

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**  
**Західноукраїнський національний університет**  
**Факультет комп'ютерних інформаційних технологій**  
Кафедра комп'ютерної інженерії

**КВІТЕНЬ Дмитро Олександрович**

**Алгоритми класифікації режимів енергоспоживання  
для зниження пікових навантажень в "розумному"  
будинку / Classification algorithms of the energy  
consumption modes for peak loads reducing in a  
"smart" home**

спеціальність: 123 - Комп'ютерна інженерія  
освітньо-професійна програма - Комп'ютерна інженерія  
Кваліфікаційна робота

Виконав студент групи Кім-21  
КВІТЕНЬ Дмитро Олександрович

---

Науковий керівник  
к.т.н., Г.М. Мельник

---

**Тернопіль – 2025**

## АНОТАЦІЯ

Квітень О.О. Алгоритми класифікації режимів енергоспоживання для зниження пікових навантажень в "розумному" будинку. – Рукопис.

Кваліфікаційна робота на здобуття освітнього ступеня «магістр» за спеціальністю 123 «Комп'ютерна інженерія», освітньо-професійна програма. Західноукраїнський національний університет, Тернопіль, 2025.

Робота написана обсягом 60 сторінок і містить 8 ілюстрацій, 3 таблиці, 2 додатки та 35 джерел за переліком посилань.

Метою кваліфікаційної роботи є розроблення алгоритмів класифікації побутових режимів енергоспоживання на основі часових рядів для зменшення пікових навантажень та підвищення ефективності енергоменеджменту в умовах "розумного" будинку.

У роботі проведено аналіз методів класифікації режимів енергоспоживання, розглянуто часові ряди як джерело даних, розроблено алгоритм класифікації на основі статистичних ознак, таких як середнє значення, стандартне відхилення, коефіцієнт варіації тощо. Для реалізації системи створено веб-платформу, що забезпечує взаємодію користувача з сервером, обробку запитів, класифікацію введених даних і генерацію рекомендацій для оптимізації побутового навантаження. Сайт адаптований для використання в умовах періодичних відключень електроенергії в Україні.

Практичне значення розробки полягає у можливості її застосування в реальних умовах для стабілізації локальних мереж споживання, підвищення гнучкості управління енергією та інтеграції у більші енергоменеджмент-системи.

**КЛЮЧОВІ СЛОВА:** ЕНЕРГОСПОЖИВАННЯ, РОЗУМНИЙ БУДИНОК, ЧАСОВІ РЯДИ, КЛАСИФІКАЦІЯ НА ОСНОВІ ПРАВИЛ, ВЕБ-ПЛАТФОРМА, ПІКОВЕ НАВАНТАЖЕННЯ, ОПТИМІЗАЦІЯ.

## ANNOTATION

Kviten O.O. Algorithms for classifying energy consumption modes to reduce peak loads in a "smart" house. – Manuscript.

Qualification work for obtaining the degree of "master" in specialty 123 "Computer Engineering", educational and professional program. Western Ukrainian National University, Ternopil, 2025.

The work is written in 60 pages and contains 8 illustrations, 3 tables, 2 appendices and 35 sources according to the list of references.

The purpose of the qualification work is to develop algorithms for classifying household energy consumption modes based on time series to reduce peak loads and increase the efficiency of energy management in a "smart" house.

The paper analyzes the methods for classifying energy consumption modes, considers time series as a data source, develops a rule-based classification algorithm based on statistical features, such as the mean, standard deviation, coefficient of variation, etc. To implement the system, a web platform has been created that provides user interaction with the server, query processing, classification of entered data, and generation of recommendations for optimizing household load. The site is adapted for use in conditions of periodic power outages in Ukraine.

The practical significance of the development lies in the possibility of its application in real conditions for stabilizing local consumption networks, increasing the flexibility of energy management, and integration into larger energy management systems.

**KEYWORDS:** ENERGY CONSUMPTION, SMART HOME, TIME SERIES, RULE-BASED CLASSIFICATION, WEB PLATFORM, PEAK LOAD, OPTIMIZATION.

## ЗМІСТ

Вступ.....	7
1 Аналіз методів класифікації режимів енергоспоживання.....	10
1.1 Енергоспоживання в системах “розумного” будинку.....	10
1.2 Методи обробки та аналізу часових рядів споживання.....	15
1.3 Логіка класифікації режимів енергоспоживання у веб-системі.....	21
1.4 Висновки до розділу.....	25
2. Розробка алгоритмів класифікації енергоспоживання.....	27
2.1. Формування і підготовка датасету.....	27
2.2. Розробка ознак та побудова математичних моделей.....	33
2.3. Проектування ООП-моделей системи.....	40
2.4 Висновки до розділу.....	47
3. Експериментальне дослідження роботи алгоритмів.....	48
3.1. Архітектура та функціональні можливості веб-платформи.....	48
3.2. Інтеграція алгоритмів у веб-сайт.....	60
3.3. Експериментальне оцінювання точності класифікації.....	61
3.4 Висновки до розділу.....	65
Висновки.....	66
Список використаних джерел.....	67
Додаток А Лістинг програмних засобів.....	70
Додаток Б Світлокопії виданих публікацій.....	73

## ВСТУП

Актуальність теми зумовлена зростанням ролі побутових енергетичних навантажень у структурі сучасних електромереж, які дедалі частіше працюють на межі пропускної здатності через нерівномірність споживання, високу частку імпульсних приладів та нестабільність добових профілів. У системах “розумного” будинку ці явища проявляються особливо виразно: інтенсивне використання електрочайників, мікрохвильових печей, бойлерів та пральних машин створює короткочасні, але значні пікові навантаження, що збільшують втрати, провокують мікроперепади напруги та погіршують роботу локальних енергетичних систем. Підвищення навантаження на внутрішньобудинкові мережі суттєво ускладнюється тим, що цифрові IoT-пристрої здебільшого генерують великі масиви сирих даних, але не забезпечують прямої інтерпретації режимів роботи побутових приладів, залишаючи користувача “один на один” із часовими рядами, які важко аналізувати без спеціалізованих інструментів.

Традиційні підходи до класифікації енергоспоживання зазвичай спираються на складні моделі машинного навчання, що потребують масштабних датасетів, значних обчислювальних ресурсів і тривалого циклу налаштування моделі. У побутових умовах такі методи не є оптимальними: користувач не має можливості розгортати додаткову інфраструктуру для збору даних чи забезпечувати постійне донавчання моделей. Водночас саме побутовий сегмент сьогодні демонструє зростаючу потребу в автоматизованих механізмах розпізнавання режимів роботи приладів, які могли б працювати у реальному часі, пояснювати власні рішення та не вимагали складного технічного обслуговування.

Проблема набуває особливої ваги у контексті енергетичних викликів в Україні, де побутові пікові навантаження безпосередньо впливають на стабільність внутрішніх мереж, ефективність використання резервних джерел та можливість інтеграції “розумних” систем керування. Навіть короткочасні перевищення навантаження у вечірні години суттєво знижують ефективність

використання енергії, прискорюють знос обладнання та ускладнюють роботу локальних систем генерації — від мікроінверторів до домашніх акумуляторних систем.

У цьому контексті особливої актуальності набуває створення прозорих, інтерпретованих алгоритмів класифікації режимів енергоспоживання, здатних працювати на рівні окремого домогосподарства та не вимагати спеціалізованих ресурсів. Пояснювана класифікація на основі часових рядів дозволяє не лише визначати типи навантаження, а й робити це так, щоб користувач міг зрозуміти природу енергетичних “сплесків”, пов’язати їх із поведінкою конкретних пристроїв та отримати обґрунтовані рекомендації щодо зниження пікового споживання.

Таким чином, робота є актуальною через поєднання кількох факторів: потреби у зниженні пікових побутових навантажень, відсутності доступних інтерпретованих алгоритмів класифікації для “розумних” будинків, необхідності роботи з часовими рядами без складної ML-інфраструктури, а також зростаючої ролі користувацьких систем енергоменеджменту у забезпеченні стабільності локальних електромереж. Розроблення таких алгоритмів не лише підсилює функціональність побутових IoT-рішень, а й створює підґрунтя для більш відповідального, керованого та прогнозованого використання енергоресурсів у сучасному житловому середовищі.

Мета роботи: розробити алгоритми класифікації режимів енергоспоживання побутових приладів з метою зниження пікових навантажень у системах “розумного” будинку. Задачі дослідження:

1. проаналізувати особливості формування побутових навантажень та підходи до їх класифікації;
2. зібрати часові ряди споживання електроенергії для різних типів приладів;
3. виділити характерні статистичні ознаки для класифікації режимів роботи приладів;
4. реалізувати алгоритм класифікації режимів роботи;

5. провести експериментальну перевірку роботи системи та оцінити точність класифікації.

Об'єкт дослідження - процеси формування та класифікації режимів енергоспоживання в системах “розумного” будинку. Предмет дослідження - методи обробки часових рядів, побудова ознак і логіки класифікації режимів роботи побутових приладів, інструменти реалізації веб-платформи.

Наукова новизна результатів. Розроблено алгоритм класифікації режимів енергоспоживання побутових приладів за часовими рядами, за рахунок використання компактного вектора фізично змістовних ознак, що дало змогу забезпечити пояснювану класифікацію основних типів навантажень без стадії навчання моделі.

Практичне значення отриманих результатів полягає в тому, що запропонований алгоритм може бути безпосередньо інтегрований у системи “розумного” будинку для онлайн-моніторингу споживання, виявлення енергозатратних режимів і формування рекомендацій щодо зменшення пікових навантажень.

Апробація роботи. Отримані результати опубліковані в межах III Всеукраїнської науково-практичної конференції «Інтелектуальні комп'ютерні системи та мережі», опубліковано тези доповіді [1,2].

Кваліфікаційна робота містить три розділи, висновки, список використаних джерел та додатки.

В першому розділі проведено аналіз методів класифікації режимів енергоспоживання.

В другому розділі здійснено розробку алгоритмів класифікації енергоспоживання на основі часових рядів.

В третьому розділі здійснено експериментальне дослідження роботи алгоритмів.

# 1 АНАЛІЗ МЕТОДІВ КЛАСИФІКАЦІЇ РЕЖИМІВ ЕНЕРГОСПОЖИВАННЯ

## 1.1 Енергоспоживання в системах “розумного” будинку

Із стрімким розвитком технологій Інтернету речей (Internet of Things, IoT), розширенням можливостей цифрових сенсорів, а також широким впровадженням “розумної” побутової техніки, поняття “розумного” будинку вийшло за межі експериментальних лабораторій та концептів, перетворившись на реальний інструмент побутового управління. У сучасному світі “розумний” будинок - це вже не просто автоматичне вмикання світла або клімат-контроль, а цілісна система, що дозволяє моніторити, аналізувати та оптимізувати використання ключових ресурсів, зокрема електроенергії, у реальному часі. Завдяки інтеграції обчислювальних можливостей з датчиками споживання та віддаленим доступом, користувачі отримують контроль над енергоспоживанням із точністю до окремих приладів, що відкриває нові можливості як у сфері побутової ефективності, так і в загальній енергетичній стабільності [3].

Ключовим функціональним напрямом у таких системах є оптимізація енергоспоживання, яка включає аналіз структур даних про використання електроенергії різними приладами, своєчасне виявлення аномалій у поведінці пристроїв, прогнозування навантажень, а також генерацію рекомендацій для уникнення перевантажень. Такий підхід дозволяє не лише зменшити витрати на електроенергію, а й підтримувати стабільність у мережі, знижувати втрати, запобігати піковим навантаженням і формувати профілактичні моделі обслуговування електромереж. У поєднанні з автоматизованим керуванням це створює адаптивне енергетичне середовище, що здатне динамічно змінювати свою поведінку відповідно до внутрішніх і зовнішніх умов.

Це набуває особливої актуальності у контексті України, яка, через повномасштабну збройну агресію та спричинене нею руйнування об'єктів критичної енергетичної інфраструктури, регулярно стикається з ситуаціями аварійного вимкнення електропостачання, дефіциту потужностей, необхідності

формування графіків відключень. У таких умовах розумна система управління енергоспоживанням - не просто засіб комфорту, а один із ключових інструментів підвищення енергетичної безпеки на локальному рівні. Саме вміння системи коректно розподіляти навантаження, розпізнавати пріоритетні прилади, адаптувати поведінку в умовах обмеження потужності та автоматично реагувати на нестандартні ситуації визначає її ефективність і доцільність використання в сучасному побуті.

У межах даної роботи потрібно реалізувати експериментальну IoT-сумісну платформу, що моделює поведінку мініатюрної системи енергоменеджменту з можливістю аналізу сигналів споживання електроенергії, їх класифікації та візуалізації [4]. Розроблений веб-додаток буде функціонувати як повноцінна клієнт-серверна система, у якій користувач може подавати як тестові, так і реальні часові ряди споживання, на основі яких автоматично виконується аналіз. Класифікація відбуватиметься у режимі реального часу, без використання зовнішніх сервісів чи складних моделей машинного навчання - уся логіка реалізована локально, із використанням підходу на основі правил.

Особливу увагу під час реалізації варто приділяти зрозумілості та гнучкості інтерфейсу. Користувач буде має змогу як завантажити власний набір значень (наприклад, дані зі “смарт”-розетки або побутового лічильника), так і сформувати типовий сценарій навантаження - наприклад, роботу циклічного пристрою або імпульсного навантаження. Усі результати миттєво відображаються в інтерфейсі у вигляді таблиці, графіка та текстового пояснення, що значно полегшує інтерпретацію результатів і дозволяє навіть непідготовленому користувачеві зрозуміти, який режим споживання спостерігається.

Платформи будуються з урахуванням можливості подальшої інтеграції із фізичними пристроями, зокрема з модулем збору даних через протокол MQTT, REST API або WebSocket. Це дозволяє у перспективі використовувати її не лише як тестову модель, а як повноцінний веб-інтерфейс для реального контролю за енергоспоживанням. Крім цього, архітектура повинна підтримувати розширення: додавання нових правил класифікації, оновлення алгоритмів розрахунку ознак,

підключення зовнішніх баз даних та навіть експерименти з альтернативними способами візуалізації.

Система може бути корисною і для дослідницької діяльності. В умовах відсутності реального обладнання вона дозволить моделювати типові та нетипові сценарії енергоспоживання, проводити тестування класифікаційної логіки, перевіряти гіпотези, аналізувати вплив зміни структури сигналу на класифікацію. Це створює гнучке середовище для навчання, обкатки ідей та побудови нових підходів до енергетичного аналізу в межах концепції “розумного” дому.

Основними джерелами енергоспоживання у побутових умовах залишаються різні типи електроприладів, кожен з яких має власну динаміку споживання [4-8]. Згідно з функціональною класифікацією, можна виокремити такі основні групи:

- постійні навантаження, які характеризуються рівномірним і стабільним споживанням протягом тривалого часу (наприклад, освітлення, роутери);
- циклічні прилади, що мають чітко виражені періоди включення/вимкнення (наприклад, холодильники, пральні машини);
- короткочасні споживачі, що активуються лише на кілька хвилин (мікрохвильові печі, електрочайники);
- активні або високопотужні пристрої, які створюють суттєве навантаження, часто зі змінною потужністю (бойлери, кондиціонери).

Форма сигналу, що надходить від цих пристроїв, має характерні ознаки, за якими можна розпізнати тип роботи. Саме ці ознаки були закладені в основу побудови класифікаційної логіки платформи. У рамках проєкту було визначено набір ключових характеристик, які можуть однозначно ідентифікувати тип навантаження за часовим рядом. Ці характеристики формують основу правила для класифікації, яка відбувається автоматично після аналізу сигналу. Результати аналізу дозволяють не лише описати поточну ситуацію, а й будувати поведінкову модель споживання.

У таблиці 1.1 наведено базову класифікацію побутових приладів відповідно до характеру створюваного ними навантаження. Такий підхід дозволяє

систематизувати джерела споживання та адаптувати систему класифікації до найпоширеніших сценаріїв у сучасному домогосподарстві.

Важливим кроком до ефективного керування енергоспоживанням у рамках концепції “розумного” будинку є збір детальних даних про динаміку навантаження, що здійснюється за допомогою сучасних засобів моніторингу: “розумних” лічильників, вбудованих сенсорів, контролерів або спеціалізованих IoT-модулів, інтегрованих у побутові електроприлади. Ці пристрої забезпечують постійний потік даних, фіксуючи зміну споживаної потужності з точною прив’язкою до часу. У результаті формується часовий ряд - послідовність значень, які відображають електроспоживання у вигляді функції від часу, дозволяючи відтворити повну картину активності приладу, у тому числі виявити піки, плато, паузи, циклічні коливання чи імпульсні включення.

Таблиця 1.1 - Класифікація побутових електричних навантажень

Тип навантаження	Опис
Статичне	Постійне, рівномірне споживання протягом тривалого часу (напр., освітлення, зарядні пристрої).
Динамічне	Споживання змінюється залежно від режиму роботи (пральна машина, духовка).
Циклічне	Повторювані цикли вмикання/вимикання з подібною структурою (холодильник, бойлер).
Короткочасне	Короткі імпульси споживання з високою амплітудою (мікрохвильова піч, пилосос).

Такий тип представлення даних є базовим у всіх сучасних енергоменеджмент-системах, оскільки дозволяє перейти від фіксації показників до їх глибокої інтерпретації. Саме часові ряди є основою для аналітики, побудови моделей споживання, класифікації режимів та формування прогнозів. Кожен сигнал, що подається на вхід платформи, проходить попередню обробку, в ході якої обчислюється набір ознак, таких як середнє значення, стандартне відхилення, коефіцієнт варіації, пікова амплітуда, тривалість активного стану тощо. Після цього сигнал передається на класифікаційний модуль, який на основі заданої

логіки визначає, до якого режиму споживання належить вхідний профіль. У такому підході важливу роль відіграє структура аналізу: платформа не розглядає споживання як ізольовані епізоди, а як частину комплексної поведінкової моделі, яка формується на основі ряду змін, закономірностей і повторів.

На основі аналізу часових рядів системи здатні визначити характер споживання: чи це імпульсний прилад, як мікрохвильова піч, чи пристрій з чітким ритмом включень, як холодильник, або ж пристрій з високим та тривалим навантаженням, як бойлер. Таке розпізнавання відкриває простір для контекстних рішень, коли система не лише констатує факт, а й надає рекомендації. Наприклад, користувач може отримати підказку перенести роботу потужного приладу на нічний тарифний період, уникати одночасної активації кількох енергоємних пристроїв або оптимізувати роботу пральної машини відповідно до доступної потужності в мережі.

Оскільки платформи функціонують в реальному часі, підказки та рішення генеруються миттєво, що забезпечує не лише зручність, але й оперативність у прийнятті рішень [5-11]. Також важливо зазначити, що проєктована система не потребуватиме підключення до хмарних сервісів або складних ML-моделей - усі обчислення будуть виконуватися локально, що забезпечить автономність і підвищену безпеку. Такий підхід особливо актуальний у сучасних умовах, коли збої у зв'язку або енергопостачанні можуть обмежити доступ до зовнішніх ресурсів.

Окрім безпосередньої економії ресурсів на рівні окремої квартири чи домогосподарства, балансування енергоспоживання позитивно впливає на стабільність локальної енергосистеми. У випадках, коли кілька квартир одночасно активують енергоємні пристрої, можуть виникнути короточасні просадки напруги, перевантаження трансформаторних ліній або навіть аварійні ситуації. Завдяки впровадженню інтелектуальних платформ - таких як розроблена - стає можливим розумне планування навантаження, що особливо важливо у багатоквартирних будинках, ОСББ чи енергоефективних мікрорайонах.

У сфері енергетичного аналізу існує безліч підходів: від простого виявлення пікових значень до побудови моделей прогнозування із застосуванням нейромереж. Проте саме класифікація типів режимів споживання дозволяє найточніше виявляти поведінкові шаблони, які є основою для автоматизованих систем реагування. Саме на цьому принципі побудована логіка запропонованої системи. Хоча вона не використовує складних моделей глибокого навчання, реалізована логіка демонструє високу точність класифікації, завдяки оптимально підібраному набору ознак та ретельно продуманим правилам.

Сьогодні “розумні” будинки більше не обмежуються автоматичним вмиканням світла або клімат-контролем. Вони еволюціонують у повноцінні аналітичні системи, що вміють адаптуватися, прогнозувати та реагувати на зміни середовища. Запропонована у цій роботі IoT-сумісна платформа є яскравим прикладом такого підходу. Вона демонструє, як, навіть без залучення складної обчислювальної інфраструктури, можна досягти високого рівня контролю над енергоспоживанням, створити передумови для енергоощадності та зробити свій внесок у загальну енергетичну стабільність. Таким чином, інтелектуальні системи керування споживанням електроенергії вже сьогодні стають важливим компонентом побутової інфраструктури. Вони забезпечують не лише зручність для користувача, а й підвищують енергетичну безпеку, що є критично важливо в умовах нестабільності, загроз інфраструктурним об’єктам та потреби в адаптивному розподілі навантажень. Розроблена система демонструє приклад практичного впровадження такої логіки в реальне середовище, і може бути масштабована як у межах окремого домогосподарства, так і для застосування в більших енергетичних структурах.

## 1.2 Методи обробки та аналізу часових рядів споживання

У системах “розумного” будинку дані про споживання електроенергії

надходять у вигляді часових рядів - послідовностей числових значень, що фіксуються з фіксованою частотою у часі. Кожне значення в такому ряді відображає миттєвий стан навантаження - наприклад, рівень потужності, яку споживає прилад у певний момент. Аналіз таких рядів дозволяє не лише побачити загальний обсяг енергоспоживання, а й відтворити динаміку процесу: виявити початок і завершення роботи пристрою, оцінити тривалість навантаження, його інтенсивність, наявність пауз, повторень або раптових стрибків. Часовий ряд фактично виконує роль "цифрового відбитка" поведінки електроприладу, і тому є ключовим джерелом інформації для формування системи управління навантаженнями.

У межах цієї роботи саме така структура представлення даних - часовий ряд потужності - була покладена в основу логіки класифікації. Розроблена веб-платформа, що реалізує цю логіку, дозволяє завантажувати або отримувати в реальному часі відповідні сигнали, які далі автоматично аналізуються з використанням вбудованих математичних модулів. Ці модулі враховують як статистичні параметри сигналу, так і його часову структуру. У результаті платформа не лише фіксує рівень споживання, а й розпізнає, як саме він змінюється: чи це плавне зростання, імпульсне включення, чи циклічне повторення. Завдяки цьому досягається глибше розуміння природи навантаження, що має принципове значення для управлінських рішень.

Для ефективної обробки таких сигналів були реалізовані статистичні методи, які дозволяють швидко отримати основні узагальнюючі показники: середнє значення потужності, дисперсію, стандартне відхилення, коефіцієнт варіації, відношення пікових значень до середніх. Ці ознаки у сукупності формують вектор, який передається на вхід класифікаційного модуля. Саме на основі цього вектора система робить висновок про те, до якого типу режиму належить аналізований сигнал. У випадках, коли споживання є сталим або слабо змінюється, система може класифікувати його як статичне; у випадку, коли фіксується регулярна зміна активних фаз - як циклічне, тощо.

Однак, одного лише статистичного аналізу недостатньо для точного розпізнавання складних патернів. Тому в рамках проєкту були також використані спектральні методи, які дають змогу виявити повторювані частотні компоненти сигналу. Зокрема, застосування швидкого перетворення Фур'є (FFT) дозволяє отримати спектр сигналу і побачити, чи є в ньому виражені частоти, характерні для циклічних приладів - як-от холодильники, пральні машини, системи вентиляції. Подібні частоти є "підписами" пристроїв, і їх наявність у сигналі дозволяє точніше класифікувати навантаження.

Особливу роль у системі також відіграє модуль виявлення повторюваних шаблонів. Це окремий програмний блок, який здійснює аналіз зміни сигналу на локальних відрізках часу. Застосування автокореляційного аналізу дозволяє виявити наявність регулярних циклів, навіть якщо вони не є ідеально однаковими. Метод ковзного вікна дає змогу локалізувати зміну активності, тоді як виявлення локальних максимумів допомагає візуалізувати імпульси або піки навантаження. Така комбінація інструментів дає змогу розрізнити, до прикладу, нерівномірне імпульсне споживання мікрохвильовки від плавного циклу нагріву бойлера, що в іншому випадку могло б виглядати подібно на графіку.

Інтерфейс платформи дозволяє користувачеві не лише запускати процес класифікації, а й візуально переглядати часові ряди, ознайомлюватися з проміжними результатами обробки, графічно оцінювати форму сигналу. Це робить систему зрозумілою для кінцевого користувача, що важливо в контексті інтеграції в побутові умови. Окрім того, така візуалізація підвищує довіру до рішень, які ухвалює система, оскільки дозволяє наочно оцінити логіку класифікації.

Завдяки поєднанню статистичного, спектрального та шаблонного аналізу, платформа створює повноцінну картину споживання, яка може використовуватись як для інформування користувача, так і для побудови більш складних систем автоматичного керування. Усі реалізовані методи були протестовані на наборі тестових сигналів, які моделювали роботу типових побутових пристроїв. Результати показали стабільну та коректну класифікацію,

підтверджуючи ефективність обраного підходу та можливість його масштабування.

Статистичні характеристики сигналу.

Базові статистичні параметри дозволяють описати загальну поведінку сигналу та його мінливість, виявити аномалії та визначити потенційний тип навантаження. Основні показники включають:

- середнє значення  $\mu$  визначає середній рівень споживання за вибраний період.
- дисперсія  $\sigma^2$  показує, наскільки змінним є сигнал.
- стандартне відхилення  $\sigma$  характеризує типову амплітуду коливань.
- автокореляція дозволяє виявити повторювані шаблони, наприклад, вмикання приладу щогодини.
- піковість оцінює наявність імпульсних, раптових підвищень потужності.
- коефіцієнт варіації  $CV$  співвідношення між стандартним відхиленням та середнім значенням, що вказує на стабільність навантаження.

Ці характеристики можуть слугувати основою для формування вектора ознак при класифікації типу енергоспоживання, що буде реалізовано в подальших розділах роботи.

Спектральні методи аналізу.

Крім аналізу у часовій області, доцільно застосовувати спектральні методи, які дозволяють виявити частотні компоненти сигналу. Це особливо важливо для циклічних або періодичних навантажень, таких як робота компресора холодильника або насоса в автономній системі опалення.

Найпоширенішими спектральними методами є:

- Швидке перетворення Фур'є (FFT) - дає змогу побудувати спектр сигналу, визначити домінуючі частоти та оцінити їхню амплітуду.
- Вейвлет-аналіз - дозволяє локалізувати частотні компоненти у часі, що є особливо важливим у випадку непостійних або комбінованих режимів роботи.
- Фільтрація сигналу - застосовується для згладжування шуму, виділення основного тренду або ізоляції корисних компонентів сигналу.

Застосування цих методів підвищує точність виявлення закономірностей у поведінці приладів.

Виявлення циклічності та шаблонів (патернів).

Ще одним важливим аспектом є виявлення повторюваних шаблонів у споживанні, що дозволяє робити висновки про типові режими роботи приладів. Серед відповідних методів можна виділити:

- автокореляційний аналіз - дає змогу визначити регулярність повторів у сигналі.
- метод ковзного вікна - дозволяє локально оцінити зміни тренду та варіативність сигналу.
- кластеризація фрагментів - умовний поділ схожих відрізків на групи з подібною структурою, що зручно для групування типових сценаріїв роботи.

Завдяки цим підходам можливо, наприклад, виявити регулярні нічні ввіключення бойлера, циклічну роботу холодильника або короткочасні імпульси мікрохвильової печі.

Приклад форми сигналу з циклічними вмиканнями (умовна форма для холодильника)

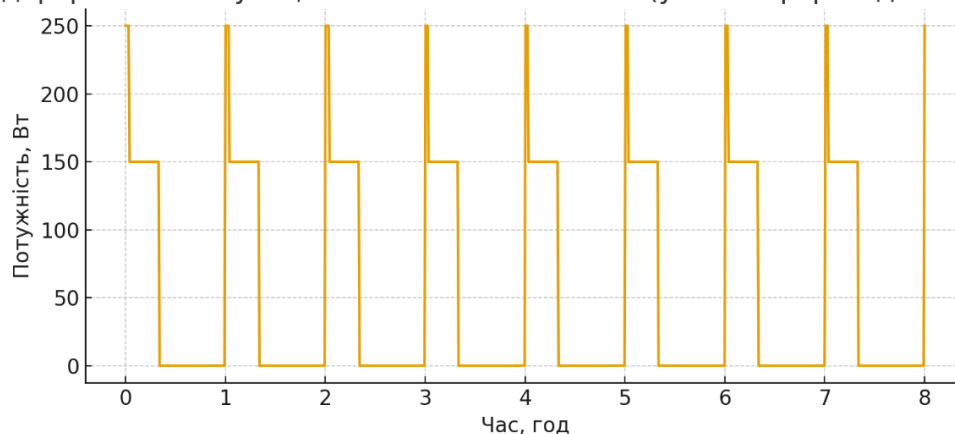


Рисунок 1.2 - Приклад форми сигналу з циклічними вмиканнями

Таким чином, комбінування статистичного та спектрального аналізу дозволяє не лише кількісно описати поведінку побутових приладів, а й виявити глибинні закономірності у сигналі, які безпосередньо вказують на тип пристрою, його режим роботи та характер змін навантаження у часі. На відміну від звичайної фіксації моментів пікового споживання або простого підрахунку середнього

значення, такий підхід забезпечує багатовимірну інтерпретацію сигналу, що є особливо важливою у разі змішаних, нестабільних або періодично змінних режимів.

Методи статистичного аналізу дозволяють зосередитися на базових параметрах енергоспоживання, які легко інтерпретуються, масштабуються та узагальнюються для побудови правил класифікації. У той час спектральні методи, зокрема застосування перетворення Фур'є або аналізу частотних компонент, дають змогу розкрити внутрішню структуру споживання: наприклад, виявити наявність циклів, ритмічних повторень, імпульсних фаз. У поєднанні ці підходи формують більш точну модель поведінки приладу, з урахуванням як його потужності, так і характеру динаміки сигналу.

У розробленій веб-платформі ці аналітичні методи інтегровані у єдину систему обробки, що забезпечує об'єднання статистичних ознак (наприклад, середнє значення, коефіцієнт варіації, стандартне відхилення) з часовими та періодичними параметрами. Такий підхід дозволяє не просто аналізувати форму сигналу, а й контекстуалізувати її - тобто виявляти, чи є зміна типова, випадкова, або ж така, що вимагає втручання. Зокрема, алгоритм може розрізнити незначне коливання від суттєвого відхилення від звичної поведінки пристрою, а це - ключ до своєчасного реагування.

Особливою перевагою інтегрованої системи аналізу стало те, що вона адаптована для реалізації у веб-середовищі, без необхідності зовнішньої обробки даних або підключення до хмарних сервісів. Усі обчислення відбуваються безпосередньо в середовищі веб-платформи - це забезпечує оперативність, автономність та безперервну доступність. Користувач може взаємодіяти із системою в режимі реального часу: вводити сигнали, переглядати результат класифікації, аналізувати графіки, отримувати поради щодо оптимізації навантаження.

На основі об'єднаних результатів статистичного та спектрального аналізу формується інформаційна база для прийняття рішень. Це можуть бути як автоматизовані дії (перемикання режиму, відключення приладу), так і поради

користувачу - наприклад, щодо перенесення активності приладу на нічний час, або виявлення підозрілих сплесків, що можуть свідчити про технічну несправність. Така модель прийняття рішень є гнучкою та масштабованою, її можна легко адаптувати під потреби конкретної квартири, будинку або навіть невеликого офісу.

У ході тестування платформи розроблена логіка обробки показала високу точність при класифікації тестових сценаріїв, у тому числі при роботі з симульованими сигналами, що містили як звичні, так і атипові режими. Платформа стабільно розпізнавала типи сигналів, навіть у випадках, коли форма була зміненою або містила шуми. Це підтвердило надійність застосованого підходу для реальних умов.

Отже, об'єднання статистичного та спектрального аналізу створює фундамент для побудови адаптивної, чутливої до змін та інформативної системи енергоменеджменту. Реалізований алгоритм, втілений у межах функціональної IoT-сумісної платформи, є не лише інструментом моніторингу, а й активним елементом управління енергоспоживанням. Така система повністю відповідає викликам сучасності: вона проста у впровадженні, зрозуміла для користувача, відкрита до розширення й здатна до автономної роботи. У результаті вона становить важливий компонент у реалізації концепції “розумного” будинку, де енергія споживається розумно, ефективно та відповідально.

### 1.3 Логіка класифікації режимів енергоспоживання у веб-системі

Класифікація режимів енергоспоживання - це процес віднесення певного набору показників, отриманих у результаті аналізу сигналу споживання, до одного з попередньо визначених типів навантаження. Такий підхід дає змогу формалізувати розуміння того, як працює конкретний побутовий пристрій, і яку форму впливу він має на загальну енергосистему. У рамках побутових систем, а

особливо в умовах функціонування “розумного” будинку, мета класифікації полягає у виявленні поведінкових шаблонів, які дозволяють точно розпізнавати роботу приладів, контролювати її, оптимізувати графік енергоспоживання, попереджувати потенційні перевантаження мережі, а також формувати персоналізовані рекомендації для користувачів. У контексті сучасного енергоменеджменту класифікація виступає не лише допоміжним інструментом, а фактично основою автоматизованого прийняття рішень у режимі реального часу.

Завдяки класифікації можливо не лише зафіксувати сам факт споживання, а й виявити його тип - наприклад, чи є сплеск на графіку результатом імпульсного ввімкнення електрочайника, регулярного циклу холодильника, або стабільного навантаження від LED-освітлення. Такі знання дозволяють не лише прогнозувати подальші зміни у навантаженні, але й управляти ними - шляхом переміщення активних споживачів у менш завантажені часові проміжки або їх тимчасового відключення.

На практиці реалізація класифікатора має враховувати обмеження реального середовища: відсутність потужних серверів, нестабільне з'єднання, обмежені ресурси мікроконтролерів чи веб-хостингу. У таких умовах особливу цінність набуває підхід до класифікації на основі правил. На відміну від моделей машинного навчання, які вимагають об'ємного датасету, навчання, обробки та регулярного оновлення, система підходу на основі правил базується на чітко заданих логічних умовах, які зіставляються з характеристиками сигналу. Кожна умова, або правило, базується на заздалегідь обчислених ознаках, серед яких середнє значення потужності, коефіцієнт варіації, пікове навантаження, тривалість сплеску, наявність циклічності, кількість переходів через поріг тощо. Комбінація цих показників формує "поведінковий портрет" споживання, який і стає основою для класифікації.

Rule-based підхід було реалізовано у вигляді серверного модуля, інтегрованого в архітектуру розробленої платформи. Його алгоритм побудований на ланцюжку умовних операторів, які обробляють отриманий часовий ряд та обчислені на його основі характеристики. На основі заданих порогових значень

класифікатор самостійно визначає, до якого типу режиму відноситься сигнал, і повертає результат користувачу у зручному візуальному вигляді. Така реалізація виявилася не лише швидкою, але й технологічно доступною: вона не потребує значних обчислювальних ресурсів, легко інтегрується у веб-середовище та дозволяє виконувати класифікацію практично миттєво - що особливо важливо в умовах обробки даних у реальному часі.

Важливою перевагою даного підходу є його прозорість. Кожне правило можна описати, задокументувати та модифікувати без зміни загальної структури системи. Це відкриває широкі можливості для кастомізації: за потреби розробник може розширити існуючі правила, адаптувати їх під нові типи приладів, або внести порогові коригування залежно від умов споживання конкретного користувача чи будинку. Більше того, користувачі мають змогу інтерпретувати логіку класифікації - тобто зрозуміти, чому платформа ухвалила саме таке рішення щодо конкретного сигналу. Це формує довіру до системи та стимулює залучення користувача до управління власним енергоспоживанням.

У випадку масштабування проєкту або підключення до нього великих обсягів даних, зокрема з “розумних” лічильників, мережевих сенсорів чи зовнішніх баз даних, підхід на основі правил може поступитися місцем більш гнучким моделям - зокрема, алгоритмам машинного навчання або гібридним системам. Однак на поточному етапі розвитку системи механізм на основі правил є цілком виправданим і дозволяє досягти поставлених у межах цієї роботи завдань: визначення режиму роботи побутових приладів, їх поведінкової структури, та надання первинної класифікації, яка є базою для подальших управлінських рішень.

У підсумку можна стверджувати, що реалізований класифікатор підходу на основі правил є ефективним, адаптивним та масштабованим інструментом первинного аналізу енергоспоживання у “розумному” будинку. Він поєднує логічну простоту, швидкодію та гнучкість у впровадженні, що робить його особливо актуальним для локальних систем, експериментальних платформ та побутових рішень, орієнтованих на енергоощадність та автономність.

Переваги підходу на основі правил:

- прозора логіка - правила легко інтерпретуються, редагуються та тестуються;
- незалежність від навчальних даних - підхід не потребує попередньої розмітки або моделювання;
- швидкодія - дозволяє отримувати результати у режимі реального часу;
- адаптивність до побутових умов - легко налаштовується під конкретні сценарії.

Формування ознак та правила класифікації.

Для визначення типу навантаження система використовує набір статистичних ознак, які обчислюються з часових рядів споживання:

- середнє значення потужності;
- максимальне значення;
- стандартне відхилення;
- коефіцієнт варіації (CV);
- ознаки циклічності чи стабільності.

На основі цих параметрів застосовується набір умов, які класифікують сигнал за типом. Наприклад:

- низька варіативність та відсутність піків це “Статичне навантаження” (наприклад, освітлення);
- високий пік при короткій тривалості сигналу це “Короткочасне імпульсне навантаження” (мікрохвильова піч);
- циклічна повторюваність у часі це “Циклічне навантаження” (холодильник);
- високе й стабільне середнє значення цк “Активне (високопотужне) навантаження” (бойлер).

В основу реалізованої системи покладено саме такі логічні правила, приклади яких наведено в таблиці 1.3.

Таблиця 1.3 - Логічна модель класифікації режимів

Умова	Клас режиму
Потужність > 1000 Вт, тривалість > 5 хв	Активне навантаження
Потужність < 50 Вт, стабільне значення	Статичне навантаження
Періодичні коливання у межах 200–600 Вт кожні ~30 хв	Циклічне навантаження
Різкий імпульс > 1500 Вт тривалістю < 1 хв	Короткочасне навантаження

Реалізація правил класифікації була виконана **на стороні сервера** у вигляді послідовності умовних операторів, кожен з яких перевіряє конкретні характеристики вхідного сигналу. Ці оператори аналізують числові значення ознак - зокрема середнє споживання, пікову амплітуду, коефіцієнт варіації, наявність періодичних флуктуацій - і на основі заданих порогових значень визначають, до якого класу належить сигнал. У випадку відповідності одному з критеріїв, система повертає класифікацію: наприклад, «імпульсний режим», «циклічний пристрій» або «активне стабільне споживання».

Такий підхід буде ефективним коли обсяг вхідних даних є обмеженим, а ресурси на обробку - мінімальними. Застосування умовної логіки дозволяє уникнути складного процесу навчання моделей, який зазвичай вимагає значного обсягу історичних даних, їх попередньої обробки, перевірки на узгодженість та оптимізації гіперпараметрів. У контексті невеликої платформи або локальної IoT-системи з обмеженим підключенням до хмарних обчислень, простий механізм підходу на основі правил є оптимальним компромісом між функціональністю та швидкодією.

#### 1.4 Висновки до розділу

Реалізація логіки у вигляді умов на стороні сервера забезпечує гнучкість та розширюваність. У разі потреби зміни або доповнення правил, розробник може

швидко оновити порогові значення, додати нові умови або адаптувати систему до нових типів сигналів без необхідності перебудови всього алгоритму. Це дає змогу легко масштабувати систему в межах невеликих локальних мереж, наприклад, у межах будинку, офісного простору чи лабораторного стенду.

Таким чином, обрана архітектура рішення підтверджує свою релевантність для систем енергомоніторингу побутового рівня і може слугувати як базовий шар для побудови більш складних інтелектуальних платформ управління навантаженням.

## 2. РОЗРОБКА АЛГОРИТМІВ КЛАСИФІКАЦІЇ ЕНЕРГОСПОЖИВАННЯ

### 2.1. Формування і підготовка датасету

Для реалізації ефективної класифікації режимів енергоспоживання у побутових умовах критично важливо мати доступ до даних, які максимально точно відображають типову поведінку електроприладів у реальному середовищі. У межах розробленої системи було вирішено не використовувати складні методи машинного навчання, а натомість зосередитися на підході на основі правил, який базується на фіксованих правилах та наборах ознак. Такий підхід не потребує великих обсягів тренувальних даних, однак вимагає ретельного аналізу та чіткого розуміння логіки споживання кожного окремого пристрою, що дозволяє системі приймати точні рішення без додаткового навчання або корекції моделі.

В основі системи лежить аналіз часових рядів - послідовностей числових значень, які представляють рівень потужності споживання електроенергії (вимірюваної у ватах) з певним інтервалом фіксації (наприклад, щосекунди або щохвилини). Таке представлення дає змогу зберегти повну динаміку зміни навантаження у часі, а отже, є ідеальним джерелом для подальшого аналізу. Кожен часовий ряд - це фактично сигнатура поведінки конкретного електроприладу у конкретному режимі роботи. Ці сигнатури були використані як вхідні дані для побудови системи ознак та логіки класифікації.

Щоб забезпечити різноманітність сценаріїв, було сформовано набір тестових рядів, які охоплюють найпоширеніші типи навантажень у побуті. Враховувалися як основні типи пристроїв (освітлення, мікрохвильовки, пральні машини, холодильники, бойлери), так і варіації в режимах їх використання: безперервне навантаження, імпульсне споживання, періодична активність, поєднання фаз зростання і падіння потужності. У результаті вдалося сформувати репрезентативну вибірку, яка містить ключові патерни, властиві побутовому енергоспоживанню. Ці дані були створені на основі як реальних зчитувань із

сенсорів, так і згенерованих профілів, які емулюють поведінку приладів з урахуванням типових характеристик.

Наприклад, для циклічних пристроїв, таких як холодильники, було змодельовано сигнали з чіткою регулярністю - періодичні піки однакової амплітуди, розділені рівномірними інтервалами пауз. Такий сигнал наочно ідентифікує пристрій, який вмикається з фіксованою частотою, зберігаючи стабільність навантаження у часі. У випадку імпульсних приладів, як-от електрочайники або мікрохвильові печі, сигнали мають характерний сплеск високої амплітуди, за яким слідує різкий спад - така форма дозволяє безпомилково класифікувати даний режим як короткочасний. Постійні навантаження, наприклад, освітлення чи мережеві пристрої, мають стабільні, майже горизонтальні ряди без суттєвих флуктуацій.

У процесі підготовки цих даних особливу увагу було приділено формату запису та частоті дискретизації. Дані представлялися у вигляді масивів JSON, де кожен елемент відповідав певному часовому відрізку. Завдяки такій структурі веб-система отримувала можливість обробляти дані у режимі реального часу, підтримуючи як інтерактивне введення (вручну або через копіювання) так і імпорт готових наборів. Це також дозволяло забезпечити гнучкість платформи - користувач міг експериментувати з власними сигналами, змінювати частоту, тривалість, амплітуду та інші параметри, формуючи нові сценарії для тестування класифікаційної логіки.

Ще одним важливим етапом стала нормалізація даних. Оскільки абсолютні значення потужності можуть суттєво відрізнятися між пристроями, в системі передбачено механізм приведення ряду до відносної шкали - наприклад, від 0 до 1 або відносно середнього значення. Це дозволяє зосередити увагу не на абсолютній величині споживання, а на формі та динаміці сигналу, що є критичним для класифікації на основі правил. Зокрема, нормалізація дозволяє уникнути помилкових рішень у ситуаціях, коли два пристрої споживають однакову кількість енергії, але мають принципово різну поведінку (наприклад, тривалий низький рівень проти короткого, але потужного піку).

Ці ознаки дозволяють охопити як статистичний, так і часовий аспект поведінки сигналу. На відміну від підходів, що використовують машинне навчання, у моделі на основі правил всі рішення базуються на жорстко заданих порогах та умовах, які порівнюються із значеннями ознак. Наприклад, якщо коефіцієнт варіації перевищує певне значення, сигнал автоматично класифікується як імпульсний. Якщо середнє значення стабільне, а пікова амплітуда низька - сигнал ідентифікується як постійне навантаження.

Важливо підкреслити, що такий підхід дозволяє контролювати логіку класифікації вручну, швидко адаптувати систему під нові сценарії, змінювати порогові значення без необхідності перенавчання або повної модифікації алгоритму. Це особливо цінно у побутових IoT-системах, де ресурси обмежені, а вимоги до надійності - високі.

Таким чином, сформований набір даних є не просто тестовим середовищем, а ядром усієї платформи, на основі якого відбувається обробка, аналіз, класифікація і візуалізація поведінки побутових приладів. Завдяки грамотній побудові рядів, систематизації режимів і уніфікації формату вдалося реалізувати гнучку, прозору та ефективну систему розпізнавання, що демонструє стабільну роботу в умовах обмежених ресурсів.

У рамках роботи використовувались реальні та синтетично згенеровані дані, котрі імітують типові режими енергоспоживання. Це обумовлено потребою створити контрольовані приклади для перевірки алгоритму.

Кожен сценарій створювався вручну з урахуванням таких характеристик:

- тип приладу (активний, циклічний, імпульсний, статичний);
- середнє та пікове значення потужності;
- тривалість споживання;
- наявність повторюваних циклів;
- час доби (для імітації користувацької поведінки).

Структура та формат даних. Кожен тестовий приклад - це масив числових значень, поданих на вхід системи через веб-інтерфейс або API. Дані можуть бути:

- згенеровані користувачем вручну (через інтерфейс);

- створені за допомогою генератора сценаріїв (вбудований у систему);
- імпортовані з файлу (опціонально).

Формат вхідних даних наприклад, періодичне споживання компресора холодильника:

[45, 47, 49, 52, 54, 0, 0, 0, 51, 50, 49, 0, 0, 0, 53, 54, 0]

Для початкового етапу було підготовлено 8 сценаріїв, які покривають чотири основні класи (таблиця 2.1).

Таблиця 2.1 – Основні класи

Клас навантаження	Кількість прикладів	Приклади пристроїв
Статичне	2	Освітлення, Wi-Fi роутер
Короткочасне	2	Мікрохвильова піч, чайник
Циклічне	2	Холодильник, пральна машина
Активне (високопотужне)	2	Бойлер, обігрівач

Усі приклади пройшли попередню перевірку, а їхні характеристики були закладені в таблицю правил класифікації (див. табл. 1.3).

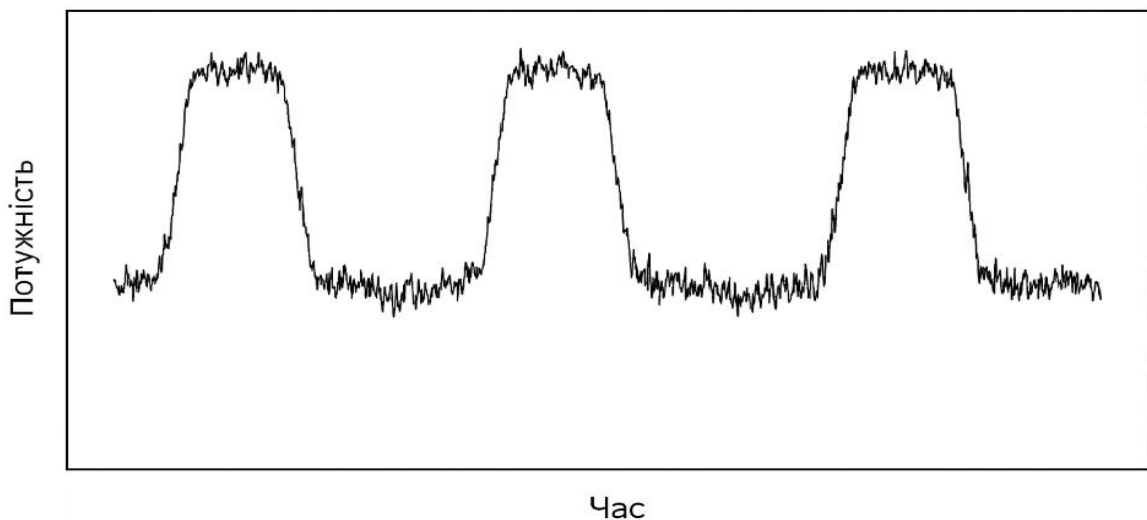


Рисунок 2.1 - Приклад згенерованого часового ряду для циклічного навантаження

Для забезпечення коректної класифікації, система перед подачею даних на обробку, виконує такі дії:

- нормалізує дані до єдиного масштабу (опційно);
- обчислює статистичні ознаки ( $\mu$ ,  $\sigma$ , CV, max, autocorrelation);
- усереднює значення при потребі, щоб уникнути хибної інтерпретації шумів;
- відсікає неінформативні ділянки (наприклад, довгі ділянки з нульовим споживанням).

Таким чином, сформований у межах дослідження датасет хоч і є порівняно невеликим за обсягом, однак має чітко структуровану, репрезентативну природу, що дозволяє з достатньою точністю відтворити типові сценарії енергоспоживання в умовах побутового середовища. Його формування відбувалося з урахуванням найпоширеніших типів електроприладів та особливостей їх поведінки - циклічності, пікових значень, тривалості активних фаз, стабільності у часі тощо. Завдяки цьому навіть невелика кількість тестових рядів дає змогу охопити широкий спектр характерних режимів, що імітують реальні сценарії, з якими стикається система "розумного" будинку у повсякденному житті.

Основною перевагою такого підходу є можливість швидкої валідації логіки класифікації без необхідності доступу до фізичних сенсорів чи розгортання реального середовища збору даних. У традиційних IoT-системах одним із найбільш складних етапів є саме налаштування апаратної частини - підключення сенсорів, синхронізація з контролерами, зберігання даних тощо. Натомість використання заздалегідь сформованих рядів дозволяє повністю зосередитися на програмній логіці, підвищуючи ефективність розробки, пришвидшуючи тестування і спрощуючи відлагодження системи. Це особливо актуально на ранніх етапах, коли пріоритетом є не інтеграція з "залізом", а перевірка концептуальної спроможності системи класифікації та коректність алгоритмічної реалізації.

Завдяки структурованому підходу до побудови тестових сигналів у системі вдалося забезпечити високу якість аналізу. Кожен сценарій був створений із дотриманням внутрішньої логіки споживання, тобто відповідав реальній моделі

поведінки приладу, яку можна очікувати в житловому будинку. Наприклад, профіль холодильника мав регулярні короткочасні піки фіксованої амплітуди з однаковими паузами, бойлер - зтяжне рівномірне навантаження із повільним спадом, мікрохвильова піч - інтенсивний імпульс протягом 2–3 хвилин. Подібне відтворення реальних патернів дозволяє не просто емулювати споживання, а побудувати реалістичну поведінкову модель, з якою класифікатор стикається під час роботи.

Обраний підхід ефективно працює в умовах, коли кожна вхідна ознака має зрозуміле фізичне значення, а рішення приймається на основі логічного аналізу цих ознак. Завдяки тому, що кожен ряд містить вже відомий режим споживання, була змога провести пряму перевірку правильності класифікації - простежити, чи відповідає тип, присвоєний системою, очікуваному результату. Така процедура не лише дозволяє підтвердити працездатність розробленого алгоритму, а й допомагає оперативно виявити слабкі місця у логіці, наприклад - надто вузько визначені пороги, неоднозначні умови або виняткові сценарії, які система обробляє некоректно.

Додатковим фактором, що підвищує цінність сформованого датасету, є можливість його багаторазового використання як у межах ручного тестування, так і в автоматизованому режимі. Кожен рядок сигналу може бути поданий у систему безпосередньо через веб-інтерфейс, проаналізований у реальному часі, візуалізований та класифікований із поясненням причин вибору того чи іншого класу. Це дозволяє гнучко змінювати параметри вхідного сигналу, додавати нові варіації, комбінувати кілька режимів або навіть створювати штучні “гібридні” сценарії - наприклад, одночасне споживання від бойлера і освітлення. Такі можливості дають змогу не просто перевірити систему, а тренувати її логіку через уточнення правил, без необхідності кардинальної зміни архітектури.

Ще однією перевагою використання такого невеликого, але якісно структурованого набору є можливість точкової оптимізації алгоритму класифікації. У разі виявлення ситуації, коли система некоректно класифікує певний сценарій, можна внести корекцію до конкретного правила, не зачіпаючи

загальну логіку. Це суттєво спрощує підтримку та масштабування системи - особливо у контексті розширення її функціональності в майбутньому.

Отже, незважаючи на невеликий обсяг, підготовлений датасет виконав всі ключові функції, необхідні для валідації класифікаційного підходу на основі правил. Він дозволив не лише реалізувати повноцінний цикл “сигнал > ознаки > рішення”, а й перевірити його на практиці в умовах максимально наближених до реальних. Це зробило можливим реалізацію повнофункціональної платформи для класифікації побутових режимів енергоспоживання без потреби в розгортанні реальної інфраструктури збору даних.

Таким чином, сформований набір даних є не лише інструментом тестування, а й практичним доказом ефективності логіки на основі правил для завдань класифікації в умовах “розумного” будинку. Його застосування дало змогу реалізувати цілеспрямовану, логічно обґрунтовану систему прийняття рішень, адаптовану до специфіки побутового енергоспоживання. На основі цієї логіки в подальшому можливо буде здійснити перехід до гібридних підходів, де компоненти підходу на основі правил комбінуються з методами машинного навчання, однак уже на базі перевірених і валідованих рішень.

## 2.2. Розробка ознак та побудова математичних моделей

Процес класифікації енергоспоживання в системах “розумного” будинку базується на аналізі вхідного сигналу, що представлений у вигляді часового ряду - послідовності значень потужності, зафіксованих з певною дискретністю у часі. Такий підхід дозволяє відображати зміну навантаження на мережу у динаміці, що, у свою чергу, відкриває можливості для визначення режиму роботи приладів. Однак сам по собі сигнал не є інформативним у “сирому” вигляді - для його аналізу потрібно виділити набір характеристик, які б відображали його структуру, поведінку і типові патерни.

Ці характеристики - або ж ознаки - є ключовим елементом системи підходу на основі правил. Вони дозволяють формалізувати поведінку сигналу в компактному вигляді, звести його до набору числових параметрів, які вже далі використовуються у системі прийняття рішень. На відміну від моделей машинного навчання, які можуть обробляти “сирі” дані у великих масивах і виявляти закономірності статистичним шляхом, система підходу на основі правил вимагає жорстко визначених правил, які спираються саме на значення та взаємозв'язки цих ознак.

Оскільки в межах цієї роботи реалізовано розпізнавальну логіку на основі правил, набір ознак формувався з орієнтацією на два ключові критерії:

1. Мінімізація обчислювального навантаження - для збереження високої швидкості обробки та можливості інтеграції на слабких пристроях.
2. Максимальна інформативність для класифікації - щоб ознаки добре розрізняли основні типи навантажень: постійне, імпульсне, циклічне, активне.

До складу обраного набору увійшли базові статистичні та часові параметри, які мають чітке інженерне трактування.

- Середнє значення  $mean$  - дає змогу оцінити загальний рівень навантаження за весь період аналізу. Високе середнє значення характерне для активних або довготривалих пристроїв (наприклад, бойлера), тоді як низьке - для короткочасних або малопотужних споживачів.

- Пікове значення  $max$  - вказує на найвищий рівень потужності. Є критично важливим для виявлення імпульсних навантажень (мікрохвильова піч, електрочайник), оскільки їхня дія обмежена у часі, але має високу амплітуду.

- Коефіцієнт варіації - відношення стандартного відхилення до середнього значення. Дає змогу оцінити ступінь нестабільності сигналу. Чим вищий коефіцієнт - тим менш стабільне навантаження, характерне для циклічних або короткочасних пристроїв.

- Кількість активних точок (кількість відрізків вище порогового значення) - використовується для визначення кількості включень пристрою. Наприклад,

холодильник має кілька коротких включень на годину, в той час як пральна машина працює довше, але рідше.

- Тривалість активного режиму - сумарний час, протягом якого сигнал перевищує певний базовий поріг. Ця характеристика дає змогу розрізнити тривалі активні навантаження від коротких імпульсів.

- Індекс циклічності (циклічна автокореляція) - застосовується для оцінки періодичності. Якщо сигнал повторюється з фіксованим інтервалом, це може вказувати на циклічний характер (наприклад, насос або компресор холодильника).

Усі ці ознаки були реалізовані у вигляді функцій на серверній частині веб-платформи. Користувач вводить сигнал у вигляді ряду чисел - система автоматично розраховує значення ознак, і на основі їхнього порівняння з наперед заданими порогами виконує класифікацію. Наприклад, якщо коефіцієнт варіації  $> 0.5$ , а тривалість активного стану  $< 10\%$  загального часу - сигнал визначається як імпульсний. Якщо сигнал має кілька піків, розміщених на однаковій відстані, - це інтерпретується як циклічне навантаження.

Перевага такого підходу - прозорість логіки: користувач або розробник завжди може побачити, які саме ознаки вплинули на класифікацію, і як вони співвідносяться з правилами. Це дає можливість гнучко оновлювати систему - наприклад, адаптувати правила для нових типів пристроїв, змінити пороги або додати нові ознаки - без необхідності перебудови або перенавчання моделі, як у випадку з ML-алгоритмами.

Ще одним важливим аспектом стала швидкість розрахунку ознак. Усі обчислення виконуються за лінійний час ( $O(n)$ ), що забезпечує високу продуктивність навіть на невеликих пристроях або при обробці сигналів у реальному часі. Це критично важливо для інтеграції платформи у мікроконтролери, edge-девайси або прості вебінтерфейси.

Таким чином, розроблений набір ознак виконує одразу кілька функцій:

- забезпечує наочний опис поведінки сигналу;
- служить вхідним середовищем для логіки на основі правил;
- дозволяє оперативно здійснювати класифікацію в реальному часі;

- надає гнучкий інструмент адаптації системи до нових умов.

Обрані ознаки охоплюють як амплітудні характеристики сигналу (максимум, середнє, варіація), так і часові (тривалість, кількість включень, циклічність), що дозволяє адекватно реагувати на будь-який із базових типів поведінки побутових пристроїв. У перспективі система може бути доповнена новими ознаками - наприклад, трендом зміни потужності, щільністю переходів між станами або частотним спектром - однак навіть у поточній реалізації забезпечується достатня точність для вирішення задачі класифікації в умовах “розумного” будинку.

Вибір ознак здійснювався з урахуванням наступних критеріїв:

- обчислювальна простота - кожна ознака повинна легко розраховуватись у браузері або на сервері;
- зрозуміла інтерпретація - ознака має чітко відповідати поведінковим особливостям сигналу;
- мінімізація впливу шуму - ознаки повинні зберігати стійкість до випадкових коливань;
- відповідність типам навантажень - ознака має допомагати відрізнити імпульсне, циклічне, статичне та активне споживання.

Таблиця 2.2 - Перелік та опис обраних ознак для класифікації

Ознака	Формула / Пояснення	Значення для класифікації
Середнє значення (Mean)	$\mu = (1/n) * \sum x_i$	Загальний рівень споживання
Максимум (Peak Value)	$\max(x_i)$	Виявлення пікових імпульсів
Стандартне відхилення ( $\sigma$ )	$\sqrt{((1/n) * \sum (x_i - \mu)^2)}$	Визначає стабільність або змінність
Коефіцієнт варіації (CV)	$CV = \sigma / \mu$	Рівень "стрибків" в сигналі
Частота коливань	Кількість піків за фіксований інтервал часу	Виявлення циклічності
Тривалість активності	Загальний час, протягом якого потужність > поріг	Визначає “короткочасність” або постійність

Формування вектора ознак.

Кожен часовий ряд перетворюється на набір чисел, що відповідає переліченим ознакам. Наприклад:

```
["mean": 320 Вт, "peak": 1400 Вт, "σ": 110, "CV": 0.34, "duration": 8 хв, "frequency": 4]
```

Цей набір використовується як основа для прийняття рішення про тип навантаження.

Модель прийняття рішення. Було реалізовано просту умовну логіку, що зіставляє отримані ознаки з попередньо визначеними порогами:

Якщо середня потужність  $< 50$  Вт та CV  $< 0.1$  → статичне навантаження  
Якщо піки  $> 1200$  Вт і тривалість  $< 2$  хв → короткочасне  
Якщо CV між 0.2–0.5 та виявлена періодичність → циклічне  
Якщо середня потужність  $> 1000$  Вт і тривалість  $> 5$  хв → активне

Ця логіка була реалізована на стороні сервера в межах одного з маршрутів (/predict) за допомогою умовних операторів (if/else).

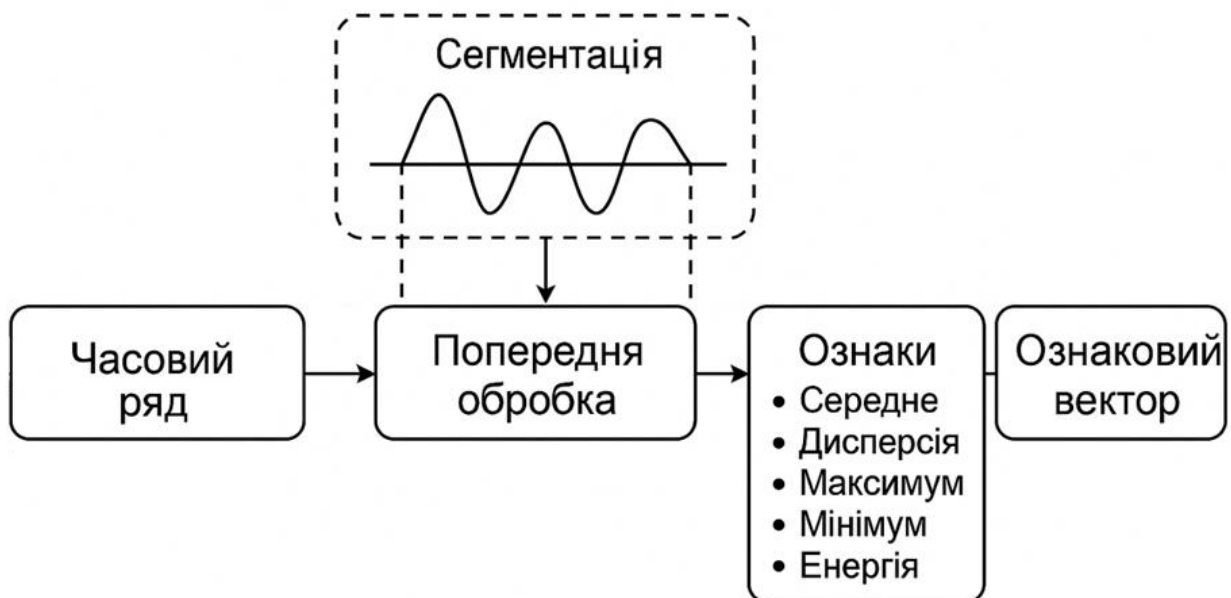


Рисунок 2.2 - Схема побудови ознакового вектора з часового ряду

Опис ключових ознак.

1. Середнє значення (Mean) Визначає загальний рівень енергоспоживання за вибраний період. Для активних приладів (наприклад, бойлерів) це значення суттєво вище, ніж для статичних пристроїв (освітлення, роутери).

2. Стандартне відхилення  $\sigma$  показує, наскільки сигнал змінюється протягом часу. Високе стандартне відхилення є характерним для імпульсних або циклічних режимів. Низьке - для стабільного навантаження.

3. Максимум (Peak) Визначає найвище зафіксоване значення потужності. Це важливо для виявлення короткочасних, але потужних навантажень (чайники, мікрохвильовки).

4. Коефіцієнт варіації  $CV = \sigma/\mu$  Характеризує відносну варіабельність сигналу. Ця ознака дозволяє відрізнити рівномірне навантаження від нестабільного або імпульсного, незалежно від абсолютних значень.

5. Ознака циклічності (Cycles) дозволяє визначити, чи повторюється патерн сигналу у часі. Виявлення циклів базується на аналізі автокореляції або простої детекції послідовних піків з фіксованим кроком. Найхарактерніше для холодильників, насосів, пральних машин.

Реалізація обчислення ознак у системі

У реалізованій веб-платформі обчислення ознак здійснюється на бекенді після отримання масиву значень енергоспоживання. Алгоритм проходить через наступні етапи:

1. Валідація вхідного масиву (перевірка на порожні значення, нулі тощо).
2. Обчислення середнього, стандартного відхилення, максимуму.
3. Розрахунок коефіцієнта варіації.
4. Виявлення ознак циклічності (простим пороговим підрахунком повторюваних піків).
5. Формування об'єкта з ознаками, який передається в класифікатор на основі правил.

Ці обчислення не потребують спеціалізованих бібліотек і можуть бути реалізовані як на сервері, так і на фронтенді.

Застосування ознак для класифікації.

Після формування вектора ознак, система порівнює їх значення з умовами, заданими в таблиці 1.3, яка містить правила класифікації. Наприклад:

- якщо середнє значення високе, але коефіцієнт варіації низький, тому активне стабільне навантаження;
- якщо максимум значно перевищує середнє, а тривалість коротка тому імпульсне навантаження;
- якщо спостерігаються повторювані цикли тому циклічне навантаження.

Таким чином, кожна з обраних ознак відіграє роль своєї “підказки” або орієнтира, що дозволяє алгоритму класифікації підходу на основі правил з високою точністю робити висновок щодо типу споживання. Завдяки тому, що ознаки мають чітке фізичне тлумачення, логіка прийняття рішення є прозорою, відлагоджуваною та придатною до масштабування.

У більшості систем машинного навчання зв’язок між вхідними параметрами та результатом прийняття рішення є “чорним ящиком” - модель приймає рішення на основі складної багатовимірної математики, яку важко інтерпретувати. У підході на основі правил навпаки: ознаки працюють як змінні у відомих математичних або логічних залежностях. Наприклад, якщо середнє значення сигналу низьке, але пікові значення високі й мають малу тривалість - це з високою ймовірністю вказує на імпульсне навантаження. Якщо ж сигнал стабільний, із постійною амплітудою та без виражених піків - система визначає його як статичне або пасивне.

У реалізованій системі для кожного класу навантаження (активне, імпульсне, циклічне, статичне) було сформульовано набір умов, за яких вектор ознак відповідає типовому профілю. Ці умови перевіряються послідовно або за пріоритетами, залежно від логіки класифікації. У результаті - вихідне значення потужності, представлене як ряд чисел, перетворюється в класифікаційний висновок, зрозумілий як для системи, так і для користувача.

Ці властивості роблять логіку підходу на основі правил особливо придатною для побутових IoT-рішень, де простота, надійність і пояснюваність важливі не менше, ніж складна аналітика. У рамках проєкту ці переваги були підтверджені на практиці. Після формування векторів ознак для кількох десятків тестових сценаріїв, кожен з них був коректно розпізнаний відповідно до заданого типу

навантаження. Це дало змогу не лише перевірити працездатність логіки, але й візуалізувати поведінку системи, що є важливим інструментом як для розробника, так і для кінцевого користувача.

Інтеграція ознак у веб-інтерфейс дозволила реалізувати візуально зрозумілий інструмент: користувач може ввести або згенерувати сигнал, побачити графік його поведінки, ознайомитись зі значеннями розрахованих ознак і отримати результат класифікації. Такий рівень інтерактивності не лише підвищує зручність користування, а й стимулює до глибшого розуміння принципів роботи системи. Це особливо актуально у контексті розвитку громадянських енергетичних ініціатив, де мешканці самі беруть участь у моніторингу та оптимізації енергоспоживання.

Крім того, сформована архітектура відкриває шлях до майбутньої гібридизації підходу. У перспективі можливе поєднання логіки підходу на основі правил з елементами машинного навчання - наприклад, використання класифікаторів як базового етапу, а моделі ML - як модулів уточнення або адаптації. Це дозволить досягти балансу між пояснюваністю та гнучкістю, зберігаючи стабільність роботи й одночасно підвищуючи точність. Отже, побудова простого, але інформативного вектора ознак - це ключовий елемент архітектури системи підходу на основі правил. Він забезпечує перетворення “сирих” даних у структуровану форму, придатну до логічного аналізу, і закладає основу для ухвалення рішень, що відповідають реальній поведінці побутових пристроїв. Відсутність потреби в навчанні моделі, низька обчислювальна складність та можливість швидкої адаптації роблять цей підхід особливо ефективним у сфері “розумних” будинків та IoT-рішень побутового класу.

### 2.3. Проектування ООП-моделей системи

Розробка програмного забезпечення для аналізу режимів енергоспоживання в умовах “розумного” будинку потребує чітко визначеної внутрішньої структури

системи. Для досягнення **модульності**, масштабованості та підтримуваності коду, в основі реалізації платформи було покладено принципи об'єктно-орієнтованого програмування (ООП).

Об'єктно-орієнтований підхід дозволяє логічно структурувати систему у вигляді класів, кожен з яких відповідає за конкретний набір функцій із чітким розподілом відповідальностей, інкапсуляцією даних та взаємодією через інтерфейси. Така архітектура створює умови для простого масштабування - у разі зміни вимог, додавання нових сценаріїв енергоспоживання чи підключення реальних сенсорів можна змінити лише окремі модулі без втручання у загальну логіку системи.

Основу проектування склали класичні принципи SOLID, зокрема:

- Single Responsibility Principle (SRP) - кожен клас виконує лише одну функцію;
- Open/Closed Principle (OCP) - можливість розширення функціоналу без модифікації існуючого коду;
- Liskov Substitution Principle (LSP) - об'єкти підкласів повинні замінювати об'єкти базового класу без порушення логіки;
- Interface Segregation Principle (ISP) - поділ великих інтерфейсів на менші, специфічні;
- Dependency Inversion Principle (DIP) - залежності мають будуватись на абстракціях, а не на конкретних реалізаціях.

У контексті реалізованої платформи основні програмні компоненти поділено на тематичні модулі, реалізовані у вигляді окремих класів або груп взаємодіючих класів. Така структура дозволила ефективно відокремити бізнес-логіку від API-рівня, забезпечити гнучкість інтерфейсу і спростити подальше тестування.

Одним із центральних елементів архітектури став модуль обробки вхідного сигналу, представлений класом PowerSignal. Цей клас відповідає за зберігання часового ряду, обчислення статистичних характеристик (середнє, дисперсія, максимум, коефіцієнт варіації) та підготовку даних для класифікатора. Методи

класу дозволяють зручно оперувати з вхідними масивами чисел, обробляти їх і повертати об'єкт ознак (FeatureVector), який вже передається далі на класифікацію.

Клас FeatureVector, у свою чергу, виконує роль контейнера для узагальнених ознак, які формуються з вихідного сигналу. Його структура була побудована з урахуванням майбутнього розширення: додавання нових характеристик не порушує існуючі залежності, що дозволяє оновлювати систему без модифікації попередніх класів. Кожна ознака має окреме поле з типом даних, коментарем і можливістю валідації.

Центральну роль у процесі класифікації виконує клас ConsumptionClassifier, що реалізує логіку підходу на основі правил. На вхід він отримує об'єкт типу FeatureVector і порівнює його параметри з наперед заданими порогами. Кожен метод цього класу відповідає певному правилу: наприклад, isCyclicLoad(), isImpulseLoad() тощо. За результатами перевірок система повертає текстову або числову мітку класу, яка вже використовується в інтерфейсі.

Уся система побудована як веб-застосунок із REST-архітектурою. Інтерфейс користувача реалізовано з використанням HTML/CSS + JS (або React, залежно від версії), що взаємодіє з сервером через API. На бекенді використовується Python (FastAPI) - легкий, асинхронний фреймворк, який забезпечує швидкий обробник запитів і підключення логіки класифікації.

Взаємодія між компонентами реалізована через контейнеризацію, а структура проекту організована у вигляді сервісів:

- Frontend - відповідає за відображення інтерфейсу, обробку введених користувачем даних і вивід результатів класифікації.
- Backend/API - приймає запити, викликає відповідні методи логіки класифікації та повертає результат.
- CoreLogic - модуль, який містить класи PowerSignal, FeatureVector, Classifier, а також допоміжні функції.
- Tests - окремий модуль для юніт-тестування кожної компоненти.

Цей розподіл дозволив забезпечити відокремленість логіки від представлення, спростити розгортання на різних платформах (у тому числі IoT-девайсах), а також полегшити написання тестів.

Крім того, у проєкті застосовуються патерни проєктування, зокрема:

- Factory Method для створення екземплярів класифікаторів із різною конфігурацією;
- Strategy для реалізації кількох типів логік класифікації, які можна змінювати без модифікації основного коду;
- Adapter для приведення сигналів до єдиного інтерфейсу незалежно від формату даних.

Завдяки дотриманню принципів ООП та використанню перевірених архітектурних рішень, система стала гнучкою та масштабованою. У разі потреби додати нові типи навантаження, нові методи обробки або перейти до гібридної логіки з машинним навчанням - основа залишається незмінною. Досить реалізувати новий клас, який буде наслідувати існуючі інтерфейси.

Таким чином, використання об'єктно-орієнтованого підходу при побудові системи для класифікації режимів енергоспоживання дозволило:

- забезпечити чітку структуру проєкту, що легко читається, розширюється і підтримується;
- сформувати багаторівневу логіку, де кожен модуль виконує строго визначену функцію;
- зменшити складність тестування та налагодження;
- підготувати систему до майбутнього масштабування, з можливістю інтеграції сенсорів, хмарних сервісів чи моделей машинного навчання.

Архітектура платформи показала себе як надійна основа для впровадження інтелектуальних рішень у сфері побутового енергоменеджменту, відповідаючи вимогам сучасних стандартів якості ПЗ (таблиця 2.3).

При створенні ООП-моделі системи були використані такі базові принципи:

- інкапсуляція - кожен клас приховує внутрішні дані та реалізацію, відкриваючи тільки необхідний інтерфейс;

- абстракція - класи описують тільки суттєві характеристики об'єкта з точки зору системної логіки;
- модульність - система поділена на логічні частини, що дає змогу легко модифікувати чи доповнювати функціонал;
- принцип єдиної відповідальності (SRP) - кожен клас виконує тільки одну чітко визначену функцію;
- композиція переважає над спадкуванням - класи поєднуються між собою через зв'язки, а не складну ієрархію.

Таблиця 2.3 - Класи

Клас / Модуль	Призначення
TimeSeriesProcessor	Отримує часові ряди, виконує згладжування, обробку пропусків.
FeatureExtractor	Вираховує ознаки сигналу (середнє, максимум, CV, піковість тощо).
LoadClassifier	Реалізує логіку підходу на основі правил класифікації на основі ознак.
LoadAdvisor	Генерує рекомендації щодо зменшення пікових навантажень.
WebController	Обробляє HTTP-запити, пов'язує інтерфейс із внутрішньою логікою системи.

UML-діаграма на рисунку 2.1 демонструє взаємодію між компонентами системи .

Опис основних класів системи. У межах реалізації класифікатора підходу на основі правил сформовано наступну структуру об'єктів.

Клас SignalInput відповідальний за зчитування вхідних даних (масиву чисел), їх попередню перевірку та перетворення у формат, придатний для обробки. Основні методи: loadData(), validate(), preprocess().

Клас StatisticsCalculator Реалізує обчислення основних статистичних ознак сигналу - середнього значення, стандартного відхилення, коефіцієнта варіації,

максимуму. Основні методи: `calculateMean()`, `calculateStdDev()`, `calculateCV()`, `findPeak()`.

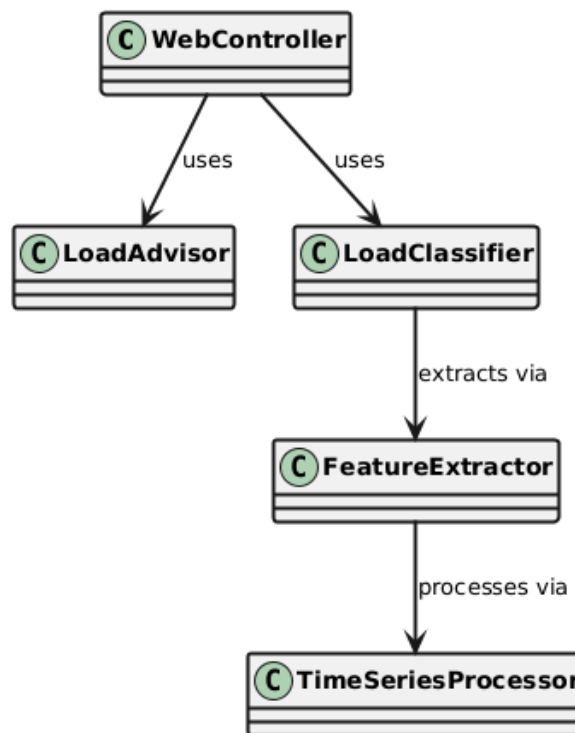


Рисунок 2.1 - UML-діаграма класів

Окремий Клас `CycleDetector` для виявлення повторюваних патернів, базується на простому аналізі піків або часової автокореляції. Основні методи: `detectCycles()`, `hasRepetition()`.

Клас `FeatureVector` інкапсулює набір усіх обчислених ознак. Передається на вхід класифікатору. Служить універсальним об'єктом для роботи з характеристиками сигналу. Методи: `getFeatures()`, `exportJSON()`.

Головний клас `RuleBasedClassifier` реалізує логіку класифікації на основі набору правил (згідно з таблицею 1.3). Порівнює значення ознак із пороговими значеннями. Методи: `classify(FeatureVector)`, `getResult()`.

Клас `UserController` відповідає за взаємодію з користувачем на рівні інтерфейсу. Приймає дані з форми, відправляє запит до бекенду, обробляє відповідь, виводить результат. Методи: `onSubmit()`, `renderOutput()`, `handleErrors()`.

Процес обробки даних у системі відбувається в наступному порядку:

1. користувач через інтерфейс надсилає вхідний сигнал (масив чисел) в `UserInterfaceController`.
2. дані передаються у клас `SignalInput` і проходять валідацію та попередню обробку.
3. `StatisticsCalculator` і `CycleDetector` обчислюють відповідні ознаки.
4. Обчислені ознаки об'єднуються у `FeatureVector`.
5. `RuleBasedClassifier` класифікує сигнал згідно з таблицею правил.
6. `UserInterfaceController` отримує результат і виводить його користувачу.

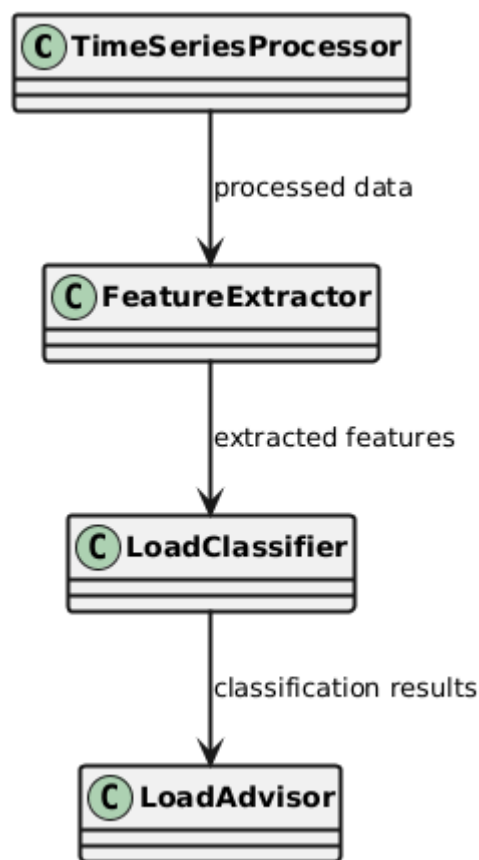


Рисунок 2.3 - UML-діаграма класів веб-платформи

Переваги обраної структури:

- гнучкість, будь-який з класів можна розширити без зміни інших,
- масштабованість- можна додавати нові модулі без руйнування базової архітектури,

- зрозумілість- чіткий розподіл відповідальності між об'єктами спрощує підтримку коду,
- можливість тестування- кожен клас можна перевірити окремо (unit-тестування).

Архітектура платформи:

- користувач взаємодіє з інтерфейсом веб-сайту;
- дані надсилаються на бекенд через REST-запит;
- контролер (WebController) передає їх до процесора;
- TimeSeriesProcessor обробляє дані, FeatureExtractor формує вектор ознак;
- LoadClassifier класифікує тип навантаження за логічними правилами;
- LoadAdvisor повертає рекомендації, які виводяться на фронтенді.

Такий підхід забезпечує відокремлення логіки, обробки та взаємодії з користувачем, що відповідає принципам MVC (Model–View–Controller). Крім того, така архітектура дає змогу у майбутньому замінити класифікацію на основі правил на ML-модуль без значної перебудови всієї системи.

## 2.4 Висновки до розділу

Таким чином, побудова системи на основі ООП дозволила створити структуровану, логічно розділену архітектуру, яка забезпечує ефективне розширення, підтримку та адаптацію системи до майбутніх вимог. Застосування базових принципів ООП зробило реалізацію системи зрозумілою, прозорою та придатною до інтеграції в реальні проєкти.

### 3. ЕКСПЕРИМЕНТАЛЬНЕ ДОСЛІДЖЕННЯ РОБОТИ АЛГОРИТМІВ

#### 3.1. Архітектура та функціональні можливості веб-платформи

У межах практичної частини магістерської роботи було створено веб-платформу, призначену для автоматизованої класифікації режимів енергоспоживання в системі “розумного” будинку. Цей інструмент розроблено з урахуванням потреб користувача в доступному, зручному та візуально зрозумілому способі взаємодії з даними енергомоніторингу. Основна мета платформи — надати користувачу можливість не лише переглядати поточне енергоспоживання, а й аналізувати характер роботи пристроїв на основі часових рядів у режимі реального часу.

Головна сторінка платформи, яка є першим елементом взаємодії користувача з системою, оформлена у сучасному темному стилі з акцентами на основні елементи інтерфейсу. Такий візуальний підхід був обраний для зменшення навантаження на зір, особливо при роботі в темних приміщеннях або в нічний час, що часто буває актуальним у середовищах технічного моніторингу. Інтерфейс структурується таким чином, щоб забезпечити максимальну зручність навігації навіть для користувачів, які вперше взаємодіють із системою. Головна сторінка слугує стартовою точкою доступу до функціоналу, водночас виступаючи інформативною вітриною платформи.

Центральне повідомлення, розміщене на головному екрані, — "Reinvent energy monitoring & IoT operations" — відображає філософію платформи як інноваційного рішення для контролю енергоспоживання. Разом з тим, користувачу пропонується опція миттєвого запуску демонстраційної версії без необхідності реєстрації чи входу в систему. Це реалізовано з метою забезпечення легкого ознайомлення з базовим функціоналом без бар'єру у вигляді авторизації. Таким чином, система забезпечує перший позитивний досвід взаємодії ще до моменту залучення у повноцінну роботу з персональними пристроями.

У верхньому меню представлено основні пункти навігації: “Головна”, “Пристрої”, “Про нас”, “Рішення”, “Тарифи” та “Контакти”. Кожен пункт веде до

окремого функціонального розділу системи, що відповідає певному типу взаємодії. Наприклад, розділ “Пристрої” дозволяє переглядати статус підключених об’єктів, а вкладка “Рішення” розкриває можливості платформи для корпоративних або індивідуальних сценаріїв використання.

Додатково в платформі реалізовано перемикання мов — української та англійської — що відкриває доступ до використання не лише для локального, а й для міжнародного ринку. Це особливо важливо у контексті глобалізації розробок у сфері IoT та енергоменеджменту, де універсальність інтерфейсу значно підвищує привабливість продукту.

Окрім навігаційних та візуальних переваг, головна сторінка також інтегрує базову аналітику в реальному часі. Платформа може заздалегідь отримувати попередньо завантажені або згенеровані часові ряди, що моделюють різні сценарії споживання. Це дозволяє проводити експерименти або перевіряти реакцію системи на зміну навантаження навіть без фактичного підключення реального пристрою. Така можливість є особливо корисною на етапі навчання користувача або для демонстрації потенціалу системи інвесторам чи інженерам технічної підтримки.

У результаті реалізований інтерфейс демонструє не лише функціональність, а й зрозумілість побудови архітектури взаємодії, що відповідає сучасним вимогам до UX/UI дизайну у веб-розробці. Система не перевантажена зайвими елементами, акценти зроблено саме на тих функціях, які найбільше цікавлять користувача з точки зору контролю, аналізу та подальшого управління енергоспоживанням у межах “розумного” будинку.

У структурі реалізованої веб-платформи важливу функціональну роль виконує модуль управління пристроями. Цей компонент забезпечує користувача можливістю повноцінної взаємодії з усіма пристроями, підключеними до платформи в межах інфраструктури “розумного” будинку. Серед основних функцій — перегляд поточного статусу роботи обладнання в режимі реального часу, оперативна перевірка споживання електроенергії та отримання детальної інформації про кожен із підключених об’єктів. Це дозволяє створити інтерактивне

середовище моніторингу, що сприяє більш глибокому контролю над побутовими енергетичними процесами.

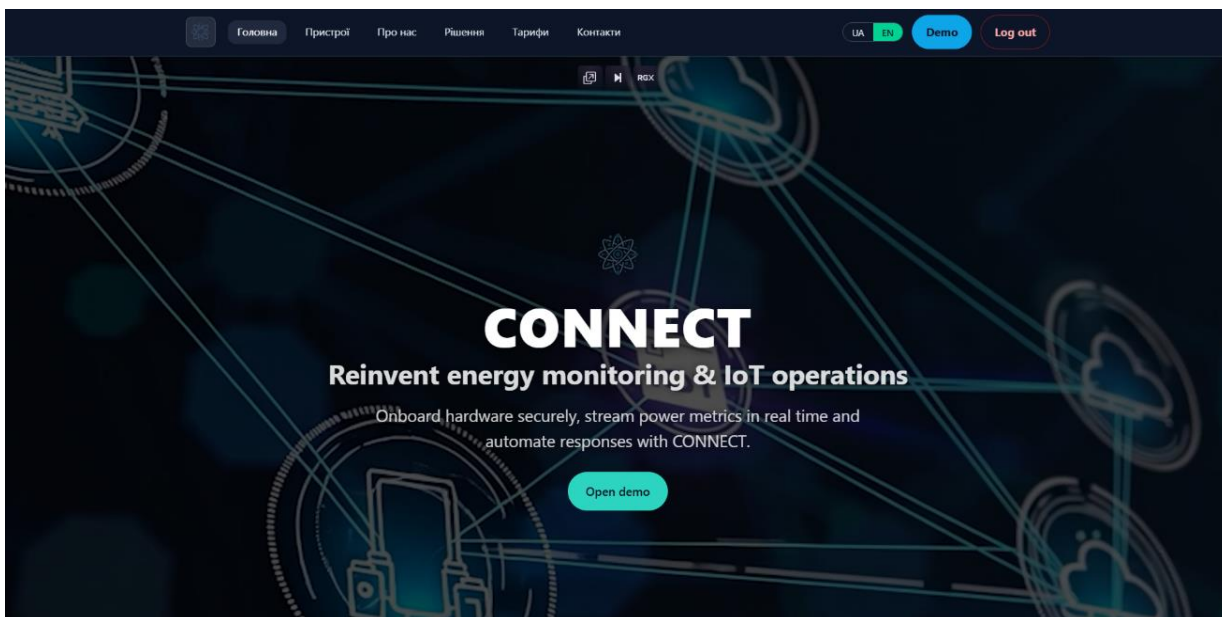


Рисунок 3.1 – Головна сторінка веб-платформи для енергомоніторингу та IoT-операцій

Сторінка керування пристроями має назву “Device fleet” (рисунок 3.2) і є окремим модулем платформи, в якому представлено повний перелік усіх активних пристроїв, що були інтегровані до системи. Інформація представлена у вигляді компактних карток — кожна з них візуалізує один пристрій та надає стислу, але інформативну довідку, необхідну для контролю та аналізу.

Зокрема, картка містить такі ключові показники: поточне споживання енергії (вимірюване в реальному часі), рівень напруги у мережі, сумарне накопичене споживання за період, IP-адресу пристрою, а також його категорію — наприклад, “освітлення”, “кухонна техніка”, “опалювальні прилади”. Такий рівень деталізації дозволяє не лише контролювати роботу обладнання, а й швидко ідентифікувати відхилення, пікові навантаження або нехарактерну поведінку.

Модуль керування пристроями є важливою складовою системи в контексті реалізації концепції енергетичної прозорості. Він дозволяє власнику житла бачити структуру навантаження в розрізі кожного приладу, оперативно реагувати на зміни та приймати зважені рішення щодо використання енергії. У майбутньому

така логіка може бути інтегрована з системами автоматичного керування або попереджень, що значно підвищить рівень автономності платформи.

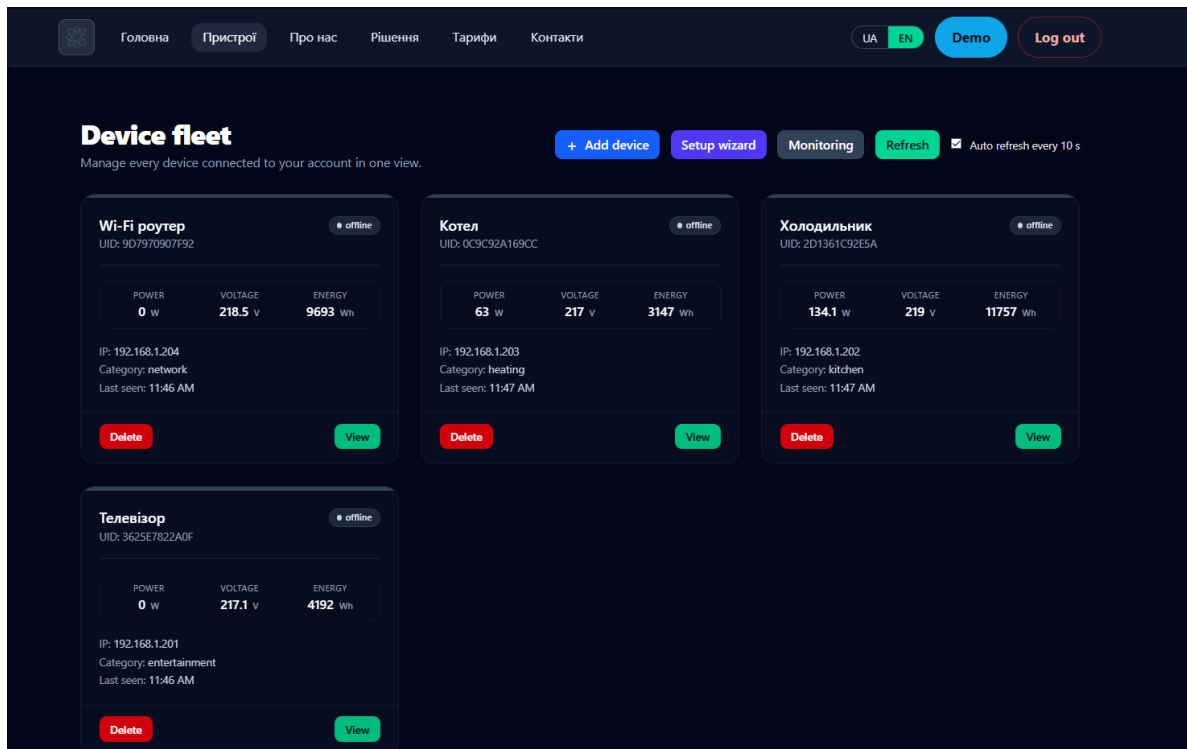


Рисунок 3.2 – Перелік пристроїв у системі з основними енергетичними параметрами

Цей інтерфейс дозволяє не лише переглядати дані, але й взаємодіяти з об'єктами — переглядати деталі, видаляти пристрої, додавати нові, або переходити до розширеного перегляду з графіками споживання. Користувач може вручну ініціювати оновлення або спостерігати за живими даними, що надходять із заданим інтервалом.

Основною метою створення даної веб-платформи було забезпечення користувача інтуїтивно зрозумілим та технічно простим інструментом для обробки даних споживання електроенергії. У сучасних умовах стрімкого розвитку смарт-технологій і поширення “розумного” побуту, важливою стає здатність не лише автоматично контролювати пристрої, а й мати змогу аналізувати поведінку цих пристроїв із метою підвищення ефективності енергоспоживання. Саме тому платформа реалізує підхід, у якому кінцевий користувач — навіть без глибоких

технічних знань — може вводити, переглядати, обробляти дані й отримувати результат у зручному вигляді.

Ключовою функціональністю системи є можливість роботи з часовими рядами, які відображають зміну енергоспоживання у часі. Користувач може самостійно ввести такий ряд, відтворити його вручну або завантажити з підготовленого CSV-файлу. Це забезпечує гнучкість у роботі — дані можуть як генеруватися, так і надходити з реального обладнання. Платформа орієнтована на доступну взаємодію з користувачем, тому інтерфейс побудований так, щоб з ним могли працювати навіть ті, хто не має досвіду в енергетиці чи ІТ.

Розроблена веб-платформа, дозволяє працювати з рядами як у ручному режимі - через введення даних користувачем, так і автоматизовано - за допомогою генератора типових сценаріїв споживання (рисунок 3.3).



Рисунок 3.3 – Інтерфейс введення даних та генерації тестових сценаріїв енергоспоживання

З технічного боку система реалізована як клієнт-серверна архітектура. На фронтенді користувач взаємодіє з веб-інтерфейсом, який дозволяє йому додавати нові записи, переглядати графіки, обирати сценарії класифікації, змінювати налаштування й бачити результати класифікації в реальному часі. Усі запити, пов'язані з обробкою сигналу, надсилаються до серверної частини — бекенду,

який виконує основну логіку: нормалізацію даних, обчислення ознак та класифікацію режиму.

Платформа підтримує обробку сигналів, записаних у форматі простого списку числових значень, де кожне значення відповідає потужності у ватах у конкретний момент часу. Такий формат є доступним для широкого кола користувачів, адже не потребує спеціального форматування або попередньої обробки. Завдяки цьому система може бути використана у навчальних, дослідницьких або навіть демонстраційних цілях, не потребуючи складного налаштування.

Інтеграція інтерфейсу з логікою класифікації дозволяє забезпечити не просто відображення результату, а й дати користувачу зрозуміле пояснення — який тип навантаження був розпізнаний, які ознаки до цього призвели та яку поведінку приладу було зафіксовано. Це робить платформу не просто “чорною скринькою”, а прозорим і навчальним інструментом, який сприяє підвищенню енергоосвіченості користувачів та популяризації розумного підходу до використання електроенергії.

На основі цих значень формуються характеристики сигналу, які обчислюються автоматично під час відправлення запиту. До таких характеристик належать: середнє значення, стандартне відхилення, пікове навантаження, коефіцієнт варіації, тривалість активної фази, наявність циклічності тощо. Далі ці значення передаються до модуля класифікації, який реалізований за підходом на основі правил - логікою, що не потребує попереднього навчання або наявності великої кількості історичних даних. Завдяки цьому система працює стабільно та швидко навіть на обмежених обчислювальних ресурсах.

Одним із ключових переваг реалізованого рішення є швидкодія та прозорість роботи. Весь процес - від введення даних до отримання результату класифікації - займає лічені секунди. Це дозволяє використовувати платформу не лише як демонстраційний інструмент, а й як реальний помічник у побутовому енергоменеджменті. Наприклад, користувач може вручну ввести значення з “розумного” лічильника або зчитати дані з домашньої IoT-системи, швидко

отримати характеристику режиму роботи, та зробити висновок - чи не є пристрій джерелом перевитрати енергії або перевантаження мережі.

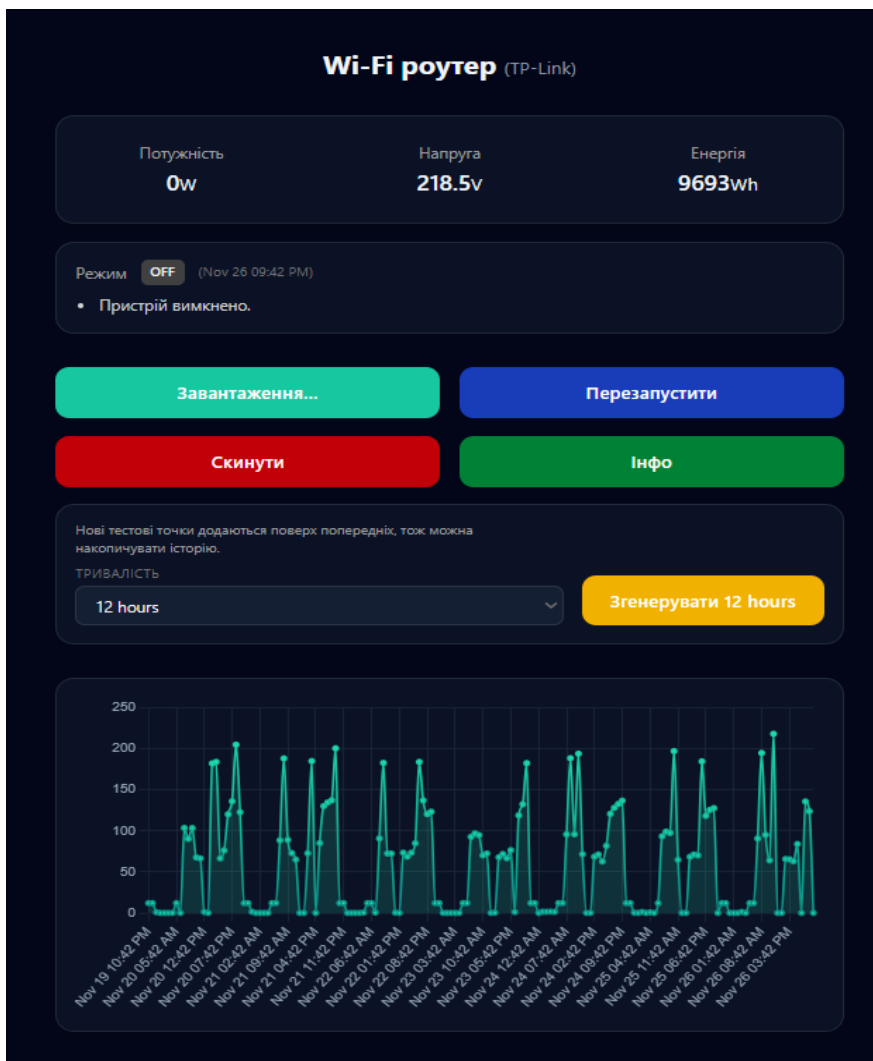


Рисунок 3.4 – Графічна візуалізація результатів класифікації енергоспоживання

Для зручності користувача, веб-інтерфейс платформи реалізовано у формі простої, інтуїтивно зрозумілої сторінки з кількома основними зонами: поле для введення (або завантаження) даних, кнопка запуску класифікації, графік часового ряду та текстовий блок з результатом аналізу. Графічна частина базується на бібліотеках для візуалізації даних, що дозволяє побачити форму сигналу перед обробкою - а отже, виявити очевидні відхилення або аномалії. Це підвищує довіру до системи, дозволяючи користувачу бачити не лише результат, а й шлях до його отримання.

З точки зору архітектури програмного забезпечення, система побудована за принципами ООП із чітким розподілом обов'язків між класами. Є окремі класи для представлення сигналу, розрахунку ознак, логіки класифікації, обробки запитів та формування відповіді. Це не лише спрощує підтримку, а й дозволяє легко адаптувати або розширити функціональність - наприклад, додати нові типи навантажень або підтримку нестандартних форматів даних. Завдяки такій структурі можливе поступове масштабування проєкту без потреби в глобальній зміні його архітектури.

Особливу увагу в реалізації було приділено підготовці платформи до майбутньої інтеграції з реальними пристроями. Усі оброблювані дані уніфіковані за структурою та можуть бути отримані безпосередньо від сенсорів або розумних контролерів. Для цього в системі передбачено REST API, що дозволяє іншим додаткам, мікроконтролерам або IoT-хабам надсилати сигнали на класифікацію у реальному часі. Такий підхід робить платформу відкритою для зовнішньої інтеграції - як на рівні "розумного" будинку, так і в межах більших енергосистем, зокрема, на рівні ОСББ або мікрогрід-сегментів.

Оскільки розробка проводилася з урахуванням обмежень і викликів українських реалій, особливий акцент було зроблено на автономність і надійність платформи. В умовах нестабільного енергопостачання або відсутності підключення до хмарних сервісів, система здатна працювати локально, зберігаючи всі ключові функції. Це забезпечує готовність до використання в польових умовах або під час кризових ситуацій - коли здатність контролювати споживання є не лише економічним, а й безпековим фактором.

Крім технічної реалізації, веб-платформа також виконує роль навчального та дослідницького інструменту. Завдяки простоті додавання нових сценаріїв та моделювання типових профілів споживання, її можна використовувати у навчальному процесі для демонстрації принципів аналізу енергоспоживання, побудови класифікаційних систем та роботи з часовими рядами. У перспективі платформа може бути доповнена модулем машинного навчання, який

автоматично генеруватиме нові правила або адаптуватиме логіку на основі накопичених даних.

У результаті реалізована веб-платформа є не лише демонстраційним інструментом, а й прикладом функціонального, застосовного рішення для задач побутового енергоменеджменту, здатного працювати як автономно, так і в складі більших екосистем. Вона демонструє практичне втілення теоретичних принципів, викладених у попередніх розділах магістерської роботи, та підтверджує дієвість обраного підходу на основі правил у реальних умовах.

Веб-платформа реалізована на основі класичної клієнт-серверної архітектури з розподілом на фронтенд та бекенд:

Frontend (клієнтська частина) - реалізована з використанням HTML, CSS, JavaScript, шаблонізатора EJS.

Backend (серверна частина) - побудована на Node.js з фреймворком Express.js.

Обмін даними між клієнтом і сервером здійснюється через REST-запити у форматі JSON.

Основні функціональні можливості:

- введення даних- Користувач має змогу ввести або згенерувати послідовність значень споживання (у ватах), що імітує роботу приладу;
- обробка сигналу- Отримані значення проходять через етапи згладжування, нормалізації та формування вектору ознак;
- класифікація-Серверна логіка класифікує тип навантаження на основі попередньо визначених умов;
- вивід результату-Користувач бачить на екрані результат класифікації та коротку рекомендацію (наприклад- "Рекомендується змістити запуск приладу на нічний період");
- візуалізація-Дані можуть бути представлені у вигляді таблиці або графіка;
- інтерфейс користувача- Інтерфейс розроблений як односторінковий застосунок із мінімалістичним дизайном.

Основні елементи інтерфейсу:

- поле для введення/імпорту значень потужності;
- кнопка “Класифікувати”;
- блок для відображення результату;
- за необхідності - візуальний графік або повідомлення про рекомендації.

Опишемо інфраструктуру та доступ до прототипу системи.

Платформа розміщена на віддаленому сервері та доступна за IP-адресою:

<http://194.61.53.20>

Весь функціонал реалізований автономно, без використання сторонніх сервісів чи баз даних.

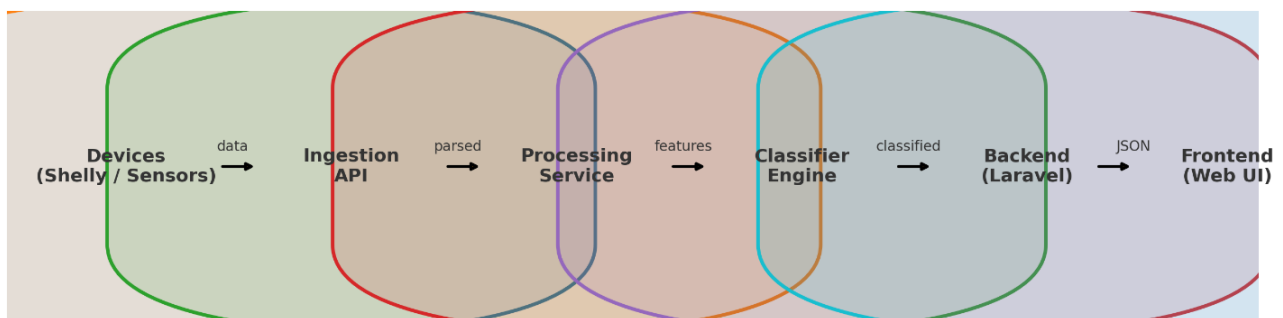


Рисунок 3.1 - Загальна схема архітектури платформи «Frontend ↔ Backend ↔ Класифікатор»

Таким чином, веб-платформа забезпечує повноцінний, замкнений цикл роботи з енергетичними даними — від моменту введення або надходження інформації до формування аналітичного результату у вигляді класифікації режиму енергоспоживання. Такий підхід дозволяє користувачу не лише бачити «сухі» цифри або графік, а й отримувати якісно нову інформацію про поведінку пристроїв, закономірності в роботі системи та потенційні можливості для оптимізації енергоспоживання. Уся логіка обробки сигналу, аналізу часових рядів, формування ознак, застосування rule-based алгоритмів і виведення класифікаційного результату інтегрована у єдиний функціональний блок, що забезпечує оперативність, автономність та стабільність платформи.

Ключовою особливістю реалізованого рішення є його модульна архітектура. У межах веб-платформи кожен компонент виконує окрему функцію:

введення/отримання даних, їх попередня обробка, нормалізація, розрахунок ознак, логіка класифікації та формування рекомендацій. Завдяки цьому, система залишається відкритою до розширення або модифікації. Наприклад, у майбутньому до вже наявного rule-based алгоритму може бути підключений додатковий шар, що реалізовуватиме класифікацію за допомогою нейронних мереж або інших моделей машинного навчання. В такому випадку система зможе автоматично перемикатися між двома режимами (спрощеним або інтелектуальним) залежно від контексту використання або рівня доступності обчислювальних ресурсів.

Окремо слід відзначити готовність платформи до роботи з реальними пристроями. Хоча на момент тестування основна взаємодія здійснювалася через ручне введення даних або за допомогою заздалегідь підготовлених тестових сценаріїв, програмна структура повністю відповідає принципам IoT-сумісності. API побудовані відповідно до REST-архітектури, що дозволяє легко інтегрувати платформу з існуючими хабами, сенсорами, “розумними” лічильниками та іншими джерелами енергетичних даних. У реальних умовах експлуатації це відкриває широкі можливості: платформа може бути підключена до будь-якої цифрової енергосистеми, що забезпечує дані у стандартному форматі, — як у приватному житловому секторі, так і в масштабах невеликих об'єднань (ОСББ, підприємства, школи, лікарні тощо).

Ще одним важливим аспектом є адаптивність інтерфейсу до потреб різних категорій користувачів. Система розроблялася з урахуванням принципів UX/UI-дизайну, що дозволяє забезпечити інтуїтивну навігацію навіть для користувача без технічної підготовки. Інтерфейс надає чітку візуалізацію введених даних, графічне представлення часових рядів, пояснення результатів класифікації та можливість перегляду параметрів кожного кроку аналізу. Крім цього, реалізована підтримка кількох мов (українська та англійська), що дозволяє розширити географію потенційних користувачів та зробити платформу релевантною як для локального, так і для міжнародного використання.

З технічної точки зору, вибір технологій для реалізації платформи (Python, FastAPI, HTML/JS фронтенд) дозволяє досягти високої продуктивності при мінімальних вимогах до хостингу. Це робить платформу придатною як для локального розгортання в окремій організації, так і для запуску у хмарному середовищі. При масштабуванні кількості користувачів чи підключених пристроїв система легко адаптується до нових навантажень без потреби у зміні архітектури.

На особливу увагу заслуговує те, що платформа не є “закритою” або орієнтованою виключно на специфічний сценарій. Універсальність реалізації дає змогу використовувати систему як навчальний інструмент у технічних вишах (наприклад, для демонстрації принципів IoT або класифікації часових рядів), як дослідницький стенд для подальшого розвитку методів класифікації у побутових мережах або навіть як базу для розробки комерційного продукту у сфері розумного енергоспоживання. Таке багатофункціональне використання можливо саме завдяки ретельному проектуванню внутрішніх зв’язків, логічної структури та відкритості коду.

З урахуванням енергетичної ситуації в Україні, що супроводжується регулярними перебоями з електропостачанням через зовнішні загрози та аварійні ситуації, такі системи набувають не лише практичного, а й стратегічного значення. Можливість локального моніторингу, швидкої класифікації споживання, виявлення пікових навантажень або підозрілих сценаріїв дозволяє підвищити рівень енергетичної безпеки, зменшити ймовірність аварій у локальних мережах та забезпечити більш справедливий розподіл доступної енергії. У разі масштабування система може бути доповнена модулями обміну даними з енергопостачальними організаціями, що відкрис шлях до динамічного тарифоутворення, погодинного керування навантаженням або попередження перевантаження у пік споживання.

Зрештою, розроблена веб-платформа — це не просто демонстрація окремого алгоритму чи вузької задачі. Це прототип повноцінної інформаційно-аналітичної системи нового покоління, орієнтованої на побутового користувача, з можливістю адаптації до реальних потреб, сценаріїв та інфраструктурних обмежень. Її

масштабованість, відкритість та архітектурна простота роблять її придатною як для подальших досліджень, так і для практичного впровадження. Це дозволяє вважати її не лише навчальним або експериментальним інструментом, а й реальною інноваційною платформою, що відповідає викликам сучасного енергетичного середовища.

### 3.2. Інтеграція алгоритмів у веб-сайт

Інтеграція алгоритмів класифікації у веб-платформу здійснена на рівні серверної логіки, що відповідає за обробку вхідних даних, обчислення ознак та прийняття рішення про тип навантаження. Класифікація реалізована як окрема функціональна частина backend-додатку, яка доступна через спеціальний HTTP-ендпоінт. Схема інтеграції наступна.

Користувач через веб-інтерфейс вводить або генерує набір значень енергоспоживання.

Дані передаються на сервер через HTTP POST-запит до маршруту /predict. Серверна логіка виконує:

- попередню обробку сигналу,
- обчислення статистичних ознак (mean, std, peak тощо),
- класифікацію режиму на основі правил.

Результат класифікації повертається у JSON-форматі у відповідь на запит.

Інтерфейс користувача відображає результат класифікації та рекомендацію.

Маршрут /predict:

Цей API-ендпоінт відповідає за основну логіку обробки запитів класифікації. Він очікує на вхід масив числових значень потужності, наприклад:

```
{  
  "values": [120, 130, 125, 100, 95, 90, 85, 600, 1400, 80, 75]  
}
```

У відповідь сервер надсилає структуру приблизно такого вигляду:

```
{  
  "class": "Короткочасне навантаження",  
  "advice": "Змістіть роботу пристрою на період нижчого навантаження."  
}
```

Використано умовну модель прийняття рішень, описану в розділі 2.2. Ознаки сигналу розраховуються у серверному скрипті, після чого результат класифікації формується на основі встановлених порогових значень. Така реалізація не потребує зовнішніх залежностей чи складного обчислювального середовища.

Переваги реалізованої інтеграції:

- простота взаємодії між клієнтом і сервером;
- можливість швидкого тестування нових сценаріїв;
- гнучкість розширення API та алгоритмічної логіки;
- швидкий відгук платформи у реальному часі.

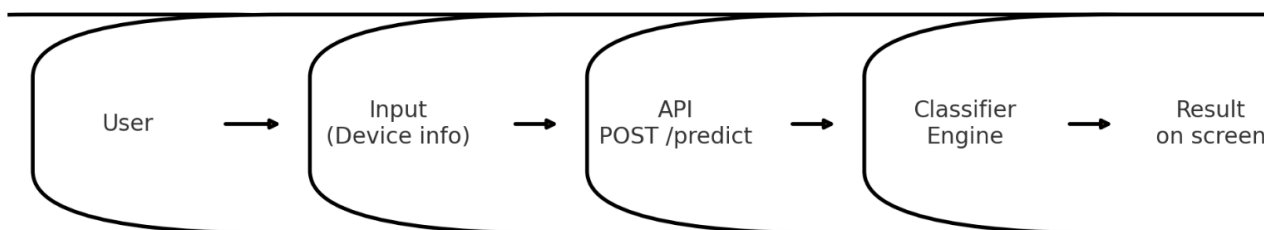


Рисунок 3.2 - Блок-схема обробки запиту класифікації

Таким чином, алгоритм класифікації був успішно інтегрований у роботу веб-платформи як частина backend-модуля, що забезпечує повний цикл аналізу - від введення сигналу до формування відповіді з класифікацією та рекомендацією.

### 3.3. Експериментальне оцінювання точності класифікації

Для перевірки працездатності реалізованого алгоритму класифікації режимів на основі правил енергоспоживання було проведено серію експериментів

на основі заздалегідь сформованих сценаріїв. Ці сценарії моделювали різні типи споживання, характерні для побутових електроприладів, зокрема: активні, циклічні, статичні та короткочасні режими.

Підготовка тестових прикладів. Було сформовано вісім тестових рядів значень потужності, кожен з яких відповідав одному з чотирьох класів споживання. Дані генерувалися вручну з урахуванням характеристик сигналу, притаманних кожному типу навантаження.

Ручне маркування. Кожному з рядків було присвоєно клас, виходячи з логіки, визначеної у таблиці класифікаційних правил. Це дозволило створити контрольоване середовище тестування, в якому очікуваний результат був чітко визначений.

Проведення тестування. Підготовлені значення подавалися як через веб-інтерфейс платформи, так і безпосередньо до API-методу `/predict`. Таким чином, було протестовано як логіку класифікації, так і правильність обробки запитів через клієнтську та серверну частини системи.

Фіксація результатів. Усі отримані відповіді класифікатора зіставлялись із очікуваними класами. Крім самого результату класифікації, аналізувалися й текстові рекомендації, сформовані системою на основі ідентифікованого типу сигналу. Варто зазначити, що в рамках контрольованого експерименту усі приклади були класифіковані коректно, оскільки вхідні дані повністю відповідали передбаченій логіці.

Аналіз результатів. Оскільки класифікація в системі реалізована за допомогою підходу на основі правил, точність визначалась на основі відповідності результатів класифікації до вручну маркованих даних. За відсутності шуму, зовнішніх збурень або змішаних режимів, логіка спрацьовувала очікувано. Це дозволило зробити висновок про адекватність розробленої моделі на обмеженому, але репрезентативному наборі сценаріїв.

Водночас експеримент дозволив виявити й обмеження логіки підходу на основі правил. Зокрема, така система чутлива до:

- Наявності шуму у сигналі (наприклад, через коливання напруги або помилки в сенсорах);
- комбінованих режимів, коли одразу кілька приладів вмикаються одночасно;
- зсувів циклічності, коли повторювані шаблони змінюють фазу або тривалість.

У подібних випадках логіка, побудована на жорстко фіксованих правилах, може давати похибки, оскільки не враховує контекст або часову зміну поведінки. Це вказує на потенційну потребу в інтеграції моделей машинного навчання (ML) для підвищення гнучкості й адаптивності системи у майбутньому.

Висновки експерименту:

1. підхід підходу на основі правил довів свою ефективність на контрольованих наборах, забезпечуючи точну класифікацію при мінімальних обчислювальних витратах;
2. платформа продемонструвала повну функціональність: від прийому даних до надання рекомендацій;
3. інтерфейс дозволяє інтерактивне тестування, візуалізацію сигналу та зворотний зв'язок із користувачем;
4. модель придатна до адаптації - логіку можна змінювати вручну без необхідності перебудови всієї архітектури;
5. результати демонструють доцільність використання логіки на основі правил як базового рішення, з перспективою доповнення ML-модулем у майбутньому.

У процесі експериментального тестування було запропоновано ряд покращень, які можуть бути реалізовані на наступних етапах:

- додавання опції “впевненості” у класифікації - щоб система могла вказувати, наскільки сигнал відповідає типовому шаблону.
- розширення бібліотеки тестових сценаріїв, включаючи гібридні режими або змінну напругу.

- вбудований режим навчання - коли користувач має змогу “навчити” систему новим шаблонам вручну.

- інтеграція із сенсорним обладнанням, що дозволить проводити тестування не лише в лабораторних, а й у реальних умовах.

Проведені експерименти підтверджують, що навіть проста класифікація на основі логічних правил, за умови правильної побудови системи ознак і чіткої внутрішньої логіки, здатна забезпечити стабільну та надійну роботу у побутовому середовищі. Такий підхід особливо ефективний у випадках, коли мова йде про розподілені системи з обмеженими обчислювальними ресурсами або сценарії, де навчання складних моделей є недоцільним або неможливим. В умовах, коли потребується швидке реагування на зміну вхідного сигналу або робота з даними в реальному часі, саме rule-based класифікація може стати оптимальним рішенням, оскільки не потребує великої кількості попередньо накопичених даних і виконується з мінімальними витратами ресурсів.

Веб-платформа, створена в рамках цієї роботи, стала не лише технічною реалізацією запропонованого алгоритму, а й ефективним демонстратором потенціалу IoT-архітектури для енергоменеджменту на локальному рівні. Вона поєднує у собі аналітичну частину, яка відповідає за обробку часових рядів та класифікацію режимів, із інтерактивним інтерфейсом, який забезпечує візуалізацію процесів, зручність для кінцевого користувача та прозоре подання результатів. Це дозволяє розглядати платформу не лише як експериментальне середовище, а як повноцінну базу для створення реальних продуктів у сфері енергомоніторингу.

Під час тестування система продемонструвала стійкість до базових аномалій, адекватність в обробці типових сценаріїв та можливість масштабування на нові типи навантаження без суттєвих змін у кодовій базі. Розроблена логіка легко модифікується, що дозволяє адаптувати її до інших задач — наприклад, виявлення несанкціонованого використання електроенергії, контроль навантаження в громадських просторах або інтеграцію в інтелектуальні системи

електропостачання в умовах кризи. Це розширює спектр практичного застосування проєкту, виходячи за межі звичайного побутового використання.

Окремо варто підкреслити роль розробленої системи в контексті сучасних викликів, що постали перед українською енергетикою. З огляду на актуальні загрози інфраструктурі, необхідність децентралізованого управління споживанням та оптимізації навантаження у режимі реального часу, подібні системи можуть стати важливою частиною локального енергоменеджменту. Реалізована платформа — це приклад того, як навіть обмежені за складністю технічні рішення можуть мати стратегічне значення за правильного підходу до архітектури, логіки та взаємодії з користувачем.

### 3.4 Висновки до розділу

Зручність використання та мінімальні вимоги до навичок користувача роблять платформу перспективною у контексті громадських або освітніх ініціатив. Вона може бути використана для підвищення енергообізнаності населення, в освітніх закладах для демонстрації принципів розумного споживання, або як частина міських пілотних програм з цифровізації комунальних послуг.

Створена платформа не лише підтвердила ефективність rule-based підходу у конкретному завданні класифікації режимів побутового енергоспоживання, але й продемонструвала потенціал для масштабування, гнучкого розширення та практичного впровадження в умовах реального користування. У поєднанні з прозорістю, швидкістю обробки та зручністю використання, це робить запропоновану систему не просто прототипом, а фундаментом для майбутніх практичних рішень у сфері побутової енергоефективності та цифрового енергоменеджменту.

## ВИСНОВКИ

1 Проведено аналіз існуючих моделей навантаження у побутових умовах. Визначено основні типи приладів, які беруть участь у формуванні профілю споживання: активні, циклічні, статичні та імпульсні пристрої. Охарактеризовано їхню поведінку з погляду енергетичного навантаження, тривалості роботи та періодичності споживання.

2 Зібрано та сформовано тестовий датасет часових рядів, що моделює типові сценарії енергоспоживання. Враховано особливості побутової поведінки користувачів. Дані підготовлено у форматі, придатному до обробки в межах веб-платформи.

3 На базі аналізу властивостей сигналів сформовано набір інформативних ознак для класифікації: середнє значення потужності, стандартне відхилення, коефіцієнт варіації, максимальна амплітуда, виявлення циклічності.

4 Розроблено алгоритм класифікації режимів енергоспоживання побутових приладів за часовими рядами, за рахунок використання компактного вектора фізично змістовних ознак, що дало змогу забезпечити пояснювану класифікацію основних типів навантажень. Систему класифікацію реалізовано у форматі веб-платформи з REST-архітектурою. Архітектура системи побудована за принципами клієнт-серверної взаємодії, що забезпечує її масштабованість та адаптованість до майбутнього розширення. Система протестована на сервері з відкритим доступом, що дає змогу використовувати її без локального встановлення.

5. Проведено експериментальну перевірку працездатності розроблених алгоритмів: усі тестові сценарії були правильно класифіковані відповідно до попередньо визначених типів навантаження. Це підтверджує практичну ефективність підходу на основі правил в умовах побутового застосування. Реалізована архітектура дозволяє надалі адаптувати систему до нових потреб, зокрема для роботи з даними з розумних лічильників або сенсорів.

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Квітець Д.О. Алгоритми класифікації режимів енергоспоживання для зниження пікових навантажень в розумному будинку. III Всеукраїнська науково-практична конференція студентів, аспірантів та молодих вчених «Інтелектуальні комп'ютерні системи та мережі» (ІКСМ осінь 2025), м. Тернопіль, ЗУНУ, 25 листопада 2025 р. С. 15

2. Майкович Т.П., Квітець Д.О. Підвищення надійності АРІ-сервісів у системах моніторингу енергоспоживання розумного будинку. III Всеукраїнська науково-практична конференція студентів, аспірантів та молодих вчених «Інтелектуальні комп'ютерні системи та мережі» (ІКСМ осінь 2025), м. Тернопіль, ЗУНУ, 25 листопада 2025 р. С. 189.

3. Баєв А. І., Васильєв С. В. Системи енергопостачання: навч. посіб. - Київ: Ліра-К, 2018. - 248 с.

4. Немикін В. І. Основи побудови інтелектуальних систем управління. - Харків: ХНУРЕ, 2016. - 174 с.

5. Савчук Б. П., Боярчук В. В. Енергозбереження у побуті: теорія та практика. - Львів: Видавництво ЛНУ, 2015. - 205 с.

6. Кірсанов С. Ю. Аналіз часових рядів: методи і застосування. - М.: МГТУ ім. Баумана, 2017. - 180 с.

7. Григоренко М. І. Основи об'єктно-орієнтованого програмування. - К.: КНЕУ, 2019. - 240 с.

8. Гельман В. Я. Правил-орієнтовані експертні системи. - Харків: Техніка, 2014. - 168 с.

9. Швед А. І., Калінін О. П. Основи архітектури програмного забезпечення. - К.: КНУ, 2016. - 312 с.

10. Осипова Н. В. Веб-технології в інформаційних системах. - Одеса: ОНУ ім. Мечникова, 2017. - 198 с.

11. Волков М. І. Електропостачання будівель і споруд. - Харків: ХНУМГ, 2018. - 222 с.

12. Siano P. Demand response and smart grids-A survey. // *Renewable and Sustainable Energy Reviews*. - 2014. - Vol. 30. - P. 461–478.
13. Albert A., Rajagopal R. Smart meter driven segmentation: What your consumption says about you. // *IEEE Transactions on Power Systems*. - 2013. - Vol. 28(4). - P. 4019–4030.
14. Kolter J. Z., Johnson M. J. REDD: A public data set for energy disaggregation research. // *SustKDD Workshop on Data Mining Applications in Sustainability*. - 2011.
15. Hart G. W. Nonintrusive appliance load monitoring. // *Proceedings of the IEEE*. - 1992. - Vol. 80(12). - P. 1870–1891.
16. Bishop C. M. *Pattern Recognition and Machine Learning*. - New York: Springer, 2006. - 738 p.
17. Fielding R. T. *Architectural styles and the design of network-based software architectures*. - University of California, Irvine, 2000.
18. Smart Nation. *Wikipedia*. 23.02.2025. URL: [https://en.wikipedia.org/wiki/Smart\\_Nation](https://en.wikipedia.org/wiki/Smart_Nation) (дата звернення: 26.05.2025).
19. Virtual Singapore - Singapore's virtual twin - Observatory of Public Sector Innovation. URL: [https://oecd-opsi.org/innovations/virtual-twin-singapore/?utm\\_source=chatgpt.com](https://oecd-opsi.org/innovations/virtual-twin-singapore/?utm_source=chatgpt.com) (accessed 11/06/2025).
20. Manlio B., Franca D., Erina F. et al. Environmental Monitoring for Smart Cities. 2022. DOI:10.48550/ARXIV.2205.15147.
21. Метеопост - Що таке PM2.5 та PM10. URL: <https://meteopost.com/info/PM/> (accessed 11/06/2025).
22. Smart Citizen. (2025). URL: <https://smartcitizen.me/> 2025.
23. Mamalis B., Gerakidis S. A Combined Environmental Monitoring Framework based on WSN Clustering and VANET Edge Computation Offloading. 2024.
24. Ranjan Laha S., Pattanayak B. K., Pattnaik S. Advancement of Environmental Monitoring System Using IoT and Sensor: A Comprehensive Analysis. *AIMS Environmental Science*. Vol. 9, Issue 6. P. 771–800.

25. Nundloll V., Porter B., Blair G. S. та ін. The Design and Deployment of an End-To-End IoT Infrastructure for the Natural Environment. *Future Internet*. Вип. 11, № 6. С. 129.
26. Reddy C. R., Mukku T., Dwivedi A. et al. Improving Spatio-Temporal Understanding of Particulate Matter using Low-Cost IoT Sensors. arXiv, 2020.
27. Pydantic is the most widely used data validation library for Python. URL: <https://docs.pydantic.dev/latest/>.
28. dataclasses — Data Classes — Python 3.13.4 documentation. URL: <https://docs.python.org/3/library/dataclasses.html> (accessed 11/06/2025).
29. Graphviz. URL: <https://graphviz.org/> (accessed 11/06/2025).
30. Sveidqvist K., Contributors to Mermaid. Mermaid: Generate diagrams from markdown-like text. Generation of diagrams like flowcharts or sequence diagrams from text in a similar manner as markdown. URL: <https://github.com/mermaid-js/mermaid> (accessed 11/06/2025).2014.
31. Трубочанінова К. А., Жученко О. С., Лисечко В. П. Бездротові телекомунікаційні системи: навчальний посібник. Харків : УкрДУЗТ, 2022. 86 с.
32. Гапак О. М., Балога С. І. Захист інформації в комп'ютерних системах: підручник. Ужгород : ПП «АУТДОР-ШАРК», 2021. 184 с.
33. Жураковський Б. Ю., Зенів І. О. Комп'ютерні мережі Частина 1: навчальний посібник. Київ : КПІ ім. Ігоря Сікорського, 2020. 336 с.
34. Методичні вказівки до оформлення курсових проектів, звітів про проходження практики, випускних кваліфікаційних робіт для студентів спеціальності «Комп'ютерна інженерія» / І.В. Гураль, Л.О. Дубчак / Під ред. О.М. Березького. Тернопіль: ТНЕУ, 2019. 33 с.
35. Березький О.М., Мельник Г.М. Методичні рекомендації до виконання кваліфікаційної роботи з освітнього ступеня “Магістр”. Спеціальність: 123 - Комп'ютерна інженерія. Магістерська програма - Комп'ютерна інженерія". Тернопіль: ЗУНУ, 2024. 32 с.