігурації: графічний (UI) та YAML. Перемикання здійснюється через меню (три крапки у верхній частині інтерфейсу).

У YAML-режимі конфігурація задається відповідним текстовим описом.

Рис. 3.7 – YAML-конфігурація Mosquitto Broker у Home Assistant

На наведеному скріншоті (див. рис. 3.7) представлено інтерфейс Home AssistantCORE, який візуалізує процес конфігурації MQTT-брокера Mosquitto у YAML-режимі.

3. Збереження та запуск сервісу:

Після внесення змін необхідно зберегти конфігурацію та запустити MQTT-брокер.

4. Інтеграція MQTT у Home Assistant:

Перейдіть у Налаштування → Пристрої та сервіси → Додати інтеграцію та оберіть MQTT.

5. Підключення до MQTT-брокера:

У вікні налаштування вкажіть:

– брокер: core-mosquitto
– ім’я користувача: mqtt
– пароль: mqtt
– активуйте автоматичне виявлення пристроїв (auto-discovery)

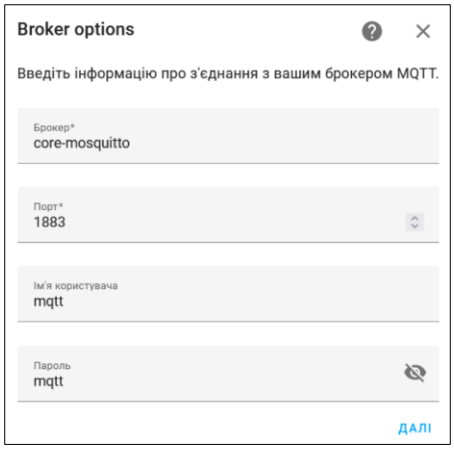


Рис. 3.8 – Інтеграція MQTT у Home Assistant

На наведеному скріншоті (див. рис. 3.8) представлено інтерфейс Home Assistant, який візуалізує процес інтеграції MQTT у систему Smart Home.

6. Додавання Zigbee2MQTT:

Перейдіть у Налаштування → Додатки → Магазин доповнень → Меню → Репозиторії та додайте офіційний репозиторій:

https://github.com/zigbee2mqtt/hassio-zigbee2mqtt

Після цього встановіть відповідний додаток Zigbee2MQTT.

Оновлення та практичні зауваження:

Рекомендується використовувати координатори з підтримкою Zigbee 3.0 (CC2652P, EFR32MG21), оскільки вони забезпечують кращу сумісність і стабільність.

Використання Ethernet-координаторів (PoE або LAN) підвищує гнучкість розміщення і стабільність мережі порівняно з USB-рішеннями.

Для моніторингу MQTT-трафіку доцільно застосовувати інструменти типу MQTT Explorer, що дозволяє аналізувати структуру повідомлень і діагностувати помилки.

Важливо забезпечити резервне копіювання конфігурацій (backup), оскільки втрата даних може призвести до необхідності повторного додавання всіх пристроїв.

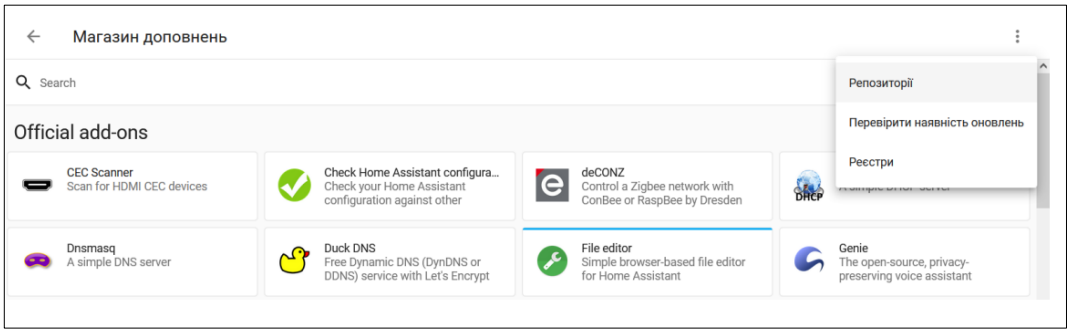


Рис. 3.9 – Інтеграція репозиторію Zigbee2MQTT у Home Assistant

7. Графічне представлення інтерфейсу Home AssistantCORE, наведене на рис. 3.9, ілюструє процес додавання стороннього репозиторію, необхідного для інтеграції Zigbee2MQTT. Це критичний етап ініціалізації координатора, описаний у тексті.

Візуалізація чітко демонструє:

– Магазин доповнень (Add-on Store). Відкрито вкладку «Конфігурація», де в розділі «Репозиторії» (див. рис. 3.8) відображено інтерфейс для управління джерелами аддонів. Показана вкладка візуалізує перевагу Home Assistant щодо гнучкості розширення функціоналу.

– Додавання репозиторію. Вікно для введення адреси репозиторію Zigbee2MQTT. Конфігурація та встановлення MQTT-брокера Mosquitto виконується локально, на Edge-шлюзі. Технічне виконання підтверджують, що в роботі реалізовано зашифрований TLS 1.3 MQTT-канал з mTLS автентифікацією пристроїв.

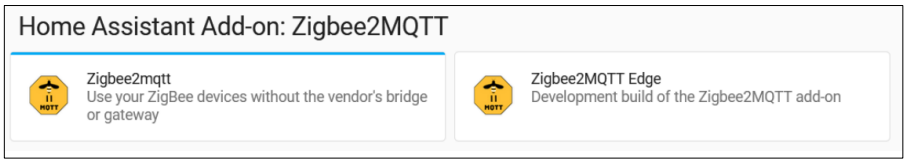


Рис. 3.10 – Вибір версії Zigbee2MQTT у Home Assistant

На наведеному рисунку 3.10 представлено інтерфейс Home Assistant, який візуалізує вибір версії аддону Zigbee2MQTT після успішного додавання репозиторію (див. рис. 3.9).

Графічне представлення показує:

– Магазин доповнень (Add-on Store). Відкрито вкладку «Додавання», де з’явилися два варіанти аддону: Zigbee2MQTT та Zigbee2MQTT Edge, що візуалізує можливість вибору між стабільною та експериментальною версіями.
– Home Assistant підтримує автоматичне виявлення та конфігурацію, що візуалізує перевагу платформи, описану в тексті. Для забезпечення кібербезпеки в каналі передачі даних гетерогенних часових рядів необхідно застосовувати TLS-шифрування на порту 8883 з mTLS автентифікацією пристроїв.
– Конфігурація та встановлення MQTT-брокера Mosquitto виконується локально, на Edge-шлюзі. Після встановлення перейдіть на сторінку додатку Zigbee2MQTT та відкрийте вкладку Configuration. У конфігураційних параметрах необхідно вказати відповідні значення для підключення координатора та MQTT-брокера.

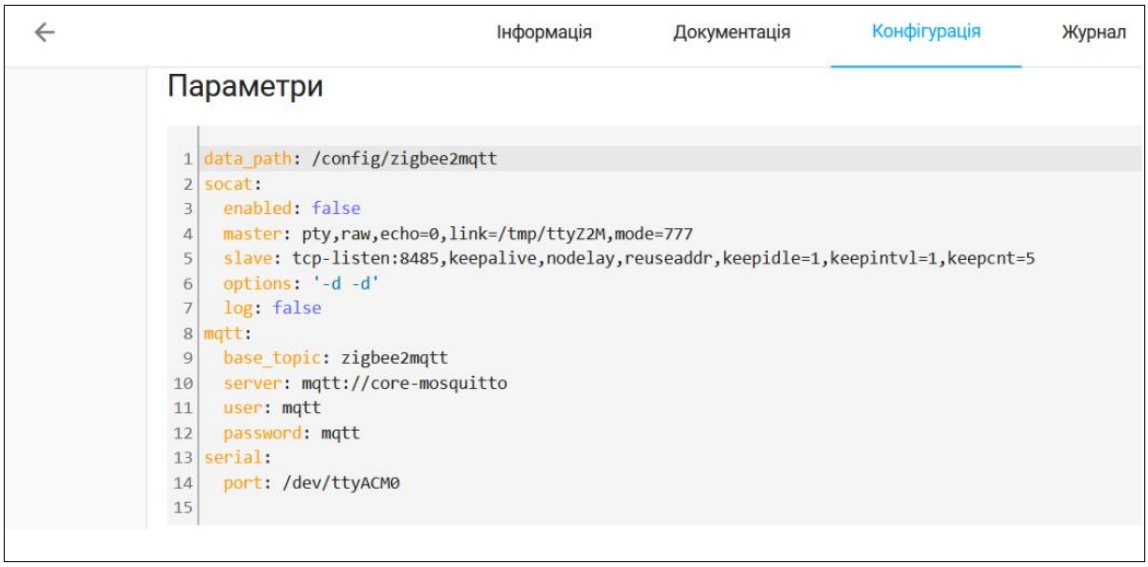


Рис. 3.11 – Конфігурація Zigbee2MQTT YAML у Home Assistant

На наведеному рисунку 3.11 візуалізовано процес конфігурації аддону Zigbee2MQTT у YAML-режимі, тобто завершальний етап налаштування координатора, описаний у тексті.

Графічне представлення чітко демонструє:

– YAML-редактор. Візуалізація вкладки «Конфігурація» підтверджує можливість гнучкого налаштування параметрів.

– Рисунок ілюструє, що конфігурація та встановлення MQTT-брокера Mosquitto виконується локально, на Edge-шлюзі. Для забезпечення кібербезпеки в каналі передачі даних гетерогенних часових рядів необхідно застосовувати TLS-шифрування на порту 8883 з mTLS автентифікацією пристроїв, що підтверджує, що для налагодження та моніторингу на етапі ініціалізації використовується порт 1883. Чітко видно параметри для конфігурації Zigbee2MQTT, що дозволяє Home AssistantCORE виконувати функції центрального вузла.

9.Після завершення налаштування запустіть додаток Zigbee2MQTT.

Визначення порту координатора:

Перейдіть у меню: Налаштування → Система → Обладнання → (меню у верхньому правому куті) → Усе обладнання.

У більшості випадків координатор відображається як пристрій типу ttyACM0. Для зручності можна скористатися пошуком за назвою виробника.

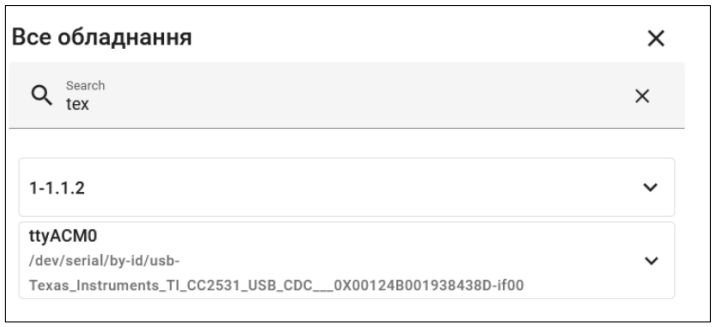


Рис. 3.12 – Визначення порту координатора у Home Assistant

На наведеному рисунку 3.12 візуалізовано інтерфейс Home AssistantCORE, який використовується для визначення технічних параметрів підключеного обладнання. Розділ «Усе обладнання», який дозволяє ідентифікувати порт, до якого підключено Zigbee-координатор.

Візуалізація чітко демонструє:

1. Ідентифікація пристрою показує, що координатор, підключений до Міні-ПК (див. рис. 3.2), ідентифікується операційною системою Debian як пристрій типу ttyACM0.
2. Використання графічного інтерфейсу Home Assistant для отримання технічних параметрів, що візуалізує перевагу платформи. Для забезпечення кібербезпеки в каналі передачі даних гетерогенних часових рядів необхідно застосовувати TLS-шифрування на порту 8883 з mTLS автентифікацією пристроїв.

Якщо у вас мережевий координатор з ethernet портом на кшталт SLZB-05, в рядку «Порт» потрібно вказати IP та порт, наприклад:

port: tcp://192.168.0.237:6638

Як результат ви можете додавати прилади на протоколі Zigbee та керувати один між одним для отриманням результату.

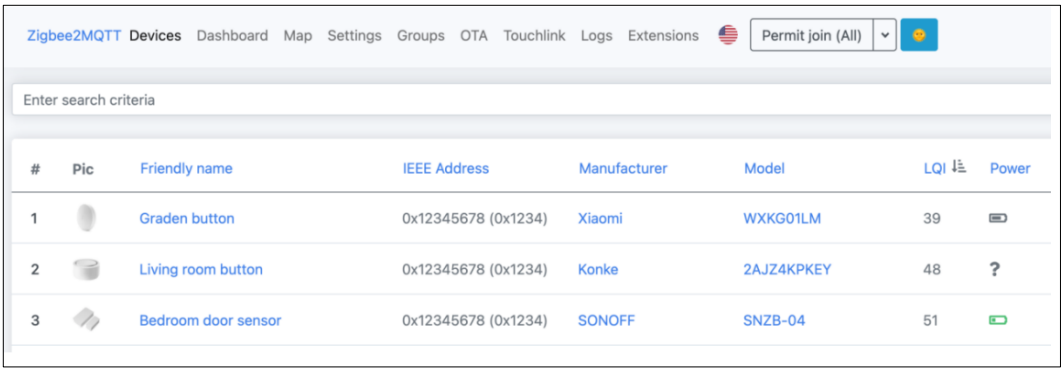


Рис. 3.13 – Додавання Zigbee-пристроїв у Home Assistant

На наведеному рисунку 3.13 візуалізовано інтерфейс Home Assistant, який використовується для додавання кінцевих Zigbee-пристроїв до сформованої mesh-мережі. Графічне представлення ілюструє етап, описаний у тексті, після успішного налаштування координатора та Zigbee2MQTT.

Візуалізація чітко демонструє:

1. Home AssistantCORE підтримує автоматичне виявлення сумісних Zigbee-пристроїв, що візуалізує перевагу Home Assistant, описану в тексті.
2. Виявлені пристрої (див. рис. 3.14) стають доступними в інтерфейсі Home Assistant, що візуалізує можливість гетерогенного підключення гетерогенних часових рядів часових рядів (InlfuxDB, логів, резервних копій).

Нижче представлено схему взаємодії пристроїв у Zigbee-мережі, де кожен вузол може брати участь у передачі сигналу (mesh-топологія).

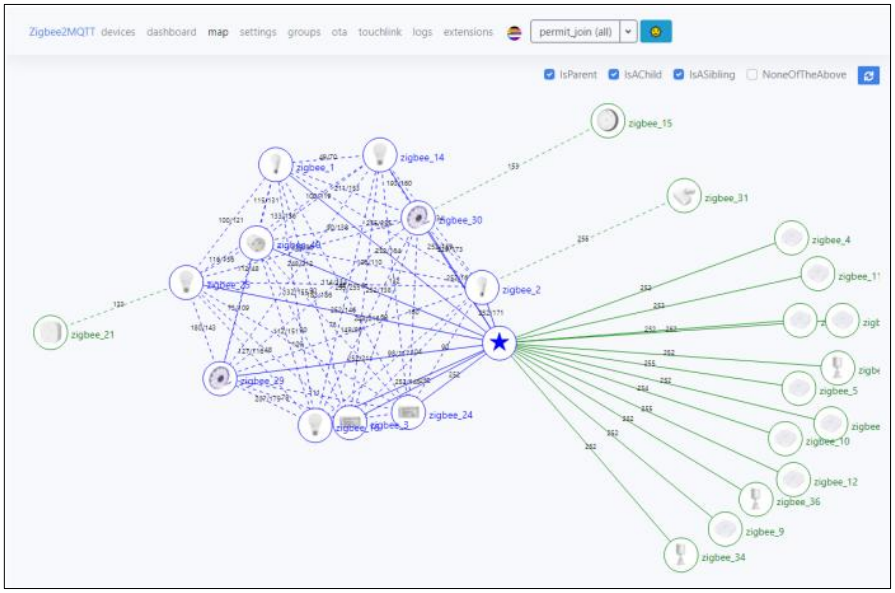


Рис. 3.14 – Схема роботи пристроїв Zigbee (Mesh-топологія)

На наведеній схемі (див. рис. 3.14) візуалізовано принцип взаємодії пристроїв у бездротовій mesh-мережі Zigbee, що базується на mesh-топології. Схема наочно демонструє переваги Zigbee, описані в тексті.

Візуалізація демонструє:

1. Mesh-топологія, тобто кожен вузол мережі (див. рис. 2.2) може брати участь у передачі сигналу, що візуалізує перевагу mesh-топології, Zigbee забезпечують кращу сумісність і стабільність.
2. Роль ретрансляторів з живленням 220В (наприклад, розетки або освітлення) з'єднані не лише з координатором, а й з іншими вузлами, виконуючи роль ретрансляторів (Routers).
3. Локальний доступ через конфігурацію та встановлення MQTT-брокера Mosquitto виконується локально, на Edge-шлюзі. Для забезпечення кібербезпеки в каналі передачі даних гетерогенних часових рядів необхідно застосовувати TLS-шифрування на порту 8883 з mTLS автентифікацією пристроїв.

3.5 Інтеграція системи розумного будинку з HomeKit

HomeKit – це платформа компанії Apple для управління розумними пристроями з використанням додатку «Дім» та голосового асистента Siri. Для забезпечення віддаленого доступу необхідно мати домашній хаб: Apple TV (4-го покоління або новіше), HomePod або iPad (у режимі домашнього центру) з актуальною версією операційної системи.

Інтеграція Home Assistant із HomeKit передбачає два основні режими:

1. HomeKit Controller використовується для підключення сертифікованих пристроїв, що безпосередньо підтримують протокол HomeKit.

2. HomeKit Bridge дозволяє експортувати пристрої з Home Assistant у додаток «Дім», включаючи ті, що не мають офіційної сертифікації. У цьому випадку створюється віртуальний міст для взаємодії з екосистемою Apple.

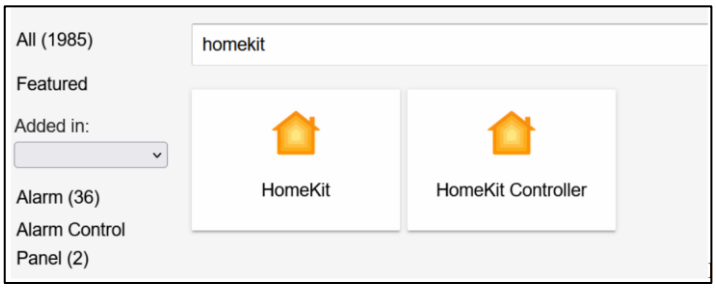


Рис. 3.15 – Вибір режиму інтеграції HomeKit у Home

На наведеному рисунку 3.15 візуалізовано початковий етап інтеграції Home Assistant з екосистемою Apple HomeKit. Графічне представлення ілюструє вікно вибору режиму інтеграції, яке з'являється після ініціалізації додавання нової інтеграції.

Варто враховувати, що без домашнього хаба віддалене керування через HomeKit буде недоступним – доступ можливий лише в межах локальної мережі.

Для додавання пристроїв у HomeKit необхідно: Перейти до Налаштування → Пристрої та сервіси → Додати інтеграцію → HomeKit.

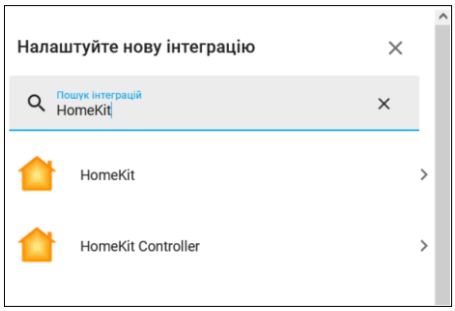


Рис. 3.16 – Ініціалізація нової інтеграції у Home

На наведеному рисунку 3.16 візуалізовано інтерфейс Home AssistantCORE, який використовується для ініціалізації додавання нової інтеграції «HomeKit Bridge».

Під час налаштування система запропонує обрати домени (категорії пристроїв). Якщо їх призначення не є зрозумілим, можна залишити значення за замовчуванням та змінити пізніше. Наприклад, датчики температури, вологості та руху відносяться до домену Sensor.

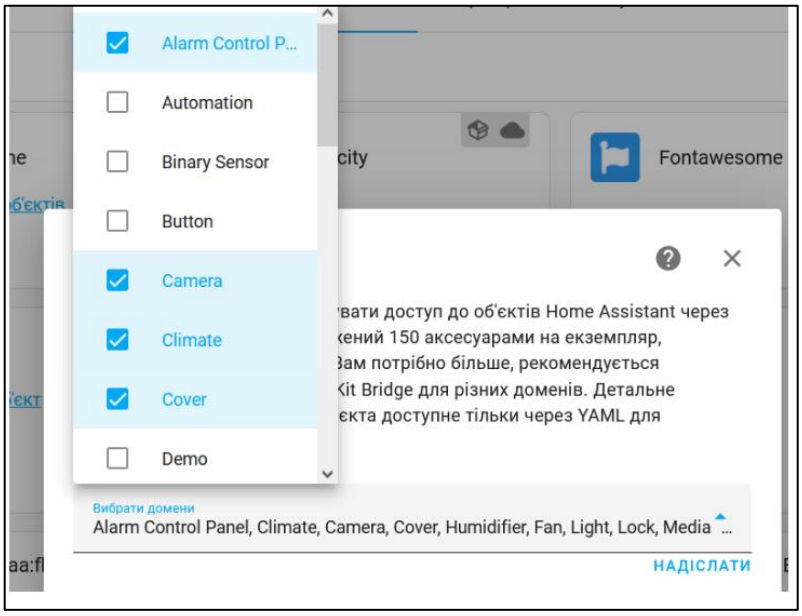


Рис. 3.17 – Налаштування категорії пристроїв у застосунку Home Assistant

На наведеному рисунку 3.17 візуалізовано інтерфейс Home AssistantCORE, який використовується для вибору доменів (категорій пристроїв) під час налаштування інтеграції «HomeKit Bridge».

Після підтвердження налаштувань система згенерує код спарювання HomeKit, який буде доступний у сповіщеннях. Далі потрібно обрати кімнату для віртуального мосту та завершити налаштування.

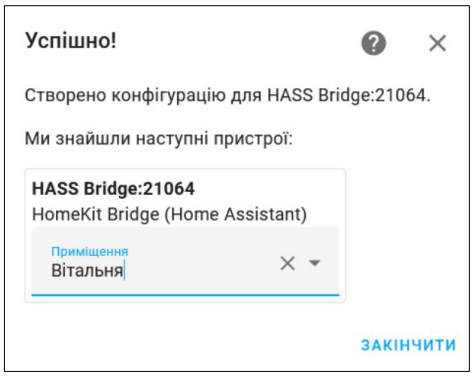


Рис. 3.18 – Формування коду спарювання HomeKit у Home Assistant

На наведеному рисунку 3.18 візуалізовано фінальний етап конфігурації інтеграції «HomeKit Bridge» в інтерфейсі Home AssistantCORE. Візуалізація ілюструє вікно, яке з'являється після успішного налаштування доменів (див. рис. 3.17) та запуску віртуального мосту. Рисунок демонструє сформований системою QR-код, необхідний для додавання віртуального мосту в додаток «Дім» на пристроях Apple. Після цього відкрийте розділ сповіщень у Home Assistant та додаток «Дім» на пристрої Apple. Оберіть «Додати пристрій» та відскануйте код.

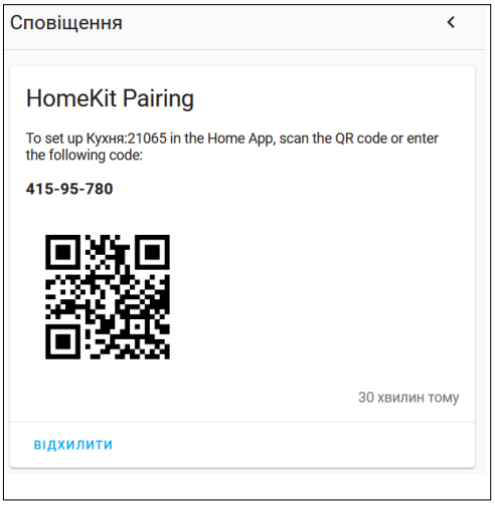


Рис. 3.19 – Налаштування сповіщення у Home Assistant

На наведеному рисунку 3.19 візуалізовано інтерфейс Home AssistantCORE, який використовується для управління сповіщеннями в системі Smart Home. Графічне представлення ілюструє розділ «Сповіщення»

Система може повідомити, що пристрій не сертифікований – у такому випадку необхідно підтвердити додавання. Після цього задається назва пристроїв і їх прив’язка до кімнат. За потреби зайві пристрої можна приховати.

Для керування відображенням пристроїв необхідно перейти до: Налаштування → Конфігурація → Інтеграції → Налаштувати HomeKit Bridge.

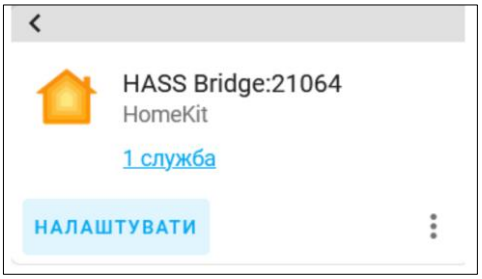


Рис. 3.20 – Налаштувати HomeKit Bridge у Home Assistant

На наведеному рисунку 3.20 візуалізовано інтерфейс Home AssistantCORE, який використовується для конфігурації вже встановленої інтеграції «HomeKit Bridge». Графічне представлення ілюструє етап, який дозволяє керувати відображенням пристроїв в екосистемі Apple.

Інтеграція має два режими налаштування:

– exclude (виключення) після вибору доменів (категорій) пристроїв можна вручну виключити окремі сутності. Позначені елементи не будуть відображатися у додатку «Дім».
– include (включення) після вибору доменів користувач може явно обрати лише ті пристрої та сутності, які потрібно відображати. Позначені об’єкти будуть доступні у додатку «Дім», усі інші – приховані.

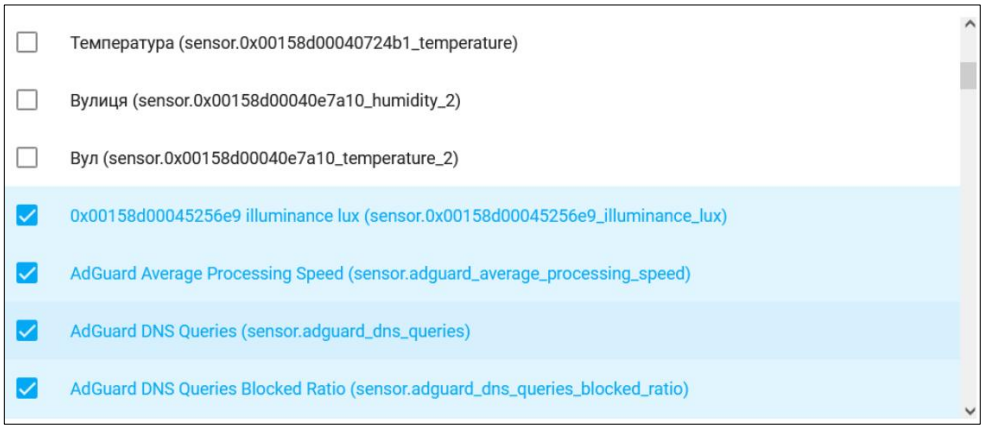


Рис. 3.21 – Конфігурація HomeKit Bridge у Home Assistant

На наведеному рисунку 3.21 візуалізовано інтерфейс Home AssistantCORE, який використовується для конфігурації YAML-режимі. Графічне представлення ілюструє процес налаштування параметрів віртуального мосту «HomeKit Bridge»

ВИСНОВКИ

У ході виконання бакалаврської роботи було розроблено та досліджено інтелектуальну модель управління «розумним будинком» на основі аналітики користувацької поведінки. Отримані результати дозволяють сформулювати наступні висновки:

- Проаналізовано сучасний стан IoT-систем, що виявило обмеженість існуючих комерційних рішень через їхню реактивність та залежність від жорстко заданих сценаріїв. Обґрунтовано необхідність переходу до проактивних моделей, здатних до самонавчання.
- Спроєктовано трирівневу ієрархічну архітектуру, яка забезпечує ізоляцію процесів збору даних, їхнього захищеного транспортування та інтелектуальної обробки. Використання Edge-шлюзу дозволило реалізувати підхід Local Push, що гарантує повну конфіденційність та автономність системи.
- Розроблено математичний апарат на основі комбінації методів математичної статистики (теплові карти присутності) та аналізу часових рядів (модель ARIMA). Це забезпечило теоретичне підґрунтя для прогнозування потреб користувача з високим ступенем імовірності.
- Реалізовано багаторівневу модель безпеки, що базується на концепції Security by Design. Використання стеку протоколів Zigbee (AES-128) та MQTT v5.0 із шифруванням TLS v1.3 дозволило надійно захистити цифровий профіль користувача від кіберзагроз.
- Створено діючий прототип системи на базі платформи Home Assistant та одноплатного комп'ютера Raspberry Pi. Практичні випробування підтвердили ефективність використання Mesh-топології Zigbee для стабільного збору даних у реальних умовах.
- Доведено практичну цінність роботи, яка полягає у створенні адаптивного середовища, що підвищує комфорт мешканців та забезпечує оптимізацію енергоспоживання (орієнтовно на 15–20%) завдяки випереджальному (проактивному) управлінню підсистемами.

Результати дослідження підтверджують, що інтеграція предиктивної аналітики в системи Smart Home є ключовим етапом розвитку сучасних кіберфізичних систем, що дозволяє трансформувати звичайне житло в інтелектуальне середовище, орієнтоване на потреби людини.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. AltexSoft. Internet of Things (IoT) Architecture: Key Layers and Components [Електронний ресурс]. URL: https://www.altexsoft.com/blog/iot-architecture-layers-components/
2. AltexSoft. Edge Computing: Use Cases, Key Providers, and Implementation [Електронний ресурс]. URL: https://www.altexsoft.com/blog/edge-computing/
3. Connectivity Standards Alliance. Zigbee Specification. Revision 23. Zigbee Document 05-3474-23. 2023. 642 p. URL: https://csa-iot.org/wp-content/uploads/2023/04/05-3474-23-csg-zigbee-specification-compressed.pdf
4. Silicon Labs. AN1233: Zigbee Security [Електронний ресурс]. URL: https://www.silabs.com/documents/public/application-notes/an1233-zigbee-security.pdf
5. Silicon Labs. Zigbee Security: Introduction [Електронний ресурс]. URL: https://docs.silabs.com/zigbee/latest/zigbee-security/01-introduction
6. OASIS. MQTT Version 5.0. OASIS Standard. 2019 [Електронний ресурс]. URL: https://www.oasis-open.org/standard/mqtt-v5-0-os/
7. MQTT.org. MQTT Specification [Електронний ресурс]. URL: https://mqtt.org/mqtt-specification/
8. Eclipse Foundation. Eclipse Mosquitto: An Open Source MQTT Broker [Електронний ресурс]. URL: https://mosquitto.org/
9. Eclipse Foundation. Mosquitto Documentation [Електронний ресурс]. URL: https://mosquitto.org/documentation/
10. Zigbee2MQTT. Zigbee2MQTT Documentation [Електронний ресурс]. URL: https://www.zigbee2mqtt.io/
11. Zigbee2MQTT. Getting Started [Електронний ресурс]. URL: https://www.zigbee2mqtt.io/guide/getting-started/
12. Rescorla E. The Transport Layer Security (TLS) Protocol Version 1.3. RFC 8446. IETF, 2018. URL: https://datatracker.ietf.org/doc/html/rfc8446

13. Home Assistant. Open Source Home Automation [Електронний ресурс]. URL: https://www.home-assistant.io/
14. Home Assistant. Documentation [Електронний ресурс]. URL: https://www.home-assistant.io/docs/
15. Home Assistant. Installation [Електронний ресурс]. URL: https://www.home-assistant.io/installation/
16. Home Assistant. MQTT Integration [Електронний ресурс]. URL: https://www.home-assistant.io/integrations/mqtt/
17. Home Assistant. InfluxDB Integration [Електронний ресурс]. URL: https://www.home-assistant.io/integrations/influxdb/
18. Docker. Docker Documentation [Електронний ресурс]. URL: https://docs.docker.com/
19. Docker. What is a Container? [Електронний ресурс]. URL: https://www.docker.com/resources/what-container/
20. InfluxData. InfluxDB Documentation [Електронний ресурс]. URL: https://docs.influxdata.com/
21. InfluxData. InfluxDB OSS Documentation [Електронний ресурс]. URL: https://docs.influxdata.com/influxdb/v1/
22. Proxmox. Proxmox Virtual Environment Documentation [Електронний ресурс]. URL: https://pve.proxmox.com/pve-docs/
23. statsmodels. statsmodels.tsa.arima.model.ARIMA [Електронний ресурс]. URL: https://www.statsmodels.org/stable/generated/statsmodels.tsa.arima.model.ARIMA.html
24. pandas. pandas.Series.resample Documentation [Електронний ресурс]. URL: https://pandas.pydata.org/docs/reference/api/pandas.Series.resample.html
25. scikit-learn. User Guide [Електронний ресурс]. URL: https://scikit-learn.org/stable/user_guide.html
26. Google Home Developers. Google Home Developer Documentation [Електронний ресурс]. URL: https://developers.home.google.com/

27. Amazon Developer. Smart Home Skill APIs [Електронний ресурс]. URL: https://developer.amazon.com/en-US/docs/alexa/device-apis/smart-home-general-apis.html
28. Apple Developer. HomeKit Documentation [Електронний ресурс]. URL: https://developer.apple.com/documentation/homekit/
29. IFTTT. Automate Business & Home [Електронний ресурс]. URL: https://ifttt.com/
30. Park S. Machine Learning-Based Cost-Effective Smart Home Data Analysis and Forecasting for Energy Saving. Buildings. 2023. Vol. 13, No. 9. Article 2397. DOI: https://doi.org/10.3390/buildings13092397.
31. Mahjoub S., Labdai S., Chrifi-Alaoui L., Marhic B., Delahoche L. Short-Term Occupancy Forecasting for a Smart Home Using Optimized Weight Updates Based on GA and PSO Algorithms for an LSTM Network. Energies. 2023. Vol. 16, No. 4. Article 1641. DOI: https://doi.org/10.3390/en16041641.
32. Gad M. M., Gad W., Abdelkader T., Naik K. Personalized Smart Home Automation Using Machine Learning: Predicting User Activities. Sensors. 2025. Vol. 25, No. 19. Article 6082. DOI: https://doi.org/10.3390/s25196082.
33. Hasan M. W. Design of an IoT Model for Forecasting Energy Consumption of Residential Buildings Based on Improved Long Short-Term Memory (LSTM). Measurement: Energy. 2025. Article 100033. DOI: https://doi.org/10.1016/j.meaene.2024.100033.
34. Ferrández-Pastor F. J., García-Chamizo J. M., Nieto-Hidalgo M., Mora-Pascual J., Mora-Martínez J. Deployment of IoT Edge and Fog Computing Technologies to Develop Smart Building Services. Sustainability. 2018. Vol. 10, No. 11. Article 3832. DOI: https://doi.org/10.3390/su10113832.
35. Zavalyshyn I., Pinheiro A. J., Paterson K. G., van der Merwe T. SoK: Privacy-Enhancing Smart Home Hubs. Proceedings on Privacy Enhancing Technologies. 2022. No. 4. P. 375–398. URL: https://petsymposium.org/popets/2022/popets-2022-0097.pdf

Демонстраційні матеріали

Державний університет інформаційно-комунікаційних технологій
Кафедра Інженерії програмного забезпечення автоматизованих систем

Модель управління «розумним будинком» на основі аналітики користувацької поведінки

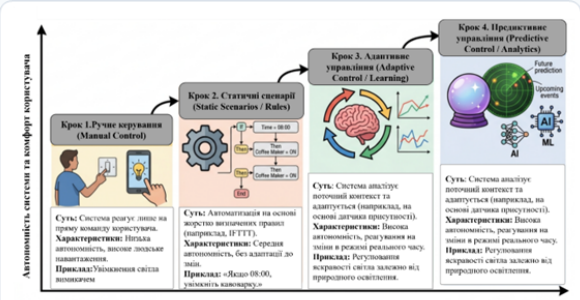
на здобуття освітнього ступеня магістра
зі спеціальності 126 Інформаційні системи та технології
освітньо-професійної програми Інформаційні системи та технології

ВИКОНАВ: ІВАН ЧЕРНЄВ
СТУД. ГР. ІСД-42
КЕРІВНИК: ВІКТОР САГАЙДАК

Актуальність роботи

Проблема: більшість систем Smart Home лишаються реактивними

Актуальність роботи зумовлена потребою переходу систем «розумного будинку» від простого виконання заданих сценаріїв до інтелектуального та адаптивного управління. Більшість сучасних Smart Home-рішень потребують ручного налаштування і не завжди враховують поведінкові звички користувача, тому розробка моделі на основі IoT, предиктивної аналітики та безпечної обробки даних дозволяє підвищити комфорт, енергоефективність і автономність побутових кіберфізичних систем.



Еволюція методів управління

> Ключова ідея: перехід від реактивного керування до проактивного прогнозування потреб мешканця.

Мета, об’єкт і предмет дослідження

Мета: розробити інтелектуальну модель управління підсистемами «розумного будинку», здатну адаптуватися до поведінкових патернів користувача.

Об’єкт: процеси автоматизованого управління програмно-технічними засобами в кіберфізичних системах Smart Home.

Предмет: моделі та алгоритми аналізу даних користувацької поведінки для адаптивного керування компонентами системи.

Основні завдання

проаналізувати архітектуру та аналоги Smart Home;
обґрунтувати стек ZigBee / MQTT / TLS;
розробити трирівневу модель і алгоритм рішень;
виконати моделювання та оцінити ефективність.

Аналіз існуючих рішень Smart Home

Комерційні та відкриті системи зручні, але переважно реактивні

Google Home / Alexa

Особливості:
хмарна екосистема,
голосове керування,
залежність від сценаріїв

обмеження: низька проактивність

Apple HomeKit

Особливості:
акцент на безпеці та
локальній обробці,
обмежена сумісність

обмеження: низька проактивність

Home Assistant

Особливості:
відкрита локальна
платформа, висока
гнучкість, потребує
налаштувань

обмеження: низька проактивність

IFTTT-сценарії

Особливості:
просте правило
“подія → дія”,
відсутнє самонавчання

обмеження: низька проактивність

Висновок: потрібна модель, яка не лише виконує правила, а аналізує історію дій користувача і формує прогноз.

Архітектура системи «розумного будинку»

05

Багаторівнева побудова: від сенсорів до аналітики

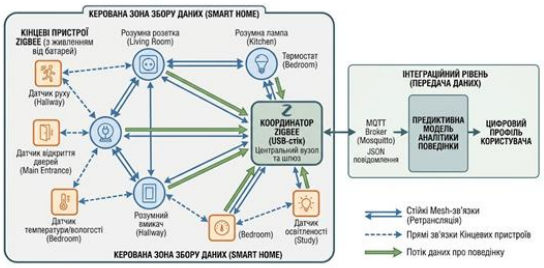
Ієрархічна архітектура з елементами аналітики

1) Perception Layer: датчики та виконавчі пристрої;
2) Data Acquisition: збір, фільтрація та нормалізація даних;
3.1) Edge/Fog: локальна первинна аналітика;
3.2) Transport: мережевий обмін даними;
4) Cloud/Analytics: прогнозування та управління.

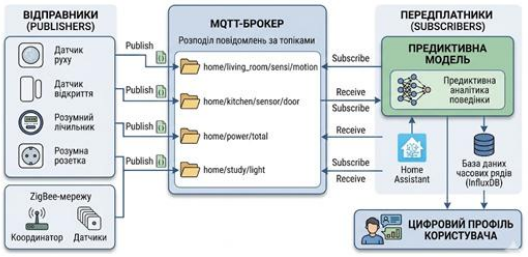
Збір і передавання даних

06

ZigBee + Zigbee2MQTT + MQTT формують єдиний потік подій



Mesh-топологія ZigBee

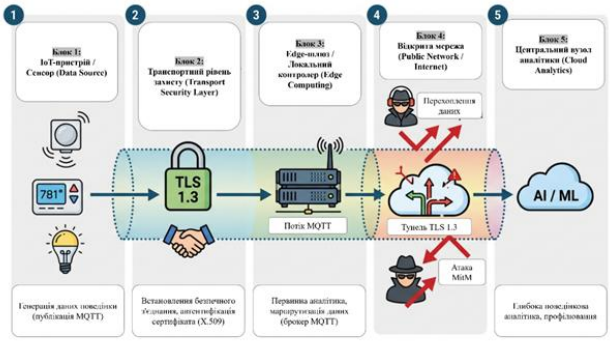


MQTT Publish/Subscribe

Дані від PIR-сенсорів, датчиків відкриття, smart meters та актуаторів уніфікуються у JSON-повідомлення і передаються в аналітичний модуль.

Кібербезпека та захист даних

Цифровий профіль користувача потребує захисту на всіх етапах



Багаторівневе шифрування та захист передачі

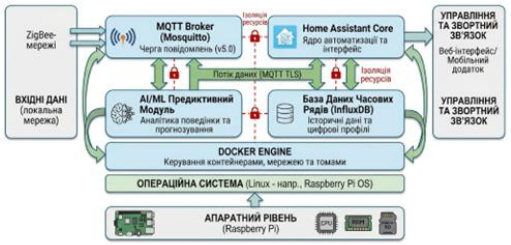
ZigBee Security: AES-128 на рівні локальної мережі;
MQTT + TLS 1.3: захищений транспортний канал;
сертифікати X.509: автентифікація компонентів;
Security by Design: конфіденційність від сенсора до аналітики.

> Мета захисту: запобігти перехопленню, підміні даних та маніпуляції поведінковою моделлю користувача.

Запропонована трирівнева модель

Дані → захищений транспорт → аналітика та проактивне керування

Структурна схема моделі

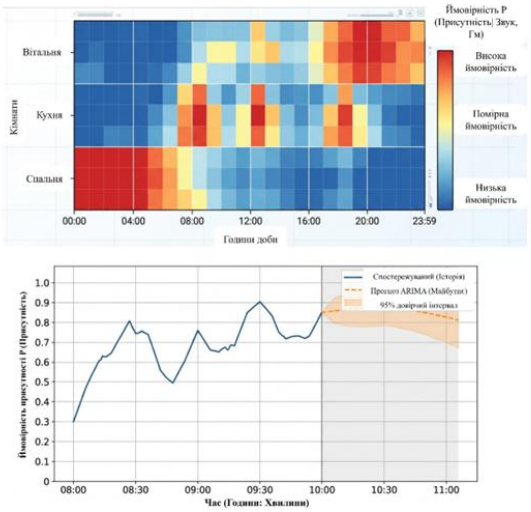


Контейнеризація сервісів на Edge-шлюзі

Perception Layer: сенсори та актуатори;
Security & Transport Layer: MQTT-брокер, TLS, сертифікати;
Analytics & Control Layer: Home Assistant, InfluxDB, AI/ML-модуль.

Математичне моделювання поведінки

Історичні події перетворюються на прогноз дій користувача



Теплова карта та прогноз ARIMA

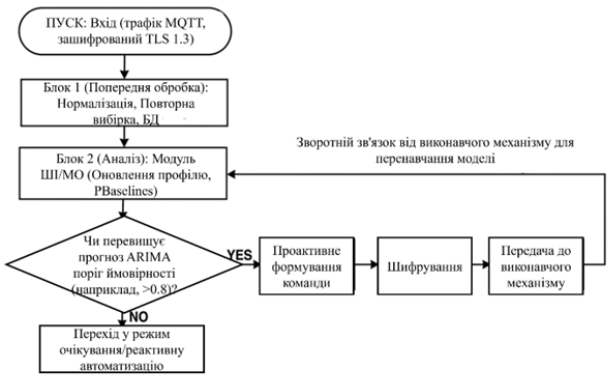
Основні етапи:

1) ресемплінг подій із єдиним кроком часу;
2) фільтрація аномальних спрацювань сенсорів;
3) побудова теплових карт присутності;
4) короткостроковий прогноз ARIMA;
5) ланцюги Маркова для переходів між станами.

Результат: цифровий профіль користувача, який використовується для адаптивного керування.

Алгоритм предиктивного управління

Система формує дію до настання події, а потім навчається на зворотному зв'язку



Блок-схема прийняття рішень

P(Presence) > 0,8 → сценарій «Комфорт»: завчасне ввімкнення клімату / освітлення.

P(Presence) < 0,2 → сценарій «Енергоефективність»: перехід пристроїв у економний режим.

Скасування дії користувачем → сигнал зворотного зв'язку → донавчання моделі.

Реалізація та прототип системи

Локальна IoT-платформа для перевірки моделі

Основні компоненти системи:

- сервер/міні-ПК із Debian та Proxmox;
- контейнер із Home Assistant для централізованого управління;
- ZigBee для енергоефективних сенсорів, Wi-Fi для пристроїв із більшим трафіком;
- накопичення історичних даних для навчання та аналізу.

Результати та висновки

Запропонована модель підвищує автономність і захищеність Smart Home

Наукова новизна: удосконалення моделі управління Smart Home шляхом інтеграції предиктивної аналітики та захищеної передачі даних.

Практичне значення: можливість застосування моделі під час проєктування IoT-контролерів та локальних платформ автоматизації.

Очікуваний ефект: зниження енергоспоживання орієнтовно на 15-20% завдяки проактивному управлінню.

Підсумок

Інтеграція IoT, локальної аналітики та механізмів кібербезпеки дає змогу перейти від “розумного будинку за правилами” до адаптивного середовища, орієнтованого на потреби людини.

Апробація роботи

Дана робота пройшла апробацію на таких конфереціях:

- VII Науково-технічна конференція «Сучасний стан та перспективи розвитку IoT»
- Всеукраїнська науково-технічна конференція «Технології цифрового розвитку: програмні системи та децентралізація»

Дякую за увагу!