

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Західноукраїнський національний університет
Факультет комп'ютерних інформаційних технологій
Кафедра інформаційно-обчислювальних систем і управління

КОСТОГЛОД Анастасія Юріївна

**Інтелектуальний модуль оптимізації режимів роботи IoT-пристроїв
для зменшення енергоспоживання в смарт-будівлях / Intelligent
Module for Optimizing the Operating Modes of IoT Devices to Reduce
Energy Consumption in Smart Buildings**

спеціальність: 122 - Комп'ютерні науки
освітньо-професійна програма - Комп'ютерні науки

Кваліфікаційна робота

Виконала студентка групи КН-42
А. Ю. Костоглод

Науковий керівник:
к.т.н., професор, В. В. Кочан

Кваліфікаційну роботу
допущено до захисту:
"___" _____ 20__ р.

Завідувач кафедри
_____ **Н.В. Дзюбановська**

Тернопіль – 2026

Факультет комп'ютерних інформаційних технологій
Кафедра інформаційно-обчислювальних систем і управління
Освітній ступінь «бакалавр»
спеціальність: 122 – Комп'ютерні науки
освітньо-професійна програма – Комп'ютерні науки

ЗАТВЕРДЖУЮ
В.о. завідувача кафедри
Н.В. Дзюбановська
«____» _____ 20__ р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТЦІ
КОСТОГЛОД Анастасії Юріївні

(прізвище, ім'я, по батькові)

1. Тема кваліфікаційної роботи

Інтелектуальний модуль оптимізації режимів роботи IoT-пристроїв для зменшення енергоспоживання в смарт-будівлях / Intelligent Module for Optimizing the Operating Modes of IoT Devices to Reduce Energy Consumption in Smart Buildings

керівник роботи к.т.н., професор В.В. Кочан

затверджені наказом по університету від 01 грудня 2025 року № 894.

2. Строк подання студентом закінченої кваліфікаційної роботи 25 травня 2026 р.

3. Вихідні дані до кваліфікаційної роботи: завдання на кваліфікаційну роботу студента, наукові статті, технічна література.

4. Основні питання, які потрібно розробити

- проаналізувати предметну область оптимізації енергоспоживання IoT-пристроїв у смарт-будівлях;
- дослідити існуючі підходи та методи зменшення енергоспоживання в IoT-системах;
- сформулювати задачу дослідження та визначити основні вимоги до інтелектуального модуля;
- розробити архітектуру модуля оптимізації режимів роботи IoT-пристроїв;
- обґрунтувати вибір методів машинного навчання та алгоритму прийняття рішень;
- реалізувати програмний модуль і підготувати дані для його тестування;
- провести експериментальне дослідження роботи модуля та оцінити ефективність запропонованого підходу.

5. Перелік графічного матеріалу в роботі

- загальна схема роботи інтелектуального модуля оптимізації режимів роботи IoT-пристроїв;
- архітектура інтелектуального модуля оптимізації режимів роботи IoT-пристроїв;
- схема процесу прийняття рішень щодо керування режимами роботи IoT-пристроїв.

6. Консультанти розділів кваліфікаційної роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		Завдання видав	Завдання прийняв

7. Дата видачі завдання 01 грудня 2025 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів кваліфікаційної роботи	Примітка
1	Затвердження теми кваліфікаційної роботи, ознайомлення з літературними джерелами та складання плану роботи	до 01.01. 2026 р.	
2	Написання 1 розділу кваліфікаційної роботи	до 01.02. 2026 р.	
3	Написання 2 розділу кваліфікаційної роботи	до 15.03.2026 р.	
4	Написання 3 розділу кваліфікаційної роботи	до 01.05. 2026 р.	
5	Представлення попереднього варіанту кваліфікаційної роботи, перевірка та внесення змін керівником	до 15.05.2026 р.	
6	Опрацювання зауважень та представлення завершеного варіанту кваліфікаційної роботи. Підготовка супроводжуючих документів	до 25.05.2026 р.	
7	Перевірка кваліфікаційної роботи на оригінальність тексту	до 30.05.2026 р.	
8	Оформлення кваліфікаційної роботи та отримання допуску до захисту	до 10.06.2026 р.	
9	Подання кваліфікаційної роботи до захисту на засіданні атестаційної комісії	до 10.06. 2026 р.	

Студентка _____ А.Ю. Костоглод
підпис

Керівник роботи _____ к.т.н., професор В.В. Кочан
підпис

АНОТАЦІЯ

Кваліфікаційна робота на тему «Інтелектуальний модуль оптимізації режимів роботи IoT-пристроїв для зменшення енергоспоживання в смарт-будівлях» на здобуття освітнього ступеня «бакалавр» зі спеціальності 122 «Комп'ютерні науки» освітньої програми «Комп'ютерні науки» написана обсягом в 78 сторінок і містить 9 ілюстрацій, 9 таблиць, 2 додатки та 38 використаних джерел.

Метою роботи є розроблення інтелектуального модуля оптимізації режимів роботи IoT-пристроїв для зменшення енергоспоживання в смарт-будівлях на основі методів аналізу даних і машинного навчання.

Методи досліджень включають аналіз предметної області та наукових джерел, методи системного аналізу, методи оброблення даних, методи машинного навчання для прогнозування та оцінювання режимів роботи пристроїв, а також методи програмного проєктування й експериментальної перевірки результатів.

Розроблено інтелектуальний програмний модуль оптимізації режимів роботи IoT-пристроїв, який забезпечує завантаження даних про параметри середовища та функціонування пристроїв, їх попереднє оброблення, формування ознак, навчання моделі прогнозування енергоспоживання на основі XGBoost, оцінювання альтернативних режимів роботи та вибір найбільш доцільного режиму з урахуванням енергоспоживання й функціональних обмежень.

Результати дослідження можуть бути використані для підтримки енергоефективного керування IoT-пристроями у смарт-будівлях навчальних, адміністративних, офісних або житлових приміщень, де важливим є поєднання автоматизації, зручності експлуатації та зниження енергетичних витрат.

Ключові слова: ЕНЕРГОСПОЖИВАННЯ, СМАРТ-БУДІВЛЯ, ІНТЕРНЕТ РЕЧЕЙ, ІОТ-ПРИСТРОЇ, GREEN ІОТ, МАШИННЕ НАВЧАННЯ, ПРОГНОЗУВАННЯ, XGBOOST, ОПТИМІЗАЦІЯ РЕЖИМІВ РОБОТИ, ЕНЕРГОЕФЕКТИВНІСТЬ, ІНТЕЛЕКТУАЛЬНИЙ МОДУЛЬ.

ANNOTATION

Qualification work on the topic «Intelligent Module for Optimizing the Operating Modes of IoT Devices to Reduce Energy Consumption in Smart Buildings» for Bachelor's degree on speciality 122 «Computer Science» educational and professional program «Computer Science» is written on 78 pages and it contains 8 figures, 9 tables, 2 annexes and 38 sources.

The purpose of the work is to develop an intelligent module for optimizing the operating modes of IoT devices to reduce energy consumption in smart buildings based on data analysis and machine learning methods.

The research methods include analysis of the subject area and scientific sources, methods of systems analysis, data processing methods, machine learning methods for forecasting and evaluating device operating modes, as well as methods of software design and experimental verification of results.

An intelligent software module for optimizing the operating modes of IoT devices has been developed. The module provides loading of data on environmental parameters and device operation, preliminary data processing, feature formation, training of an energy consumption forecasting model based on XGBoost, evaluation of alternative operating modes, and selection of the most appropriate mode taking into account energy consumption and functional constraints.

The research results can be used to support energy-efficient control of IoT devices in smart buildings of educational, administrative, office, or residential premises, where the combination of automation, ease of operation, and reduction of energy costs is important.

Keywords: ENERGY CONSUMPTION, SMART BUILDING, INTERNET OF THINGS, IOT DEVICES, GREEN IOT, MACHINE LEARNING, FORECASTING, XGBOOST, OPTIMIZATION OF OPERATING MODES, ENERGY EFFICIENCY, INTELLIGENT MODULE.

ЗМІСТ

Вступ.....	7
1 Аналіз проблемної області та постановка задачі	10
1.1 Опис предметної області оптимізації енергоспоживання IoT-пристроїв у смарт-будівлях	10
1.2 Аналіз існуючих підходів і методів зменшення енергоспоживання в IoT- системах	14
1.3 Постановка задачі дослідження.....	18
2 Проєктування інтелектуального модуля оптимізації режимів роботи IoT- пристроїв	21
2.1 Формалізація задачі оптимізації та вимоги до інтелектуального модуля	21
2.2 Розроблення архітектури модуля оптимізації режимів роботи IoT- пристроїв.....	26
2.3 Обґрунтування вибору методів машинного навчання та алгоритму оптимізації	30
2.4 Моделювання процесу прийняття рішень щодо керування режимами роботи пристроїв.....	35
3 Реалізація та експериментальні дослідження програмного модуля	41
3.1 Програмна реалізація інтелектуального модуля	41
3.2 Організація експерименту та опис тестових даних	47
3.3 Аналіз результатів роботи модуля та оцінювання ефективності оптимізації	52
Висновки	52
Список використаних джерел	60
Додаток А Код програмного забезпечення	65
Додаток Б Копія публікації	73

ВСТУП

У сучасних умовах цифровізації будівельної інфраструктури зростає кількість IoT-пристроїв, що використовуються для моніторингу стану приміщень, керування освітленням, вентиляцією, опаленням, безпекою та іншими підсистемами smart-будівель. Такі пристрої забезпечують підвищення комфорту, автоматизацію рутинних процесів і розширення можливостей диспетчеризації, однак одночасно формують нову проблему – зростання сукупного енергоспоживання розподіленої мережі сенсорів, контролерів і виконавчих модулів. Особливої актуальності ця проблема набуває в тих системах, де пристрої функціонують безперервно або працюють у змінних умовах навантаження.

Енергоспоживання в smart-будівлях визначається не лише характеристиками окремих пристроїв, а й режимами їх роботи, частотою обміну даними, сценаріями активації, параметрами середовища та поведінкою користувачів. Через це традиційні підходи, що ґрунтуються на фіксованих правилах або ручному налаштуванні, часто виявляються недостатньо гнучкими. У таких умовах доцільним є використання інтелектуальних методів, здатних аналізувати дані функціонування IoT-пристроїв, виявляти закономірності енергоспоживання та формувати рішення щодо вибору більш економних режимів роботи [2, 18, 26].

Актуальність теми зумовлена кількома чинниками. По-перше, smart-будівлі поступово переходять від окремих автоматизованих функцій до комплексних кіберфізичних систем, у яких важливого значення набуває енергоефективність. По-друге, зростання кількості IoT-вузлів призводить до підвищення навантаження на мережеву й обчислювальну інфраструктуру, а також до додаткових витрат електроенергії. По-третє, сучасні методи машинного навчання дають змогу не лише прогнозувати споживання енергії, а й підтримувати прийняття рішень щодо оптимізації роботи пристроїв з урахуванням реальних умов експлуатації [5, 16, 32]. Отже, розроблення інтелектуального модуля, орієнтованого на зменшення енергоспоживання в smart-будівлях, є практично значущим і науково обґрунтованим завданням.

Аналіз наукових джерел свідчить, що питання прогнозування та керування

енергоспоживанням у смарт-будівлях активно досліджується. У працях розглядаються моделі прогнозування на основі дерев рішень, ансамблевих методів, нейронних мереж, LSTM-моделей та гібридних підходів [1, 4, 11, 19]. Окрему увагу приділено концепціям Green IoT, edge AI та інтелектуальному керуванню енергетичними процесами у цифрових середовищах [3, 13, 17, 30, 33]. Водночас значна частина наявних рішень або орієнтована переважно на прогнозування енергоспоживання, або потребує складної інфраструктури впровадження. Це зумовлює доцільність створення прикладного програмного модуля, який поєднує аналіз даних, оцінювання режимів роботи IoT-пристроїв і формування рекомендацій або керувальних дій для зменшення енергетичних витрат.

Метою роботи є розроблення інтелектуального модуля оптимізації режимів роботи IoT-пристроїв для зменшення енергоспоживання в смарт-будівлях на основі методів аналізу даних і машинного навчання.

Для досягнення поставленої мети необхідно розв'язати такі завдання:

- проаналізувати предметну область оптимізації енергоспоживання IoT-пристроїв у смарт-будівлях;
- дослідити існуючі підходи та методи зменшення енергоспоживання в IoT-системах;
- сформулювати задачу дослідження та визначити основні вимоги до інтелектуального модуля;
- розробити архітектуру модуля оптимізації режимів роботи IoT-пристроїв;
- обґрунтувати вибір методів машинного навчання та алгоритму прийняття рішень;
- реалізувати програмний модуль і підготувати дані для його тестування;
- провести експериментальне дослідження роботи модуля та оцінити ефективність запропонованого підходу.

Об'єктом дослідження є процеси керування режимами роботи IoT-пристроїв у смарт-будівлях.

Предметом дослідження є методи, моделі та програмні засоби інтелектуальної оптимізації енергоспоживання IoT-пристроїв.

У роботі використано такі методи дослідження: аналіз предметної області та наукових джерел, методи системного аналізу, методи оброблення даних, методи машинного навчання для прогнозування та оцінювання режимів роботи пристроїв, а також методи програмного проєктування й експериментальної перевірки результатів.

Практичне значення одержаних результатів полягає в можливості використання розробленого модуля для підтримки енергоефективного керування IoT-пристроями у смарт-будівлях. Запропонований підхід може бути застосований у навчальних, адміністративних, офісних або житлових приміщеннях, де важливим є поєднання автоматизації, зручності експлуатації та зниження енергетичних витрат.

Результати кваліфікаційної роботи апробовані на міжнародній науковій інтернет-конференції «Інформаційне суспільство: технологічні, економічні та технічні аспекти становлення», м. Тернопіль, Україна, м. Ополе, Польща, 7-8 травня 2026 р. (додаток Б).

Кваліфікаційна робота складається із вступу, трьох розділів, висновків, списку використаних джерел та додатків.

1 АНАЛІЗ ПРОБЛЕМНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАДАЧІ

1.1 Опис предметної області оптимізації енергоспоживання IoT-пристроїв у смарт-будівлях

Смарт-будівля є складною технічною системою, у межах якої поєднуються засоби збору даних, виконавчі механізми, мережеві комунікації, програмні сервіси та модулі автоматизованого керування. Її функціонування ґрунтується на безперервній взаємодії між фізичним середовищем і цифровою інфраструктурою. У такій системі IoT-пристрої виконують роль базових елементів спостереження, передавання даних і реалізації керувальних дій. Саме тому ефективність смарт-будівлі значною мірою визначається не лише якістю алгоритмів керування, а й раціональністю режимів роботи підключених пристроїв [2, 18].

До складу IoT-інфраструктури смарт-будівлі зазвичай входять датчики температури, вологості, освітленості, руху, якості повітря, споживання електроенергії, а також контролери, реле, інтелектуальні лічильники, камери, термостати та інші вузли автоматизації. Частина таких пристроїв працює безперервно, частина – періодично або подієво. Унаслідок цього формується багаторівнева система енергоспоживання, у якій витрати електроенергії залежать від кількості пристроїв, інтенсивності обміну даними, частоти вимірювань, обчислювального навантаження та тривалості активного режиму [3, 30].

Особливість смарт-будівлі полягає в тому, що енергоспоживання тут має розподілений характер. Електроенергія витрачається не лише на основні підсистеми, наприклад освітлення, опалення, вентиляцію чи кондиціювання, а й на численні периферійні IoT-вузли, які забезпечують збір і передавання інформації. Окремо взятий сенсор споживає небагато, проте сукупне навантаження великої кількості таких пристроїв стає відчутним, особливо за умови постійної роботи, частих передач даних або використання бездротових протоколів зв'язку. Через це задача зменшення енергоспоживання повинна розглядатися не лише на рівні будівлі в цілому, а й на рівні окремих пристроїв та їхніх режимів функціонування.

У предметній області дослідження доцільно виділити кілька основних компонентів. Першим компонентом є середовище смарт-будівлі, у якому

функціонують технічні підсистеми та перебувають користувачі. Другим є мережа IoT-пристроїв, що фіксує параметри середовища та реалізує команди керування. Третім компонентом виступає інформаційно-аналітичний рівень, де відбувається накопичення даних, їх оброблення, прогнозування та вироблення рішень. Четвертим є модуль керування режимами роботи, який змінює параметри функціонування пристроїв відповідно до поставлених цілей, зокрема з метою енергозбереження.

У загальному вигляді логіку цієї предметної області можна подати як цикл: спостереження за станом середовища, збирання телеметрії, аналіз даних, оцінювання доцільного режиму роботи пристроїв, передавання команд і повторне вимірювання результату. Такий цикл є основою інтелектуального керування, оскільки дає змогу не просто реагувати на події, а адаптувати поведінку системи до змінних умов експлуатації.

У смарт-будівлях IoT-пристрої найчастіше обслуговують такі функціональні напрями:

- моніторинг мікроклімату приміщень;
- керування освітленням;
- керування системами опалення, вентиляції та кондиціонування;
- контроль доступу та безпеки;
- облік енергоресурсів;
- виявлення присутності користувачів і побудова сценаріїв автоматизації.

Кожен із цих напрямів безпосередньо пов'язаний з енергоспоживанням. Наприклад, надто часте зчитування сенсорних даних або постійне перебування пристрою у стані високої активності може бути технічно необґрунтованим. Аналогічно, передавання даних у реальному часі в ситуаціях, де достатньо пакетної або періодичної синхронізації, збільшує витрати енергії без суттєвого підвищення якості сервісу. Саме тому в центрі дослідження перебуває питання вибору такого режиму роботи пристрою, який забезпечує потрібну функціональність за менших енергетичних витрат.

Поняття режиму роботи IoT-пристрою в цій роботі доцільно розуміти як

сукупність параметрів його функціонування в певний проміжок часу. До таких параметрів можуть належати частота зчитування даних, інтервал передавання повідомлень, період сну, тривалість активного стану, рівень обчислювального навантаження, використання локальної або хмарної обробки, а також правила переходу між станами. Зміна хоча б одного з цих параметрів може впливати як на якість роботи системи, так і на рівень споживання енергії.

У практичних системах зазвичай використовують кілька базових режимів роботи: активний, економний, черговий, адаптивний або подієвий. Активний режим орієнтований на максимальну швидкість реагування та найвищу точність моніторингу, але супроводжується підвищеним енергоспоживанням. Економний режим, навпаки, зменшує частоту операцій і передавання даних, що дозволяє заощаджувати енергію, проте може погіршувати оперативність отримання інформації. Адаптивний режим є більш доцільним для інтелектуальних систем, оскільки передбачає динамічне налаштування параметрів залежно від умов середовища, часу доби, присутності людей або історичних даних споживання [13, 25].

Задача оптимізації режимів роботи IoT-пристроїв у смарт-будівлі має багатокритеріальний характер. З одного боку, необхідно мінімізувати енергоспоживання. З іншого – потрібно зберегти належний рівень якості функціонування системи. Це означає, що рішення про перехід пристрою в енергозберігаючий режим не може прийматися ізольовано від контексту. Наприклад, надмірне зниження частоти моніторингу в системі контролю температури може призвести до погіршення мікроклімату, а надто агресивне переведення сенсорів руху в сплячий режим – до втрати важливих подій. Отже, предметна область охоплює не лише енергетичний аспект, а й баланс між енергоефективністю, функціональною повнотою та комфортом користувачів.

Важливою особливістю цієї сфери є наявність великого обсягу різномірних даних. Смарт-будівля генерує часові ряди сенсорних вимірювань, журнали подій, відомості про стани пристроїв, дані про присутність користувачів, команди керування, а також параметри зовнішнього середовища. Такі дані мають різну структуру, різну періодичність оновлення та різний рівень повноти. Саме тому

традиційні статичні правила не завжди дають змогу отримати достатньо гнучке рішення. У свою чергу, методи машинного навчання дозволяють виявляти приховані закономірності між параметрами середовища, режимами роботи пристроїв і фактичним енергоспоживанням [5, 16].

У контексті даної теми особливу роль відіграють системи, орієнтовані на концепцію Green IoT. Її суть полягає в тому, щоб забезпечити функціонування IoT-інфраструктури з мінімальним негативним впливом на енергетичні ресурси та довкілля. Це досягається шляхом застосування енергоощадних протоколів зв'язку, оптимізації режимів обчислень, використання edge-підходів, скорочення надлишкових передач даних і впровадження інтелектуальних механізмів прийняття рішень [3, 13, 30]. Для смарт-будівель така концепція є особливо актуальною, оскільки саме ці об'єкти мають значний потенціал для локального керування ресурсами.

Для кращого розуміння функціональних елементів предметної області доцільно узагальнити типові категорії IoT-пристроїв та їхній вплив на енергоспоживання (таблиця 1.1).

Таблиця 1.1 – Основні категорії IoT-пристроїв у смарт-будівлях та їхні характеристики

Категорія пристроїв	Основна функція	Приклади	Фактори енергоспоживання
Сенсори середовища	Вимірювання параметрів приміщення	Датчики температури, вологості, освітленості, CO ₂	Частота вимірювань, інтервал передавання даних
Сенсори присутності	Виявлення руху або присутності людей	PIR-датчики, ультразвукові датчики	Постійний моніторинг, чутливість, очікування режим
Виконавчі пристрої	Реалізація керувальних дій	Реле, термостати, розумні розетки	Частота спрацювань, тривалість активного стану
Обчислювальні вузли	Локальне оброблення даних	Контролери, edge-шлюзи	Обсяг обчислень, передавання даних, режим процесора
Комунікаційні модулі	Передавання даних у мережі	Wi-Fi, ZigBee, BLE, LoRa-модулі	Потужність сигналу, частота сеансів зв'язку

Дані, наведені в таблиці 1.1, показують, що джерелом енергоспоживання є не

лише сам факт роботи пристрою, а й конкретні параметри його використання. Це підтверджує доцільність переходу від статичного керування до інтелектуального модуля, який враховує контекст функціонування системи.

У межах предметної області важливо також враховувати обмеження, що впливають на процес оптимізації. До них належать апаратні обмеження пристроїв, нестабільність бездротового зв'язку, обмежений ресурс автономного живлення, неоднорідність протоколів, потреба в сумісності з наявною інфраструктурою будівлі, а також вимоги до надійності й безпеки. Крім того, система керування не повинна порушувати базові функції будівлі, пов'язані з безпекою, комфортом і безперервністю основних сервісів.

Таким чином, предметна область дослідження охоплює сукупність технічних, інформаційних та алгоритмічних засобів, спрямованих на раціональне керування режимами роботи IoT-пристроїв у смарт-будівлях. Її ключовою особливістю є поєднання розподіленого енергоспоживання, динамічного середовища, великого обсягу телеметричних даних і потреби в адаптивному прийнятті рішень. У таких умовах інтелектуальний модуль оптимізації виступає не допоміжним елементом, а центральною ланкою, що забезпечує зменшення енергоспоживання без істотного погіршення функціональних характеристик системи.

1.2 Аналіз існуючих підходів і методів зменшення енергоспоживання в IoT-системах

Зменшення енергоспоживання в IoT-системах є однією з ключових умов побудови сучасних смарт-будівель і смарт-міст. Це пояснюється тим, що цифровізація інфраструктури супроводжується стрімким зростанням кількості підключених пристроїв, сенсорів, контролерів і комунікаційних вузлів. У результаті навіть відносно мале енергоспоживання окремого IoT-пристрою на рівні всієї системи перетворюється на суттєвий фактор, що впливає на загальні витрати енергії, навантаження на мережу та екологічну стійкість цифрового середовища [3, 30].

У наукових працях останніх років простежується перехід від локальних енергоощадних рішень до комплексних підходів, орієнтованих на інтегроване керування енергетичними процесами. Значна увага приділяється концепції Smart Green Energy, у межах якої IoT-засоби використовуються для моніторингу, аналізу та оптимізації споживання енергії в міській і будівельній інфраструктурі. Такі системи передбачають використання датчиків, комунікаційних платформ та інтелектуальних алгоритмів для прогнозування попиту на енергію, коригування режимів роботи обладнання та зменшення експлуатаційних витрат [33]. У цьому контексті важливого значення набувають моделі, здатні працювати не лише як засоби спостереження, а і як інструменти прийняття рішень.

Окремий напрям досліджень пов'язаний із концепцією Green IoT, яка спрямована на зниження енергетичного та екологічного навантаження, створюваного самою IoT-інфраструктурою. На відміну від традиційного підходу, де IoT розглядається переважно як засіб автоматизації, Green IoT акцентує увагу на тому, що цифрові сервіси також мають власну вартість з точки зору енергоспоживання, викидів та використання обчислювальних ресурсів. У межах цього підходу аналізуються енергоефективні архітектури, оптимізація мережевих протоколів, раціональне використання ресурсів, інтеграція відновлюваних джерел енергії та формування стратегій сталого функціонування IoT-систем [3, 30].

Для смарт-будівель Green IoT є особливо важливим, оскільки будівельний сектор залишається одним із найбільших споживачів енергії. За цієї умови впровадження лише традиційних засобів автоматизації вже не є достатнім. Необхідними стають рішення, які поєднують IoT, машинне навчання, edge computing та аналітичні сервіси в межах єдиної архітектури керування енергоспоживанням. Такі системи дають змогу збирати телеметрію в реальному часі, оцінювати поточний стан об'єкта, прогнозувати майбутнє навантаження та адаптувати режими роботи обладнання. Перспективність цього підходу підтверджується роботами, у яких запропоновано наскрізні архітектури енергоменеджменту будівель із використанням IoT-моніторингу, ML-моделей та зрозумілих інтерфейсів для користувача [17, 26].

Серед існуючих методів зменшення енергоспоживання в IoT-системах

доцільно виокремити кілька основних груп.

Першу групу становлять апаратно-мережеві методи. До них належать використання енергоощадних протоколів зв'язку, зниження потужності передавачів, регулювання частоти сеансів передавання, оптимізація режимів сну та пробудження пристроїв, а також застосування технологій на кшталт NB-IoT, що забезпечують ширше покриття та більший час автономної роботи. Перевагою цих методів є відносна простота впровадження на рівні інфраструктури. Водночас їхній недолік полягає в тому, що вони здебільшого не враховують функціональний контекст використання пристрою та не формують інтелектуальних стратегій керування [30].

Другу групу утворюють архітектурні методи, що передбачають зміну загальної організації оброблення даних у системі. У сучасних роботах значна увага приділяється перенесенню частини обчислень із централізованих хмарних сервісів на периферійні вузли, тобто до edge-рівня. Такий підхід зменшує затримки, скорочує обсяг переданих даних і в багатьох випадках дозволяє ефективніше використовувати енергетичні ресурси. Разом із цим виникає нова суперечність: локальне виконання AI-алгоритмів також потребує енергії, тому позитивний ефект edge AI залежить від збалансованого розподілу функцій між пристроєм, шлюзом і серверною частиною [13, 17]. Отже, саме по собі перенесення обчислень на периферію ще не гарантує енергоефективності.

Третю групу становлять алгоритмічні та інтелектуальні методи. Саме вони нині розглядаються як найперспективніші, оскільки дозволяють перейти від жорстко заданих правил до адаптивного керування. У цій групі використовуються методи машинного навчання, глибокого навчання, генетичні алгоритми, багатоагентні моделі та підходи на основі підкріплення. Їхнє призначення полягає в тому, щоб на основі історичних та поточних даних визначати оптимальні режими роботи пристроїв, прогнозувати енергоспоживання і мінімізувати надлишкові витрати [25, 28, 32].

Особливий інтерес становлять рішення, у яких IoT-сенсори поєднуються з методами deep reinforcement learning. У таких системах алгоритм не лише аналізує накопичені дані, а й навчається вибирати кращу дію в конкретній ситуації,

наприклад змінювати частоту вимірювань, переводити пристрої в економний режим або коригувати графік споживання енергії. Перевага цього підходу полягає в адаптивності та здатності працювати в динамічному середовищі. Недоліком є складність навчання, потреба в якісних даних і підвищені вимоги до обчислювальних ресурсів [33].

У низці досліджень задача енергоменеджменту розглядається як проблема оптимізації для великої кількості взаємопов'язаних IoT-пристроїв. За такого підходу система аналізує не лише параметри окремого вузла, а й поведінку всієї мережі, намагаючись досягти енергетичної прозорості, погодженості дій і зниження сумарного навантаження. Для цього пропонуються багатоагентні моделі, генетичні алгоритми та інші евристичні процедури. Перевагою цих рішень є врахування децентралізованої природи IoT-середовища. Разом із тим складність таких моделей зростає зі збільшенням кількості пристроїв і сценаріїв взаємодії [25].

У сучасній літературі також підкреслюється, що розвиток Green IoT неможливий без урахування ширшого контексту сталого розвитку. Йдеться не лише про економію електроенергії, а й про зменшення вуглецевого сліду цифрової інфраструктури, раціональне використання апаратних ресурсів, підвищення прозорості рішень AI та можливість поєднання IoT-систем із відновлюваними джерелами енергії [3, 13]. Саме тому дедалі частіше як перспективний напрям розглядається інтеграція Green IoT, edge AI, локального прогнозування та інтелектуального керування навантаженням.

Попри значну кількість досліджень, аналіз літератури дозволяє виділити низку обмежень існуючих підходів. По-перше, частина рішень орієнтована переважно на рівень смарт-міста або енергосистеми в цілому, тоді як рівень окремої смарт-будівлі та окремих IoT-пристроїв розглядається менш детально. По-друге, у багатьох роботах переважає або моніторинг, або прогнозування, але не повний цикл інтелектуального керування режимами роботи. По-третє, практичне впровадження складних моделей часто стримується апаратними обмеженнями периферійних пристроїв, неоднорідністю протоколів і вимогами до надійності. По-четверте, залишається актуальною проблема балансу між енергоощадністю та якістю сервісу: надмірне скорочення активності пристроїв може погіршити

функціонування системи в цілому [17, 30].

Для узагальнення розглянутих підходів подамо їх у вигляді таблиці 1.2.

Таблиця 1.2 – Порівняльна характеристика підходів до зменшення енергоспоживання в IoT-системах

Підхід	Сутність	Переваги	Недоліки
Апаратно-мережевий	Оптимізація протоколів зв'язку, режимів сну, потужності передавання	Простота впровадження, зменшення базових витрат енергії	Обмежена адаптивність, слабе врахування контексту
Архітектурний	Перерозподіл обчислень між хмарою, edge-вузлами та пристроями	Зменшення затримок, скорочення трафіку, локальна обробка	Можливе зростання навантаження на edge-пристрої
Алгоритмічний	Використання ML, DL, генетичних алгоритмів, багатоагентних моделей	Адаптивність, прогнозування, підтримка рішень	Вища складність реалізації та налаштування
Green IoT-орієнтований	Поєднання енергоефективності, екологічності та сталого розвитку	Комплексне бачення проблеми, перспективність для смарт-будівель	Потребує інтеграції багатьох технологій і критеріїв

Отже, проведений аналіз показує, що існуючі підходи до зменшення енергоспоживання в IoT-системах поступово еволюціонують від окремих технічних рішень до інтегрованих інтелектуальних систем керування. Найбільш перспективними для смарт-будівель є підходи, у яких поєднуються IoT-моніторинг, аналітика даних, прогнозування та адаптивна оптимізація режимів роботи пристроїв. Саме такі рішення дозволяють враховувати реальний контекст експлуатації, зберігати належну якість функціонування та одночасно знижувати енергетичні витрати. Це створює підстави для формулювання задачі розроблення інтелектуального модуля, здатного підтримувати прийняття рішень щодо енергоефективного керування IoT-пристроями у смарт-будівлях.

1.3 Постановка задачі дослідження

Проведений аналіз предметної області та існуючих підходів до зменшення енергоспоживання в IoT-системах показав, що смарт-будівлі є складними динамічними середовищами, у яких енергоефективність залежить не лише від технічних характеристик обладнання, а й від способу його використання, частоти

взаємодії з мережею, умов середовища та сценаріїв експлуатації. Існуючі рішення забезпечують певний рівень автоматизації, однак далеко не завжди враховують змінність контексту та потребу в адаптивному керуванні режимами роботи пристроїв. У результаті виникає практична потреба у створенні інтелектуального модуля, здатного аналізувати параметри функціонування IoT-пристроїв і формувати рішення, спрямовані на зменшення енергоспоживання без суттєвого погіршення якості роботи системи [17, 25].

У межах даної роботи розглядається задача оптимізації режимів роботи IoT-пристроїв у смарт-будівлі. Її суть полягає у виборі таких параметрів функціонування пристроїв, за яких забезпечується раціональний баланс між енергоефективністю та функціональною доцільністю. До таких параметрів можуть належати частота вимірювань, інтервали передавання даних, тривалість активного стану, використання режимів сну, інтенсивність локального оброблення інформації та правила переходу між робочими станами. Зміна цих параметрів безпосередньо впливає на рівень споживання енергії, швидкодію системи, точність моніторингу та стабільність сервісу.

З технічної точки зору поставлена задача має кілька важливих особливостей. По-перше, вхідні дані є різномірними та змінними в часі. До них належать показники сенсорів, стани пристроїв, часові характеристики, параметри середовища, інформація про активність користувачів та історія енергоспоживання. По-друге, рішення повинно прийматися з урахуванням обмежень, пов'язаних із ресурсами IoT-пристроїв, специфікою бездротового зв'язку та вимогами до безперервності функціонування системи. По-третє, процес оптимізації не може зводитися лише до мінімізації енергоспоживання, оскільки надмірне обмеження активності пристроїв може погіршити точність моніторингу, швидкість реакції або комфорт користувачів.

Отже, задача дослідження формулюється як задача розроблення інтелектуального модуля, який на основі аналізу вхідних даних про середовище смарт-будівлі та параметри роботи IoT-пристроїв забезпечує вибір доцільного режиму функціонування з метою зменшення енергоспоживання.

Тому, метою роботи є розроблення інтелектуального модуля оптимізації

режимів роботи IoT-пристроїв для зменшення енергоспоживання в смарт-будівлях на основі методів аналізу даних і машинного навчання.

Для досягнення поставленої мети необхідно розв'язати такі завдання:

- проаналізувати предметну область оптимізації енергоспоживання IoT-пристроїв у смарт-будівлях;
- дослідити існуючі підходи та методи зменшення енергоспоживання в IoT-системах;
- сформулювати задачу дослідження та визначити основні вимоги до інтелектуального модуля;
- розробити архітектуру модуля оптимізації режимів роботи IoT-пристроїв;
- обґрунтувати вибір методів машинного навчання та алгоритму прийняття рішень;
- реалізувати програмний модуль і підготувати дані для його тестування;
- провести експериментальне дослідження роботи модуля та оцінити ефективність запропонованого підходу.

Об'єктом дослідження є процеси керування режимами роботи IoT-пристроїв у смарт-будівлях.

Предметом дослідження є методи, моделі та програмні засоби інтелектуальної оптимізації енергоспоживання IoT-пристроїв.

2 ПРОЄКТУВАННЯ ІНТЕЛЕКТУАЛЬНОГО МОДУЛЯ ОПТИМІЗАЦІЇ РЕЖИМІВ РОБОТИ ІОТ-ПРИСТРОЇВ

2.1 Формалізація задачі оптимізації та вимоги до інтелектуального модуля

Проектування інтелектуального модуля оптимізації режимів роботи ІоТ-пристроїв потребує чіткого формалізованого опису задачі, який дає змогу визначити склад вхідних даних, цільові показники, обмеження та логіку прийняття рішень. У межах смарт-будівлі така задача не зводиться лише до технічного керування окремими сенсорами чи виконавчими вузлами. Вона охоплює аналіз взаємозв'язку між параметрами середовища, режимами функціонування пристроїв і фактичним рівнем енергоспоживання. Саме тому на етапі проектування необхідно визначити формальну модель системи, функціональні можливості модуля та основні вимоги до його роботи [1, 18].

У загальному вигляді система розглядається як сукупність ІоТ-пристроїв, що функціонують у межах смарт-будівлі та взаємодіють із середовищем і модулем керування. Нехай існує множина пристроїв

$$D = \{d_1, d_2, d_3, \dots, d_n\}, \quad (2.1)$$

де d_i – окремий ІоТ-пристрій, який характеризується певним набором параметрів стану, функціонування та енергоспоживання.

Для кожного пристрою формується вектор ознак

$$X_i = \{x_{i1}, x_{i2}, x_{i3}, \dots, x_{im}\}, \quad (2.2)$$

де компонентами вектора можуть бути температура, вологість, рівень освітленості, час доби, поточний режим роботи, інтенсивність передавання даних, тривалість активного стану, частота вимірювань, рівень завантаження пристрою, а також інші характеристики, що впливають на споживання енергії. Таким чином, кожен ІоТ-пристрій описується не одним параметром, а сукупністю чинників, які в реальних умовах змінюються в часі та залежать від сценарію використання.

Цільовим показником у межах дослідження є значення енергоспоживання пристрою або групи пристроїв за певний проміжок часу. Формально це можна подати у вигляді функції

$$E = f(X), \quad (2.3)$$

де E – енергоспоживання;

$f(X)$ – залежність енергоспоживання від сукупності вхідних параметрів.

У практичному сенсі ця функція не задається аналітично в готовому вигляді, а визначається на основі даних, отриманих із системи моніторингу або з експериментального набору даних. Саме тут доцільним є використання методів машинного навчання, які дають змогу виявляти приховані закономірності між ознаками й цільовим показником [16, 32].

Завдання інтелектуального модуля полягає у визначенні такого режиму роботи R , за якого енергоспоживання буде мінімізоване. Формально це можна записати так:

$$R^* = \arg \min E(R, X), \quad (2.4)$$

де R^* – оптимальний режим роботи пристрою або сукупності пристроїв.

У цьому випадку режим роботи розуміється як сукупність параметрів функціонування, зокрема частота опитування сенсорів, інтервали передавання даних, тривалість активного стану, періоди сну, рівень локального обчислювального навантаження та інші характеристики, які можуть змінюватися в межах допустимого діапазону.

Разом із цим оптимізація не може виконуватися без урахування функціональних обмежень. Надмірне зниження активності пристрою може зменшити енергоспоживання, але водночас призвести до втрати якості моніторингу, уповільнення реакції системи або погіршення умов експлуатації смарт-будівлі. Тому задача оптимізації розглядається за умови збереження мінімально допустимого рівня функціональності:

$$Q(R, X) \geq Q_{min}, \quad (2.5)$$

де Q – показник якості функціонування системи;

Q_{min} – мінімально допустимий рівень цієї якості.

Показник Q може характеризувати точність збору даних, стабільність передавання, своєчасність реагування або інший критерій, важливий для конкретної підсистеми будівлі.

Отже, задача має ознаки багатокритеріальної оптимізації. Необхідно не просто зменшити енергетичні витрати, а забезпечити таке керування, за якого

система зберігатиме працездатність, інформативність і практичну корисність. У цьому полягає принципова відмінність інтелектуального підходу від жорстко фіксованих правил. Якщо традиційна схема працює за наперед визначеним сценарієм, то інтелектуальний модуль повинен враховувати змінний контекст середовища та приймати рішення на основі аналізу даних.

Для реалізації такого підходу в роботі передбачається створення програмного модуля, який виконує послідовність функціональних операцій. На першому етапі модуль повинен здійснювати збір і приймання вхідних даних від IoT-пристроїв або з підготовленого набору даних. На другому етапі необхідним є попереднє оброблення інформації, що охоплює очищення, структурування, перетворення та формування набору ознак. Далі виконується оцінювання або прогнозування енергоспоживання для заданого стану системи. Після цього модуль має визначити доцільний режим роботи пристрою та сформулювати результат у вигляді рекомендації або керувальної дії. Окремо повинна забезпечуватися можливість накопичення даних для подальшого аналізу ефективності роботи модуля.

Таким чином, логіка функціонування модуля має такий узагальнений вигляд (рисунок 2.1):

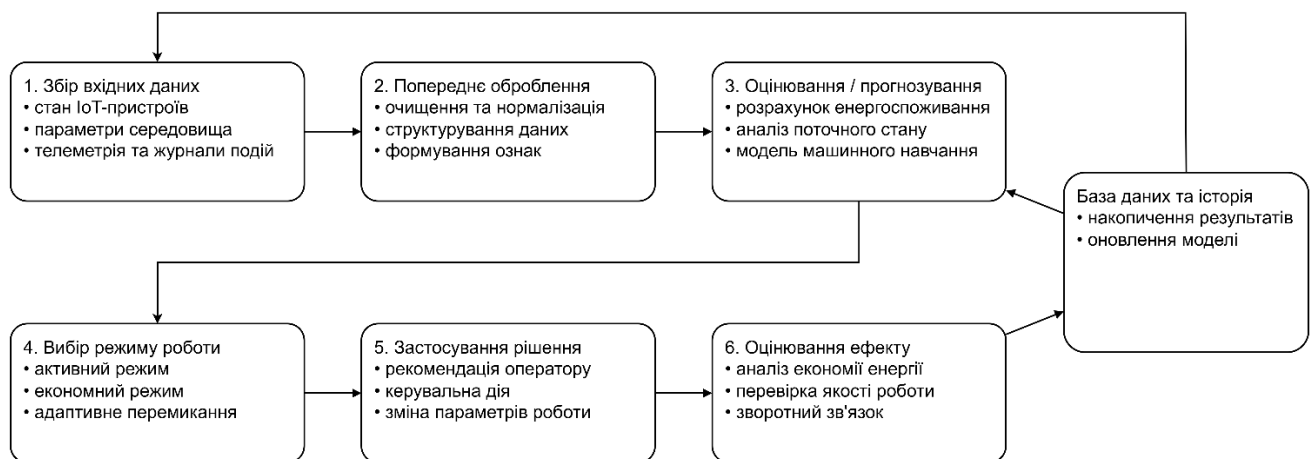


Рисунок 2.1 – Загальна схема роботи інтелектуального модуля оптимізації режимів роботи IoT-пристроїв

- збір вхідних даних про стан пристроїв і середовища;
- попереднє оброблення даних;
- формування набору ознак;

- оцінювання або прогнозування енергоспоживання;
- вибір доцільного режиму роботи;
- застосування рішення;
- оцінювання отриманого ефекту.

Формалізація задачі дозволяє перейти до визначення основних вимог до модуля. Насамперед модуль повинен бути адаптивним. Це означає, що він має враховувати зміни вхідних параметрів і реагувати на різні умови функціонування системи. Така вимога є базовою, оскільки смарт-будівля не є статичним середовищем: у ній змінюються параметри мікроклімату, наявність користувачів, інтенсивність обміну даними та режими навантаження.

Другою важливою вимогою є енергоорієнтованість. Логіка прийняття рішень повинна прямо спрямовуватися на зменшення витрат електроенергії, а не лише на підтримання стандартної функціональності. Це означає, що показник енергоспоживання має розглядатися як один із ключових критеріїв вибору режиму роботи, а не як другорядна характеристика.

Третьою вимогою є пояснюваність на прикладному рівні. У межах бакалаврської роботи доцільно використовувати такі підходи, результати яких можна інтерпретувати в контексті задачі керування пристроями. Це особливо важливо в системах смарт-будівель, де адміністратор або користувач повинен розуміти, чому модуль рекомендує саме певний режим роботи, а не інший [17].

Четвертою вимогою є масштабованість. У реальних умовах кількість IoT-пристроїв у будівлі може змінюватися, а система повинна зберігати працездатність як для невеликого набору вузлів, так і для більшої мережі сенсорів і контролерів. Звідси випливає потреба у такій архітектурі, яка не залежить жорстко від фіксованої кількості пристроїв.

П'ятою вимогою є простота експериментальної реалізації та розширення. Оскільки в роботі розробляється прототип, програмний модуль повинен бути достатньо простим для тестування, модифікації та подальшого розвитку. Це дає змогу перевірити працездатність запропонованого підходу без надмірного ускладнення системи.

Доцільно також враховувати низку додаткових вимог. Модуль повинен бути

сумісним із даними, що надходять із різних типів пристроїв, підтримувати базове логування результатів, забезпечувати можливість повторного навчання або переналаштування моделі, а також не потребувати надлишкових обчислювальних ресурсів. Остання вимога є особливо важливою в контексті Green IoT, де енергоефективність самої аналітичної підсистеми також має значення [3, 13].

У межах кваліфікаційної роботи доцільно прийняти певні обмеження, які звужують задачу до практично реалізованого варіанта. По-перше, дослідження орієнтоване не на повномасштабну промислову систему керування будівлею, а на програмний прототип інтелектуального модуля. По-друге, основний акцент робиться на аналізі даних і виборі режимів роботи окремих IoT-пристроїв або їхніх типових груп, без охоплення всіх підсистем смарт-будівлі одночасно. По-третє, у роботі розглядаються насамперед інформаційно-аналітичні та програмні аспекти оптимізації, тоді як детальне проєктування апаратної частини не є основною метою дослідження. Такий підхід є цілком обґрунтованим для бакалаврської роботи, оскільки дозволяє зосередитися на ключовій задачі – зменшенні енергоспоживання шляхом інтелектуальної оптимізації режимів функціонування.

Для узагальнення вимог до розроблюваного рішення подано їх у вигляді таблиці 2.1.

Таблиця 2.1 – Основні вимоги до інтелектуального модуля

Вимога	Зміст вимоги
Адаптивність	Урахування змін параметрів середовища та стану пристроїв
Енергоорієнтованість	Спрямованість рішень на зменшення енергоспоживання
Пояснюваність	Можливість інтерпретації результатів роботи модуля
Масштабованість	Працездатність при зміні кількості пристроїв
Простота реалізації	Можливість створення й перевірки програмного прототипу
Розширюваність	Можливість подальшого вдосконалення та додавання нових функцій

Отже, у підрозділі виконано формалізацію задачі оптимізації режимів роботи IoT-пристроїв у смарт-будівлі та визначено основні вимоги до інтелектуального модуля.

2.2 Розроблення архітектури модуля оптимізації режимів роботи IoT-пристроїв

Після формалізації задачі та визначення основних вимог до інтелектуального модуля необхідно перейти до розроблення його архітектури. Саме архітектурне проектування дозволяє встановити склад функціональних компонентів системи, їхнє призначення, послідовність оброблення даних і взаємозв'язки між окремими блоками. Для даної роботи архітектура модуля повинна забезпечувати не лише приймання та аналіз інформації від IoT-пристроїв, а й формування рішень щодо вибору енергоефективного режиму роботи з урахуванням умов функціонування смарт-будівлі [18, 26].

У межах дослідження доцільно використовувати модульний підхід до побудови архітектури. Такий підхід є виправданим з кількох причин. По-перше, він дозволяє логічно відокремити функції збору даних, аналітичного опрацювання, прогнозування та прийняття рішень. По-друге, модульна структура спрощує реалізацію програмного прототипу, тестування окремих компонентів і подальше розширення системи. По-третє, така організація відповідає практиці побудови сучасних IoT-рішень, де окремі рівні системи можуть розгортатися на різних апаратних або програмних платформах [13, 31].

З позиції функціонального призначення архітектура інтелектуального модуля повинна охоплювати повний цикл роботи з даними: від їх надходження до формування рішення та оцінювання результату. У зв'язку з цим доцільно виділити кілька основних компонентів: модуль збору даних, модуль попереднього оброблення, аналітичний модуль, модуль прийняття рішень, підсистему збереження даних і модуль взаємодії з користувачем або системою керування.

Першим елементом архітектури є модуль збору даних. Його призначення полягає в отриманні вхідної інформації від IoT-пристроїв, датчиків, лічильників або з підготовленого набору даних, якщо робота виконується в режимі моделювання. На цьому рівні система приймає показники сенсорів, часові мітки, відомості про режим роботи пристроїв, інтенсивність мережевої активності, а також інші параметри, що характеризують стан середовища та поточне

функціонування вузлів. Оскільки в бакалаврській роботі розробляється програмний прототип, цей модуль може бути реалізований у спрощеному вигляді, наприклад як засіб імпорту табличних даних із файлу або приймання потоку телеметрії від емулятора.

Другим компонентом є модуль попереднього оброблення даних. Його роль полягає в підготовці інформації до подальшого аналітичного опрацювання. На цьому етапі виконуються перевірка повноти записів, усунення пропусків, нормалізація або масштабування числових ознак, перетворення категоріальних параметрів, формування часових характеристик і узгодження даних, що надходять із різних джерел. У контексті даної задачі цей блок має особливе значення, оскільки якість роботи моделі оптимізації безпосередньо залежить від якості сформованого набору ознак.

Третім компонентом виступає аналітичний модуль, який є центральним елементом архітектури. Саме тут реалізуються методи машинного навчання або інші процедури, що дозволяють оцінювати енергоспоживання та виявляти залежності між станом середовища, параметрами роботи пристроїв і рівнем витрат електроенергії. Аналітичний модуль може працювати у двох основних режимах. Перший режим передбачає прогнозування споживання енергії для заданого набору параметрів. Другий режим полягає в оцінюванні декількох можливих режимів роботи пристрою та виборі найбільш доцільного варіанта з точки зору енергоефективності. У такому випадку модель не лише описує наявний стан системи, а й підтримує формування керувального рішення [1, 18].

Четвертим елементом архітектури є модуль прийняття рішень. Його призначення полягає у перетворенні результатів аналітичного блоку на конкретне рішення щодо режиму роботи пристрою. У межах цього компонента можуть застосовуватися правила вибору режиму, порогові умови, логіка порівняння альтернативних варіантів або алгоритм оптимізації, що враховує не лише очікуване енергоспоживання, а й допустимий рівень якості функціонування системи. Саме цей блок пов'язує модель прогнозування з практичним керуванням, тому його наявність є принципово важливою для переходу від аналітичної оцінки до прикладного використання результатів.

П'ятим компонентом архітектури є підсистема збереження даних. Вона забезпечує накопичення телеметрії, збереження проміжних результатів, історії прийнятих рішень та отриманих ефектів. Такий компонент виконує одразу кілька функцій. По-перше, він дозволяє використовувати історичні дані для подальшого навчання або переналаштування моделі. По-друге, забезпечує можливість аналізу ефективності модуля за різні проміжки часу. По-третє, створює основу для аудиту рішень та повторного відтворення експериментів.

Шостим компонентом є модуль взаємодії з користувачем або зовнішньою системою. У найпростішому випадку він реалізується як інтерфейс відображення результатів, де подаються поточні значення параметрів, оцінка споживання енергії, рекомендований режим роботи та узагальнені показники ефективності. Якщо система використовується в режимі автоматизованого керування, цей блок може також виконувати роль інтерфейсу передавання команд до виконавчих пристроїв.

З урахуванням зазначених компонентів архітектуру модуля можна подати як багаторівневу структуру (рисунок 2.2).

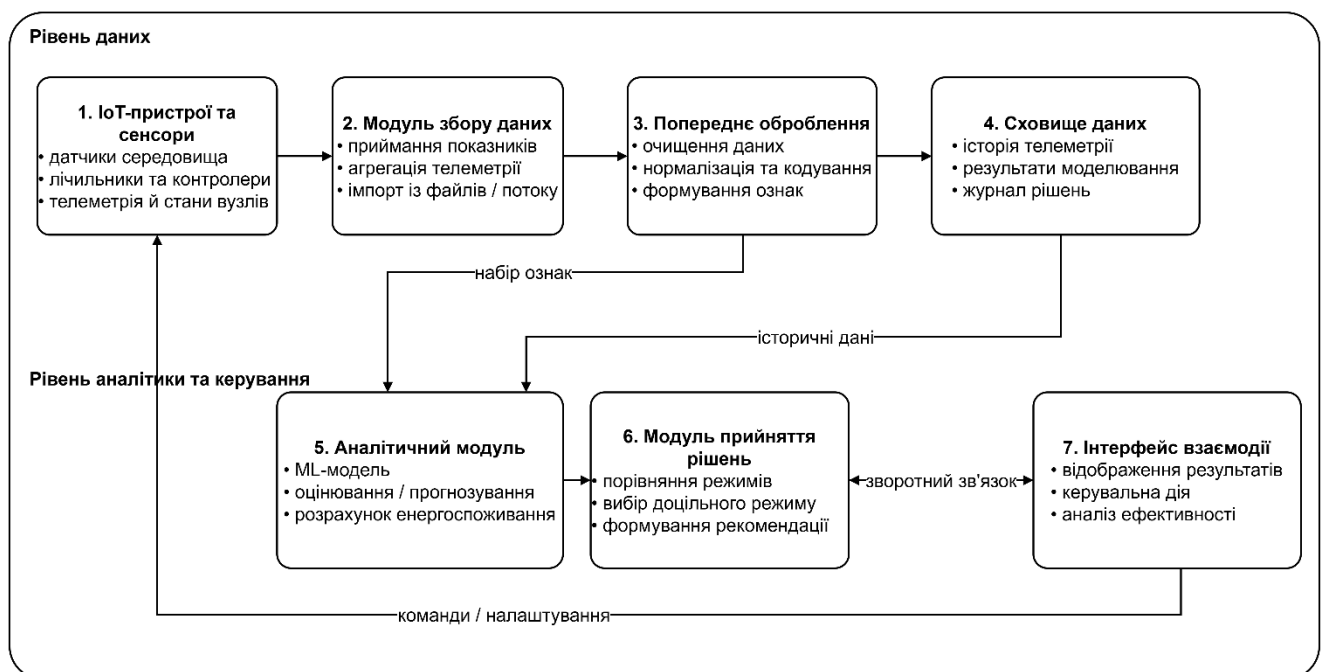


Рисунок 2.2 – Архітектура інтелектуального модуля оптимізації режимів роботи IoT-пристроїв

На нижньому рівні розміщуються джерела даних – IoT-пристрої, датчики, лічильники та інші вузли. Середній рівень охоплює попереднє оброблення,

збереження та аналітичне опрацювання. Верхній рівень включає логіку прийняття рішень і взаємодію з користувачем або системою керування. Така організація є доцільною, оскільки дозволяє відокремити рівень фізичних даних від рівня інтелектуального аналізу та прикладного керування.

Важливо зазначити, що розроблювана архітектура не є жорстко прив'язаною до конкретної технології реалізації. У межах дадної роботи вона повинна залишатися достатньо універсальною, щоб її можна було реалізувати у вигляді програмного прототипу з використанням доступних інструментів оброблення даних і машинного навчання. Водночас структура архітектури має відображати реальну логіку побудови інтелектуальної IoT-системи, що може бути масштабована або адаптована до інших сценаріїв використання.

Окремої уваги потребує питання розміщення аналітичного модуля. У практиці IoT-систем існують принаймні три базові варіанти: хмарна обробка, edge-обробка та гібридна архітектура. Хмарний підхід забезпечує вищу обчислювальну потужність і зручність централізованого керування, але супроводжується залежністю від мережевих каналів і підвищенням обсягу переданих даних. Edge-підхід, навпаки, дозволяє обробляти дані ближче до джерела їх генерації, що зменшує затримки та скорочує трафік, однак створює додаткове навантаження на локальні вузли [13, 17]. Для прототипу, розроблюваного в цій роботі, найбільш доцільною є спрощена гібридна схема, за якої основне навчання моделі може виконуватися централізовано, а використання моделі для оцінювання режимів роботи – локально або в межах прикладного програмного середовища.

Ще одним важливим аспектом архітектурного проєктування є організація зворотного зв'язку. У задачах інтелектуального керування система не повинна обмежуватися одноразовим виробленням рішення. Після застосування рекомендованого режиму необхідно оцінити, чи справді відбулося зменшення енергоспоживання та чи не було порушено допустимий рівень функціональності. Саме тому архітектура повинна включати замкнений цикл, у межах якого результати попередніх рішень повертаються до бази даних і можуть використовуватися для подальшого уточнення або переналаштування моделі.

З точки зору програмної реалізації архітектура модуля має бути побудована

так, щоб окремі блоки могли функціонувати незалежно, але при цьому взаємодіяти через чітко визначені інтерфейси. Наприклад, модуль збору даних передає підготовлені записи до модуля попереднього оброблення, той формує набір ознак для аналітичного блоку, а результати прогнозування надходять до модуля прийняття рішень. Така організація дозволяє не лише підвищити прозорість структури системи, а й спрощує відлагодження та експериментальну перевірку.

Отже, розроблена архітектура інтелектуального модуля охоплює всі ключові етапи роботи з даними та підтримки керування режимами роботи IoT-пристроїв у смарт-будівлі. Її побудова ґрунтується на принципах модульності, адаптивності, енергоорієнтованості та можливості подальшого розширення.

2.3 Обґрунтування вибору методів машинного навчання та алгоритму оптимізації

Ефективність інтелектуального модуля оптимізації режимів роботи IoT-пристроїв значною мірою визначається правильністю вибору методів оброблення даних, прогнозування енергоспоживання та формування керувального рішення. У межах даної роботи цей вибір повинен враховувати не лише точність моделі, а й практичну доцільність її використання в умовах смарт-будівлі. Йдеться про роботу з табличними даними, змінними параметрами середовища, порівняно обмеженим набором ознак, а також потребу в інтерпретованому та достатньо простому для реалізації програмному рішенні [2, 18].

На відміну від задач суто теоретичного моделювання, у даному випадку метод машинного навчання розглядається як прикладний інструмент, що має виконувати дві взаємопов'язані функції. Перша функція полягає в оцінюванні або прогнозуванні енергоспоживання IoT-пристрою залежно від параметрів його функціонування. Друга функція пов'язана з підтримкою прийняття рішення щодо вибору доцільного режиму роботи. Отже, модель повинна бути придатною до роботи з числовими та категоріальними ознаками, стійкою до неоднорідності даних і достатньо гнучкою, щоб відображати нелінійні залежності між параметрами системи та рівнем енергоспоживання.

У сучасних дослідженнях для задач прогнозування енергоспоживання в смарт-будівлях застосовуються різні класи моделей: лінійна регресія, дерева рішень, ансамблеві методи, штучні нейронні мережі, рекурентні мережі та гібридні підходи [1, 4, 11, 16, 32]. Проте придатність кожного з цих методів визначається не лише теоретичними перевагами, а й відповідністю конкретній постановці задачі.

Лінійна регресія є одним із найпростіших базових підходів. Її перевагами є прозорість, швидкість навчання та простота інтерпретації результатів. Разом із тим для задачі оптимізації режимів роботи IoT-пристроїв цього методу часто недостатньо, оскільки зв'язок між параметрами середовища, режимами роботи та споживанням енергії переважно має нелінійний характер. Лінійна модель може бути корисною як базова для порівняння, однак навряд чи забезпечить достатню точність для практичного використання в інтелектуальному модулі.

Дерева рішень є більш гнучкими, оскільки дають змогу виявляти нелінійні залежності та працювати з різними типами ознак. Крім того, вони мають добру пояснюваність, що є важливим для задач прикладного керування. Проте окреме дерево рішень може виявитися нестійким до зміни вибірки та схильним до перенавчання. Саме тому в задачах прогнозування частіше використовуються ансамблеві підходи, зокрема Random Forest та XGBoost [6, 9].

Метод Random Forest є доцільним кандидатом для задач даного класу. Він добре працює з табличними даними, стійкий до шуму, дозволяє оцінювати важливість ознак і зазвичай забезпечує кращу узагальнювальну здатність, ніж окреме дерево рішень. Для задачі прогнозування енергоспоживання в IoT-середовищі це є суттєвою перевагою, оскільки вхідні дані можуть містити нерівномірні залежності, пропуски або різні масштаби значень. Водночас Random Forest не завжди забезпечує найвищу точність у випадках, коли необхідно вловити тонкі взаємозв'язки між параметрами.

Метод XGBoost у контексті даної роботи є особливо привабливим. Його перевага полягає в тому, що він поєднує високу точність, стійкість до складних нелінійних залежностей, ефективну роботу з табличними наборами даних і можливість гнучкого налаштування. Саме такі характеристики роблять його доцільним для моделювання енергоспоживання IoT-пристроїв, де результат

залежить від сукупності багатьох чинників: параметрів середовища, режимів роботи, часових характеристик та історії функціонування [9, 18]. Додатковою перевагою є можливість оцінювання важливості ознак, що дозволяє не лише одержати прогноз, а й зрозуміти, які параметри найбільше впливають на рівень енергоспоживання.

Нейронні мережі, зокрема багатошарові перцептрони та LSTM-моделі, також широко використовуються для задач прогнозування енергії [4, 11, 32]. Їхньою перевагою є здатність моделювати складні нелінійні та часові залежності. Особливо це важливо у випадках, коли дані мають виражену часову структуру або прогноз формується на основі довгих послідовностей спостережень. Однак для бакалаврської роботи використання глибоких нейронних мереж не завжди є найбільш доцільним. По-перше, вони вимагають більшого обсягу даних і ретельнішого налаштування. По-друге, їхні результати складніше інтерпретувати. По-третє, програмна реалізація та аналіз таких моделей істотно ускладнюють роботу, зміщуючи акцент із прикладної оптимізації на складну модельну інженерію.

З урахуванням поставленої задачі доцільно обрати ансамблевий підхід на основі XGBoost як основний метод прогнозування енергоспоживання, а Random Forest або лінійну регресію – як додаткові базові моделі для порівняння. Такий вибір є виправданим з кількох причин. По-перше, XGBoost добре працює з табличними даними, які є типовими для телеметрії IoT-пристроїв. По-друге, цей метод забезпечує достатню точність без потреби у надмірно складній архітектурі моделі. По-третє, він дозволяє одержати не лише прогноз, а й аналітичну інформацію про значущість ознак. По-четверте, його відносно просто інтегрувати у програмний модуль як ядро оцінювання енергоспоживання [9, 18].

Для узагальнення доцільності різних підходів подано порівняльну характеристику методів (таблиця 2.2).

Після вибору методу прогнозування необхідно обґрунтувати алгоритм оптимізації режиму роботи пристрою. У межах даної роботи доцільно застосувати алгоритм вибору режиму на основі прогнозованого енергоспоживання. Його суть полягає в тому, що для кожного пристрою або групи пристроїв формується декілька

допустимих варіантів режиму роботи, наприклад активний, економний та адаптивний. Для кожного варіанта визначається набір параметрів, після чого модель оцінює очікуване енергоспоживання за цих умов. Режим, що забезпечує мінімальне значення цільового показника за дотримання встановленого обмеження якості, обирається як доцільний.

Таблиця 2.2 – Порівняльна характеристика методів машинного навчання для задачі прогнозування енергоспоживання

Метод	Переваги	Недоліки	Доцільність у роботі
Лінійна регресія	Простота, швидкість, інтерпретованість	Слабо описує нелінійні залежності	Доцільна як базова модель
Дерево рішень	Простота інтерпретації, робота з різними ознаками	Схильність до перенавчання	Обмежено доцільне
Random Forest	Стійкість, добра узагальнювальна здатність, важливість ознак	Менша гнучкість порівняно з бустингом	Доцільний як модель порівняння
XGBoost	Висока точність, робота з нелінійностями, придатність до табличних даних	Потребує налаштування параметрів	Доцільний як основна модель
LSTM нейронні мережі	Добре працюють із часовими рядами та складними залежностями	Складність реалізації, нижча інтерпретованість	Перспективні, але надмірні для прототипу

Такий алгоритм має низку переваг. Насамперед він є достатньо простим для програмної реалізації та експериментальної перевірки. Крім того, він добре узгоджується з архітектурою модуля, де аналітичний блок відповідає за оцінювання, а модуль прийняття рішень – за порівняння можливих режимів. Важливо й те, що цей підхід дозволяє відокремити прогнозування від оптимізації: модель оцінює наслідки, а правило вибору формує керувальне рішення. Для бакалаврської роботи така схема є більш доцільною, ніж повноцінні методи підкріплення чи складні багатоагентні моделі, оскільки зберігає баланс між інтелектуальністю рішення та його практичною реалізованістю [17, 25].

У спрощеному вигляді алгоритм оптимізації можна подати так:

1. Отримати поточні параметри середовища та стану пристрою.
2. Сформулювати набір допустимих режимів роботи.
3. Для кожного режиму розрахувати або спрогнозувати очікуване

енергоспоживання.

4. Перевірити виконання функціональних обмежень.
5. Обрати режим із мінімальним прогнозованим енергоспоживанням серед допустимих варіантів.
6. Передати рекомендацію або застосувати рішення.

У контексті даної роботи доцільно також врахувати критерії оцінювання якості обраної моделі. Для задачі прогнозування енергоспоживання найчастіше використовуються такі показники, як середньоквадратична помилка, середня абсолютна помилка та коефіцієнт детермінації. Вони дозволяють оцінити, наскільки точно модель відтворює реальні або тестові значення енергоспоживання. Поряд із цим важливо оцінювати і прикладний результат, тобто фактичне зниження енергоспоживання після застосування запропонованого алгоритму вибору режиму. Отже, ефективність рішення в роботі доцільно розглядати у двох вимірах: як якість прогнозування та як якість оптимізації.

Не менш важливим є питання інтерпретованості обраного підходу. Для систем керування смарт-будівлями бажано, щоб рішення модуля не сприймалося як суто «чорна скринька». У цьому аспекті XGBoost має перевагу над складнішими нейронними мережами, оскільки дозволяє оцінювати внесок окремих ознак у результат прогнозування. Це дає можливість пояснити, чому в певний момент було рекомендовано саме економний або адаптивний режим, наприклад через підвищену частоту передавання даних, зміну температурних умов або низьке поточне навантаження.

Отже, у межах даної роботи як основний метод машинного навчання доцільно обрати XGBoost, а як алгоритм оптимізації – процедуру вибору режиму роботи на основі порівняння прогнозованого енергоспоживання для допустимих альтернатив. Така комбінація забезпечує достатню точність, практичну реалізованість, інтерпретованість результатів і відповідність задачі інтелектуальної оптимізації режимів роботи IoT-пристроїв у смарт-будівлях.

2.4 Моделювання процесу прийняття рішень щодо керування режимами роботи пристроїв

Процес прийняття рішень щодо керування режимами роботи IoT-пристроїв є центральним елементом функціонування розроблюваного рішення, оскільки забезпечує перехід від аналізу даних до практичної дії. У межах смарт-будівлі такий перехід повинен бути достатньо обґрунтованим, адаптивним і водночас технічно простим для реалізації в програмному прототипі.

У загальному вигляді процес прийняття рішень можна розглядати як послідовність етапів, у межах яких система отримує вхідні дані, оцінює поточний стан пристрою та середовища, прогнозує наслідки можливих режимів роботи, порівнює альтернативи та обирає той варіант, що забезпечує найкраще співвідношення між енергоспоживанням і функціональною доцільністю. Отже, у цій роботі модель прийняття рішень ґрунтується не на жорстко заданому правилі, а на процедурі аналітичного вибору серед кількох допустимих режимів.

Для розв'язання поставленої задачі доцільно використати дискретну модель вибору режиму роботи. Це означає, що для кожного пристрою або типу пристроїв визначається скінченний набір можливих режимів функціонування. Наприклад, такими режимами можуть бути:

- активний режим, у якому пристрій працює з максимальною частотою вимірювань і передавання даних;
- економний режим, у якому зменшуються частота вимірювань, інтенсивність передавання або тривалість активного стану;
- адаптивний режим, у якому параметри пристрою змінюються залежно від поточного контексту.

Такий підхід є доцільним для цієї роботи, оскільки дозволяє уникнути надмірної складності без втрати логіки інтелектуального керування. Замість безперервної оптимізації великої кількості параметрів пропонується більш практичний варіант: система оцінює обмежений набір реалістичних сценаріїв роботи пристрою та обирає серед них найкращий.

У межах моделі процес прийняття рішень починається з формування

вхідного опису стану системи. Такий опис повинен охоплювати як поточні параметри середовища, так і характеристики функціонування самого IoT-пристрою. До складу вхідного вектора можуть входити значення температури, вологості, освітленості, час доби, поточний режим роботи, кількість переданих пакетів, тривалість активного стану, історія попереднього енергоспоживання, а також інші ознаки, що використовуються аналітичною моделлю.

Нехай поточний стан пристрою описується вектором

$$S_t = \{x_1, x_2, x_3, \dots, x_m\}, \quad (2.6)$$

де S_t – стан системи в момент часу t ;

$x_1, x_2, x_3, \dots, x_m$ – ознаки, що характеризують умови функціонування пристрою.

Для цього стану формується множина допустимих альтернативних режимів роботи

$$R = \{r_1, r_2, r_3, \dots, r_k\}. \quad (2.7)$$

Кожен режим r_j визначається своїм набором параметрів, наприклад частотою зчитування, інтервалом передавання даних або тривалістю перебування у сплячому стані.

Далі для кожної альтернативи виконується оцінювання очікуваного енергоспоживання. Якщо використовується навчена модель прогнозування, то для кожного режиму формується модифікований вектор ознак $S_t^{(r_j)}$, який відповідає умовам застосування цього режиму, після чого обчислюється прогноз:

$$\hat{E}_j = f(S_t^{(r_j)}) \quad (2.8)$$

де $\hat{E}_j$ – прогнозоване енергоспоживання для режиму r_j ;

f – модель машинного навчання.

Проте одного лише прогнозу енергоспоживання недостатньо для вибору рішення. Необхідно також оцінити, чи не порушує певний режим мінімальні вимоги до якості функціонування системи. Саме тому для кожного режиму перевіряється виконання обмеження

$$Q_j \geq Q_{min}, \quad (2.9)$$

де Q_j – оцінка якості функціонування для режиму r_j ;

Q_{min} – мінімально допустиме значення.

У спрощеній реалізації ця перевірка може виконуватися у вигляді набору правил. Наприклад, не допускається вибір режиму, якщо інтервал між вимірюваннями перевищує припустиме значення, якщо затримка реакції стає надто великою або якщо для певного типу пристрою економний режим не забезпечує необхідної точності моніторингу.

Таким чином, процедура вибору режиму може бути описана як пошук такого варіанта, для якого прогнозоване енергоспоживання є мінімальним серед усіх допустимих альтернатив:

$$r^* = \arg \min_{r_j \in R_{adm}} \hat{E}_j, \quad (2.10)$$

де R_{adm} – множина режимів, що задовольняють функціональні обмеження.

Запропонована модель прийняття рішень має кілька важливих переваг. Насамперед вона дозволяє чітко розмежувати етап прогнозування та етап вибору. Це означає, що аналітична модель відповідає за оцінювання наслідків, а модуль прийняття рішень – за порівняння альтернатив і застосування обмежень. Такий поділ спрощує структуру програмного модуля, підвищує прозорість його логіки та полегшує відлагодження.

Крім того, ця модель є достатньо гнучкою. За потреби кількість режимів можна змінювати, набір правил – уточнювати, а метод оцінювання якості функціонування – розширювати. Водночас загальна структура процесу залишається сталою: збір даних, формування альтернатив, прогнозування, перевірка допустимості, вибір найкращого режиму.

У контексті розроблюваного модуля доцільно подати процес прийняття рішень як послідовність функціональних кроків:

1. Отримання поточного стану пристрою та середовища.
2. Формування набору допустимих режимів роботи.
3. Генерування ознак для кожної альтернативи.
4. Прогнозування енергоспоживання для кожного режиму.
5. Перевірка функціональних обмежень.
6. Вибір режиму з мінімальним прогнозованим енергоспоживанням серед

допустимих варіантів.

7. Передавання рекомендації або керуючої дії.
8. Збереження результату та подальший аналіз ефективності.

Для наочного подання такої логіки розроблено схему (рисунок 2.3).

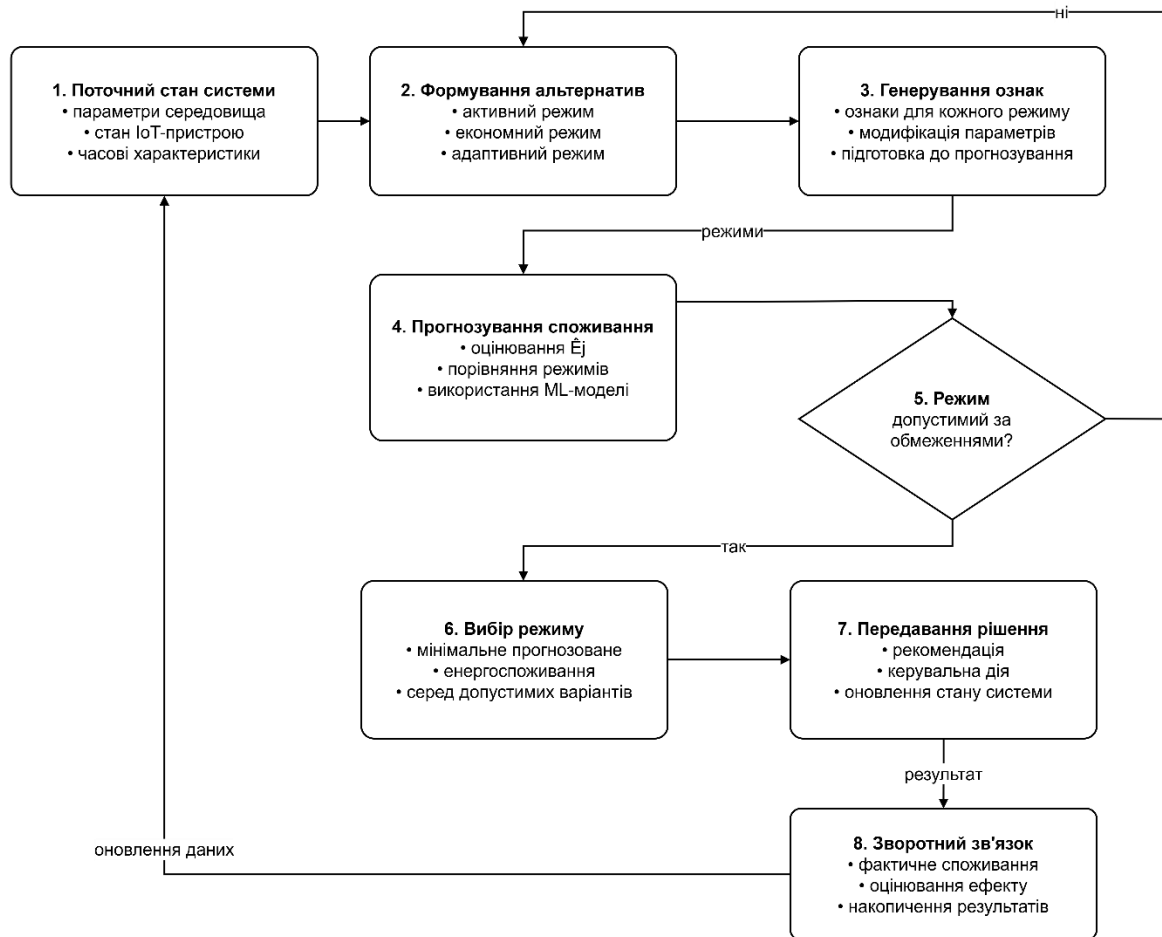


Рисунок 2.3 – Модель процесу прийняття рішень щодо керування режимами роботи IoT-пристроїв

У межах цієї моделі доцільно також визначити роль окремих критеріїв вибору. Основним критерієм є мінімізація прогнозованого енергоспоживання, проте він не є єдиним. На практиці враховуються також:

- збереження мінімального рівня якості роботи системи;
- допустимість режиму для конкретного типу пристрою;
- стабільність переходу між режимами;
- відсутність конфлікту з попереднім станом або сценарієм роботи.

Наприклад, навіть якщо економний режим забезпечує найменше прогнозоване енергоспоживання, його не слід застосовувати в умовах, коли в

приміщенні активно змінюються параметри середовища або коли пристрій виконує критично важливу функцію. У такому випадку пріоритет може бути надано адаптивному режиму, який забезпечує компроміс між економією та функціональною повнотою.

З огляду на це доцільно розглядати рішення не як суто математичний мінімум, а як практично доцільний вибір. Саме такий підхід є найбільш виправданим для систем смарт-будівель, де енергозбереження не повинно досягатися ціною втрати корисності системи.

Для кращого узагальнення логіки вибору режимів доцільно подати характеристику можливих альтернатив у таблиці 2.3.

Таблиця 2.3 – Узагальнена характеристика режимів роботи IoT-пристроїв у моделі прийняття рішень

Режим роботи	Характеристика	Переваги	Обмеження використання
Активний	Висока частота вимірювань і передавання даних	Максимальна оперативність і точність	Підвищене енергоспоживання
Економний	Зменшена інтенсивність роботи пристрою	Мінімізація витрат енергії	Може знижувати оперативність і точність
Адаптивний	Динамічна зміна параметрів залежно від умов	Баланс між якістю та економією	Потребує аналітичної підтримки

У межах програмної реалізації описану модель доцільно представити у вигляді алгоритму. Його спрощений псевдокод можна подати так:

Лістинг 2.1 – Узагальнений алгоритм вибору режиму роботи IoT-пристрою

вхід: поточний стан S_t , множина режимів R
 вихід: обраний режим r^*

```

1. best_mode := null
2. best_energy := +∞
3. для кожного режиму  $r_j$  з множини  $R$ :
4.     сформувати ознаки  $S_t^{(r_j)}$ 
5.     обчислити прогноз енергоспоживання  $\hat{E}_j$ 
6.     оцінити допустимість режиму  $Q_j$ 
7.     якщо  $Q_j \geq Q_{\min}$  і  $\hat{E}_j < \text{best\_energy}$ :
8.         best_energy :=  $\hat{E}_j$ 
9.         best_mode :=  $r_j$ 
10. повернути best_mode
    
```

Наведений алгоритм не є складним, проте саме така простота є його перевагою в межах кваліфікаційної роботи. Він дозволяє реалізувати інтелектуальне керування без побудови надмірно ускладненої системи, зберігаючи при цьому логіку аналітичного вибору. У подальшому цей підхід може бути розширений, наприклад шляхом введення вагових коефіцієнтів для кількох критеріїв, використання більш складних функцій корисності або переходу до підкріплювального навчання.

Важливим елементом моделі є також контур зворотного зв'язку. Після застосування вибраного режиму система повинна фіксувати фактичне енергоспоживання та оцінювати, наскільки результат відповідає прогнозу. Це дозволяє:

- аналізувати якість роботи моделі;
- уточнювати параметри алгоритму;
- виявляти випадки, коли рекомендація виявилася недостатньо ефективною;
- накопичувати дані для подальшого вдосконалення системи.

Саме завдяки зворотному зв'язку модель прийняття рішень набуває адаптивного характеру. У такому випадку вона працює не як одноразовий механізм вибору, а як процес, що поступово накопичує досвід і може бути вдосконалений на основі нових спостережень.

Отже, у даному підрозділі розроблено модель процесу прийняття рішень щодо керування режимами роботи IoT-пристроїв у смарт-будівлі. Її основу становить дискретний вибір серед допустимих режимів за критерієм мінімізації прогнозованого енергоспоживання з урахуванням функціональних обмежень.

3 РЕАЛІЗАЦІЯ ТА ЕКСПЕРИМЕНТАЛЬНІ ДОСЛІДЖЕННЯ ПРОГРАМНОГО МОДУЛЯ

3.1 Програмна реалізація інтелектуального модуля

Програмна реалізація інтелектуального модуля є завершальним етапом переходу від теоретичної моделі до практичного рішення, придатного для експериментальної перевірки. У межах даної роботи реалізація орієнтована не на створення повномасштабної промислової системи керування смарт-будівлею, а на розроблення програмного прототипу, який дозволяє виконати основні функції аналізу даних, прогнозування енергоспоживання та вибору доцільного режиму роботи IoT-пристрою. Такий підхід є достатнім для бакалаврської роботи, оскільки забезпечує перевірку працездатності запропонованих рішень у прикладному середовищі.

Під час програмної реалізації було враховано архітектуру модуля, описану в попередньому розділі. Відповідно до неї програмний прототип повинен охоплювати кілька основних функціональних блоків: завантаження та підготовку даних, формування ознак, навчання або використання моделі прогнозування, процедуру вибору режиму роботи пристрою та відображення отриманих результатів. Реалізація цих компонентів у межах єдиного програмного рішення дає змогу експериментально перевірити логіку інтелектуального керування режимами роботи IoT-пристроїв у смарт-будівлі.

Для створення прототипу доцільно використовувати мову Python, оскільки вона є однією з найпоширеніших у задачах аналізу даних, машинного навчання та швидкого створення дослідницьких програмних модулів. Вибір саме цього середовища зумовлений наявністю великої кількості бібліотек для оброблення табличних даних, навчання моделей, оцінювання результатів і побудови візуалізацій. Крім того, Python забезпечує достатню простоту реалізації, що є важливим для створення зрозумілого та модифікованого програмного прототипу.

Структурна схема програмної реалізації наведена на рисунку 3.1.

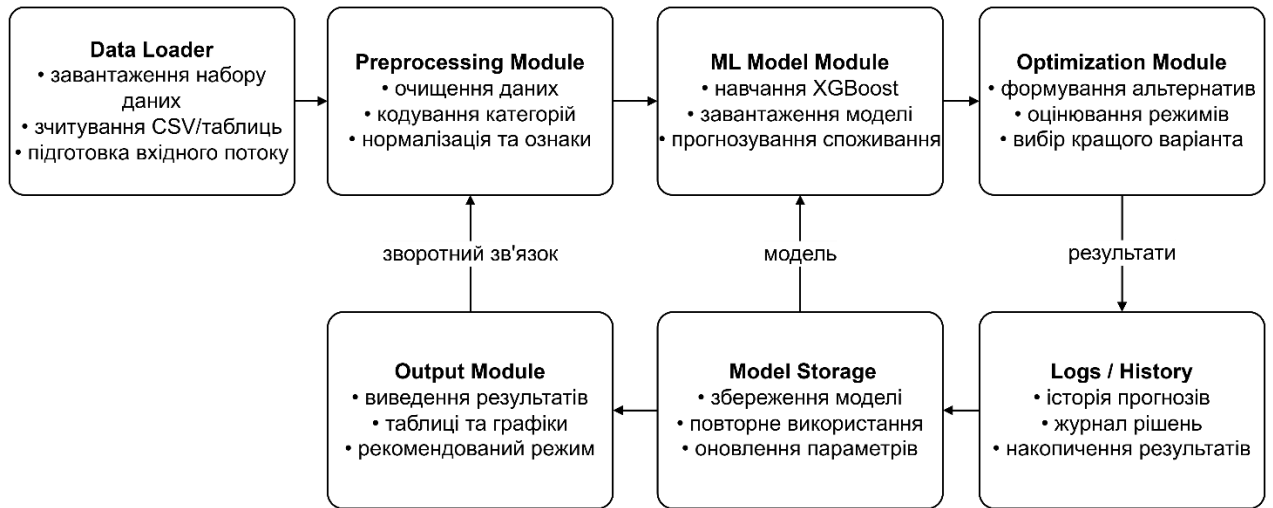


Рисунок 3.1 – Структурна схема програмної реалізації інтелектуального модуля

Першим компонентом програмної реалізації є модуль завантаження даних. Його призначення полягає у зчитуванні вхідної інформації з файлів або іншого джерела даних. У межах прототипу таким джерелом виступає підготовлений набір записів, що містить параметри середовища, характеристики роботи пристрою та значення енергоспоживання. На цьому етапі реалізується імпорт даних у табличну структуру, яка надалі використовується всіма іншими компонентами системи.

Другим компонентом є модуль попереднього оброблення. У ньому реалізуються операції очищення даних, усунення пропусків, кодування категоріальних змінних, нормалізація числових ознак та формування підсумкового вектора параметрів для подальшого моделювання. Саме цей блок забезпечує узгоджене представлення вхідної інформації. У практичному аспекті він є одним із найважливіших, оскільки якість моделі значною мірою залежить від правильності підготовки даних.

Третім блоком виступає модуль машинного навчання, у якому реалізовано навчання або використання моделі XGBoost для прогнозування енергоспоживання. У межах цього модуля виконується розподіл даних на навчальну та тестову вибірки, побудова моделі, її налаштування, а також формування прогнозів для нових записів. Якщо модель уже навчена, цей модуль використовується як аналітичне ядро, що обчислює очікуване значення енергоспоживання для заданого

режиму роботи пристрою.

Четвертим блоком є модуль оптимізації режимів роботи. Він реалізує логіку, описану в підрозділі 2.4. Для кожного поточного стану пристрою формується набір альтернативних режимів, після чого для кожного з них оцінюється прогнозоване енергоспоживання. Далі виконується перевірка допустимості режиму та вибір варіанта з мінімальним очікуваним споживанням серед допустимих альтернатив. У програмному сенсі цей блок є зв'язувальною ланкою між прогнозною моделлю та прикладним рішенням щодо керування пристроєм.

П'ятим блоком виступає модуль формування результатів, який відповідає за виведення прогнозованих значень, вибраного режиму роботи та узагальнених показників ефективності. У спрощеному варіанті результати можуть відображатися в консольному інтерфейсі, таблицях або графіках. Для експериментальної перевірки такого рівня подання цілком достатньо.

У межах програмної реалізації доцільно застосувати модульний принцип, за якого окремі функції винесені в самостійні процедури або класи. Це підвищує читабельність коду, спрощує відлагодження та дозволяє легко вносити зміни до логіки роботи системи. Наприклад, попереднє оброблення даних, навчання моделі, прогнозування та вибір режиму можуть бути реалізовані як окремі функціональні блоки з чітко визначеними вхідними та вихідними параметрами.

Загальна логіка роботи програмного модуля має такий вигляд:

- 1) зчитування вхідного набору даних;
- 2) підготовка та перетворення ознак;
- 3) навчання або завантаження прогнозної моделі;
- 4) формування альтернативних режимів роботи;
- 5) оцінювання енергоспоживання для кожного режиму;
- 6) вибір оптимального або доцільного варіанта;
- 7) подання результату користувачеві.

Для наочного подання реалізації доцільно включити короткі фрагменти програмного коду, а повні лістинги винести до додатка. Такий підхід є методично правильним, оскільки в основному тексті роботи достатньо подати лише ключові частини програмної реалізації.

Нижче наведено приклад структури завантаження та підготовки даних.

Лістинг 3.1 – Фрагмент програмної реалізації завантаження та підготовки даних

```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder

data = pd.read_csv("iot_energy_data.csv")

categorical_columns = ["device_type", "usage_pattern", "environmental_condition"]
encoders = {}

for column in categorical_columns:
    encoder = LabelEncoder()
    data[column] = encoder.fit_transform(data[column])
    encoders[column] = encoder

X = data.drop("energy_consumption", axis=1)
y = data["energy_consumption"]

X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.2, random_state=42
)
```

Наведений фрагмент показує, що програмна реалізація починається з імпорту даних та перетворення категоріальних ознак у числовий формат. Далі формується навчальна та тестова вибірки. Така підготовка є стандартною для задач регресійного моделювання, але в даному випадку вона має прикладне значення, оскільки створює основу для побудови моделі оцінювання енергоспоживання.

Приклад реалізації прогнозовної моделі можна подати так.

Лістинг 3.2 – Фрагмент програмної реалізації навчання моделі XGBoost

```
from xgboost import XGBRegressor
from sklearn.metrics import mean_squared_error, r2_score

model = XGBRegressor(
    n_estimators=100,
    max_depth=6,
    learning_rate=0.1,
    random_state=42
)

model.fit(X_train, y_train)

predictions = model.predict(X_test)

mse = mean_squared_error(y_test, predictions)
r2 = r2_score(y_test, predictions)

print("MSE:", mse)
print("R2:", r2)
```

У цьому фрагменті реалізовано базову процедуру навчання моделі XGBoost та оцінювання її якості на тестовій вибірці. Саме цей модуль надалі

використовується для визначення прогнозованого енергоспоживання при зміні режиму роботи пристрою.

Окрему увагу слід приділити процедурі вибору режиму роботи. Її узагальнений варіант може бути реалізований так.

Лістинг 3.3 – Фрагмент програмної реалізації вибору режиму роботи пристрою

```
def choose_best_mode(base_state, possible_modes, model):
    best_mode = None
    best_energy = float("inf")

    for mode in possible_modes:
        state = base_state.copy()
        state["mode"] = mode["mode"]
        state["sampling_rate"] = mode["sampling_rate"]
        state["transmission_interval"] = mode["transmission_interval"]

        predicted_energy = model.predict(pd.DataFrame([state]))[0]

        if mode["allowed"] and predicted_energy < best_energy:
            best_energy = predicted_energy
            best_mode = mode

    return best_mode, best_energy
```

Наведений фрагмент відображає основну ідею інтелектуального модуля: для кількох альтернативних режимів формується модифікований опис стану, для кожного з них розраховується прогнозоване енергоспоживання, після чого обирається найкращий допустимий варіант.

У межах реалізації також важливо забезпечити можливість повторного використання моделі. Для цього доцільно передбачити збереження навченої моделі у файл і подальше її завантаження без повторного навчання. Такий підхід спрощує тестування та дозволяє використовувати модуль у режимі окремого прогнозування.

Ще одним важливим аспектом є оброблення помилок та перевірка коректності вхідних даних. У реальних умовах IoT-середовище може генерувати пропущені, неповні або аномальні значення. Тому програмна реалізація повинна містити базові механізми перевірки вхідних параметрів, щоб уникнути некоректної роботи моделі.

З точки зору інтерфейсу, розроблюваний прототип може бути реалізований у спрощеному вигляді. Для цієї роботи достатньо текстового інтерфейсу або запуску через скрипт, який приймає вхідні дані, виконує прогнозування та виводить

рекомендацію щодо режиму роботи пристрою. За потреби такий прототип може бути розширений графічним інтерфейсом або веб-панеллю.

На рисунку 3.2 подано головне вікно розробленого інтелектуального модуля, у якому відображено загальну інформацію про набір даних, поточний стан IoT-пристрою, рекомендований режим роботи та очікуване зниження енергоспоживання.

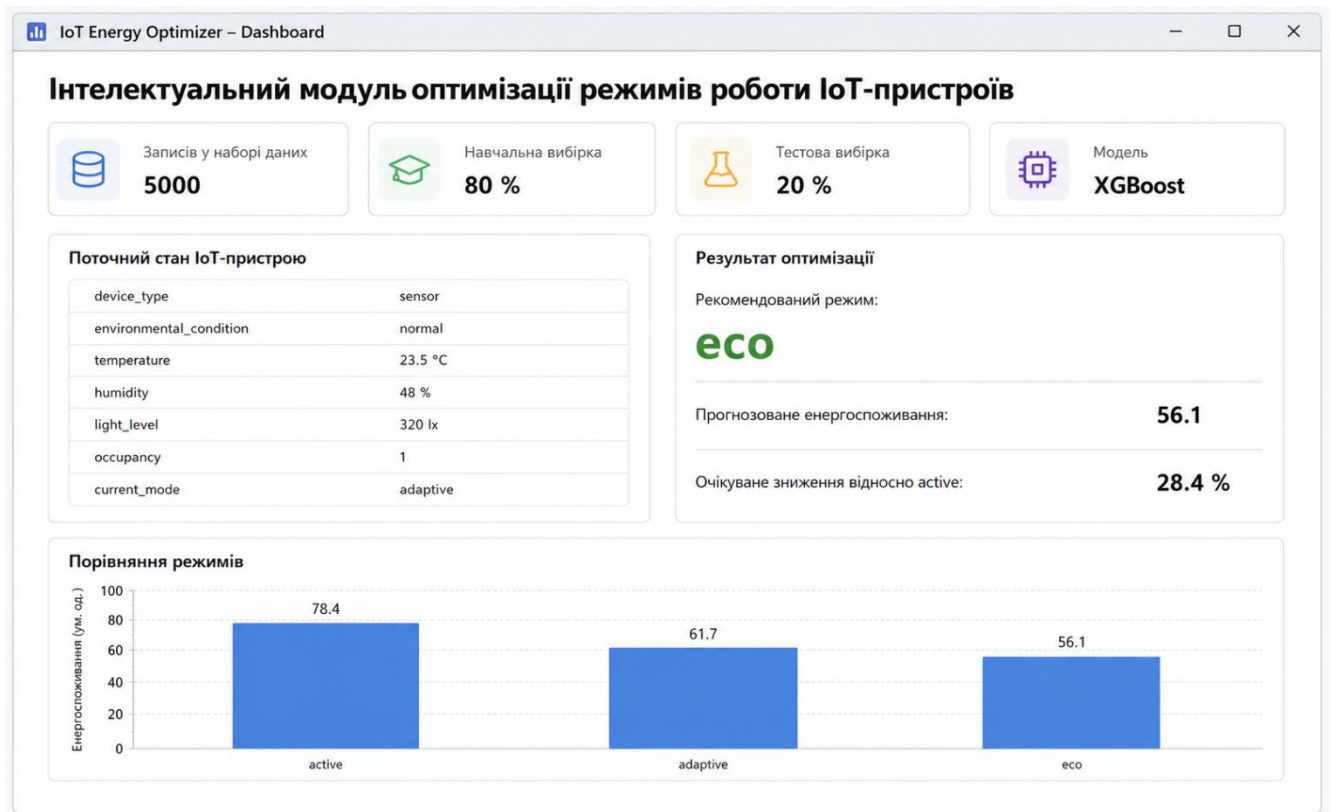


Рисунок 3.2 – Головне вікно інтелектуального модуля оптимізації режимів роботи IoT-пристроїв

Повні лістинги програмного забезпечення подано в додатку А.

Таким чином, програмна реалізація інтелектуального модуля охоплює всі ключові етапи, необхідні для практичного тестування запропонованого підходу: завантаження даних, їх підготовку, побудову прогнозної моделі, оцінювання альтернативних режимів роботи та формування рекомендації щодо оптимізації енергоспоживання. Розроблений прототип є достатньо простим для експериментальної перевірки, але водночас відображає основну логіку інтелектуального керування IoT-пристроями в смарт-будівлі.

3.2 Організація експерименту та опис тестових даних

Після програмної реалізації інтелектуального модуля виконано експериментальну перевірку його працездатності. Метою експерименту було оцінювання здатності розробленого модуля прогнозувати енергоспоживання IoT-пристроїв та формувати рекомендацію щодо вибору більш енергоефективного режиму роботи. У межах дослідження перевірено повний цикл функціонування програмного прототипу: завантаження даних, попереднє оброблення, формування ознак, навчання моделі XGBoost, оцінювання альтернативних режимів роботи та вибір рекомендованого режиму.

Для експериментального дослідження використано тестовий набір даних `iot_energy_data.csv`, сформований програмно з метою моделювання типових режимів роботи IoT-пристроїв у смарт-будівлі. Такий підхід дав змогу отримати контрольований набір спостережень для перевірки працездатності інтелектуального модуля без підключення до реальної інфраструктури смарт-будівлі. Дані мають модельний характер і використовуються для імітаційного експерименту, тому вони не подаються як реальні телеметричні дані з фізичних пристроїв.

На рисунку 3.3 наведено фрагмент завантаженого тестового набору даних у програмному модулі.

Кожен запис у наборі даних описує один стан IoT-пристрою з урахуванням параметрів середовища, режиму функціонування та відповідного значення енергоспоживання. До набору включено різні типи пристроїв, зокрема сенсори, контролери, інтелектуальні лічильники та розумні розетки. Для кожного пристрою задано умови середовища, часові параметри, режим роботи, частоту зчитування даних, інтервал передавання та тривалість активного стану.

У межах експерименту використано три базові режими роботи IoT-пристроїв: `active`, `eco` та `adaptive`. Режим `active` моделює інтенсивну роботу пристрою з високою частотою вимірювань, коротким інтервалом передавання даних і тривалим активним станом. Режим `eco` передбачає зниження інтенсивності роботи пристрою, збільшення інтервалу передавання та скорочення активного стану.

Режим adaptive займає проміжне положення між активним та економним режимами й імітує контекстно-залежне налаштування параметрів.

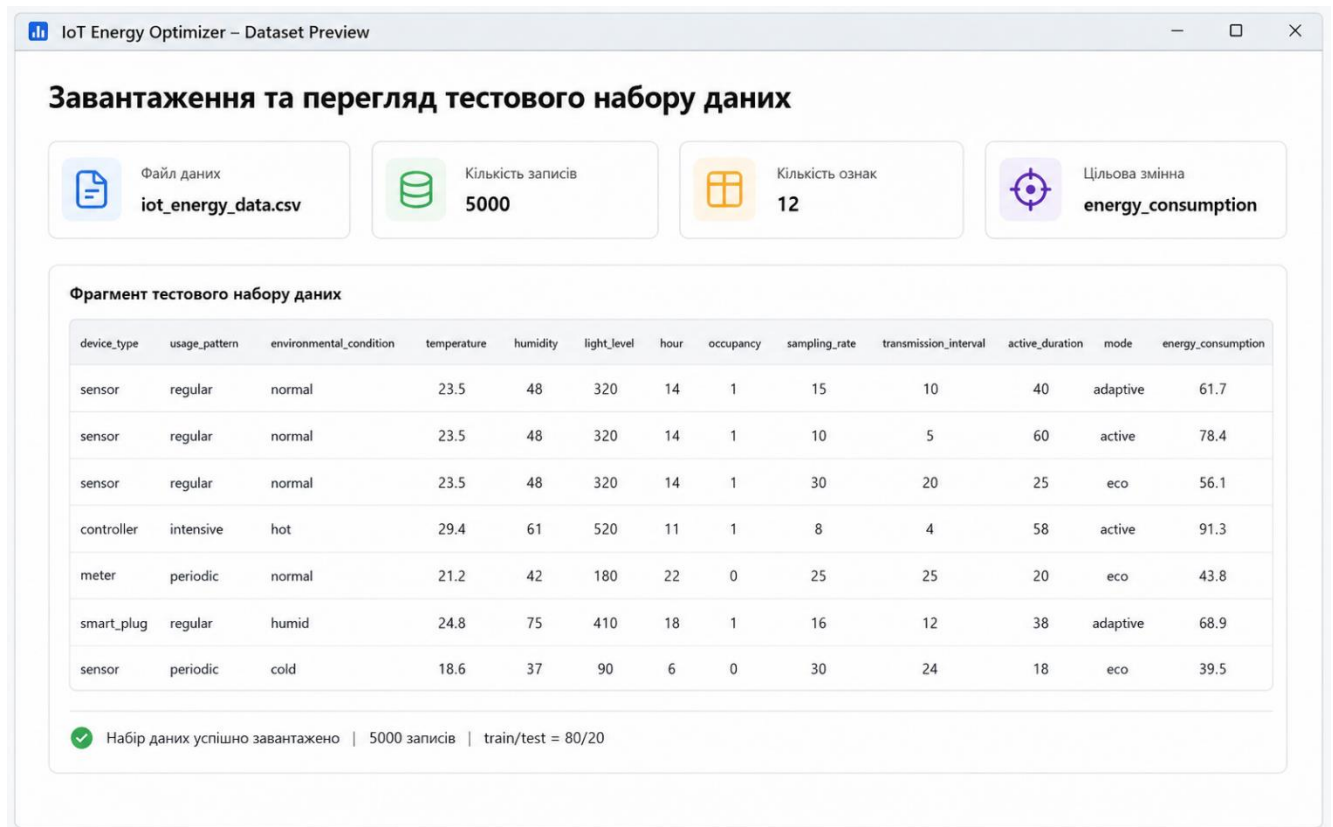


Рисунок 3.3 – Завантаження та перегляд тестового набору даних у програмному модулі

Цільовою змінною в експерименті є `energy_consumption`, тобто значення енергоспоживання пристрою за певний проміжок часу. Саме цей показник прогнозується моделлю машинного навчання та використовується надалі для порівняння альтернативних режимів роботи.

Із таблиці 3.1 видно, що тестовий набір даних містить як числові, так і категоріальні ознаки. Це зумовило необхідність попереднього оброблення даних перед навчанням моделі. Категоріальні змінні було перетворено у числове подання, числові параметри підготовлено до моделювання, а цільову змінну `energy_consumption` відокремлено від вхідного набору ознак.

Таблиця 3.1 – Характеристика основних ознак тестового набору даних

Ознака	Тип даних	Зміст
device_type	категоріальний	тип IoT-пристрою: sensor, controller, meter, smart_plug
usage_pattern	категоріальний	характер використання пристрою: regular, intensive, periodic
environmental_condition	категоріальний	умови середовища: normal, hot, cold, humid
temperature	числовий	температура середовища
humidity	числовий	відносна вологість
light_level	числовий	рівень освітленості
hour	числовий	час доби
occupancy	числовий бінарний	наявність користувача в зоні дії
sampling_rate	числовий	частота зчитування даних
transmission_interval	числовий	інтервал передавання даних
active_duration	числовий	тривалість активного стану
mode	категоріальний	режим роботи пристрою: active, eco, adaptive
energy_consumption	числовий	цільовий енергоспоживання показник

Загальну характеристику тестового набору даних подано в таблиці 3.2.

Таблиця 3.2 – Загальна характеристика тестового набору даних

Параметр	Значення
Назва файлу	iot_energy_data.csv
Характер даних	синтетично сформовані модельні дані
Кількість записів	5000
Кількість вхідних ознак	12
Тип задачі	регресія
Цільова змінна	energy_consumption
Частка навчальної вибірки	80 %
Частка тестової вибірки	20 %
Основна модель	XGBoost
Альтернативні режими роботи	active, eco, adaptive

Експериментальне дослідження виконано у кілька етапів. На першому етапі згенеровано та завантажено тестовий набір даних. На другому етапі виконано перевірку структури даних, попереднє оброблення, кодування категоріальних

ознак і формування вхідного ознакового простору. На третьому етапі дані поділено на навчальну та тестову вибірки у співвідношенні 80 % до 20 %. На четвертому етапі навчено модель XGBoost для прогнозування енергоспоживання. На п'ятому етапі здійснено оцінювання якості моделі та перевірено алгоритм вибору режиму роботи пристрою.

Навчання моделі XGBoost виконано в програмному середовищі Python. У процесі навчання було використано сформовані ознаки, що описують параметри середовища, тип пристрою, режим його роботи та режимні характеристики функціонування. Результати виконання програми в консолі наведено на рисунку 3.4.

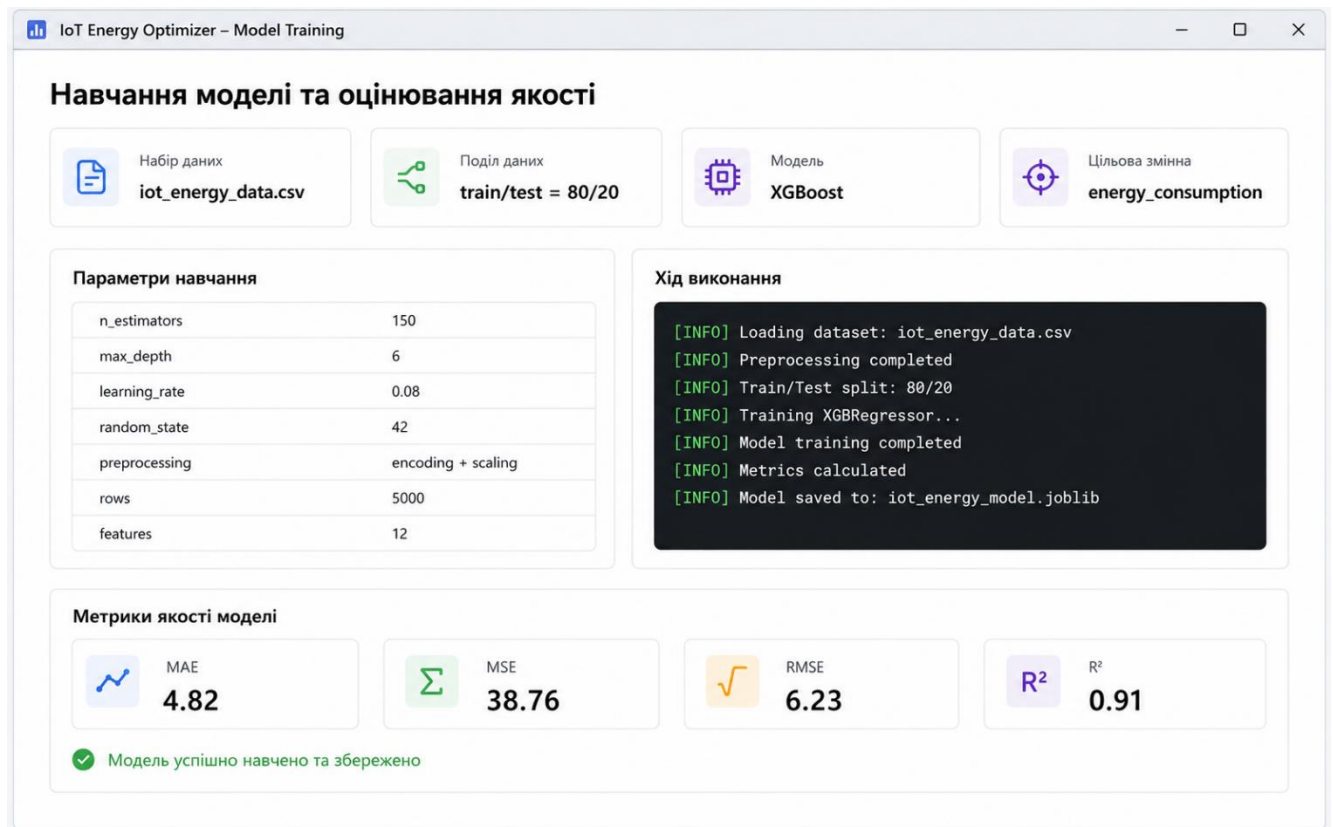


Рисунок 3.4 – Результати навчання моделі XGBoost у програмному модулі

На рисунку 3.4 показано, що програмний модуль виконує завантаження набору даних, попереднє оброблення, поділ на навчальну і тестову вибірки, навчання моделі та виведення основних метрик якості. Для оцінювання моделі використано такі показники:

– MAE – середня абсолютна похибка;

- MSE – середньоквадратична похибка;
- RMSE – корінь із середньоквадратичної похибки;
- R^2 – коефіцієнт детермінації.

За результатами тестового запуску отримано такі значення метрик: MAE = 4.82, MSE = 38.76, RMSE = 6.23, R^2 = 0.91. Отримані значення свідчать про достатню узгодженість прогнозованих і тестових значень енергоспоживання в межах сформованого модельного набору даних. Водночас ці результати слід розглядати як підтвердження працездатності програмного прототипу, а не як виробничу оцінку для реальної смарт-будівлі.

Окрім перевірки якості прогнозування, проведено сценарне тестування алгоритму вибору режиму роботи IoT-пристрою. Для цього задано поточний стан пристрою та сформовано три альтернативні режими: active, adaptive і eco. Для кожного режиму модель спрогнозувала очікуване енергоспоживання, після чого модуль перевіряв допустимість режимів і вибрав варіант із найменшим прогнозованим споживанням.

У межах сценарної перевірки розглянуто такі типові ситуації:

- стандартне навантаження в робочий час;
- знижене навантаження у вечірній період;
- підвищене навантаження за активної присутності користувачів;
- зміна параметрів середовища, зокрема температури, вологості та рівня освітленості.

Такий підхід дозволив перевірити, чи змінюється рекомендація модуля залежно від контексту функціонування пристрою. Основна увага приділялася не лише точності прогнозування, а й тому, чи здатний модуль вибирати режим, який зменшує очікуване енергоспоживання без порушення функціональних обмежень.

У ході експерименту було зафіксовано параметри навчання моделі, склад навчальної та тестової вибірок, перелік альтернативних режимів, правила перевірки допустимості режиму та формат збереження результатів. Це забезпечило відтворюваність експерименту та можливість повторного запуску програмного модуля з тими самими параметрами.

Окремо слід зазначити, що експеримент виконано в межах програмного

прототипу, а не на реальній промисловій інфраструктурі smart-будівлі. Тому результати дослідження мають демонстраційний і дослідницький характер. Вони підтверджують, що запропонована логіка поєднання прогнозної моделі та алгоритму вибору режиму є працездатною і може бути використана як основа для подальшої інтеграції з реальними IoT-пристроями.

Таким чином, у підрозділі описано організацію експериментального дослідження, охарактеризовано синтетично сформований тестовий набір даних, визначено склад ознак, описано процес навчання моделі XGBoost та порядок сценарної перевірки алгоритму оптимізації режимів роботи IoT-пристроїв.

3.3 Аналіз результатів роботи модуля та оцінювання ефективності оптимізації

Після організації експерименту та навчання моделі XGBoost виконано перевірку роботи інтелектуального модуля в режимі оцінювання альтернативних режимів функціонування IoT-пристрою. Основною метою цього етапу було визначення того, чи здатний розроблений модуль не лише прогнозувати енергоспоживання, а й формувати практично корисну рекомендацію щодо вибору енергоефективного режиму роботи.

Аналіз результатів проведено у двох напрямках. Перший напрям стосувався оцінювання альтернативних режимів роботи пристрою за прогнозованим рівнем енергоспоживання. Другий напрям передбачав аналіз якості прогнозування та визначення ознак, які найбільше впливають на результат роботи моделі. Такий підхід дозволив оцінити як прикладну ефективність алгоритму оптимізації, так і обґрунтованість результатів, отриманих за допомогою моделі машинного навчання.

Для сценарної перевірки було задано поточний стан IoT-пристрою. У прикладі використано пристрій типу sensor, який працює в режимі adaptive. Для нього задано такі параметри середовища: температура становить 23.5 °C, вологість – 48 %, рівень освітленості – 320 lx, значення ознаки присутності користувача – 1. На основі цих даних програмний модуль сформував три альтернативні режими роботи: active, adaptive та eco.

На рисунку 3.5 подано екран програмного модуля, на якому відображено результат оцінювання альтернативних режимів роботи IoT-пристрою.

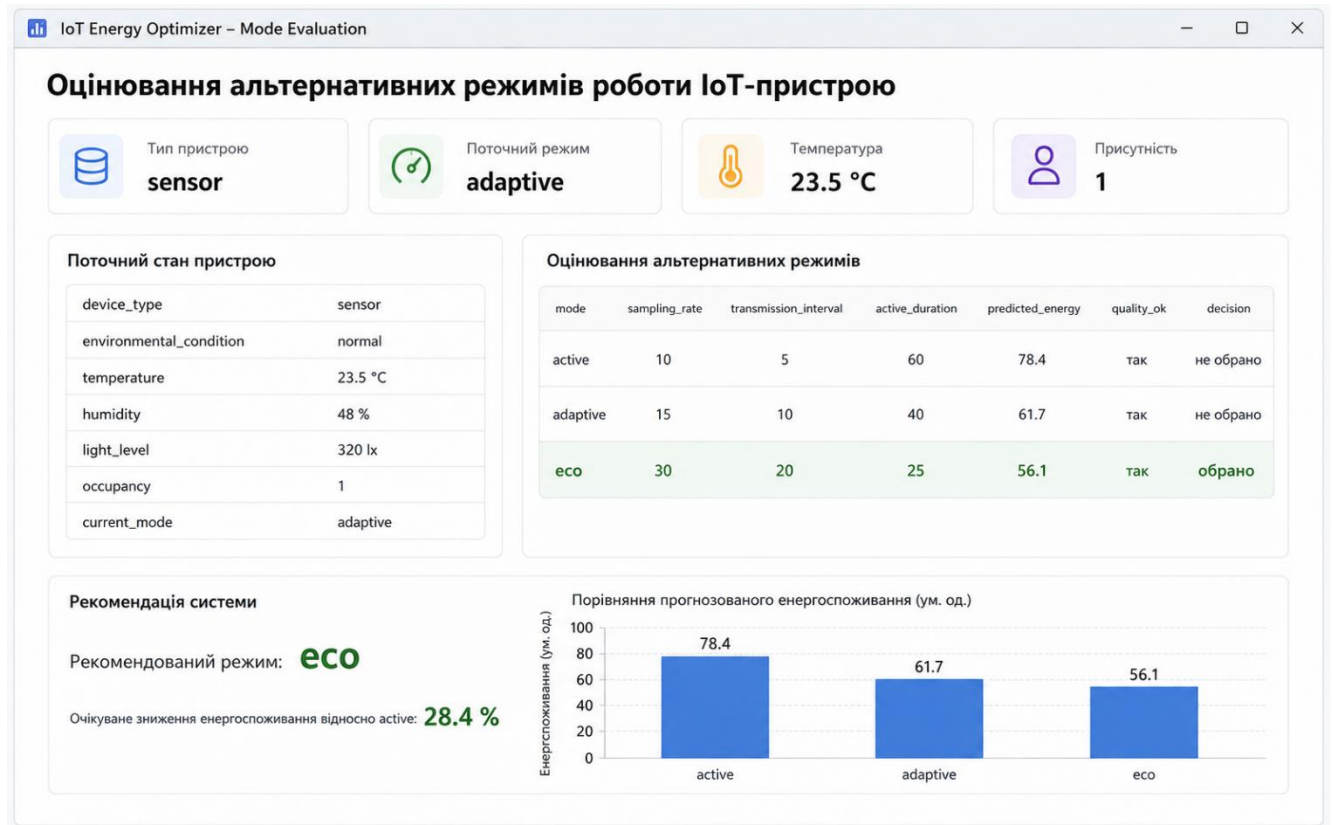


Рисунок 3.5 – Оцінювання альтернативних режимів роботи IoT-пристрою

З рисунка 3.5 видно, що модуль виконує порівняння трьох режимів роботи за набором параметрів: частота зчитування даних, інтервал передавання, тривалість активного стану, прогнозоване енергоспоживання, допустимість режиму та підсумкове рішення. Для режиму active встановлено частоту зчитування 10, інтервал передавання 5 та тривалість активного стану 60. Прогнозоване енергоспоживання для цього режиму становить 78.4 умовних одиниці. Для режиму adaptive відповідні параметри становлять 15, 10 і 40, а прогнозоване енергоспоживання – 61.7 умовних одиниці. Для режиму eco задано частоту зчитування 30, інтервал передавання 20 та тривалість активного стану 25, що дало прогнозоване енергоспоживання 56.1 умовних одиниці.

Усі три режими в межах розглянутого сценарію задовольняють функціональні обмеження, тому вибір виконувався за критерієм мінімального прогнозованого енергоспоживання. Найменше значення отримано для режиму eco,

тому саме цей режим було обрано як рекомендований. Система сформувала підсумкову рекомендацію щодо переходу до режиму eco, а також обчислила очікуване зниження енергоспоживання відносно режиму active.

Результати оцінювання альтернативних режимів узагальнено в таблиці 3.3.

Таблиця 3.3 – Результати оцінювання альтернативних режимів роботи IoT-пристрою

Режим роботи	Частота зчитування	Інтервал передавання	Тривалість активного стану	Прогнозоване енергоспоживання	Допустимість режиму	Рішення
active	10	5	60	78.4	так	не обрано
adaptive	15	10	40	61.7	так	не обрано
eco	30	20	25	56.1	так	обрано

За результатами таблиці 3.3 встановлено, що режим eco забезпечує найменше прогнозоване енергоспоживання серед допустимих альтернатив. У наведеному сценарії застосування запропонованого алгоритму дозволило зменшити прогнозоване енергоспоживання на 28.4 % порівняно з режимом active. Водночас вибраний режим залишився допустимим за встановленими обмеженнями, що підтверджує коректність логіки прийняття рішення.

Важливо, що розроблений модуль не виконує механічного вибору режиму з найменшим споживанням без перевірки додаткових умов. У процедурі оцінювання враховується допустимість режиму, тому економний режим може бути відхилений, якщо він не забезпечує потрібний рівень функціональності. У розглянутому прикладі режим eco є допустимим, тому його вибір є обґрунтованим. У випадках, коли економний режим не задовольняє вимоги до частоти вимірювань або оперативності передавання даних, система може обрати режим adaptive як компромісний варіант між енергоефективністю та якістю функціонування.

Другим етапом аналізу було оцінювання якості прогнозування та визначення важливості ознак, які впливають на результат моделі. Для цього використано метрики MAE, RMSE та коефіцієнт детермінації R^2 . Крім того, виконано побудову графіка співвідношення фактичних і прогнозованих значень енергоспоживання, а також діаграми важливості ознак.

На рисунку 3.6 подано екран аналізу результатів прогнозування та важливості ознак у програмному модулі.

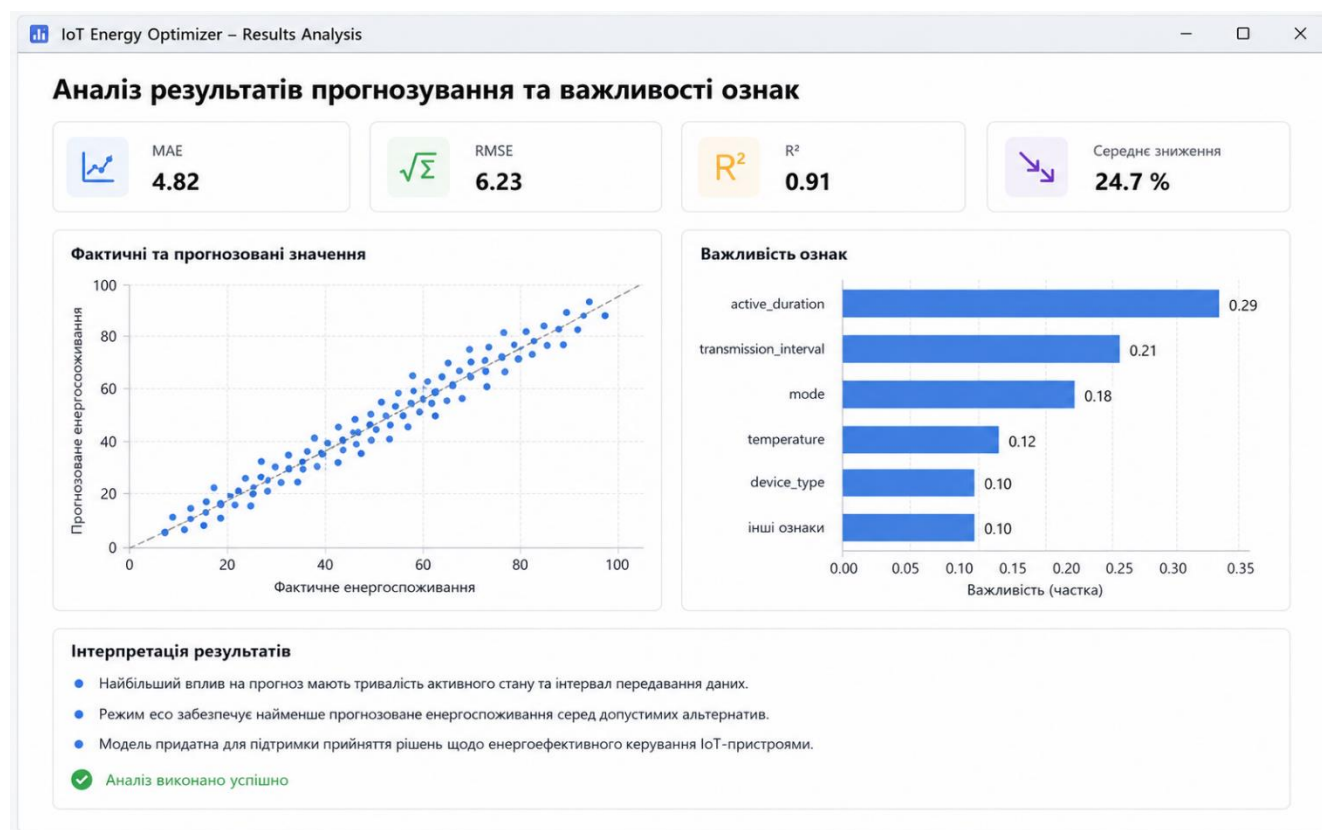


Рисунок 3.6 – Аналіз результатів прогнозування та важливості ознак

На рисунку 3.6 відображено основні метрики якості моделі: $MAE = 4.82$, $RMSE = 6.23$, $R^2 = 0.91$. Значення MAE показує середню абсолютну похибку прогнозування, RMSE характеризує середній рівень відхилення з урахуванням більших помилок, а коефіцієнт R^2 свідчить про те, що модель пояснює значну частину варіації цільового показника в межах сформованого тестового набору даних. Отримані значення підтверджують достатню точність моделі для використання в експериментальному програмному прототипі.

Графік фактичних і прогнозованих значень демонструє, що більшість точок розміщується поблизу діагональної лінії. Це означає, що прогнозовані значення енергоспоживання загалом узгоджуються з тестовими значеннями. Наявність окремих відхилень є очікуваною для регресійної задачі, оскільки енергоспоживання формується під впливом кількох параметрів одночасно.

Окремо проаналізовано важливість ознак, що використовувалися моделлю

XGBoost для прогнозування енергоспоживання. Результати такого аналізу подано в таблиці 3.4.

Таблиця 3.4 – Ранжування ознак за важливістю для моделі прогнозування

Ознака	Відносна важливість
active_duration	0.29
transmission_interval	0.21
mode	0.18
temperature	0.12
device_type	0.10
Інші ознаки	0.10

З таблиці 3.4 видно, що найбільший вплив на результат прогнозування має ознака `active_duration`, тобто тривалість активного стану IoT-пристрою. Це є логічним, оскільки чим довше пристрій перебуває в активному режимі, тим більшим є його енергоспоживання. Друге місце займає ознака `transmission_interval`, яка відображає інтенсивність передавання даних. Менший інтервал передавання зазвичай означає частішу мережеву активність, що також збільшує витрати енергії. Важливою є й ознака `mode`, оскільки саме режим роботи визначає загальну стратегію функціонування пристрою.

Температура та тип пристрою мають менший, але також помітний вплив на результат прогнозування. Це свідчить про те, що модель враховує не лише режимні параметри, а й контекст функціонування пристрою. Такий результат є важливим для задачі смарт-будівель, де енергоспоживання залежить від поєднання технічних характеристик пристрою, умов середовища та сценарію використання.

Для оцінювання загального ефекту оптимізації в модулі також визначено середнє зниження енергоспоживання, яке в межах тестових сценаріїв становило 24.7 %. Це значення характеризує очікуваний аналітичний ефект від застосування алгоритму вибору режиму роботи. Його не слід трактувати як гарантовану економію для реальної смарт-будівлі, оскільки експеримент виконано на синтетично сформованому модельному наборі даних. Водночас отриманий показник підтверджує, що запропонована логіка оптимізації здатна зменшувати прогнозоване енергоспоживання в контрольованих умовах експерименту.

У ході аналізу встановлено, що розроблений модуль виконує не лише прогнозування, а й прикладну підтримку прийняття рішень. Він порівнює альтернативні режими, перевіряє їх допустимість, вибирає режим із найменшим прогнозованим енергоспоживанням і подає результат у зрозумілому для користувача вигляді. Це відрізняє модуль від звичайної прогнозовної моделі, яка лише обчислює значення цільового показника, але не формує керувальної рекомендації.

Разом із тим результати експерименту мають певні обмеження. Передусім використано синтетично сформований тестовий набір даних, тому результати відображають працездатність запропонованої логіки в модельному середовищі. Крім того, у роботі розглянуто три базові режими роботи пристрою, тоді як у реальній смарт-будівлі кількість режимів і параметрів може бути більшою. Також у програмному прототипі реалізовано рекомендаційний механізм вибору режиму, а не повноцінне автоматизоване керування фізичними IoT-пристроями.

Незважаючи на ці обмеження, отримані результати підтверджують доцільність запропонованого підходу. Розроблений модуль показав здатність оцінювати енергоспоживання, визначати найвпливовіші параметри, порівнювати альтернативні режими та формувати рекомендацію щодо вибору енергоефективного варіанта роботи пристрою. Це створює основу для подальшого розвитку системи, зокрема для використання реальних телеметричних даних, розширення набору режимів, інтеграції з edge- або cloud-середовищами та підключення до фізичних IoT-пристроїв.

Таким чином, у підрозділі виконано аналіз результатів роботи інтелектуального модуля та оцінено ефективність запропонованого алгоритму оптимізації. Встановлено, що в межах тестового сценарію модуль обрав режим есо, який забезпечив найменше прогнозоване енергоспоживання серед допустимих альтернатив. Оцінювання якості моделі та аналіз важливості ознак підтвердили придатність реалізованого підходу для підтримки прийняття рішень щодо енергоефективного керування IoT-пристроями в смарт-будівлях.

ВИСНОВКИ

У процесі виконання роботи досягнуто поставленої мети, яка полягала у розробленні інтелектуального модуля оптимізації режимів роботи IoT-пристроїв у смарт-будівлях на основі методів аналізу даних і машинного навчання. Для цього було послідовно виконано всі поставлені завдання та отримано наступні висновки:

1. Проаналізовано предметну область оптимізації енергоспоживання IoT-пристроїв у смарт-будівлях. Встановлено, що енергоспоживання в таких системах визначається не лише технічними характеристиками обладнання, а й параметрами середовища, частотою вимірювань, інтенсивністю передавання даних, тривалістю активного стану та сценаріями використання пристроїв. Обґрунтовано, що задача зменшення енергоспоживання повинна розглядатися на рівні режимів роботи окремих IoT-вузлів, а не лише на рівні будівлі в цілому.

2. Досліджено існуючі підходи та методи зменшення енергоспоживання в IoT-системах. Показано, що сучасні рішення охоплюють апаратно-мережеві, архітектурні та алгоритмічні підходи, а найбільш перспективними для смарт-будівель є методи, що поєднують IoT-моніторинг, Green IoT, edge AI та машинне навчання. Водночас встановлено, що багато наявних рішень орієнтовані переважно на прогнозування споживання енергії або мають надто складну інфраструктуру впровадження, що підтвердило доцільність створення окремого прикладного програмного модуля.

3. Сформульовано задачу дослідження та визначено основні вимоги до інтелектуального модуля. Показано, що розроблюване рішення повинно бути адаптивним, енергоорієнтованим, масштабованим, достатньо простим для реалізації та придатним до інтерпретації результатів. Виконано формалізацію задачі як задачі вибору такого режиму роботи IoT-пристрою, який мінімізує енергоспоживання за умови дотримання допустимого рівня якості функціонування системи.

4. Розроблено архітектуру інтелектуального модуля. Запропонована структура охоплює модуль збору даних, модуль попереднього оброблення, аналітичний модуль машинного навчання, модуль прийняття рішень, підсистему

збереження результатів і контур зворотного зв'язку. Така архітектура забезпечує логічну послідовність оброблення інформації – від надходження телеметричних даних до формування рекомендації щодо оптимального режиму роботи пристрою.

5. Обґрунтовано вибір методів машинного навчання та алгоритму прийняття рішень. Встановлено, що для задачі прогнозування енергоспоживання найбільш доцільним є використання XGBoost як основної моделі, оскільки цей метод поєднує точність, здатність працювати з табличними даними, стійкість до нелінійних залежностей і відносну простоту практичного використання. Запропоновано алгоритм оптимізації, у межах якого для кожного поточного стану пристрою формуються альтернативні режими роботи, виконується прогнозування енергоспоживання для кожного з них, перевіряються функціональні обмеження та обирається найбільш доцільний варіант.

6. Реалізовано програмний прототип інтелектуального модуля. Програмна реалізація виконана мовою Python та охоплює основні функціональні блоки: завантаження даних, попереднє оброблення, навчання або використання моделі, вибір режиму роботи пристрою та формування результатів.

7 Проведено експериментальне дослідження роботи модуля та оцінено ефективність запропонованого підходу. Для цього організовано обчислювальний експеримент на основі тестового набору даних, що містить параметри середовища, характеристики функціонування пристроїв і цільовий показник енергоспоживання. Оцінювання виконано за допомогою метрик MAE, MSE, RMSE та R^2 , а також за прикладним критерієм – здатністю системи вибирати більш енергоефективний режим роботи серед допустимих альтернатив. Результати дослідження показали, що розроблений модуль є придатним до використання як засіб підтримки прийняття рішень щодо енергоефективного керування IoT-пристроями.

8. Узагальнення результатів дозволяє стверджувати, що запропонований підхід має практичну цінність. Його використання дає змогу не лише оцінювати очікуване енергоспоживання IoT-пристроїв, а й формувати рекомендації щодо вибору режиму роботи, орієнтовані на зменшення енергетичних витрат.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Afzal S., Ziapour B. M., Shokri A., Shakibi H., Sobhani B. Building energy consumption prediction and optimization using different neural network-assisted models; comparison of different networks and optimization algorithms. *Engineering Applications of Artificial Intelligence*. 2024. Vol. 127. Art. 107356.
2. Alanne K., Sierla S. An overview of machine learning applications for smart buildings. *Sustainable Cities and Society*. 2022. Vol. 76. Art. 103445.
3. Almalki F. A., Alsamhi S. H., Sahal R., Hassan J., Hawbani A., Rajput N. S., Saif A., Morgan J., Breslin J. G. Green IoT for Eco-Friendly and Sustainable Smart Cities: Future Directions and Opportunities. *Mobile Networks and Applications*. 2023. Vol. 28, no. 1. P. 178–202.
4. Anan M., Kanaan K., Benhaddou D., Nasser N., Qolomany B., Talei H., Sawalmeh A. Occupant-Aware Energy Consumption Prediction in Smart Buildings Using a LSTM Model and Time Series Data. *Energies*. 2024. Vol. 17, no. 24. Art. 6451.
5. Bourdeau M., Zhai X. Q., Nefzaoui E., Guo X., Chatellier P. Modeling and forecasting building energy consumption: A review of data-driven techniques. *Sustainable Cities and Society*. 2019. Vol. 48. Art. 101533.
6. Breiman L. Random Forests. *Machine Learning*. 2001. Vol. 45, no. 1. P. 5–32.
7. Bykovyy P., Sachenko A., Kochan V., Osolinskyi O., Kochan R. Reducing Power Consumption of Measurement and Control Modules Fed with Autonomous Power Supply. *CEUR Workshop Proceedings*. 2018. Vol. 2104. P. 544–554.
8. Candanedo L. M., Feldheim V., Deramaix D. Data driven prediction models of energy use of appliances in a low-energy house. *Energy and Buildings*. 2017. Vol. 140. P. 81–97.
9. Chen T., Guestrin C. XGBoost: A Scalable Tree Boosting System. In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. New York : ACM, 2016. P. 785–794.

10. Djenouri D., Laidi R., Djenouri Y., Balasingham I. Machine Learning for Smart Building Applications: Review and Taxonomy. *ACM Computing Surveys*. 2019. Vol. 52, no. 2. Art. 24. P. 1–36.
11. Durand D., Aguilar J., R-Moreno M. D. An Analysis of the Energy Consumption Forecasting Problem in Smart Buildings Using LSTM. *Sustainability*. 2022. Vol. 14, no. 20. Art. 13358.
12. Fang M., Misnan M. S., Halim N. H. F. A. A Systematic Literature Review on Energy Efficiency Analysis of Building Energy Management. *Buildings*. 2024. Vol. 14, no. 10. Art. 3136.
13. Fraga-Lamas P., Lopes S. I., Fernández-Caramés T. M. Green IoT and Edge AI as Key Technological Enablers for a Sustainable Digital Transition towards a Smart Circular Economy: An Industry 5.0 Use Case. *Sensors*. 2021. Vol. 21, no. 17. Art. 5745.
14. Ghasemkhani B., Yilmaz R., Birant D., Kut R. A. Machine Learning Models for the Prediction of Energy Consumption Based on Cooling and Heating Loads in Internet-of-Things-Based Smart Buildings. *Symmetry*. 2022. Vol. 14, no. 8. Art. 1553.
15. Hochreiter S., Schmidhuber J. Long Short-Term Memory. *Neural Computation*. 1997. Vol. 9, no. 8. P. 1735–1780.
16. Huotari M., Malhi A., Främling K. Machine Learning Applications for Smart Building Energy Utilization: A Survey. *Archives of Computational Methods in Engineering*. 2024. Vol. 31, no. 5. P. 2537–2556.
17. Kök İ., Ergun Y., Uğur N. Explainable AI-powered edge computing solution for smart building energy management in green IoT. *IET Conference Proceedings*. 2023. P. 150–157. (Low-Cost Digital Solutions for Industrial Automation, LoDiSA 2023).
18. Mariano-Hernández D., Hernández-Callejo L., García F. S., Duque-Perez O., Zorita-Lamadrid A. L. A Review of Energy Consumption Forecasting in Smart Buildings: Methods, Input Variables, Forecasting Horizon and Metrics. *Applied Sciences*. 2020. Vol. 10, no. 23. Art. 8323.
19. Natarajan Y., K. R. S. P., Wadhwa G., Choi Y., Chen Z., Lee D.-E., Mi Y. Enhancing Building Energy Efficiency with IoT-Driven Hybrid Deep Learning Models for Accurate Energy Consumption Prediction. *Sustainability*. 2024. Vol. 16, no. 5. Art. 1925.

20. Osolinskyi O., Molga A., Kochan V., Sachenko A. Method of Ensuring the Noise Immunity at Measurement of Single-Board Microcontroller Average Energy Within IoT Environment. In: *2020 IEEE 40th International Conference on Electronics and Nanotechnology (ELNANO)*. 2020. P. 807–810.
21. Osolinskyi O., Sachenko A., Kochan V., Kolodiichuk L. Measurement and Optimization Methods of Energy Consumption for Microcontroller Systems Within IoT. In: *2022 12th International Conference on Dependable Systems, Services and Technologies (DESSERT)*. 2022. P. 1–7.
22. Ozadowicz A. A New Concept of Active Demand Side Management for Energy Efficient Prosumer Microgrids with Smart Building Technologies. *Energies*. 2017. Vol. 10, no. 11. Art. 1771.
23. Pan H., Yin Z., Jiang X. High-Dimensional Energy Consumption Anomaly Detection: A Deep Learning-Based Method for Detecting Anomalies. *Energies*. 2022. Vol. 15, no. 17. Art. 6139.
24. Poyyamozhi M., Murugesan B., Rajamanickam N., Shorfuzzaman M., Aboelmagd Y. IoT–A Promising Solution to Energy Management in Smart Buildings: A Systematic Review, Applications, Barriers, and Future Scope. *Buildings*. 2024. Vol. 14, no. 11. Art. 3446.
25. Raval M., Bhardwaj S., Aravelli A., Dofe J., Gohel H. A. Smart energy optimization for massive IoT using artificial intelligence. *Internet of Things*. 2021. Vol. 13. Art. 100354.
26. Shah S. F. A., Iqbal M., Aziz Z., Rana T. A., Khalid A., Cheah Y.-N., Arif M. The Role of Machine Learning and the Internet of Things in Smart Buildings for Energy Efficiency. *Applied Sciences*. 2022. Vol. 12, no. 15. Art. 7882.
27. Shapi M. K. M., Ramli N. A., Awalin L. J. Energy consumption prediction by using machine learning for smart building: Case study in Malaysia. *Developments in the Built Environment*. 2021. Vol. 5. Art. 100037.
28. Shi Q., Liu C., Xiao C. Machine learning in building energy management: A critical review and future directions. *Frontiers of Engineering Management*. 2022. Vol. 9, no. 2. P. 239–256.

29. Sievers J., Blank T. A Systematic Literature Review on Data-Driven Residential and Industrial Energy Management Systems. *Energies*. 2023. Vol. 16, no. 4. Art. 1688.
30. Tabassum H., Ben Ghorbel M., ElSawy H., Guibène W., Guruacharya S. Green Internet of Things (IoT): Enabling Technologies, Architectures, Performance, and Design Issues. *Wireless Communications and Mobile Computing*. 2018. Vol. 2018. Art. 3747562.
31. Teixeira N., Barreto R., Gomes L., Faria P., Vale Z. A Trustworthy Building Energy Management System to Enable Direct IoT Devices' Participation in Demand Response Programs. *Electronics*. 2022. Vol. 11, no. 6. Art. 897.
32. Yin Q., Han C., Li A., Liu X., Liu Y. A Review of Research on Building Energy Consumption Prediction Models Based on Artificial Neural Networks. *Sustainability*. 2024. Vol. 16, no. 17. Art. 7805.
33. Zhang X., Manogaran G., Muthu B. IoT-enabled integrated system for green energy into smart cities. *Sustainable Energy Technologies and Assessments*. 2021. Vol. 46. Art. 101208.
34. Гайдукевич В. О., Дорошенко А. Ю. Застосування моделей машинного навчання для прогнозування енергоспоживання в системах розумного дому. *Проблеми програмування*. 2025. № 3. С. 29–39.
35. Дяченко Д. О., Кайда В. В., Левченко А. О., Міхаль О. П. Методи функціонування пристроїв IoT з використанням машинного навчання. *Системи управління, навігації та зв'язку*. 2022. № 4. С. 78–81.
36. Костоглод А. Ю. Архітектура інтелектуального модуля оптимізації режимів роботи IoT-пристроїв. Інформаційне суспільство: технологічні, економічні та технічні аспекти становлення (випуск 110): матеріали Міжнародної наукової інтернет-конференції, (м. Тернопіль, Україна, м. Ополь, Польща, 7-8 травня 2026 р.) / редкол. : О. Патряк та ін. ГО "Наукова спільнота", WSZIA w Opolu. Тернопіль : ФО-П Шпак В.Б. 2026. С. 38-40.
37. Комар М.П., Саченко А.О., Васильків Н.М та ін. Методичні рекомендації до виконання кваліфікаційної роботи з освітньо-професійної

програми «Комп'ютерні науки» спеціальності 122 «Комп'ютерні науки» за першим (бакалаврським) рівнем вищої освіти. Тернопіль: ЗУНУ, 2024. 52 с.

38. Островерхов В. М., Біловус Л. І., Возьний К. З. та ін. Загальні методичні рекомендації з підготовки, оформлення, захисту та оцінювання кваліфікаційних робіт здобувачів вищої освіти першого (бакалаврського) і другого (магістерського) рівнів. Тернопіль: ЗУНУ, 2024. 83 с.

Додаток А

Код програмного забезпечення

Основний програмний модуль інтелектуальної оптимізації режимів роботи IoT-пристроїв

```
import json
import joblib
from dataclasses import dataclass
from pathlib import Path
from typing import Dict, List, Tuple, Any

import numpy as np
import pandas as pd
from sklearn.compose import ColumnTransformer
from sklearn.metrics import mean_absolute_error, mean_squared_error, r2_score
from sklearn.model_selection import train_test_split
from sklearn.pipeline import Pipeline
from sklearn.preprocessing import OneHotEncoder, StandardScaler
from xgboost import XGBRegressor
```

```
@dataclass
```

```
class ModeConfig:
```

```
    """Опис альтернативного режиму роботи IoT-пристрою."""
```

```
    name: str
```

```
    sampling_rate: int
```

```
    transmission_interval: int
```

```
    active_duration: int
```

```
    allowed: bool = True
```

```
class IoTEnergyOptimizer:
```

```
    """
```

```
    Інтелектуальний модуль прогнозування та оптимізації режимів роботи IoT-пристроїв.
```

```
    """
```

```
    def __init__(self) -> None:
```

```
        self.model: Pipeline | None = None
```

```
        self.feature_columns: List[str] = []
```

```
        self.target_column: str = "energy_consumption"
```

```
    def load_data(self, file_path: str) -> pd.DataFrame:
```

```
        """
```

```
        Завантаження даних із CSV-файлу.
```

```
        """
```

```
        path = Path(file_path)
```

```
        if not path.exists():
```

```
            raise FileNotFoundError(f"Файл не знайдено: {file_path}")
```

```
        data = pd.read_csv(path)
```

```
        if data.empty:
```

```
            raise ValueError("Завантажений файл не містить даних.")
```

```
        return data
```

```
    def preprocess_data(self, data: pd.DataFrame) -> Tuple[pd.DataFrame, pd.Series]:
```

```
        """
```

```
        Підготовка даних до навчання моделі.
```

```

"""
if self.target_column not in data.columns:
    raise ValueError(
        f"У наборі даних відсутня цільова змінна '{self.target_column}'."
    )

data = data.copy()

# Усунення пропусків
for column in data.columns:
    if data[column].dtype == "object":
        data[column] = data[column].fillna("unknown")
    else:
        data[column] = data[column].fillna(data[column].median())

# Формування ознак і цільової змінної
X = data.drop(columns=[self.target_column])
y = data[self.target_column]

self.feature_columns = list(X.columns)
return X, y

def build_pipeline(self, X: pd.DataFrame) -> Pipeline:
    """
    Побудова конвеєра оброблення даних і моделі.
    """
    categorical_features =
X.select_dtypes(include=["object"]).columns.tolist()
    numerical_features = X.select_dtypes(exclude=["object"]).columns.tolist()

    preprocessor = ColumnTransformer(
        transformers=[
            (
                "num",
                StandardScaler(),
                numerical_features,
            ),
            (
                "cat",
                OneHotEncoder(handle_unknown="ignore"),
                categorical_features,
            ),
        ],
    )

    model = XGBRegressor(
        n_estimators=150,
        max_depth=6,
        learning_rate=0.08,
        subsample=0.9,
        colsample_bytree=0.9,
        objective="reg:squarederror",
        random_state=42,
    )

    pipeline = Pipeline(
        steps=[
            ("preprocessor", preprocessor),
            ("regressor", model),
        ]
    )
    return pipeline

def train_model(
    self, X: pd.DataFrame, y: pd.Series
) -> Dict[str, float]:

```

```

"""
Навчання моделі та обчислення показників якості.
"""
X_train, X_test, y_train, y_test = train_test_split(
    X,
    Y,
    test_size=0.2,
    random_state=42,
)

self.model = self.build_pipeline(X)
self.model.fit(X_train, y_train)

predictions = self.model.predict(X_test)

metrics = {
    "MAE": float(mean_absolute_error(y_test, predictions)),
    "MSE": float(mean_squared_error(y_test, predictions)),
    "RMSE": float(np.sqrt(mean_squared_error(y_test, predictions))),
    "R2": float(r2_score(y_test, predictions)),
}
return metrics

def save_model(self, model_path: str) -> None:
    """
    Збереження навченої моделі.
    """
    if self.model is None:
        raise ValueError("Модель не навчено. Немає що зберігати.")

    payload = {
        "model": self.model,
        "feature_columns": self.feature_columns,
        "target_column": self.target_column,
    }
    joblib.dump(payload, model_path)

def load_model(self, model_path: str) -> None:
    """
    Завантаження раніше збереженої моделі.
    """
    path = Path(model_path)
    if not path.exists():
        raise FileNotFoundError(f"Файл моделі не знайдено: {model_path}")

    payload = joblib.load(model_path)
    self.model = payload["model"]
    self.feature_columns = payload["feature_columns"]
    self.target_column = payload["target_column"]

def predict_energy(self, device_state: Dict[str, Any]) -> float:
    """
    Прогнозування енергоспоживання для одного стану пристрою.
    """
    if self.model is None:
        raise ValueError("Модель не завантажено або не навчено.")

    row = pd.DataFrame([device_state])

    # Гарантування наявності всіх ознак
    for column in self.feature_columns:
        if column not in row.columns:
            row[column] = 0

    row = row[self.feature_columns]
    prediction = self.model.predict(row)[0]

```

```

        return float(prediction)

def choose_best_mode(
    self,
    base_state: Dict[str, Any],
    possible_modes: List[ModeConfig],
    max_sampling_rate: int = 60,
) -> Tuple[ModeConfig | None, float, List[Dict[str, Any]]]:
    """
    Вибір найкращого режиму роботи пристрою.
    """
    if self.model is None:
        raise ValueError("Модель не завантажено або не навчено.")

    best_mode = None
    best_energy = float("inf")
    evaluation_log: List[Dict[str, Any]] = []

    for mode in possible_modes:
        state = base_state.copy()
        state["mode"] = mode.name
        state["sampling_rate"] = mode.sampling_rate
        state["transmission_interval"] = mode.transmission_interval
        state["active_duration"] = mode.active_duration

        # Перевірка базових обмежень
        quality_ok = mode.allowed and mode.sampling_rate <= max_sampling_rate

        predicted_energy = self.predict_energy(state)

        evaluation_log.append(
            {
                "mode": mode.name,
                "sampling_rate": mode.sampling_rate,
                "transmission_interval": mode.transmission_interval,
                "active_duration": mode.active_duration,
                "predicted_energy": predicted_energy,
                "quality_ok": quality_ok,
            }
        )

        if quality_ok and predicted_energy < best_energy:
            best_energy = predicted_energy
            best_mode = mode

    return best_mode, best_energy, evaluation_log

def export_results(
    self,
    results: List[Dict[str, Any]],
    output_path: str,
) -> None:
    """
    Збереження результатів оцінювання у CSV-файл.
    """
    df = pd.DataFrame(results)
    df.to_csv(output_path, index=False, encoding="utf-8-sig")

def build_default_modes() -> List[ModeConfig]:
    """
    Формування стандартного набору режимів роботи.
    """
    return [
        ModeConfig(
            name="active",

```

```

        sampling_rate=10,
        transmission_interval=5,
        active_duration=60,
        allowed=True,
    ),
    ModeConfig(
        name="eco",
        sampling_rate=30,
        transmission_interval=20,
        active_duration=25,
        allowed=True,
    ),
    ModeConfig(
        name="adaptive",
        sampling_rate=15,
        transmission_interval=10,
        active_duration=40,
        allowed=True,
    ),
]

def main() -> None:
    data_path = "iot_energy_data.csv"
    model_path = "iot_energy_model.joblib"
    results_path = "mode_evaluation_results.csv"

    optimizer = IoTEnergyOptimizer()

    # Завантаження та підготовка даних
    data = optimizer.load_data(data_path)
    X, y = optimizer.preprocess_data(data)

    # Навчання моделі
    metrics = optimizer.train_model(X, y)
    optimizer.save_model(model_path)

    print("Показники якості моделі:")
    for key, value in metrics.items():
        print(f"{key}: {value:.4f}")

    # Приклад поточного стану пристрою
    current_state = {
        "device_type": "sensor",
        "usage_pattern": "regular",
        "environmental_condition": "normal",
        "temperature": 23.5,
        "humidity": 48,
        "light_level": 320,
        "hour": 14,
        "occupancy": 1,
        "sampling_rate": 15,
        "transmission_interval": 10,
        "active_duration": 40,
        "mode": "adaptive",
    }

    modes = build_default_modes()
    best_mode, best_energy, log = optimizer.choose_best_mode(
        base_state=current_state,
        possible_modes=modes,
    )

    optimizer.export_results(log, results_path)

    print("\nРезультати оцінювання режимів:")

```

```

for item in log:
    print(item)

if best_mode is not None:
    print("\nОбраний режим:")
    print(f"Назва режиму: {best_mode.name}")
    print(f"Очікуване енергоспоживання: {best_energy:.4f}")
else:
    print("\nНе вдалося знайти допустимий режим.")

if __name__ == "__main__":
    main()

```

Програма формування тестового набору даних

```

import random
import pandas as pd

def generate_iot_dataset(num_records: int = 5000) -> pd.DataFrame:
    """
    Генерування тестового набору даних для IoT-пристроїв.
    """
    random.seed(42)

    device_types = ["sensor", "controller", "meter", "smart_plug"]
    usage_patterns = ["regular", "intensive", "periodic"]
    environmental_conditions = ["normal", "hot", "cold", "humid"]
    modes = ["active", "eco", "adaptive"]

    rows = []

    for _ in range(num_records):
        device_type = random.choice(device_types)
        usage_pattern = random.choice(usage_patterns)
        env_condition = random.choice(environmental_conditions)
        mode = random.choice(modes)

        temperature = round(random.uniform(16, 32), 2)
        humidity = random.randint(25, 85)
        light_level = random.randint(50, 900)
        hour = random.randint(0, 23)
        occupancy = random.randint(0, 1)

        if mode == "active":
            sampling_rate = random.randint(5, 12)
            transmission_interval = random.randint(3, 8)
            active_duration = random.randint(45, 60)
            base_energy = 70
        elif mode == "eco":
            sampling_rate = random.randint(20, 35)
            transmission_interval = random.randint(15, 30)
            active_duration = random.randint(15, 30)
            base_energy = 35
        else:
            sampling_rate = random.randint(10, 18)
            transmission_interval = random.randint(8, 15)
            active_duration = random.randint(30, 45)
            base_energy = 50

        # Спрощена модель генерації енергоспоживання
        energy = (
            base_energy
            + 0.7 * temperature
            + 0.15 * humidity

```

```

        + 0.02 * light_level
        + 1.5 * occupancy
        + 0.9 * active_duration
        - 0.4 * transmission_interval
        + random.uniform(-8, 8)
    )

    rows.append(
        {
            "device_type": device_type,
            "usage_pattern": usage_pattern,
            "environmental_condition": env_condition,
            "temperature": temperature,
            "humidity": humidity,
            "light_level": light_level,
            "hour": hour,
            "occupancy": occupancy,
            "sampling_rate": sampling_rate,
            "transmission_interval": transmission_interval,
            "active_duration": active_duration,
            "mode": mode,
            "energy_consumption": round(energy, 2),
        }
    )

    return pd.DataFrame(rows)

if __name__ == "__main__":
    dataset = generate_iot_dataset(5000)
    dataset.to_csv("iot_energy_data.csv", index=False, encoding="utf-8-sig")
    print("Тестовий набір даних збережено у файл iot_energy_data.csv")

```

Модуль візуалізації результатів експерименту

```

import matplotlib.pyplot as plt
import pandas as pd

def plot_mode_comparison(results_file: str) -> None:
    """
    Побудова стовпчикової діаграми порівняння режимів.
    """
    data = pd.read_csv(results_file)

    plt.figure(figsize=(8, 5))
    plt.bar(data["mode"], data["predicted_energy"])
    plt.xlabel("Режим роботи")
    plt.ylabel("Прогнозоване енергоспоживання")
    plt.title("Порівняння прогнозованого енергоспоживання для різних режимів")
    plt.tight_layout()
    plt.show()

def plot_actual_vs_predicted(actual: list[float], predicted: list[float]) -> None:
    """
    Побудова графіка порівняння фактичних і прогнозованих значень.
    """
    plt.figure(figsize=(7, 5))
    plt.scatter(actual, predicted)
    plt.xlabel("Фактичні значення")
    plt.ylabel("Прогнозовані значення")
    plt.title("Порівняння фактичного та прогнозованого енергоспоживання")
    plt.tight_layout()
    plt.show()

```

```

if __name__ == "__main__":
    plot_mode_comparison("mode_evaluation_results.csv")
Допоміжний модуль збереження параметрів і результатів
import json
from pathlib import Path
from typing import Any

def save_json(data: dict[str, Any], file_path: str) -> None:
    """
    Збереження словника у JSON-файл.
    """
    path = Path(file_path)
    path.write_text(
        json.dumps(data, ensure_ascii=False, indent=4),
        encoding="utf-8",
    )

def load_json(file_path: str) -> dict[str, Any]:
    """
    Завантаження JSON-файлу.
    """
    path = Path(file_path)
    if not path.exists():
        raise FileNotFoundError(f"Файл не знайдено: {file_path}")

    return json.loads(path.read_text(encoding="utf-8"))

if __name__ == "__main__":
    example_result = {
        "best_mode": "eco",
        "predicted_energy": 54.27,
        "quality_ok": True,
    }

    save_json(example_result, "best_mode_result.json")
    print(load_json("best_mode_result.json"))

```

Приклад конфігураційного файлу режимів роботи

```

{
    "modes": [
        {
            "name": "active",
            "sampling_rate": 10,
            "transmission_interval": 5,
            "active_duration": 60,
            "allowed": true
        },
        {
            "name": "eco",
            "sampling_rate": 30,
            "transmission_interval": 20,
            "active_duration": 25,
            "allowed": true
        },
        {
            "name": "adaptive",
            "sampling_rate": 15,
            "transmission_interval": 10,
            "active_duration": 40,
            "allowed": true
        }
    ]
}

```


www.konferenciaonline.org.ua

**Міжнародна наукова
інтернет-конференція**

**Інформаційне суспільство:
технологічні, економічні
та технічні аспекти становлення**

Випуск 110

ISSN 2522-932X

Google Scholar



AKADEMIA NAUK STOSOWANYCH
WYŻSZA SZKOŁA ZARZĄDZANIA I ADMINISTRACJI
W OPOLE

7-8 травня 2026 р.

м. Тернопіль, Україна – м. Ополе, Польща
2026

УДК 001 (063)

Інформаційне суспільство: технологічні, економічні та технічні аспекти становлення (випуск 110): матеріали Міжнародної наукової інтернет-конференції, (м. Тернопіль, Україна, м. Ополе, Польща, 7-8 травня 2026 р.) / редкол.: О. Патряк та ін. ГО "Наукова спільнота", WSZIA w Opolu. Тернопіль: ФО-П Шпак В.Б. 2026. 195 с. – ISSN 2522-932X

Збірник доповідей підготовлено за матеріалами Міжнародної наукової інтернет-конференції (випуск 110) 7-8 травня 2026 р. на сайті www.konferenciaonline.org.ua

Оргкомітет ГО Наукова спільнота:

Патряк Олександра Тарасівна, кандидат економічних наук, ЗУНУ;

Шевченко (Огінська) Анастасія Юріївна, кандидат економічних наук, директор ТОВ «Школа майбутнього»;

Назарчук Оксана Михайлівна, доктор філософії (Ph.D.), ННІ «Юридичний інститут КНЕУ імені Вадима Гетьмана»;

Гомотюк Оксана Євгенівна, доктор історичних наук, професор, ЗУНУ;

Біловус Леся Іванівна, доктор історичних наук, кандидат філологічних наук, професор, ЗУНУ;

Рибуха Лілія Зіновіївна, доктор педагогічних наук, кандидат психологічних наук, професор, ЗУНУ;

Недошитко Ірина Романівна, кандидат історичних наук, доцент, ЗУНУ;

Стефанишин Олена Василівна, кандидат історичних наук, доцент, ЗУНУ;

Яблонська Наталія Мирославівна, кандидат філологічних наук, старший викладач, ЗУНУ;

Рудакевич Оксана Мирославівна, кандидат філософських наук, ЗУНУ;

Русенко Святослав Ярославович, викладач ВСІП ФКЕПТ ЗУНУ.

Тексти матеріалів конференції подаються в авторській редакції. Відповідальність за точність, достовірність і зміст поданих матеріалів несуть автори. Всі роботи ліцензуються відповідно до Creative Commons Attribution 4.0 International License.

Автори зберігають авторське право, а також надають збірнику право першого опублікування оригінальних наукових статей на умовах ліцензії Creative Commons Attribution 4.0 International License, що дозволяє іншим розповсюджувати роботу з визнанням авторства твору та першої публікації в цьому збірнику.

Наша адреса: Оргкомітет МНІК "Конференція онлайн"

а/с 797, м. Тернопіль 46005

тел. моб. 068 366 0 525

e-mail: inetkonf@ukr.net

URL Інтернет-конференції: <http://www.konferenciaonline.org.ua/>

ISSN 2522-932X

© ГО "Наукова спільнота" 2026

© Автори статей 2026



Віпшовський Юрій Андрійович, Грицюк Юрій Іванович ПРОСТОРОВА ЛОКАЛІЗАЦІЯ АНОМАЛІЙ НА МЕТАЛЕВИХ ПОВЕРХНЯХ ЗА ДОПОМОГОЮ АНАЛІЗУ ОДНОВИМІРНИХ ПРОЕКЦІЙ ІНТЕНСИВНОСТІ.....	24
Гуцул Олександр Станіславович, Танасюк Юлія Володимирівна ІНТЕЛЕКТУАЛЬНИЙ МОБІЛЬНИЙ ЗАСТОСУНОК ДЛЯ ЦИФРОВОЇ АВТОМАТИЗАЦІЇ ТРЕНАЖЕРНИХ ЗАЛПВ.....	27
Деревягин Ярослав Вікторович АПРОКСИМАЦІЯ КРИВИХ ДЛЯ АВТОМАТИЗАЦІЇ ГРАФОАНАЛІТИЧНОГО РОЗРАХУНКУ.....	32
Комар Мирослав Петрович, Горбаль Юлія-Марія Мар'янівна ПРОЄКТУВАННЯ ВЕБ-ОРІЄНТОВАНОЇ ІНФОРМАЦІЙНОЇ СИСТЕМИ КЕРУВАННЯ КОНТЕНТОМ МІЖНАРОДНОГО ОСВІТНЬОГО ПРОЄКТУ.....	35
Костоглод Анастасія Юріївна АРХІТЕКТУРА ІНТЕЛЕКТУАЛЬНОГО МОДУЛЯ ОПТИМІЗАЦІЇ РЕЖИМІВ РОБОТИ ІОТ-ПРИСТРОЇВ.....	38
Красовський Андрій Олександрович ПОРІВНЯЛЬНИЙ АНАЛІЗ МОДЕЛЕЙ МАШИННОГО НАВЧАННЯ ДЛЯ ВИЯВЛЕННЯ НАРАТИВІВ.....	41
Майків Ігор Мирославович, Стецюк Арсен Ігорович АВТОМАТИЗАЦІЯ ЗРОШЕННЯ НА БАЗІ ДАНИХ ІОТ-СЕНСОРІВ ТА МОДЕЛЕЙ МАШИННОГО НАВЧАННЯ.....	46
Майків Ігор Мирославович, Цьомик Олег Ігорович МОНІТОРИНГ СТАНУ СІЛЬСЬКОГОСПОДАРСЬКИХ КУЛЬТУР У ТЕПЛИЦІ НА ОСНОВІ ГЛИБОКОЇ НЕЙРОННОЇ МЕРЕЖІ.....	50
Мельник Назар Борисович, Стрижеус Матвій Юрійович ВИЯВЛЕННЯ ФЕЙКОВИХ НОВИН ТА ФОРМУВАННЯ РЕКОМЕНДАЦІЙ ДОСТОВІРНОГО КОНТЕНТУ НА ОСНОВІ ГЛИБОКИХ НЕЙРОННИХ МЕРЕЖ.....	54

АРХІТЕКТУРА ІНТЕЛЕКТУАЛЬНОГО МОДУЛЯ ОПТИМІЗАЦІЇ РЕЖИМІВ РОБОТИ ІoT-ПРИСТРОЇВ

Інтернет-адреса публікації на сайті:

<http://www.konferenciaonline.org.ua/ua/article/id-2585/>

Енергоспоживання в смарт-будівлях визначається не лише характеристиками окремих пристроїв, а й режимами їх роботи, частотою обміну даними, сценаріями активації, параметрами середовища та поведінкою користувачів. Через це традиційні підходи, що ґрунтуються на фіксованих правилах або ручному налаштуванні, часто виявляються недостатньо гнучкими. У таких умовах доцільним є використання інтелектуальних методів, здатних аналізувати дані функціонування ІoT-пристроїв, виявляти закономірності енергоспоживання та формувати рішення щодо вибору більш економних режимів роботи [1-3].

З позиції функціонального призначення архітектура інтелектуального модуля повинна охоплювати повний цикл роботи з даними: від їх надходження до формування рішення та оцінювання результату. У зв'язку з цим доцільно виділити кілька основних компонентів: модуль збору даних, модуль попереднього оброблення, аналітичний модуль, модуль прийняття рішень, підсистему збереження даних і модуль взаємодії з користувачем або системою керування (рисунк 1).

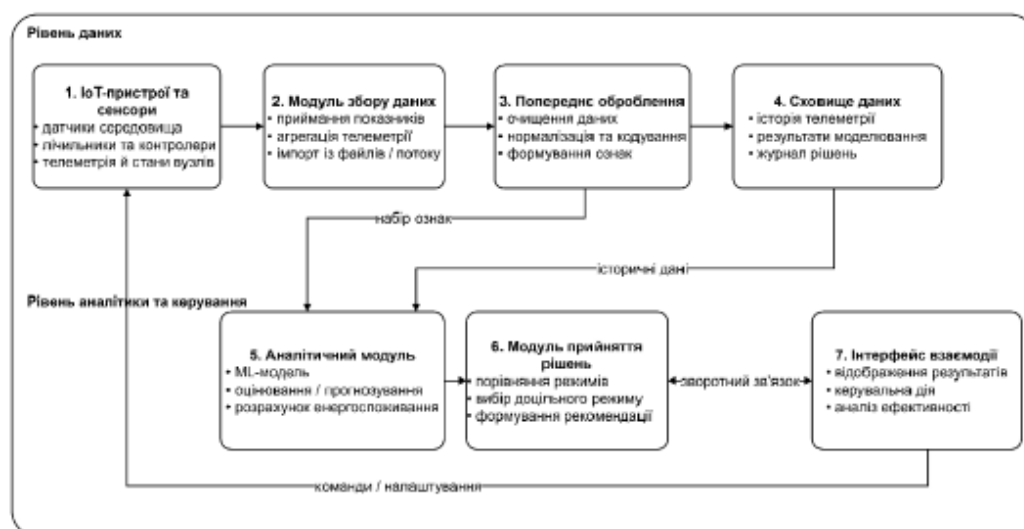


Рисунок 1 – Архітектура інтелектуального модуля

Першим елементом архітектури є модуль збору даних. Його призначення полягає в отриманні вхідної інформації від IoT-пристроїв, датчиків, лічильників або з підготовленого набору даних, якщо робота виконується в режимі моделювання. На цьому рівні система приймає показники сенсорів, часові мітки, відомості про режим роботи пристроїв, інтенсивність мережевої активності, а також інші параметри, що характеризують стан середовища та поточне функціонування вузлів. Цей модуль може бути реалізований у спрощеному вигляді, наприклад як засіб імпорту табличних даних із файлу або приймання потоку телеметрії від емулятора.

Другим компонентом є модуль попереднього оброблення даних. Його роль полягає в підготовці інформації до подальшого аналітичного опрацювання. На цьому етапі виконуються перевірка повноти записів, усунення пропусків, нормалізація або масштабування числових ознак, перетворення категоріальних параметрів, формування часових характеристик і узгодження даних, що надходять із різних джерел. У контексті даної задачі цей блок має особливе значення, оскільки якість роботи моделі оптимізації безпосередньо залежить від якості сформованого набору ознак.

Третім компонентом виступає аналітичний модуль, який є центральним елементом архітектури. Саме тут реалізуються методи машинного навчання або інші процедури, що дозволяють оцінювати енергоспоживання та виявляти залежності між станом середовища, параметрами роботи пристроїв і рівнем витрат електроенергії. Аналітичний модуль може працювати у двох основних режимах. Перший режим передбачає прогнозування споживання енергії для заданого набору параметрів. Другий режим полягає в оцінюванні декількох можливих режимів роботи пристрою та виборі найбільш доцільного варіанта з точки зору енергоефективності. У такому випадку модель не лише описує наявний стан системи, а й підтримує формування керуючого рішення [2, 4].

Четвертим елементом архітектури є модуль прийняття рішень. Його призначення полягає у перетворенні результатів аналітичного блоку на конкретне рішення щодо режиму роботи пристрою. У межах цього компонента можуть застосовуватися правила вибору режиму, порогові умови, логіка порівняння альтернативних варіантів або алгоритм оптимізації, що враховує не лише очікуване енергоспоживання, а й допустимий рівень якості функціонування системи.

П'ятим компонентом архітектури є підсистема збереження даних. Вона забезпечує накопичення телеметрії, збереження проміжних результатів, історії прийнятих рішень та отриманих ефектів. Такий компонент виконує одразу кілька функцій. По-перше, він дозволяє використовувати історичні дані для подальшого навчання або переналаштування моделі. По-друге, забезпечує можливість аналізу ефективності модуля за різні проміжки часу. По-третє, створює основу для аудиту рішень та повторного відтворення експериментів.

Шостим компонентом є модуль взаємодії з користувачем або зовнішньою системою. У найпростішому випадку він реалізується як інтерфейс відображення результатів, де подаються поточні значення параметрів, оцінка споживання енергії, рекомендований режим роботи та узагальнені показники ефективності. Якщо система використовується в режимі автоматизованого керування, цей блок може також виконувати роль інтерфейсу передавання команд до виконавчих пристроїв.

Література:

1. Alanne K., Sierla S. An overview of machine learning applications for smart buildings. *Sustainable Cities and Society*. 2022. Vol. 76. Art. 103445.
2. Mariano-Hernández D., Hernández-Callejo L., García F. S., Duque-Perez O., Zorita-Lamadrid A. L. A Review of Energy Consumption Forecasting in Smart Buildings: Methods, Input Variables, Forecasting Horizon and Metrics. *Applied Sciences*. 2020. Vol. 10, no. 23. Art. 8323.
3. Shah S. F. A., Iqbal M., Aziz Z., Rana T. A., Khalid A., Cheah Y.-N., Arif M. The Role of Machine Learning and the Internet of Things in Smart Buildings for Energy Efficiency. *Applied Sciences*. 2022. Vol. 12, no. 15. Art. 7882.
4. Afzal S., Ziapour B. M., Shokri A., Shakibi H., Sobhani B. Building energy consumption prediction and optimization using different neural network-assisted models; comparison of different networks and optimization algorithms. *Engineering Applications of Artificial Intelligence*. 2024. Vol. 127. Art. 107356.