КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ ТАРАСА ШЕВЧЕНКА
Факультет інформаційних технологій

Кафедра технологій управління

Спеціальність 122 – Комп'ютерні науки,
освітня програма «Інформаційна аналітика та впливи»

КВАЛІФІКАЦІЙНА РОБОТА МАГІСТРА
на тему:

"Розробка технології для оптимізації енергоспоживання в розумних будинках методами Data Science"

Студента 2-го курсу групи ІАВ-21
Мініна Ігоря Борисовича

Науковий керівник
к.т.н., асистент кафедри технологій управління
Андрій ХЛЕВНИЙ

_____ _____ _____
(підпис студента) (дата) (підпис)

Попередній захист:		
(Висновок: «До захисту в Екзаменаційній комісії»)		
Завідувач кафедри технологій управління		
(підпис)	(прізвище, ініціали)	(дата)

Київ – 2025

КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІМЕНІ ТАРАСА ШЕВЧЕНКА
Факультет інформаційних технологій

Кафедра технологій управління
Освітньо-кваліфікаційний рівень Магістр
Спеціальність 122 – Комп'ютерні науки
Освітня програма Інформаційна аналітика та впливи

ЗАТВЕРДЖУЮ
Завідувач кафедри
професор Віктор МОРОЗОВ
_____

«___» _____ 20__ р.

ЗАВДАННЯ
НА ВИКОНАННЯ КВАЛІФІКАЦІЙНОЇ РОБОТИ

Студент Мінін Ігор Борисович
Група ІАВ–21

1. **Тема кваліфікаційної роботи**

Розробка технології для оптимізації енергоспоживання в розумних будинках методами Data Science
Затверджена наказом по від «__»_____ 20__р. No__.

Строк подання студентом готової роботи – "19" травня 2025р.

1. Цільова установка та вихідні дані до роботи

Прототип аналітичної системи для прогнозування енергоспоживання та виявлення пікових навантажень на основі обробки історичних даних з використанням алгоритмів машинного навчання. Система реалізована як модуль прогнозної аналітики, що приймає дані файлів запису, визначає ключові фактори споживання, будує короткостроковий прогноз та формує рекомендації.

2. Зміст роботи.

Визначення впливу застосування алгоритмів машинного навчання на точність прогнозування енергоспоживання в умовах розумного будинку, а також оцінка ефективності аналітичних рекомендацій для оптимізації роботи побутових приладів. Виокремлення сучасних підходів і методів обробки даних у сфері енергетичного аналізу, зокрема для побудови інтерпретованих моделей і виявлення періодів пікового навантаження. Обґрунтування вибору технологічного стеку для реалізації прототипу аналітичної системи з урахуванням обмежень щодо доступності даних, обчислювальних ресурсів та потреб користувача. Аналіз існуючих досліджень і рішень щодо прогнозування енергоспоживання, а також оцінка переваг застосування відкритих інструментів і фреймворків для створення адаптивних систем. Опис методики розробки прототипу системи, що дозволяє здійснювати аналіз, прогноз і візуалізацію енергоспоживання, інтегруючи її до локального цифрового середовища мешканця або дослідника без потреби в хмарній інфраструктурі.

3. Перелік графічного матеріалу (слайдів).

10 рисунків, 20 слайдів. Перелік слайдів: мета (1 слайд), актуальність (2 слайди), класифікація методів (1 слайд), методологія (1 слайд), оптимізація моделей (1 слайд), датасет (7 слайдів), імплементація моделей (3 слайди), прогнозування (1 слайд), висновки (1 слайд)

6. Календарний план виконання роботи:

№ п/п	Назва частин роботи		Виконання роботи	
			За планом	Фактично
1.	Вибір теми дипломної роботи	3	01.10.24	01.10.24
2.	Протокол кафедри ТУ про затвердження тем дипломних робіт та призначення наукових керівників	2	27.12.24	27.12.24

3.	Формування переліку нормативних матеріалів, літератури з проблематики дипломної роботи	10	24.02.25	24.02.25
4.	Складання розгорнутого плану кваліфікаційної роботи	5	25.02.25	25.02.25
5.	Ознайомлення наукового керівника з розгорнутим планом кваліфікаційної роботи. Внесення змін.	5	27.02.25	27.02.25
6.	Підготовка розділу 1 «Аналіз предметної галузі та постановка задачі.	10	09.04.25	09.04.25
7.	Підготовка розділу 2 «Аналіз методів Data Science»	14	18.04.25	18.04.25
8.	Підготовка розділу 3 «Розробка методів Data Science»	14	30.04.25	30.04.25
9.	Підготовка розділу 4 «Імплементація моделей Data Science»	13	09.05.25	09.05.25
10.	Оформлення кваліфікаційної роботи. Підготовка висновків і пропозицій	15	10.05.25	10.05.25
11.	Передача кваліфікаційної роботи науковому керівникові	2	12.05.25	12.05.25
12.	Передача кваліфікаційної роботи рецензенту для рецензування	2	12.05.25	12.05.25
13.	Попередній захист кваліфікаційної роботи	5	12.05.25	12.05.25

Дата видачі завдання «____» _____ 2025 р.

Керівник роботи: к.т.н., асистент Хлевний А.О.

_____

(підпис)

Завдання прийняв до виконання студент групи ІАВ-21 Мінін І.Б.

_____

(підпис)

ЗМІСТ

АНОТАЦІЯ

КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ ТАРАСА ШЕВЧЕНКА

Факультет інформаційних технологій

Кафедра технологій управління

Спеціальність 122 - Комп’ютерні науки,

освітня програма “Інформаційна аналітика та впливи”

Дипломна робота магістра Мініна І.Б.

Тема роботи – «Розробка технології для оптимізації енергоспоживання в розумних будинках методами Data Science».

Мета дипломної роботи магістра – створення методики прогнозування енергоспоживання та виявлення пікових навантажень на основі методів Data Science, а також розробка прототипу інформаційної системи для аналізу, візуалізації та генерації рекомендацій щодо оптимізації споживання.

Об’єкт дослідження – енергоспоживання в житлових будинках у контексті впровадження інтелектуальних систем управління.

Предмет дослідження – методи обробки даних, побудови прогнозних моделей та алгоритми прийняття рішень для оптимізації енергоспоживання у розумному будинку.

Наукова новизна роботи – полягає в поєднанні інтерпретованих методів машинного навчання з аналізом ключових драйверів навантаження та адаптивними алгоритмами прогнозування споживання в умовах нестабільних даних. Запропонована технологія враховує погодні фактори, активність пристроїв і часові закономірності, а також включає автоматичну генерацію рекомендацій на основі структурованих JSON-звітів.

У роботі проведено огляд сучасних архітектур систем розумного будинку, класифіковано методи оптимізації енергоспоживання, реалізовано прототип аналітичної системи з функціями прогнозу, візуалізації та пояснення впливу

факторів. Побудовані моделі оцінено за метриками MAPE, R^2 та SHAP. Сформовано рекомендації щодо перенесення пікових навантажень, підвищення ефективності та адаптації моделей до змінних умов.

Дипломна робота складається зі вступу, чотирьох розділів, висновків і списку використаних джерел. Загальний обсяг – 90 сторінок, кількість джерел – 33, перелік додатків: А, Б.

Ключові слова: розумний будинок, енергоспоживання, прогнозування, машинне навчання, оптимізація, інтерпретовані моделі, SHAP, MAPE.

ПЕРЕЛІК ВИКОРИСТАНИХ СКОРОЧЕНЬ

AI — Artificial Intelligence / Штучний інтелект

ARIMA — AutoRegressive Integrated Moving Average / Авторегресійна інтегрована модель ковзного середнього

CNN — Convolutional Neural Network / Згорткова нейронна мережа

CSV — Comma-Separated Values / Формат табличних даних

DNN — Deep Neural Network / Глибока нейронна мережа

ETS — Exponential Smoothing / Експоненціальне згладжування

GAN — Generative Adversarial Network / Генеративна змагальна мережа

IoT — Internet of Things / Інтернет речей

JSON — JavaScript Object Notation / Формат обміну даними

LSTM — Long Short-Term Memory / Довготривала короткочасна пам’ять

LP — Linear Programming / Лінійне програмування

MAPE — Mean Absolute Percentage Error / Середня абсолютна відносна похибка

MPC — Model Predictive Control / Управління з прогнозуванням моделі

PID — Proportional-Integral-Derivative Пропорційно-інтегрально-диференціальний регулятор

PSO — Particle Swarm Optimization / Оптимізація рою частинок

PV — Photovoltaic / Фотоелектричний (сонячна енергія)

RL — Reinforcement Learning / Навчання з підкріпленням

RNN — Recurrent Neural Network / Рекурентна нейронна мережа

SHAP — SHapley Additive exPlanations / Пояснення на основі значень Шеплі

SVM — Support Vector Machine / Метод опорних векторів

SVR — Support Vector Regression / Регресія на основі методу опорних векторів

XGBoost — Extreme Gradient Boosting / Метод бустингу з екстремальним градієнтом

ВСТУП

У XXI столітті цивілізаційний розвиток дедалі тісніше пов’язаний із питаннями стійкого енергоспоживання та раціонального використання природних ресурсів [1]. Змінні кліматичні умови, глобальні тенденції урбанізації, зростання доходів населення й насичення життєвого простору технологіями призводять до постійного зростання енергетичного попиту. Одночасно посилюються виклики, пов’язані зі зменшенням вуглецевого сліду, декарбонізацією економіки й необхідністю дотримання цілей Паризької кліматичної угоди , згідно з якою країни-учасниці мають утримати підвищення глобальної температури на рівні «значно нижче 2 °C» [2]. З геополітичної та економічної точок зору Україна, яка інтегрується в європейський простір і поступово синхронізує енергетичне законодавство з Директивою 2012/27/ЄС «Про енергоефективність», стикається з подвійним завданням: підвищити енергетичну незалежність і скоротити витрати кінцевих споживачів, не пожертвувавши комфортом громадян [3].

У такому контексті концепція «розумного будинку» (Smart Home), що поєднує Інтернет речей (IoT), сенсорні мережі, системи автоматизації та алгоритми обробки даних у реальному часі, виходить на передній план [4]. Водночас більшість існуючих комерційних продуктів використовує статичні або псевдостатичні правила (rule-based): «вимкни світло після 23:00», «знизь температуру, коли нікого немає вдома», «увімкни бойлер з 05:00 до 07:00» тощо. Такі підходи ігнорують складну динаміку поведінки мешканців , миттєву зміну погодних умов, нерівномірність тарифної сітки, а також синергетичні ефекти між різними видами енергії (електрична, теплова, потенційно — газова) [5]. Зростаюча доступність багатоканальних часових рядів, хмарних обчислень і засобів штучного інтелекту створює передумови для переходу від простих «розкладів» до адаптивних систем, що навчаються на даних у процесі експлуатації [6].

Саме методи Data Science — зокрема машинне навчання , статистичне моделювання, оптимізаційні алгоритми та візуальна аналітика — здатні забезпечити таку адаптивність [7]. Вони не лише описують історичні закономірності, а й надають прогностичні інсайти, рекомендуючи оптимальні сценарії дій. Таким чином, інтеграція новітніх інструментів аналітики у систему керування житлом відкриває дорогу до створення «розумних енергоефективних екосистем», де кожен пристрій, датчик чи користувач стає джерелом даних для спільної оптимізації [8][9].

Актуальність дослідження зумовлена необхідністю розробки інноваційних підходів до управління енергоспоживанням у контексті розумних будинків. Традиційні методи оптимізації часто базуються на статичних алгоритмах і не враховують динамічні зміни умов експлуатації об’єктів нерухомості. Використання ж методів Data Science дозволяє врахувати комплексний вплив різних факторів: погодних умов, поведінки мешканців, сезонних коливань та інших змінних, що впливають на рівень споживання енергії. Таким чином, застосування сучасних технологій є ключовим чинником для досягнення високої ефективності та економії енергії у розумних будинках.

Метою даної магістерської роботи є розробка технології оптимізації енергоспоживання в розумних будинках із застосуванням методів Data Science, що дозволить створити гнучкий і адаптивний інструментарій для аналізу, прогнозування та управління споживанням енергії. Основними завданнями роботи є:

- Проведення аналізу сучасного стану проблеми енергоспоживання в житлових будинках та огляд існуючих рішень у сфері «розумного дому».
- Розробка концепції технології, яка включає збір, обробку та аналіз даних, а також створення моделей прогнозування споживання енергії.

- Реалізація прототипу системи оптимізації енергоспоживання із використанням обраних методів аналізу даних.
- Оцінка ефективності розробленого рішення шляхом порівняння з традиційними методами управління енергоспоживанням та аналізу отриманих результатів.

Наукова новизна дослідження полягає у поєднанні сучасних методів Data Science з практичними аспектами управління енергоспоживанням у розумних будинках. Запропонований підхід дозволяє не лише проводити аналіз історичних даних, а й здійснювати прогнозування змін споживання енергії в режимі реального часу, що забезпечує можливість своєчасного прийняття рішень для оптимізації витрат енергії. Важливим аспектом є адаптивність системи до різних сценаріїв експлуатації житлових комплексів, що сприяє підвищенню її універсальності та практичної цінності.

Практичне значення роботи обумовлено можливістю використання розробленої технології у реальних умовах експлуатації розумних будинків. Реалізація прототипу системи оптимізації енергоспоживання може стати основою для подальшого впровадження інтелектуальних систем управління енергетичними ресурсами як у приватних будинках, так і у великих житлових комплексах.

Методологічною основою дослідження є комплексний підхід, що об’єднує методи аналізу даних, машинного навчання та системного моделювання. Використання сучасних інструментів для обробки великих даних дозволяє забезпечити високу точність прогнозування та прийняття рішень у режимі реального часу. Ретельний аналіз отриманих результатів сприятиме виявленню ключових закономірностей, які можуть бути використані для розробки рекомендацій щодо подальшого покращення систем управління енергоспоживанням.

Отже, сучасний етап розвитку інформаційних технологій відкриває нові перспективи для вирішення проблем енергозбереження та оптимізації споживання енергії. Розробка технології, що базується на методах Data Science, є важливим кроком у напрямку створення ефективних та адаптивних систем управління розумними будинками. Запропоноване дослідження сприятиме поглибленню розуміння механізмів впливу різних факторів на енергоспоживання, а також дозволить розробити практичні рекомендації для впровадження інноваційних технологій у сфері енергозбереження.

РОЗДІЛ 1
АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ ТА ПОСТАНОВКА ЗАДАЧІ

1.1 Фактичний стан енергоспоживання житлового сектору

Світова енергетична криза останніх років різко загострила питання ефективності використання енергоресурсів у житловому секторі. Згідно з останніми дослідженнями, проведеними міжнародними організаціями, будівлі споживають значну частину всієї електроенергії, при цьому значна частка цих ресурсів використовується неефективно [10]. В Україні ця проблема має особливу актуальність через фізичне старіння житлового фонду, низький рівень автоматизації та недостатню енергетичну культуру населення.

Міські агломерації стикаються з особливо гострими викликами. Наприклад, у столиці України за останні п'ять років спостерігається стабільне зростання споживання електроенергії, що в окремі періоди призводило до критичного навантаження на мережі. Світовий досвід свідчить, що інтелектуальні системи управління енергоспоживанням можуть забезпечити значну економію ресурсів. У країнах Європи та Північної Америки впровадження таких систем EcoGrid EU у Данії, SmartHome Initiative у Німеччині та Google Nest у США дозволило досягти вражаючих результатів зі зниженням енерговитрат.

Світова тенденція стабільного зростання попиту на енергоресурси у житловому секторі підтверджується останнім звітом *International Energy Agency – Energy Efficiency 2024* (IEA EE 2024) [11]. За останні три десятиліття світове споживання енергії зросло майже вдвічі(рис. 1.1), причому суттєва частина цього обсягу припадає на будівлі житлового та громадського секторів [12].

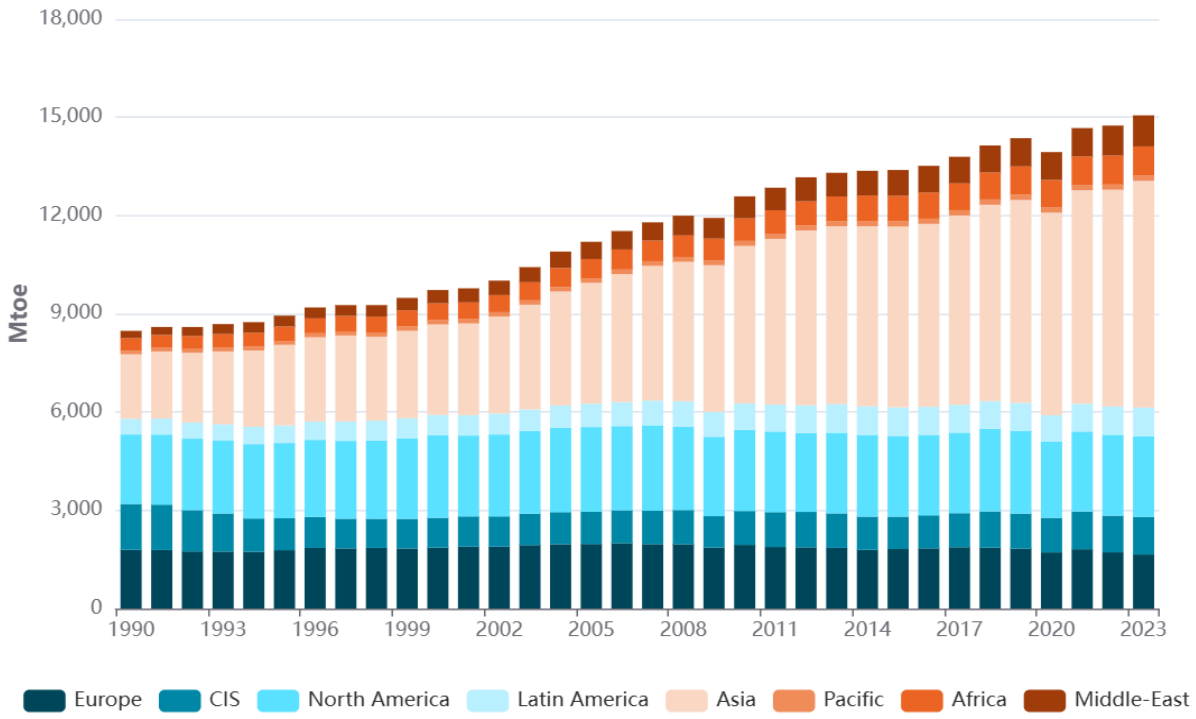

Рисунок 1.1 - зростання світового енергоспоживання у 1990-2023 роках

Сучасна енергетична політика, спрямована на раціональне використання ресурсів та зниження впливу на навколишнє середовище, стимулює розвиток технологій, що забезпечують оптимізацію енергоспоживання [13]. У цьому контексті поняття «розумний будинок» набуває особливої актуальності, адже поєднання систем автоматизації, інтернету речей (IoT) та аналітики даних створює нові можливості для управління енергетичними ресурсами.

Останні роки відзначаються стрімким поширенням технологій IoT у житловому секторі, що суттєво прискорює еволюцію концепції «розумного будинку». Якщо на початкових етапах подібні рішення здавалися елітними і були доступні лише вузькому колу користувачів, то сьогодні активний розвиток електронних компонентів та зниження їхньої вартості сприяють масовому впровадженню інтелектуальних систем у широкому спектрі будівель — від приватних осель до багатоквартирних комплексів.

Динаміку зростання кількості розумних будинків ілюструють прогнози на період із 2019 по 2028 рік. Якщо у 2019 році нараховувалося близько 191 млн будинків, обладнаних технологіями «розумного дому», то до 2028 року ця кількість може зрости до 785 млн [14]. Отже, йдеться про потенційне збільшення більш ніж у чотири рази лише за дев’ять років.

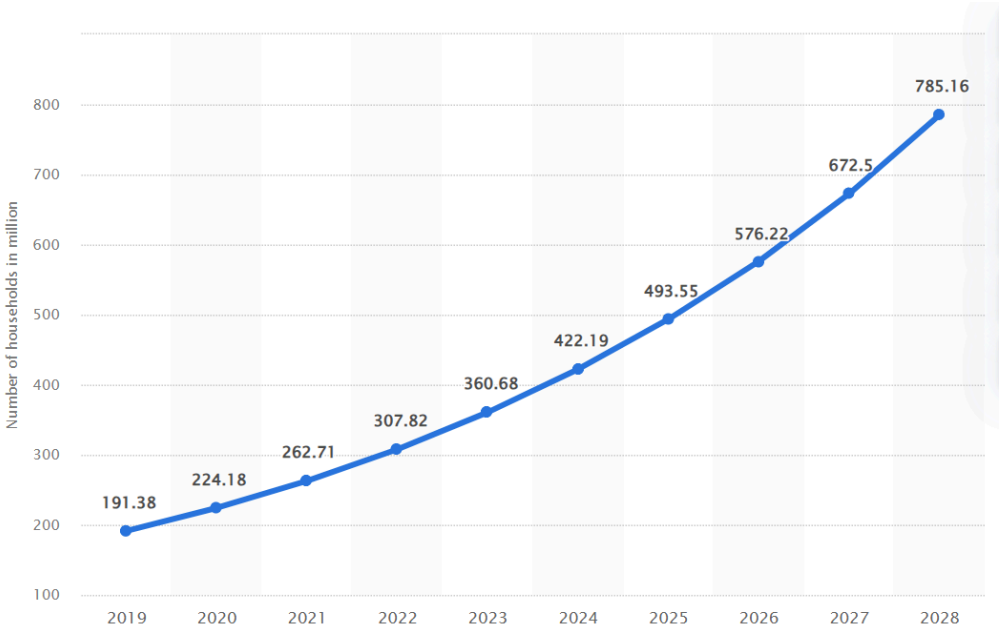

Рисунок 1.2 - кількість будинків із системами розумного дому

Одночасно з цим розширюється й економічна складова ринку: за даними дослідницьких компаній, світовий дохід у сфері розумних будинків, який у 2020 році становив приблизно 181,5 млрд дол. США, у 2024-му перевищив 390,5 млрд, а до 2033-го може зрости до 934,2 млрд [15]. Таким чином, з огляду на темпи, що перевищують середні показники IT-сектора, ринок розумних будинків має потенціал збільшитися в п’ять разів упродовж трохи більше ніж десятиріччя.

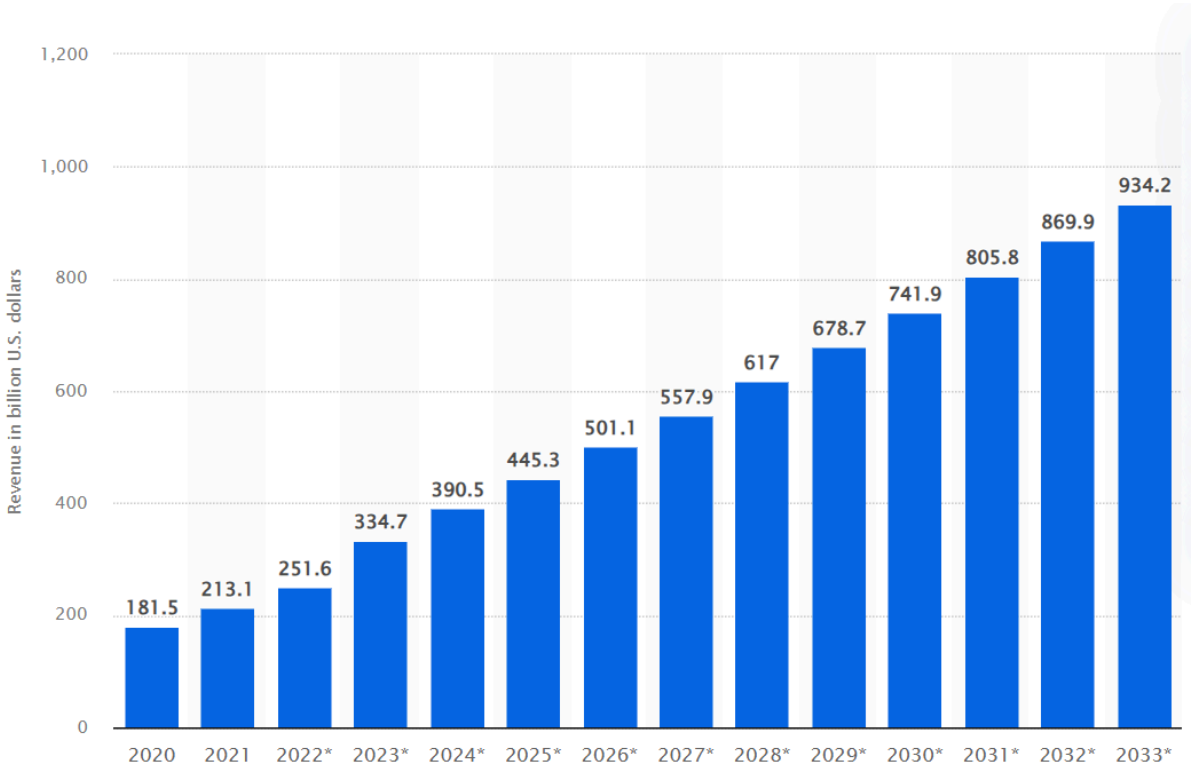

Рисунок 1.3 - Прибуток від IoT

Відсутність ознак уповільнення цієї тенденції свідчить про те, що попит на рішення з централізованого моніторингу та управління обладнанням у житлових приміщеннях продовжить зростати. Йдеться не лише про традиційні системи безпеки, освітлення чи клімат-контролю, а й про широке впровадження хмарних платформ, сервісів аналітики даних і алгоритмів машинного навчання. Вони дають змогу виявляти аномалії у споживанні енергії, прогнозувати майбутні потреби та оперативно оптимізувати роботу різних підсистем будинку. Усе це створює сприятливе середовище для подальших інновацій, залучення інвестицій і формування нових стандартів комфорту й енергоефективності, перетворюючи звичайне житло на високотехнологічний простір.

1.2. Архітектурні рішення для розумних будинків

Сучасні архітектурні рішення для систем управління енергоспоживанням в розумних будинках можна класифікувати за кількома ключовими критеріями, кожен з яких має свої особливості реалізації та сфери застосування.

Централізовані архітектури представляють класичний підхід до побудови систем автоматизації. В основі таких систем лежить єдиний центральний контролер (часто на базі мікроконтролерів типу Arduino або Raspberry Pi), який відповідає за збір даних з усіх датчиків та управління виконавчими пристроями. Головною перевагою такого підходу є відносна простота проектування та налагодження, а також низька вартість рішення. Однак централізовані системи мають ряд суттєвих недоліків: низьку надійність (вихід з ладу центрального контролера паралізує всю систему), обмежену масштабованість (додавання нових пристроїв часто вимагає значних змін у програмному забезпеченні) та високу залежність від якості проводки. В умовах української дійсності, де багато будинків мають застарілу електромережу, ці обмеження стають особливо критичними.

Децентралізовані архітектури є більш сучасним і перспективним рішенням. В таких системах кожен інтелектуальний пристрій (розумні вимикачі, термостати, лічильники тощо) має власний процесор і здатний функціонувати автономно. Пристрої комунікують між собою за допомогою бездротових технологій (Zigbee, Z-Wave, Bluetooth Mesh), що дозволяє відмовитися від додаткової проводки. Важливою перевагою децентралізованих систем є їхня висока надійність - вихід з ладу окремого вузла не впливає на роботу всієї системи. Крім того, такі рішення легко масштабуються - додавання нового пристрою часто зводиться до його простого підключення до мережі. Проте децентралізовані системи мають і свої недоліки: більш високу вартість окремих компонентів, складність первинного налаштування та потенційні проблеми з сумісністю пристроїв різних виробників.

В Україні такі системи поки що не отримали широкого поширення через їхню відносно високу вартість.

Хмарні архітектури передбачають винесення основної логіки роботи системи на віддалені сервери. Пристрої в будинку виконують лише функції збору даних та виконання команд, тоді як вся аналітика та прийняття рішень відбуваються в хмарі. Такі рішення пропонують безпрецедентну обчислювальну потужність і можливість використання складних алгоритмів машинного навчання. Відомими прикладами є системи типу Google Nest або Amazon Alexa. Однак хмарні архітектури мають критичний недолік - залежність від якості інтернет-з'єднання. В умовах України, з існуючими проблемами відключень електроенергії та неповного покриття швидкісним інтернетом, це може призвести до серйозних проблем у роботі системи. Крім того, виникають питання безпеки даних, оскільки вся інформація про життя мешканців передається на сторонні сервери.

Гібридні архітектури поєднують переваги локальних і хмарних рішень. В таких системах критично важливі функції (наприклад, управління опаленням або аварійними системами) реалізуються на місці, тоді як складні аналітичні задачі виконуються в хмарі. Це дозволяє досягти оптимального балансу між надійністю та функціональністю. Сучасні промислові рішення (наприклад, платформи типу Home Assistant або OpenHAB) дозволяють реалізувати гібридну архітектуру з використанням відносно недорогого обладнання. Особливо важливим аспектом таких систем є можливість роботи в офлайн-режимі при втраті інтернет-з'єднання, що робить їх особливо привабливими для українських умов.

Важливим напрямком розвитку архітектур розумних будинків є стандартизація протоколів взаємодії. В останні роки набувають популярності такі відкриті стандарти, як Matter, які дозволяють забезпечити взаємодію пристроїв різних виробників. Це актуальна проблема, оскільки ринок розумних пристроїв

представлений продукцією багатьох виробників, і проблема сумісності стоїть особливо гостро.

Окремо варто відзначити архітектурні рішення, орієнтовані на енергоефективність. Сучасні системи все частіше включають:

- Локальні обчислювальні вузли з низьким енергоспоживанням
- Оптимізовані протоколи передачі даних
- Механізми адаптивного живлення пристроїв
- Інтеграцію з альтернативними джерелами енергії

В умовах України, де питання енергозбереження стають все більш актуальними, такі рішення набувають особливого значення. Наприклад, використання технології energy harvesting (збору енергії з навколишнього середовища) для живлення датчиків дозволяє значно знизити енергоспоживання системи в цілому. Перспективним напрямком є розробка архітектур, орієнтованих на поступове впровадження. Для багатьох українських будинків повний перехід на "розумні" технології є недосяжним через фінансові обмеження. Тому особливу цінність мають рішення, які дозволяють починати з невеликої кількості пристроїв і поступово розширювати систему, зберігаючи при цьому її цілісність і функціональність.

1.3. Класифікація методів оптимізації енергоспоживання в розумних будинках

Останні п’ять років позначилися справжнім «вибухом» публікацій і пілотних проектів, тому спроби класифікувати всі існуючі підходи без чіткої таксономічної рамки неминуче призводять до плутанини. На основі систематичних оглядів – Springer Sustain. Cities & Society (2025),

ScienceDirect Energy AI (2024) і метарев'ю ResearchGate (2025) – у роботі прийнято класифікацію за алгоритмічним парадигмами (яка теорія лежить в основі ухвалення рішень).

Таблиця 1 - Алгоритмічні парадигми

Категорія	Теоретична база	Типові методи	Приклади реалізацій
Детерміновані	Логіка, класичне управління	Rule-based сценарії, PID	Nest v3, Tado°, Ajax LifeQuality
Математична оптимізація	ЛП, QP, МІП, MPC	Convex-MPC, MILP диспет-черизація	FLEXCoop, Siemens \ Desigo CC
Мета-евристичні	Генетичні алгорит-Ми, PSO, ACO	GA для графіка бойлера, PSO + PV-storage	Huawei FusionSolar Optimizer
Навчання з учителем (ML)	Статистика, DL	LSTM, GRU, XG Prophet	Ecoisme Pilot (Kyiv), Sense Labs
Підкріплювальне навчання (RL)	MDP, Q-Learning, Policy Gradient	DQN, PPO, SAC	NREL RLEMS, TEPCO HomeLab
Гібридні	Комбінації вище	MPC-guided RL, LSTM + Rule fuse	NTNU SARLEM 2024; Learning-Based MPC

1) Детерміновані (rule-based, PID, таймери)

Це найдавніша і найдоступніша парадигма, що формально не містить елементів навчання чи оптимізації. Управління ґрунтується на жорстко зафіксованих логічних правилах типу if–then–else або на локальних регуляторах (PID) з фіксованими коефіцієнтами.

Переваги. Мінімальний CAPEX, інтуїтивна пояснюваність, нульові вимоги до історичних даних.

Недоліки. Не реагує на зміну поведінки мешканців і динамічні тарифи; щоб наблизитися до адаптивності, необхідно створювати сотні правил - це веде до “комбінаторного вибуху” та зростання ризику конфліктів.

2) Математична оптимізація (MPC, лінійне/квадратичне програмування, MILP)

Model Predictive Control, лінійні (LP) та змішано-цілі (MILP) оптимізації працюють із детальною фізичною чи квазі-емпіричною моделлю будинку. Вони мінімізують суму витрат/дискомфорту на прогнозному горизонті, обмежуючи допустимі дії.

Переваги. Гарантоване дотримання обмежень (температура, частота перемикань, потужність мережі); можливість безпосередньо включити багатозонні тарифи та CO_2 інтенсивність.

Недоліки. Вимагає трудомісткої ідентифікації параметрів будівлі; при великих об’єктах задача стає нелінійною й обчислювально складною.

3) Мета-евристичні алгоритми (GA, PSO, ACO)

Коли фізична модель невідома, а простір рішень дискретний і негладкий, використовують еволюційні й рійові евристики. Генетичні алгоритми перебирають коди «хромосоми» графіка роботи приладів; Particle Swarm — шукає мінімум витрат як глобальний потік частинок [16].

Переваги. Не потребує градієнта; легко працює з нелінійними тарифами та взаємодією кількох джерел (PV + тепловий насос + EV-зарядка).

Недоліки. Гарантій глобального оптимуму немає; швидкість збіжності залежить від тонкого підбору параметрів популяції й операторів схрещення/мутації.

4) Навчання з учителем (машинне навчання для прогнозу)

Тут оптимізація розділяється: спершу ML-модель (LSTM, GRU, XGBoost, Prophet) прогнозує майбутнє навантаження або температуру, потім окрема евристика (часто linear-programming чи even rule) будує план.[17][18]

Переваги. Висока точність на складних нелінійних даних; добре масштабується як хмарний сервіс.

Недоліки. Потребує ≥ 6 місяців чистих історичних даних; сама по собі не видає оптимальної політики — потрібен додатковий планувальник.

Типові KPI. $10 – 15\,\%$ економії, залежно від точності прогнозу та якості евристики.

5) Підкріплювальне навчання (RL: DQN, PPO, SAC)

Розглядає будинок як марковське середовище; агент приймає дії й одразу отримує винагороду, що карає за витрати, CO_2 і дискомфорт. Ідеально підходить для мультиоб'єктних цілей.

Переваги. Адаптивність до зміни поведінки, можливість самостійно вивчати складні стратегії (відкласти нагрів бойлера, коли CO_2-інтенсивність висока) [19].

Недоліки. "Cold start" — потрібен симулятор; ризик некоректних дій без safety-layer; складна пояснюваність.

6) Гібридні підходи (Learning-Based MPC, MPC-safe RL, LSTM + Rule Fuse)

Об'єднують сильні сторони вищеописаних класів: ML-прогноз слугує «м'якою» моделлю всередині MPC, RL-агент працює поверх rule-based safety, або MPC задає «коридор» для RL-дослідження.

Переваги. Гарантії безпеки від MPC + гнучкість RL; може працювати з неповними фізичними моделями.

Недоліки. Найвища складність реалізації, потреба у двох типах даних (для ML і для MPC), злиття моделей підвищує вимоги до обчислювальних ресурсів.

Таким чином, кожна парадигма має чітку «нішу застосування»:

- Rule-based — швидкий старт, малий бюджет.
- MPC / MILP — там, де відома фізика й потрібні гарантії.
- Метаевристики — для складних дискретних задач
- ML-прогноз — масштабовані хмарні сервіси.
- RL — адаптивні prosumer-моделі з багатьма джерелами енергії.
- Гібриди — комбінація переваг кількох підходів.

1.4. Проблематика енергоефективності в сучасних розумних будинках

Попри бурхливий розвиток технологій IoT, широкодоступні сенсори та хмарні сервіси аналітики, рівень фактичної енергоефективності більшості Smart Home-установок суттєво відстає від потенційного. Нижче систематизовано ключові бар'єри, що стримують досягнення глибокої економії у житловому секторі.

1.4.1 Техніко-інфраструктурні обмеження

Попри те, що на ринку пропонується сотні моделей «розумних» термостатів, ламп, реле й теплових насосів, більшість квартир і приватних будинків залишається технологічно неготовою до просунутого енергокерування. Передусім це пов'язано з фрагментацією протоколів. На практиці навіть у новобудовах можна виявити одночасну присутність Zigbee-датчиків освітленості, Wi-Fi камер спостереження і Z-Wave розеток. Кожна екосистема тягне за собою власний шлюз, власні оновлення прошивки й окремий мобільний застосунок. У результаті створюється хаотична класифікація пристроїв, який важко інтегрувати

в єдине середовище для оптимізації навантажень. Для об’єднання даних доводиться встановлювати багатопротокольні хаби типу Home Assistant з одночасним навантаженням на мережу та підвищеним ризиком кіберзагроз.

Друга інфраструктурна проблема — обмеження по електропостачанню. Наприклад, типовий радянський багатоквартирний будинок із проектним приєднаним навантаженням 5 кВт на квартиру не може одночасно живити індукційну плиту (7 кВт) і тепловий насос (3–4 кВт) без ризику вибивання автомата. Додайте до цього старі алюмінієві дроти в стояках та перевантажені трансформаторні підстанції — і стає зрозуміло, що алгоритмічні новації без модернізації мережевої інфраструктури неможливі.

Третя проблема — теплотехнічна інерційність. Низький опір теплопередачі стін ($R < 1{,}5$ м²·К/Вт) означає, що будь-яка економія, отримана завдяки оптимізації графіка опалення, швидко «витікає» через огороджувальні конструкції. Хоча за новими стандартами для нових будівель опір теплопередачі стін має бути не менше 4 м²·К/Вт, більшість житлового фонду - це старі радянські будинки, які не відповідають таким високим нормам [20]. Спираючись на це знання, стає очевидним, що цифрові рішення повинні йти разом з будівельними заходами, інакше ефекту від алгоритмічного енергозбереження не буде.

1.4.2 Дані та алгоритмічний розрив

Сучасні системи навчання з підкріплення й глибинні рекурентні мережі теоретично здатні давати передбачення навантаження з похибкою $< 5\,\%$ і оптимізувати керування так, щоб економити до чверті енергії. На практиці їм бракує якісних даних. Деякі старі лічильники пишуть CSV-файли на локальну карту SD, тож при перепаді живлення інформація за ніч може не записатися; Wi-Fi-сенсори часто втрачають зв’язок і повертають NaN-значення. Навіть якщо дані повні, виникає явище concept drift — зміни у поведінці мешканців та тарифній політиці. Уявімо сім’ю, яка перевела робочу активність з дистанційного

формату 2022 р. на офісний у 2024-му: ранковий та вечірній пік зміщуються, а денне фонове споживання падає. Модель, натренована на «карантинних» даних, починає систематично переоцінювати дні тижня, і профіт від оптимізації зникає. Без процедур онлайн перенавчання або автоматичного детектора дрейфу ML-система втрачає перевагу над звичайним календарним таймером буквально за пів року експлуатації.

1.4.3 Регуляторно-правові та кібербезпекові ризики

Збір детальних даних побутового споживання неминуче наштовхується на вимоги GDPR. У 200-тисячному масиві показників з інтервалом 1 хв можна виявити, коли мешканець приймає душ або повертається з роботи, що вважається персональною інформацією. Для сертифікації системи Smart Home доводиться впроваджувати end-to-end шифрування (TLS 1.3) і робити агрегацію даних на локальному шлюзі, що збільшує вимоги до «заліза» та вартість прошивки.

Окрему загрозу становлять уразливості типу MQTT misconfiguration. Дослідження Ruhr-Uni Bochum (2024) проаналізувало 337 000 бекендів протоколів MQTT, CoAP, and XMPP, і виявило, що 99.84% MQTT-брокерів є незахищенними, що означає, що через них можна дистанційно вимикати чи вмикати бойлери, по суті організовуючи DDoS мережі живлення.

“99.84% of MQTT- and XMPP-speaking backends use insecure transport protocols (only 0.16% adopt TLS, of which 70.93% adopt a vulnerable version)” [22, с.1]

1.4.4 Дефіцит локальних даних

Більшість відкритих датасетів — Pecan Street (Техас), IDEAS (Середземномор’я), UK-DALE (Оксфорд) — описують інший клімат, інший розподіл тарифів і поведінки споживачів. Український споживач опалює житло з

жовтня по квітень, тоді як техаський — охолоджує з травня по вересень; це робить прямий перенос моделей некоректним. Локальних повних наборів з відкритою ліцензією просто не існує. Таким чином, будь-який серйозний дослідник змушений або купувати дані в оператора (і отримує неповні щоденні агрегати), або встановлювати власний стенд і пів року чекати накопичення вибірки.

1.5 Постановка задачі

Під час огляду літератури з'ясовано, що головний недолік багатьох смарт-систем полягає не у відсутності «розумних» алгоритмів, а у банальній неспроможності — в реальному часі й на малій вибірці — передбачати, коли будинок увійде в режим підвищеного навантаження. Якщо мати короткостроковий прогноз споживання й розуміння, які прилади чи погодні фактори створюють пік, навіть найпростіший інтерактивний сценарій дає відчутну економію. Саме тому в центр дослідження поставлено задачу прогнозування загального навантаження і пошуку ключових драйверів цього споживання.

1.5.1 Мета роботи

Метою роботи є зменшення пікових навантажень та покращення ефективності енергоспоживання в розумних будинках шляхом розробки адаптивної технології аналізу та прогнозування на основі методів Data Science.

Запропонована методика дозволяє будувати інтерпретовані моделі споживання енергії, виявляти ключові чинники впливу, ідентифікувати критичні періоди навантаження та формувати рекомендації щодо їх зниження.

Виявлення чинників впливу передбачає комплексний аналіз даних. На першому етапі досліджується взаємозв'язок між активністю окремих приладів і загальним споживанням енергії. Особливу увагу приділяється погодним параметрам: температурі, яка впливає на роботу систем опалення та охолодження,

вологості, що може збільшувати навантаження на вентиляцію, та швидкості вітру, яка іноді пов’язана з енерговтратами. Окремо аналізуються часові закономірності.

Побудова прогнозної моделі орієнтована на досягнення середньої абсолютної процентної похибки (MAPE) не гірше 10% на п’ятнадцяти хвилинному горизонті. Попередня обробка даних включає нормалізацію, кодування категоріальних змінних та заповнення пропусків. Валідація моделі проводиться з урахуванням часової залежності даних через крос-валідацію TimeSeriesSplit, що імітує реальні умови прогнозування. Після навчання аналізуються залишки моделі, щоб виявити систематичні помилки та вдосконалити її точність.

Ідентифікація періодів надмірного навантаження базується на двох критеріях: перевищенні порогового значення споживання та нерівномірному розподілі енергії між приладами. Наприклад, одночасна робота печі та посудомийки може створювати критичне навантаження. Система також генерує автоматичні рекомендації: запропонувати перенести запуск енергоємних приладів на нічний тариф, скорегувати температурні налаштування або активізувати сонячну генерацію для покриття пікових потреб.

Репродукованість методики забезпечується через публікацію коду у відкритих репозиторіях (наприклад, GitHub) з детальним описом кроків. Це дозволяє іншим дослідникам або користувачам відтворити результати без необхідності настроювання специфічного середовища. Практичне застосування методики охоплює різні сценарії: домогосподарства можуть знизити витрати на енергію завдяки оптимізації графіків роботи приладів, енергокомпанії – прогнозувати навантаження на мережу, а дослідники – адаптувати підхід для аналізу комерційних будівель.

Наукова новизна роботи полягає в інтеграції інтерпретованих моделей машинного навчання з методами часового аналізу, що дозволяє будувати детальні енергетичні профілі. Розроблені критерії надмірного навантаження враховують реальні умови експлуатації, на відміну від узагальнених статистичних підходів.

Акцент на практичну реалізацію робить методику доступною для широкого застосування без необхідності використовувати спеціалізоване програмне забезпечення.

Обмеження дослідження пов’язані з годинною деталізацією даних, яка не враховує хвилинні коливання, що може бути важливим для деяких сценаріїв. Модель не включає зовнішні економічні фактори, такі як динаміка тарифів, які впливають на поведінку споживачів. Крім того, валідація проведена на даних одного будинку, тому масштабованість методики потребує додаткових перевірок на різноманітних датасетах.

1.5.2 Дослідницькі завдання

Таблиця 2 - Дослідницькі завдання

Шаг	Зміст	Очікуваний результат
Аналіз даних	Описова статистика, теплова карта кореляцій «прилад ↔ house overall», пошук аномалій	Перелік підозрілих піків, діаграми розподілів

Продовження

таблиці 2

Шаг	Зміст	Очікуваний результат
Відбір ознак	Комбінування експертних (часові індикатори) з автоматичними (Permutation Importance, SHA	Ранжований список 10–15 ключових параметрів
Навчання моделей	Порівняння різних класів регресорів	Модель-переможець із MAPE ≤ 10 % і $R^2 \geq 0{,}85$
Інтерпретація	SHAP-пояснення	Візуальний сюжет
Рекомендації	JSON файл з загальним споживанням, “важкими” пристроями та характеристи-ками погоди	JSON файл

1.5.3 Метрики успіху

Для оцінки ефективності запропонованої методики використовуються три ключові показники, які охоплюють аспекти точності, інтерпретованості та практичної корисності моделі. Ці метрики забезпечують комплексний підхід до верифікації результатів, враховуючи як технічні вимоги до алгоритмів, так і потреби кінцевих користувачів.

Точність прогнозу вимірюється за допомогою середньої абсолютної процентної похибки (MAPE). Вимога до моделі полягає в тому, щоб MAPE на тестовій вибірці (30% даних) не перевищувала 10%. Цей поріг обрано з урахуванням практичних стандартів у галузі енергетичного прогнозування, де

похибка у межах 10% вважається прийнятною для оперативного планування. Наприклад, якщо реальне споживання становить 50 кВт·год, прогнозоване значення має бути в діапазоні 45–55 кВт·год. Досягнення цього рівня точності забезпечує достатню надійність для рекомендацій щодо оптимізації, зокрема перенесення пікових навантажень або коригування роботи приладів. Для розрахунку MAPE використовується формула(1.1):

$$MAPE = \frac{1}{n}\sum_{i=1}^{n} |\frac{y_i - \hat{y}_i}{y_i}| \times 100\%,$$

(1.1)

де y_i - реальне значення,

$\hat{y}_i$ - прогнозоване значення,

n - кількість спостережень.

Пояснюваність моделі оцінюється через аналіз внеску ознак за допомогою SHAP-значень (SHapley Additive exPlanations). Критерій успіху полягає в тому, що п’ять провідних ознак (наприклад, активність печі, температура зовнішнього середовища, година доби) мають сукупно пояснювати не менше 60% дисперсії прогнозу. Ця вимога забезпечує, що модель не є "чорною скринькою": користувачі можуть зрозуміти, які фактори найбільше впливають на споживання. Наприклад, якщо SHAP-аналіз показує, що температура пояснює 35% дисперсії, а робота посудомийки — 25%, це дає підстави зосередитися на оптимізації цих параметрів.

Енергетичний ефект визначається як різниця між фактичним споживанням та сценарієм "із рекомендацією". Для перевірки статистичної значущості цієї економії застосовується бутстреп-тест з рівнем значущості $p < 0.05$. Це означає, що ймовірність випадкового досягнення подібного результату становить менше 5%, що підтверджує ефективність рекомендацій. Наприклад, якщо середнє пікове споживання — 100 кВт·год, модель має забезпечити зниження до 95 кВт·год. Для розрахунку використовується імітація 1000 бутстреп-вибірок з подальнім порівнянням розподілів "реального" та "оптимізованого" споживання.

РОЗДІЛ 2
АНАЛІЗ МЕТОДІВ DATA SCIENCE

2.1 Комплексний огляд методів аналізу даних у контексті енергетичного прогнозування

Енергетичне прогнозування, як наука, поєднує в собі точність математичних моделей, глибину аналізу даних і практичні знання про специфіку енергетичних систем. Його мета — не лише передбачити майбутнє, а й забезпечити стабільність, ефективність і екологічну безпеку. Сучасні методи аналізу даних у цій галузі розвиваються стрімко, інтегруючи досягнення статистики, машинного навчання та глибокого навчання. Кожен підхід має свої унікальні риси, які роблять його привабливим для конкретних сценаріїв. Розглянемо їх детальніше, акцентуючись на теоретичних основах, практичних застосуваннях і викликах, з якими стикаються дослідники та інженери.

2.1.1 Статистичні методи

Серед класичних підходів до аналізу часових рядів виділяються моделі ARIMA (AutoRegressive Integrated Moving Average) та їх сезонні модифікації SARIMA [22]. Ці методи базуються на ідеї розкладання ряду на авторегресійні компоненти, інтеграції та ковзні середні.

1. Авторегресія (AR): Прогнозування майбутніх значень як лінійної комбінації минулих:

$$y_t = c + \phi_1 y_{t-1} + \phi_2 y_{t-2} + \ldots + \phi_p y_{t-p} + \varepsilon_t, \quad (2.1)$$

де ϕ - коефіцієнти,

ε_t - шум

2. Інтеграція (I): Диференціювання ряду для усунення трендів. Наприклад, перший порядок:

$$\Delta y_t = y_t - y_{t-1}. \quad (2.2)$$

3. Ковзне середнє (MA): Врахування помилок попередніх прогнозів:

$$y_t = \mu + \varepsilon_t + \theta_1 \varepsilon_{t-1} + \ldots + \theta_q \varepsilon_{t-q}. \quad (2.3)$$

Наприклад, авторегресійна частина (AR) описує залежність поточного значення ряду від його минулих значень, тоді як інтеграція (I) допомагає усунути нестаціонарність через диференціювання. Сезонна версія SARIMA додає параметри, $(P, D, Q)_m$, де m — період сезонності, що моделюють періодичні коливання, наприклад, добові або тижневі цикли.

Такі моделі ідеально підходять для стабільних систем із чіткими закономірностями. Наприклад, у міській енергосистемі, де споживання електроенергії щодня досягає піку о 19:00, SARIMA може врахувати цю сезонність і забезпечити прогноз з похибкою 8-10%. Однак їхній слабкий бік — нездатність адаптуватися до раптових змін, таких як аварії або екстремальні погодні умови. Крім того, процес налаштування параметрів (p, d, q для ARIMA та сезонних P, D, Q для SARIMA) залишається трудомістким і часто вимагає ручного аналізу автокореляційних функцій.

Експоненційне згладжування (ETS) — інший класичний інструмент, який розкладає ряд на рівень(базове значення ряду), тренд(лінійне або експоненційне зростання/спад) і сезонність(періодичні коливання). На відміну від ARIMA, ETS більш гнучкий у роботі з нестаціонарними даними, але його точність різко падає при довгострокових прогнозах.

Наприклад модель ETS(A,A,A) описується як на формулі (2.4)

$$\begin{cases} y_t = l_{t-1} + b_{t-1} + s_{t-m} + \epsilon_t \\ l_t = l_{t-1} + b_{t-1} + \alpha\epsilon_t \\ b_t = b_{t-1} + \beta\epsilon_t \\ s_t = s_{t-m} + \gamma\epsilon_t \end{cases}, \quad (2.4)$$

де α, β, γ — коефіцієнти згладжування,

m — період сезонності.

2.1.2 Машинне навчання

Перехід від статистичних методів до машинного навчання відкриває нові горизонти для роботи з нелінійними залежностями та великими наборами даних. Випадкові ліси (Random Forest), наприклад, будують ансамбль дерев, кожне з яких тренується на випадковій підмножині даних. Цей підхід дозволяє ефективно обробляти відсутні значення, категоріальні змінні та шум. Наприклад, у задачі прогнозування пікового навантаження взимку Random Forest може врахувати взаємодію між температурою, вологістю і часом доби, досягаючи R^2 0.85. Важлива перевага — здатність визначати ключові фактори впливу через feature importance. Однак ігнорування часової послідовності обмежує його застосування в реальному часі.

Градієнтний бустинг вирішує цю проблему завдяки послідовній побудові дерев, де кожна наступна модель виправляє помилки попередньої. Цей алгоритм особливо ефективний для складних даних, таких як прогнозування виробництва сонячної енергії, де потрібно враховувати хмарність, кут падіння сонячних променів і технічні параметри панелей. Регуляризація (L1/L2) дозволяє уникнути перенавчання, а підтримка інкрементного навчання робить модель адаптивною до нових даних. Наприклад, система, що прогнозує ціни на електроенергію, може оновлюватися щогодини, враховуючи зміни попиту та погоди. Недолік — висока

чутливість до гіперпараметрів: невірний вибір швидкості навчання (learning rate) або глибини дерев (max_depth) може погіршити точність на 10-15%.

Метод опорних векторів, відомий в англомовній літературі як support vector machine (SVM), є машинним алгоритмом, котрий навчається на прикладах та використовується для класифікації об’єктів [23]. Наприклад, SVM може розрізнити аварійний режим роботи електромеханічної системи та класифікувати його за наявності попередніх досліджень, можливих за технологічними вимогами режимів роботи. Такий підхід розкриває значні можливості для побудування адаптивних систем автоматичного керування[]. Метод заснований на ядерних функціях, пропонує альтернативу для роботи у високовимірних просторах. Наприклад, у задачах прогнозування цін на газ SVR з радіальним ядром (RBF) може виявити нелінійні залежності між попитом, сезонністю та геополітичними факторами. Однак масштабування на великі набори даних залишається проблемою: навчання моделі на мільйоні точок може зайняти години навіть на потужних серверах. SVR будує гіперплощину, яка мінімізує відхилення прогнозів від реальних значень. Формула виглядає наступним чином:

$$\min\left(\frac{1}{2}\|\omega\|^2\right) + C\sum_{i=1}^{n}(\xi_i + \xi_i^*), \tag{2.5}$$

де ξ_i, ξ_i^* - вільні змінні,

C - параметр регуляризації

Також SVR має деякі обмеження:

- Масштабування: На великих даних алгоритм стає неефективним.
- Вибір ядра: Потрібен експеримент з різними типами (поліноміальне, сигмоїдне).

2.1.3 Глибоке начання

Глибоке навчання, засноване на нейронних мережах, стало потужним інструментом для аналізу складних часових рядів, які характерні для енергетичних систем. На відміну від класичних методів, які покладаються на ручне виділення ознак, глибокі моделі автоматично виявляють приховані залежності, що робить їх незамінними для прогнозування споживання, оптимізації генерації та управління розподіленими ресурсами. Далі розглядаються основні види нейронних мереж.

Згорткові нейронні мережі (CNN)

Згорткові нейронні мережі, спочатку розроблені для обробки зображень, стали ефективними для аналізу часових рядів завдяки здатності виявляти локальні паттерни. В енергетиці вони застосовуються для виявлення аномалій, класифікації режимів роботи обладнання та прогнозування короткострокових коливань споживання [24].

Математична основа:

Згортковий шар застосовує фільтри до вхідних даних, обчислюючи зважену суму локальних ділянок. Для часового ряду x(t) згортка з ядром K розміром k виражається як:

$$(I \cdot K)_i = \sum_{m=1}^{k} I_{i-m} \cdot K_m, \tag{2.6}$$

де I — вхідний сигнал,

K — ядро фільтра розміром k.

Після згортки пулінг-шар (наприклад, max-pooling) зменшує розмірність даних, виділяючи найважливіші ознаки.

У енергетиці CNN ефективні для виявлення аномалій у мережі. Наприклад, модель, натренована на даних зі смарт-лічильників, може класифікувати короткочасні спалахи напруги, пов’язані з несправностями обладнання. Однак CNN мають обмеження в роботі з довгостроковими залежностями, оскільки фокусуються на локальних, а не глобальних паттернах.

Обмеження:

- Неефективність для довгострокових залежностей через фокус на локальних паттернах.
- Вимагає великої кількості даних для навчання складних фільтрів.

Глибока нейронна мережа(DNN)

Глибокі нейронні мережі (DNN) складаються з декількох шарів нейронів, де кожен нейрон попереднього шару зв’язаний із усіма нейронами наступного. Вони є універсальними апроксиматорами, здатними моделювати складні нелінійні залежності, що робить їх корисними для багатовимірних задач у енергетиці [25].

Математична основа:

Кожен шар перетворює вхідні дані, застосовуючи нелінійну функцію активації (наприклад, ReLU):

$$a^{(l)} = max(0,\ W^{(l)}a^{(l-1)} + b^{(l)}), \tag{2.7}$$

де W - ваги,

b - зміщення

Для боротьби з перенавчанням використовують техніки регуляризації, такі як dropout, який випадково "вимикає" частину нейронів під час навчання:

$$a^{(l)} = dropout(a^{(l)}, p), \tag{2.8}$$

де p - ймовірність відключення нейрона.

Застосування в енергетиці:

- Прогнозування цін на енергоносії: DNN може обробляти сотні факторів, включаючи попит, погоду, політичні рішення.
- Оптимізація енергоефективності будівель: Модель аналізує взаємодію між опаленням, вентиляцією та зовнішніми умовами

Обмеження:

- Висока обчислювальна складність для мереж із великою кількістю шарів.
- Схильність до перенавчання при малих обсягах даних.

Генеративні змагальні мережі(GAN)

Генеративні змагальні мережі — це клас алгоритмів штучного інтелекту, що використовуються в некерованому навчанні, реалізовані системою двох штучних нейронних мереж, які змагаються одна з одною в рамках гри з нульовою сумою.

GAN складаються з двох мереж: генератора, який створює синтетичні дані, і дискримінатора, який намагається відрізнити їх від реальних. Ця архітектура особливо корисна для генерування даних у умовах їхнього дефіциту, що актуально для енергетики, де історичні дані часто обмежені [27].

Математична основа:

Генератор G перетворює випадковий шум z у синтетичні дані G(z), а дискримінатор D оцінює їх правдоподібність. Мінімаксна задача оптимізації виглядає як:

$$\min \max E_x \left[\log D\left(x\right)\right] + E_z \left[\log \left(1 - D\left(G\left(z\right)\right)\right)\right]. \tag{2.9}$$

Застосування в енергетиці:

- Синтез даних для тренування: GAN генерують реалістичні дані про споживання, що дозволяє покращити точність прогнозних моделей.

- Виявлення аномалій: Модель, натренована на "нормальних" даних, виявляє відхилення, такі як крадіжки електроенергії або аварії.

Обмеження:

- Складність балансування між генератором і дискримінатором.
- Ризик генерації нефізичних даних, якщо модель недостатньо натренована.

Рекурентні нейронні мережі (RNN)

Рекурентні нейронні мережі — це клас нейронних мереж, спеціально розроблених для роботи з послідовними даними, де порядок і контекст грають вирішальну роль. На відміну від звичайних мереж, RNN зберігають інформацію про попередні стани через механізм внутрішньої пам’яті, що робить їх ідеальними для аналізу часових рядів, мови, відео та інших динамічних процесів. У енергетиці вони знаходять застосування в прогнозуванні споживання, аналізі даних з сенсорів і навіть управлінні розподіленими енергоресурсами [27].

Основна ідея RNN полягає у введенні скритого стану (hidden state), який переносить інформацію між кроками послідовності. На кожному кроці t мережа отримує вхідні дані x_t та попередній скритий стан h_{t-1} , обчислюючи новий стан:

$$h_t = \sigma(W_h * x_t + U_h * h_{t-1} + b_h), \quad (2.10)$$

де W_h, U_h - матриці ваг для входу та скритого стану,

b_h - функція активації(наприклад tanh або ReLU).

Вихід мережі y_t формується на основі поточного скритого стану:

$$y_t = \sigma(W_y * h_t + b_y). \quad (2.11)$$

Ця структура дозволяє RNN враховувати залежності між віддаленими елементами послідовності. Наприклад, при прогнозуванні споживання

електроенергії RNN може зв’язати вечірній пік із часом доби та погодними умовами.

Незважаючи на теоретичні переваги, класичні RNN стикаються з проблемою затухаючого градієнта. При навчанні на довгих послідовностях градієнти, які передаються через час, експоненційно зменшуються, що робить неможливим оновлення ваг на ранніх шарах. Це обмежує здатність RNN запам’ятовувати довгострокові залежності.

Для подолання обмежень класичних RNN були розроблені архітектури з механізмами контролю пам’яті, а саме довга короткочасна пам’ять (LSTM).

LSTM (Long Short-Term Memory) – це різновид рекурентних нейронних мереж, спеціально розроблений для роботи з часовими рядами. Основною перевагою LSTM є здатність зберігати інформацію протягом тривалих проміжків часу, завдяки чому мережа може враховувати довгострокові залежності у послідовностях даних [28]. Це робить LSTM надзвичайно корисним для задач прогнозування енергоспоживання та генерації, де історичні дані та зовнішні впливи (наприклад, погодні умови) мають значний вплив на майбутні показники. Модель LSTM здатна адаптуватися до складних нелінійних залежностей у даних, що забезпечує високу точність прогнозів. Крім того, LSTM може інтегрувати додаткові ознаки, що дозволяє враховувати мультифакторні впливи на енергоспоживання [29].

Застосування RNN в енергетиці

- Прогнозування споживання енергії: RNN аналізують історичні дані, враховуючи сезонність, погоду та економічні фактори.
- Оптимізація генерації енергії: RNN можуть прогнозувати виробництво вітрової та сонячної енергії, інтегруючи дані з метеостанцій.
- Виявлення аномалій: RNN можуть виявляти нестандартні режими роботи обладнання (наприклад, перегрів трансформаторів) через аналіз часових рядів з датчиків.

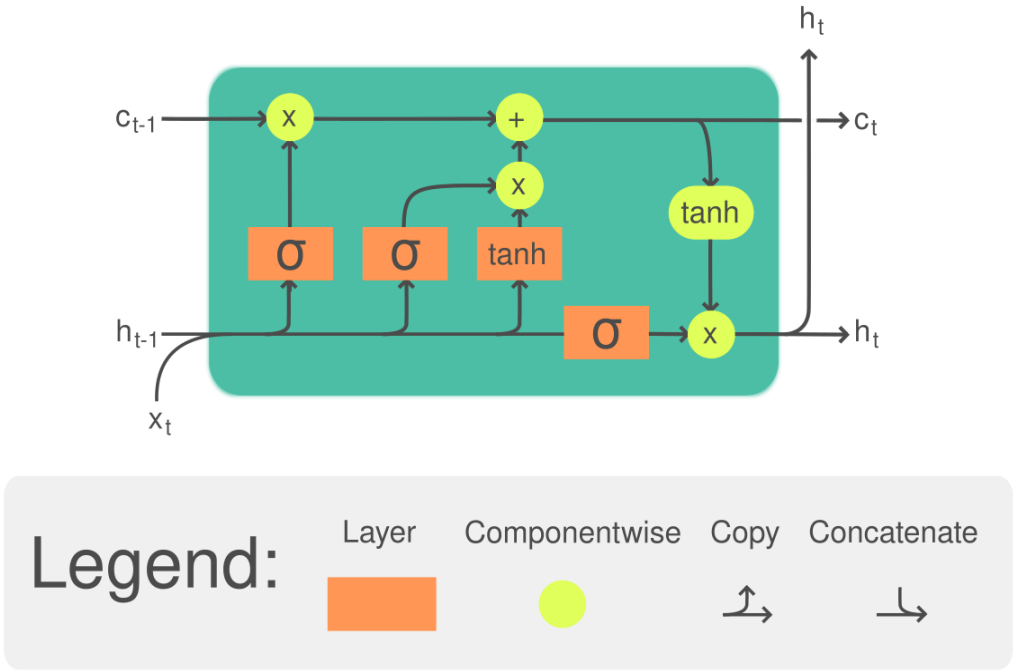

Рисунок 2.1 - Схема роботи LSTM

Обмеження:

- Неможливість перегляду рішень щодо зберігання: одним із основних обмежень LSTM є його боротьба за перегляд збережених значень, коли зустрічається більш подібний вектор. Це може призвести до неоптимальної продуктивності в завданнях, які вимагають динамічного оновлення збереженої інформації.
- Обмежена ємність зберігання: LSTM стискають інформацію в скалярні стани комірки, що може обмежити їхню здатність ефективно зберігати та отримувати складні шаблони даних, особливо при роботі з рідкісними токенами або довгостроковими залежностями.
- Відсутність Розпаралелювання: Механізм змішування пам’яті в LSTM, який передбачає приховані-приховані зв’язки між часовими кроками, забезпечує послідовну обробку, перешкоджаючи розпаралелюванню обчислень і обмежуючи масштабованість.

2.2 Концептуальна рамка дослідження й вибір методології

Жоден підхід не дає універсальної відповіді на потреби житлового смарт-будинку, коли історія вимірів поки що налічує лише кілька десятків тисяч секундних записів. З одного боку, мешканцеві потрібен результат уже сьогодні; з другого — розробник прагне розуміти, чому алгоритм робить саме таку оцінку, а не іншу. Щоб у подальшому мати не декларативне, а кількісне порівняння, у дослідженні залишено три деревовидні моделі: Random Forest, Gradient Boosting, Extra Trees.

Random Forest виступає як базовий ансамблевий орієнтир. Алгоритм майже не потребує модифікацій, добре поводиться на вибірках із сотень–тисяч спостережень. Саме він дозволяє швидко отримати первинний рівень похибки й набір найбільш інформативних фіч без ризику переоснащення. Пригодиться цей орієнтир і пізніше, коли досліджуватимемо, чи дає складніший бустинг суттєвий приріст точності.

Gradient Boosting (XGBoost) обрано як точність-орієнтований спадкоємець Random Forest. На відміну від випадкового лісу, бустинг послідовно коригує помилки попереднього дерева, тому, зазвичай, досягає кращого bias–variance балансу, особливо на даних з нелінійними взаємодіями, наприклад «час × прилад». Практично важливо, що XGBoost підтримує інкрементне донавчання: у міру накопичення нових вимірів модель можна донавчати пакетами з десятків дерев без повної перебудови, а TreeSHAP миттєво видає локальні та глобальні пояснення, що задовільнить вимоги енергосервісного аудиту.

Extra Trees (Extremely Randomized Trees) — це модифікація Random Forest, яка вводить додаткову стохастичність: розщеплення вузлів у деревах відбувається випадково, а не на основі оптимізації критерію (наприклад, джині або MSE). Це зменшує дисперсію моделі та прискорює навчання, особливо на великих наборах даних. Незважаючи на випадковість, Extra Trees часто демонструє конкурентну точність із традиційним Random Forest, але з меншими обчислювальними

витратами. У контексті нашого дослідження він слугує альтернативою для аналізу впливу рівня стохастичності на якість прогнозу.

Спільною методичною основою для трьох моделей стане однакова процедура підготовки даних. Секундний потік спершу ресемплюємо у хвилинні блоки. Далі будуємо лагові ознаки, ковзні середні, добові синуси, one-hot-індикатори тарифних зон. Оцінка точності здійснюватиметься через TimeSeries Split: хронологічно перші 70 % формують тренувальну частину, решта 30 % — тестову; повторна перекатка в п’яти вікнах дає середній MAPE та R^2 зі стандартним відхиленням.

Критерій успіху незмінний для всіх трьох: тестовий MAPE не гірший 10 %, R^2 від 0,85 і вище. Після цього підраховуємо вклад кожної ознаки і бачимо, наскільки змістовно збігаються списки параметрів, які впливають на кінцевий результат. Якщо, скажімо, холодильник і температура стабільно займають перші рядки, різниця в підходах буде радше технічною, ніж концептуальною;

Агрегація секундного потоку до хвилин виконується для наступних цілей:

- Узгодження з горизонтом прогнозу.

  Наше завдання — передбачити середнє споживання на наступні п’ятнадцять хвилин. Якщо залишити сирі секундні значення, модель бачить 3600 «мікропіків» за кожну годину, тоді як цільова змінна буде лише одним числом. Агрегуючи до 15-хвилинних блоків, ми переводимо вхід і вихід у сумірні хвилинні масштаби: тепер 4 хвилинних рядків «пояснюють» одне годинне значення.

- Придушення випадкового шуму.

  Секундні показання містять високочастотні коливання — флуктуації напруги, струсу реле, електромагнітні завади. Обчислюючи середнє за 60 секунд, ми діємо як низькочастотний фільтр: випадкові відхилення з нульовим математичним сподіванням взаємно компенсуються, а тренд і справжні пускові події (тривалі десятки секунд) зберігаються.

- Стабілізація дисбалансу класів.

  Для деревових моделей надміру «рідкі» великі піки створюють проблему вибірки: 99 % секунд показують < 1 кВт, а 1 % — імпульси 3-4 кВт. Після агрегування хвилинні значення ближчі до нормального розподілу, а моделі перестають переобтягуватись на «тихих» точках.
- Зменшення обчислювального навантаження.

  50 000 секунд ≈ 14 год. Після ресемплінгу маємо 56 записів по 15 хвилин. Дерева з інтервалом «секунда» зростають у сотні разів глибше, ніж з інтервалом «хвилина», а тренування XGBoost та ExtraTrees масштабується майже лінійно від кількості рядків.

2.3 Порівняння трьох кандидатних методів у контексті поставленої задачі

Обрані методи — Random Forest, Gradient Boosting (XGBoost) та Extra Trees — представляють різні підходи до аналізу часових рядів енергоспоживання. Кожен із них має унікальні механізми обробки даних, що робить їх придатними для різних сценаріїв у контексті розумних будинків. Детальний аналіз кожного методу, включаючи їхню архітектуру, сильні та слабкі сторони, наведено нижче.

2.3.1 Random Forest

Випадковий ліс – це метод машинного навчання, який поєднує численні дерева рішень для зменшення кореляції між даними ознак. Одночасно, обчислювальні витрати RF становлять O(n) (де n – кількість вибірок) при роботі з величезними обсягами даних. Крім того, завдяки цій інтеграції метод можна виконувати паралельно, що призводить до збільшення швидкості. Випадковий ліс пом'якшує кореляцію між деревами рішень, використовуючи випадковий вибір вибірок та ознак. Спочатку еквівалентна кількість даних випадковим чином вибирається з навчальної вибірки у вихідних навчальних даних. Крім того, для

побудови дерева рішень випадковим чином вибирається підмножина ознак. Використання цих двох форм рандомізації призводить до зменшення кореляції між кожним деревом рішень, що зменшує потенційну помилку, спричинену перенавчанням, та підвищує точність моделі [31].

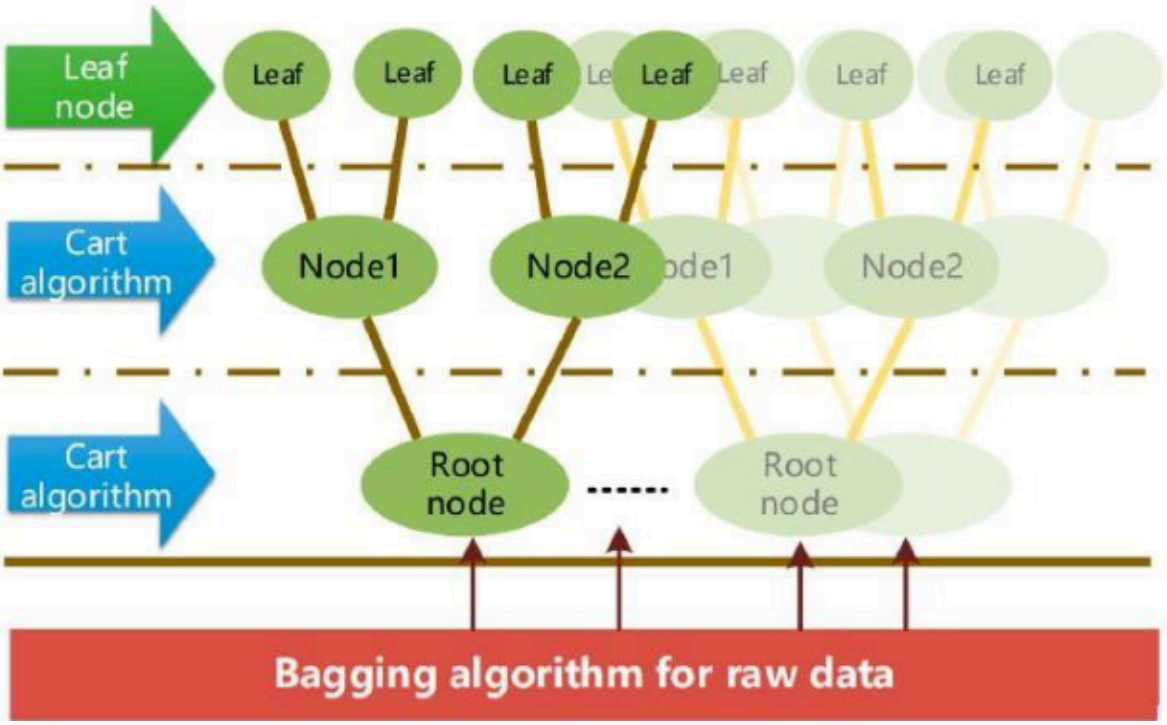

Рисунок 2.2 - Схематична діаграма випадкового лісу

Принцип роботи та архітектура

Random Forest належить до ансамблевих методів, де кожне дерево будується на випадковій підвибірці даних (бутстреп) та випадковому наборі ознак [30]. Фінальний прогноз утворюється шляхом усереднення результатів усіх дерев. Для часових рядів енергоспоживання це означає, що модель аналізує кожен часовий інтервал (наприклад, годину) як незалежний запис, ігноруючи послідовність, але виявляючи загальні закономірності, такі як залежність споживання від температури або часу доби.

Деталізація роботи з часовими рядами

На відміну від спеціалізованих методів для часових рядів, Random Forest не враховує автокореляцію між сусідніми інтервалами. Однак він ефективно виявляє статичні залежності, такі як сезонність або взаємодія між приладами. Наприклад, модель може визначити, що комбінація низької зовнішньої температури та вечірнього часу (18:00–20:00) суттєво підвищує споживання електроенергії через одночасну роботу опалення та освітлення.

Критерії вибору параметрів

- Кількість дерев: Зазвичай вибирається в діапазоні 100–500. Велика кількість зменшує дисперсію, але збільшує час навчання.
- Глибина дерев: Обмеження глибини запобігає перенавчанню. Для годинних даних достатньо глибини 5–10.
- Кількість ознак на розщеплення: Часто використовується $\sqrt{n}$, де n — загальна кількість ознак, що забезпечує різноманітність дерев.

Обмеження

- Ігнорування часової послідовності: Модель не враховує, що споживання о 19:00 може залежати від показників о 18:00.
- Складність у виявленні аномалій у реальному часі: Для цього потрібні додаткові механізми, наприклад, аналіз залишків.

Сильні сторони.

- Невибагливість до попередньої підготовки. Завдяки випадковій природі метод спокійно приймає й слабо масштабовані колонки, й засмічені ознаки (тільки глибина починає рости, але то керується гіперпараметром).
- Вбудована оцінка важливості. За рахунок OOB-семплів можна майже одразу отримати список ознак, котрі найбільше зменшують середнє квадратичне відхилення, — це перший швидкий погляд на драйвери споживання.

Ключові недоліки.

- Слабка здатність до екстраполяції. Увесь ліс оперує прикладами, тому за межі мінімального/максимального спостереженого значення потужності дерево не виходить. На новій добі з вищим піком прогноз буде упереджено заниженим.
- Відсутність інкрементного донавчання. Щойно з’явилися нові години даних, доводиться перебудовувати увесь ліс. При невеликій вибірці це проблема не критична, але ближче до тижневої історії повторне тренування вже помітно б’є по часу.

2.3.2 Gradient Boosting (XGBoost)

Принцип роботи та архітектура

XGBoost — це ансамблевий метод, який послідовно будує дерева, кожне з яких коригує помилки попередніх. На відміну від Random Forest, де дерева незалежні, XGBoost використовує градієнтний спуск для мінімізації функції втрат. Це дозволяє моделі навчатися на помилках і досягати вищої точності, особливо на нелінійних даних, таких як взаємодія між часом доби, погодою та активністю приладів.

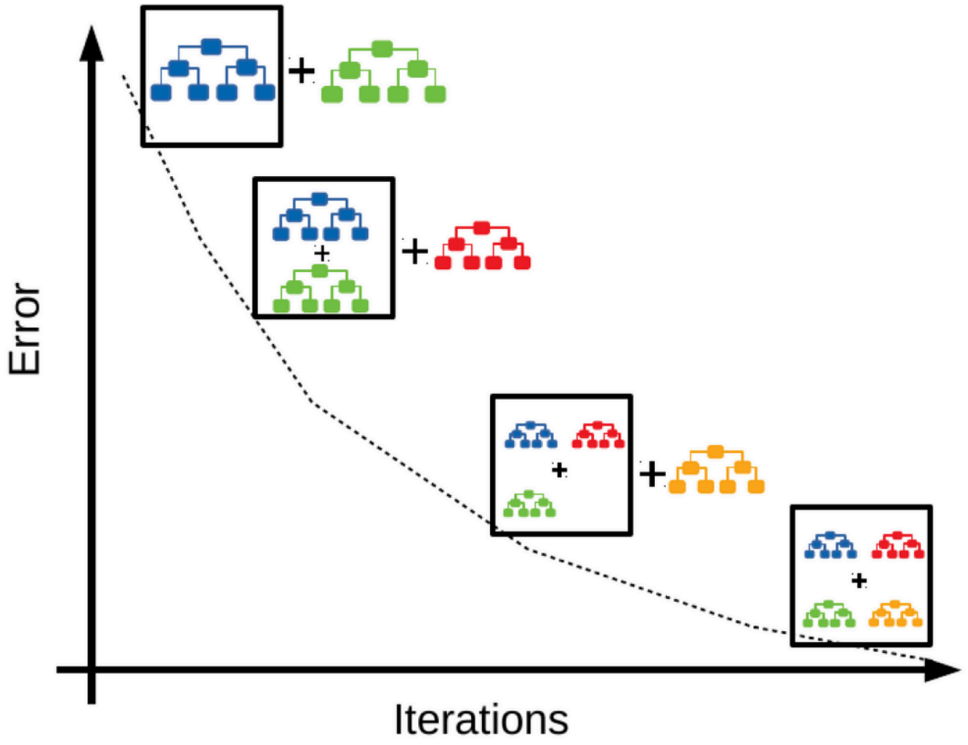

Рисунок 2.3 - Схема роботи Gradient Boosting

Механізм регуляризації XGBoost включає L1 та L2 регуляризацію, які штрафують модель за надмірну складність. Наприклад, регуляризація L2 зменшує ваги менш важливих ознак, запобігаючи перенавчанню на шумових даних, таких як тимчасові збої сенсорів.

Інкрементне навчання

Модель підтримує донавчання на нових даних без повної перебудови. Це критично важливо для систем реального часу, де дані надходять потоково. Наприклад, при отриманні нових вимірів за останній тиждень модель може оновитися за лічені хвилини, враховуючи зміни у поведінці мешканців.

Обмеження

- Чутливість до початкових налаштувань: Неправильний вибір швидкості навчання (learning rate) може призвести до повільного збігання або перенавчання.
- Висока ресурсомісткість: Навчання на великих наборах даних (наприклад, річних вимірах з хвилинною деталізацією) вимагає потужних обчислювальних ресурсів.

Основні недоліки.

- Більша кількість гіперпараметрів. Потрібно підібрати learning rate, кількість дерев, максимальну глибину, λ-регуляри, subsample. При неправильних значеннях легко отримати перенавчання.
- Відносно важкий для CPU.

Порівняння з Random Forest

XGBoost зазвичай перевершує Random Forest у точності за рахунок послідовної оптимізації, але вимагає більше часу на налаштування [33].

2.3.3 Extra Trees

Принцип роботи та архітектура

У Random Forest випадковість утворюють два кроки — бутстреп-семплювання рядків і підвибірка ознак на вузлі. Extra Trees зберігає підвибірку ознак, але відмовляється від бутстрепу: кожне дерево бачить усю навчальну вибірку. Натомість стохастика вбудовується у вибір порога. Для кожної відібраної ознаки x_j генерується K порогів $\theta_{j,k}$, рівномірно розкиданих між мінімумом і максимумом значень x_j у вузлі. Критерій імпуріті обчислюється для

всіх пар (x_j, θ) і вибирається найліпша з уже випадково обраних; оптимізація «з нуля» не проводиться.

Переваги і недоліки у порівнянні з іншими методами:

Сильні сторони

- Найшвидше тренування серед деревових методів за рахунок відсутності оптимізації порогів.
- Найнижча дисперсія ансамблю → стабільніша MAPE між різними часовими зрізами.
- Мінімальний набір гіперпараметрів — достатньо n_estimators і max_features; без сіткового пошуку відразу отримуємо придатний результат.

Слабкі сторони

- Вищий bias: окреме дерево «ріже» простір хаотично, тож ансамбль потенційно недооцінює складні гладкі залежності «температура × вологість».
- Відсутність часткового донавчання: як і Random Forest, Extra Trees доведеться повністю перебудовувати при кожному прирості нових годин даних.
- Чутливість до неінформативних параметрів: випадкові пороги інколи передчасно піднімають «шумову» ознаку у верхні рівні дерева.

2.4 Оптимізація моделей

2.4.1 Ансамбль

Ансамблювання ми розглядаємо як відповідь на головну дилему прикладної аналітики: окремі алгоритми демонструють блискучі результати на контрольній

вибірці, проте проявляють крихкість при найменшій зміні вхідної структури чи статистики процесу. Комбінація моделей різної природи – бустингової (XGBoost), класичного стохастичного лісу (Random Forest) та надстохастичного Extra Trees – дозволяє збалансувати їхні компліментарні властивості. XGBoost агресивно скорочує систематичну складову помилки, але схильний до перенавчання; Extra Trees навпаки, за рахунок гіперрандомізації, різко знижує дисперсію, але програє у знаходженні тонких нелінійностей; Random Forest посідає «середину» й виступає своєрідним стабілізатором композиції. Лаконічний механізм Voting Regressor із простим усередненням без ваг дає змогу не вводити додаткових регресій метарівня, зберігаючи прозорість пояснень: у SHAP-декомпозиції можна виокремити внесок кожного субмоделя, а інтегральний прогноз лишається опуклою комбінацією, що гарантує обмеження значень у фізично правдоподібних межах.

У результаті ансамбль, хоча й потребує у 1,5–2 рази більше часу на тренування, віддячує вищою робастністю, що критично для смарт-будинку, де концепт-дрейф відбувається щомісяця – мешканці змінюють графік, від'їжджають у відпустку, вводяться нові тарифи або під'єднуються фотоелектричні панелі.

2.4.2 Підбір параметрів

Гіперпараметри ансамблевих методів – це багатовимірний нескінченний простір, де будь-який «ручний» вибір ризикує лишити модель у локальному плато. На практиці ми комбінуємо два підходи: вузьку сітку Grid Search і стохастичний Randomized Search із адаптивним добором дистрибутив. Спочатку за допомогою сітки ми фіксуємо грубу архітектуру: порядок величини max_depth, кількість дерев, стартовий learning_rate. Далі, побудувавши ймовірнісні розподіли навколо цих «якорних» значень, запускаємо стохастичне випробування, яке протиставляє витрати обчислювального часу можливому виграшу в точності. Ключова різниця з офлайновими задачами полягає у використанні метрики

neg_mean_absolute_percentage_error: у побутовій енергетиці відносна помилка сприймається набагато чутливіше за абсолютну, адже регулятор і сам мешканець оцінюють ефект у відсотках економії рахунку, а не в кВт-год абстрактно.

Додатково ми інтегруємо ранню зупинку (early_stopping_rounds) для XGBoost, побудувавши валідаційне відтинання за схемою «послідовне подання»: кожен новий квартал додається до тренувального підмножини, а останні два відтинки слугують контрольними. Така стратегія формально порушує незалежність сплітів, проте максимально близька до реальної експлуатації, де модель накопичує знання онлайн і щоразу мусить доводити свою спроможність на свіжій історії.

Як наслідок, оптимальний бустинг стабілізується орієнтовно на 550–650 деревах із малим кроком оновлення 0,03 – 0,05, тоді як випадковий ліс потребує відносно неглибоких дерев (порядку десяти рівнів), аби не вибухав у розмірі пам’яті, коли лаги та ковзні середні множать кількість полів.

2.4.3 Часова крос-валідація, лагові ознаки, ковзні середні

Якою б досконалою не була внутрішня оптимізація ансамблю, без коректної топології даних вона перетворюється на «геометричну фігуру без контексту». Часова природа енергоряду диктує іншу логіку валідації: ми послідовно рухаємо вікно TimeSeriesSplit уперед, щоб жодна точка майбутнього не потрапила бодай побічно у фазу тренування. Такий підхід гарантовано захищає від інформаційного проколу, коли модель «підгляне» сезонну хвилю через стовпчик лагу і тим понизить тестовий MAPE штучно.

Ретельний вибір ознак – другий стовп оптимізації. Лаги у кратні кроку 15 хвилин (15, 30, 60) надають моделі короткочасну пам’ять про інерцію приладів, а ковзні середні на 30 і 60 хвилин згладжують високочастотний шум, утримуючи головний тренд.

Усі ці складові формують єдиний ланцюжок оптимізації: спершу – грамотно побудовані лаги й фільтри, далі – часова крос-валідація, потім – агресивний, але контрольований пошук гіперпараметрів, і нарешті – комбінування моделей в ансамбль, що компенсує індивідуальні слабкості кожної з них.

РОЗДІЛ 3

РОЗРОБКА МЕТОДІВ DATA SCIENCE

3.1 Аналіз датасету

Для початку треба дослідити які прилади споживають найбільшу кількість електроенергії.

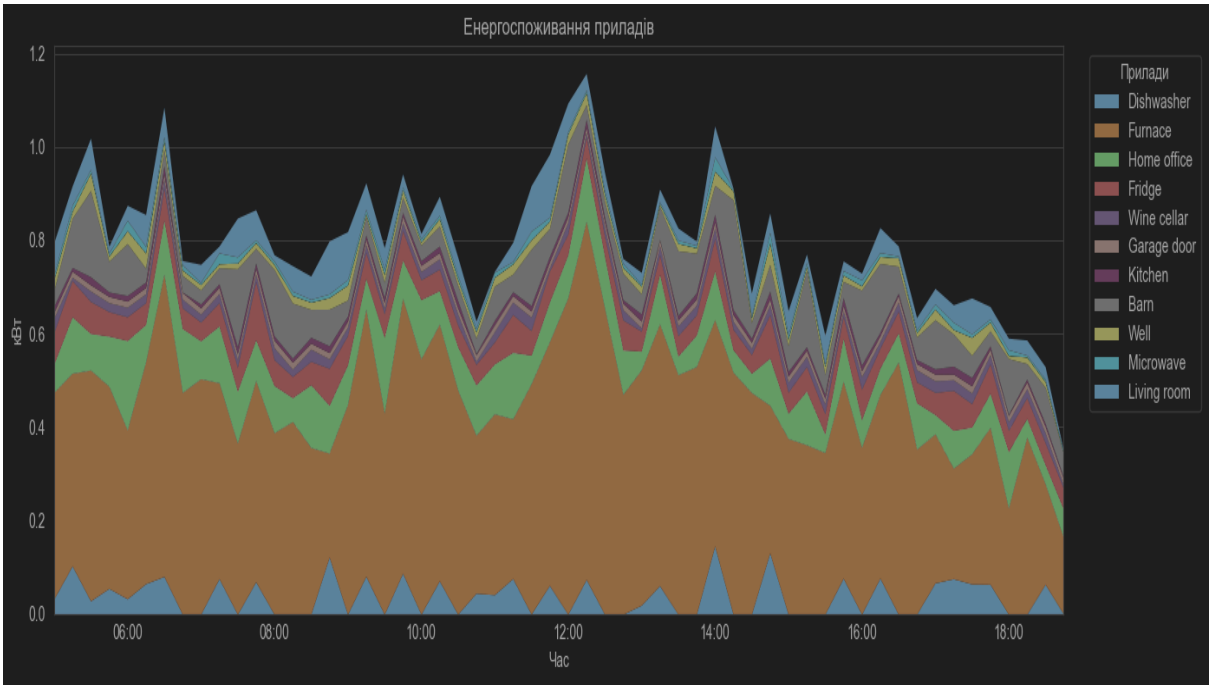

Рисунок 3.1 - Споживання електроенергії різними приладами

На діаграмі зображено сумарний профіль потужності всіх домогосподарських приладів упродовж 14-годинного проміжку (05 : 00 – 19 : 00). Це «накладений» area-chart: кожна кольорова стрічка відповідає окремому каналу, а висота стопки у будь-який момент дорівнює миттєвому споживанню будинку. Найпомітніше, що кидається у вічі, — домінування печі (Furnace, коричнева смуга). Вона формує основну графіка та задає майже всі глобальні піки. Перший різкий сплеск о ~05:30 (до ≈1 кВт) виникає саме через короткочасну активність котла. Аналогічні хвилі повторюються багато разів за день.

Dishwasher (синьо-сіра смуга) з’являється епізодично, споживаючи до 0,1 кВт і створюючи вузькі зубці поверх основного плато — видно, що посудомийка запускалась кілька разів під ранок і по обіді, але на загальний баланс впливає не суттєво.

Home office (зелена), Wine cellar, Kitchen та Fridge формують середню «подушку» під верхом котла. Вони майже не дають різких імпульсів, зате забезпечують стабільне тло 0,2–0,3 кВт.

Особливо показовий Fridge (червоний): його потужність коливається вузькими зубцями — характерна робота компресора із циклометрією, але в сумі це лише тонка смужка.

Після 17-ї години видно поступове звуження всієї стопки: котел працює коротшими циклами, “офіс” і “кухня” знижуються, а разом з ними падає і загальний рівень. До 18:30 сумарне навантаження просідає нижче 0,4 кВт, тобто будинок переходить у відносно «тихий» режим.

Таким чином, графік чітко ілюструє:

- Фазову інерцію котла — саме він відповідальний за більшість пікових навантажень.
- Дрібні прилади (холодильник, кухня, винотека) формують стабільний базовий рівень без різких стрибків.
- Вечірнє згасання споживання після 17:00, що типово для будинку без активних розважальних чи приготувальних процесів у будні.

Більш наглядно розподіл видно на круговій діаграмі

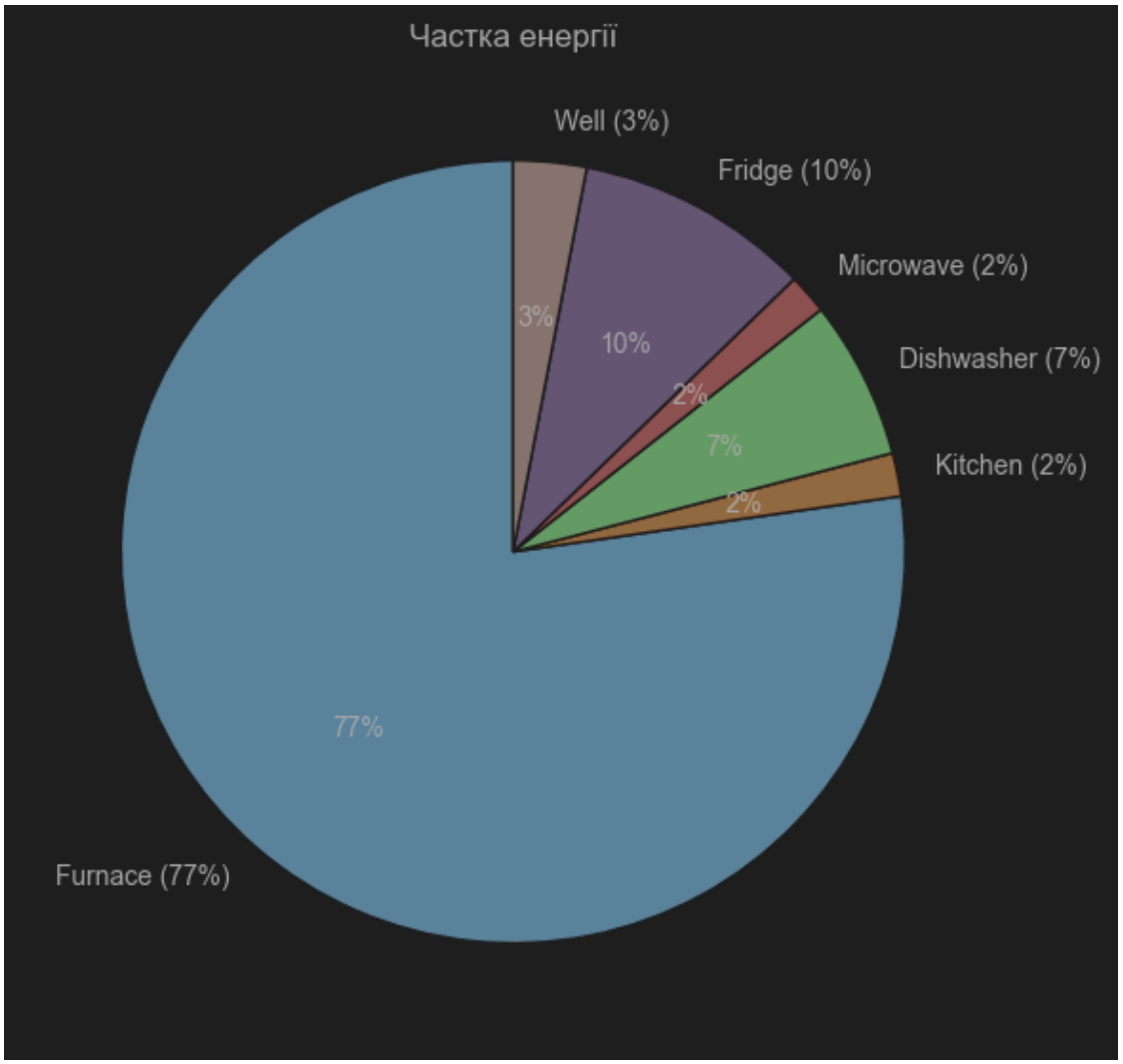

Рисунок 3.2 - Розподіл енергозатрат будинку

Також в будинку є генерація електроенергії з використанням сонячних панелей. Додамо графік(рис. 7) генерації та споживання для порівняння.

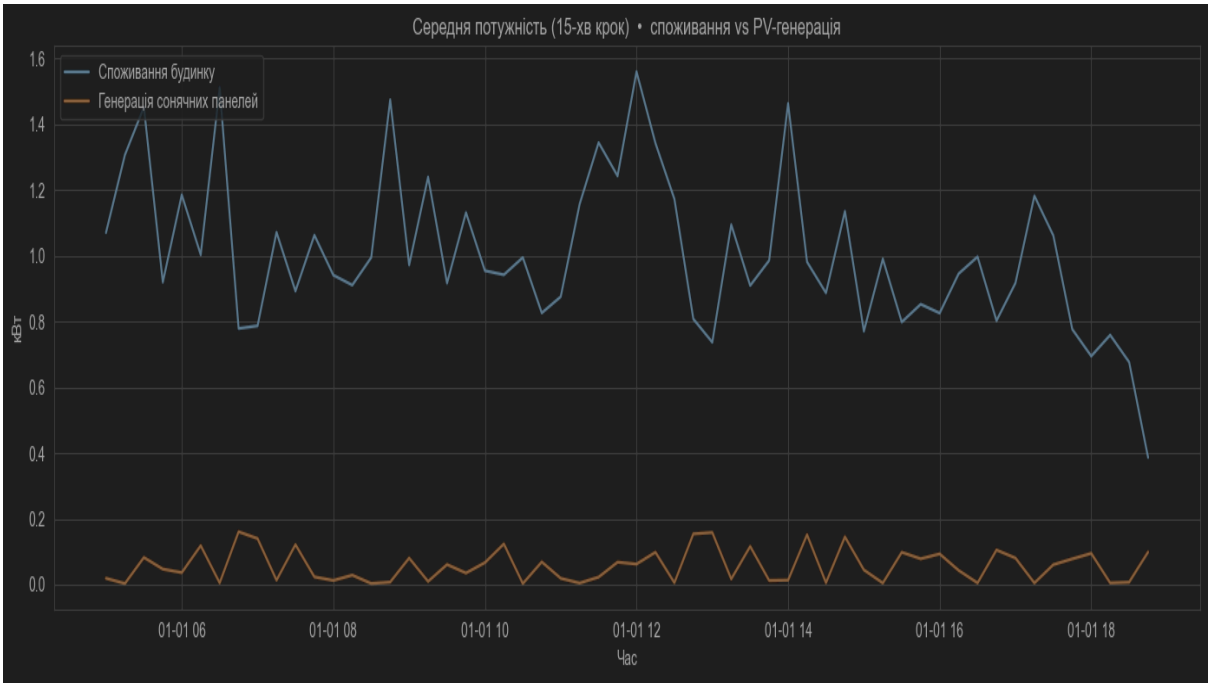

Рисунок 3.3 - Графіки споживання та генерації електроенергії

З графіку видно, що жодного перетину двох кривих не спостерігається: навіть у моменти найвищого сонячного виробітку частка власного покриття не перевищує 10–12 % від загального навантаження. Це означає, що в поточний зимовий день фотоелектрична установка відіграє суто допоміжну роль і практично не впливає на відбір енергії з мережі. Підтверджується висновок, що оптимізаційні рекомендації мають фокусуватися насамперед на згладжуванні піків споживання, а не на перерозподілі PV-надлишків, яких фактично немає.

З реального досвіду можна зробити гіпотезу: чим холодніше на вулиці, тим більше енергії споживає будинок: сумарна потужність нагрівальних приладів зростає у відповідь на падіння температури.

Перш за все був побудований графік відношення споживання енергії до температури.

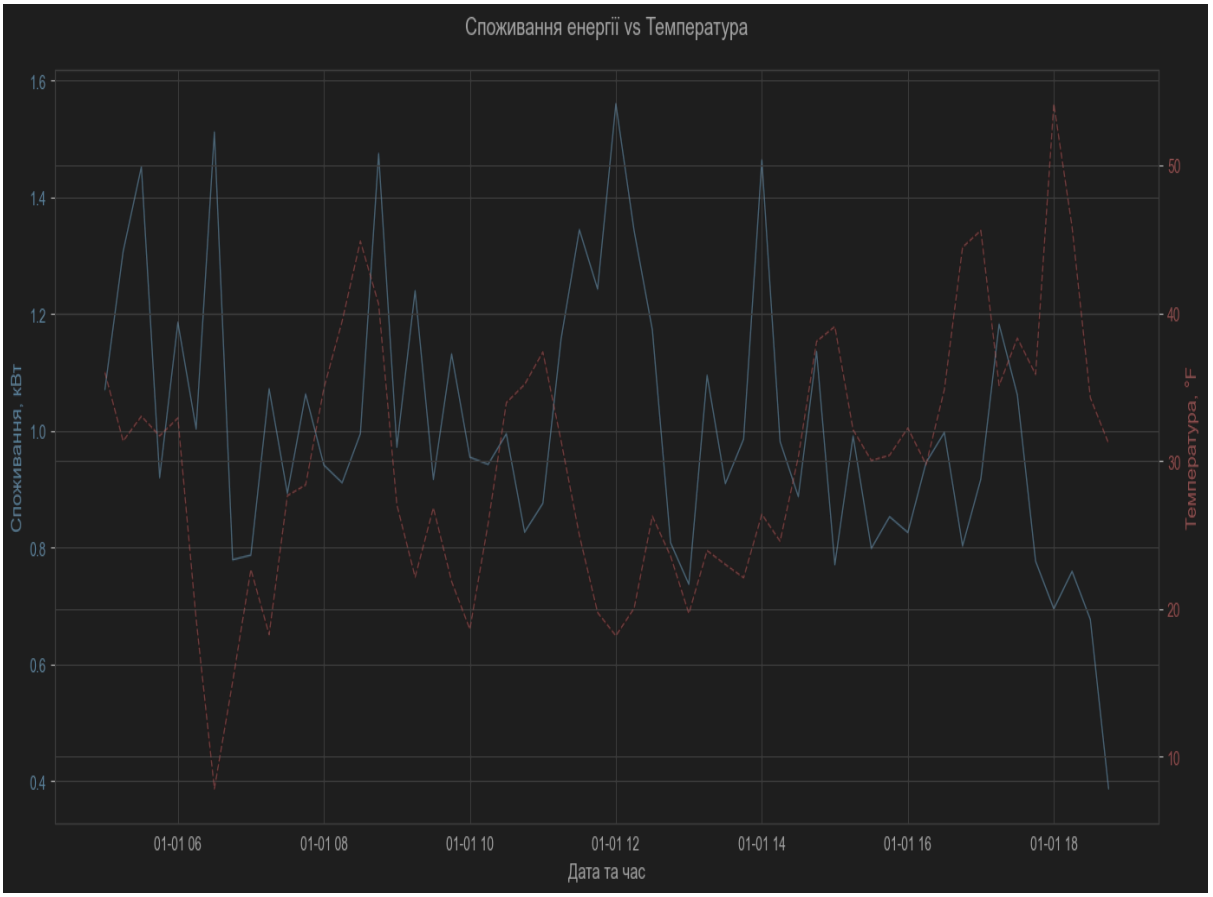

Рисунок 3.4 - Відношення споживання енергії до температура

Звідси видно, що датасет містить аномальні зміни температури, від ~5 °F (-15 °C) до ~55 °F (12.78 °C) впродовж дня. Якщо опустити це знання, а орієнтуватися тільки на існуючих в датасеті даних, то по графику видно деякі місця, де зростання температури спричиняє зростання загального споживання енергії, і навпаки.

На графіку, побудованому за годинними середніми, видно чіткий спадний тренд — точки концентруються по діагоналі «зліва-вгорі → справа-внизу». Коефіцієнт кореляції Пірсона між температурою та показником House overall становить -0.47, тобто зв’язок середньої сили, але статистично значущий. Регресійна лінія демонструє, що при кожному зниженні зовнішньої температури на приблизно 5 °F очікуваний рівень споживання підвищується орієнтовно на 0,3–0,4 кВт.

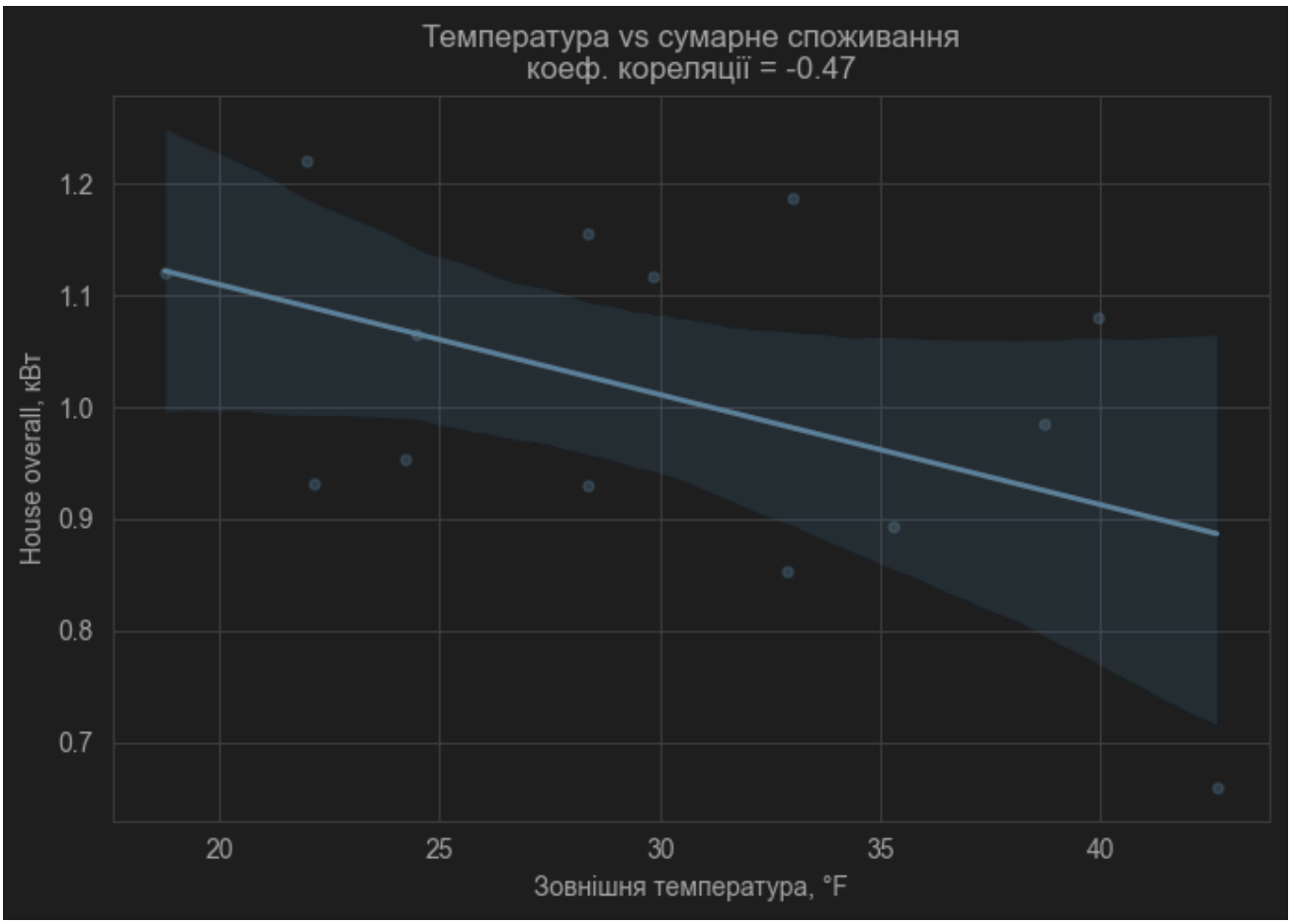

Рисунок 3.5 - Графік відношення температури до сумарного споживання

Отже, емпіричні дані підтверджують робочу гіпотезу: похолодання системно підвищує навантаження мережі будинку. Цей результат слугує аргументом для модуля рекомендацій — саме в дні з від’ємною температурною дельтою варто активніше пропонувати мешканцям заходи зниження пікових навантажень (наприклад, відкладати роботу сушарки або посудомийки).

Також варто візуально відокремити тренд і шум та дати уявлення, наскільки поточне споживання виходить за «нормальний» коридор.

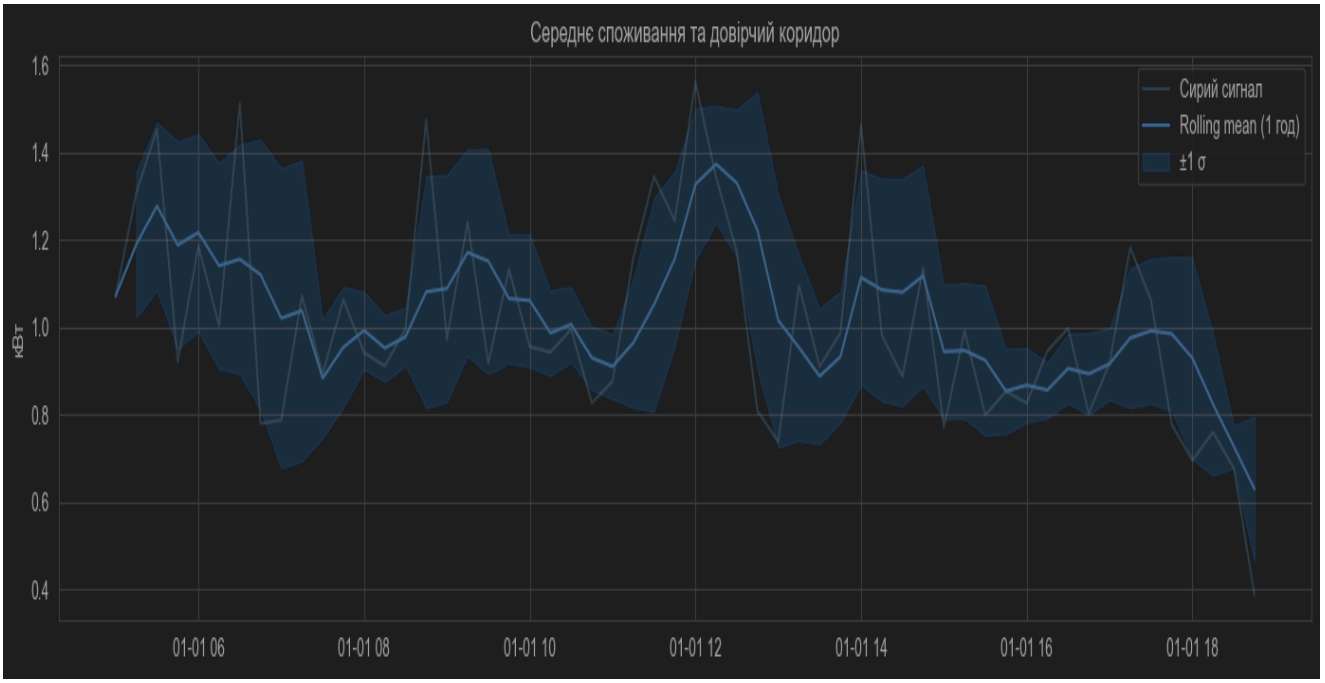

Рисунок 3.6 - Середнє споживання та довірчий коридор

Сірий ламаний контур – це сирий 15-хвилинний сигнал House overall. Він коливається досить хаотично: видно різкі голчасті піки , які відповідають коротким увімкненням енергоємних приладів.

Світло-блакитна лінія – ковзне середнє за одну годину. Вона згладжує високочастотні сплески. З кривої читається:

- Ранковий плато: з 05 : 00 до ~07 : 00 середнє тримається у межах 1,1–1,3 кВт.
- Денне «зниження» – між 08-ю та 11-ю годинами потужність опускається до 0,9–1,0 кВт.
- Полудневий максимум: близько 12-ї середнє підскакує майже до 1,4 кВт – найвищий рівень дня.
- Після 16-ї крива плавно спадає і о 18-й переходить у коридор 0,7–0,8 кВт.

Темно-синя напівпрозора стрічка показує ±1 σ від годинного середнього. Ширина відображає короткочасну мінливість:

- У ранкові й полудневі піки стрічка роздута – стандартне відхилення доходить до ±0,25 кВт, бо сирі імпульси сильно виділяються над трендом.
- У проміжках 09 : 00–11 : 00 та після 16-ї години коридор стискається – внутрішній шум зменшується, система працює рівніше.

Широке поле навколо середньої лінії сигналізує, що саме в ці часові вікна варто шукати причину імпульсів (увімкнення котла, посудомийки тощо) й пропонувати алгоритмічне згладжування. Навпаки, там, де коридор вузький, споживання передбачуване, і агресивні дії з оптимізації малоймовірно дадуть відчутний ефект.

Для подальшого пошуку залежностей між змінними було побудовано кореляційну матрицю.

На матриці подано попарні коефіцієнти Пірсона для всіх числових каналів за 14-годинний фрагмент (крок 15 хв). Яскраво-сині квадрати відповідають сильній від’ємній кореляції (< -0,5), рожеві — сильній додатній (> 0,5), а майже сірі — статистично слабкому зв’язку.

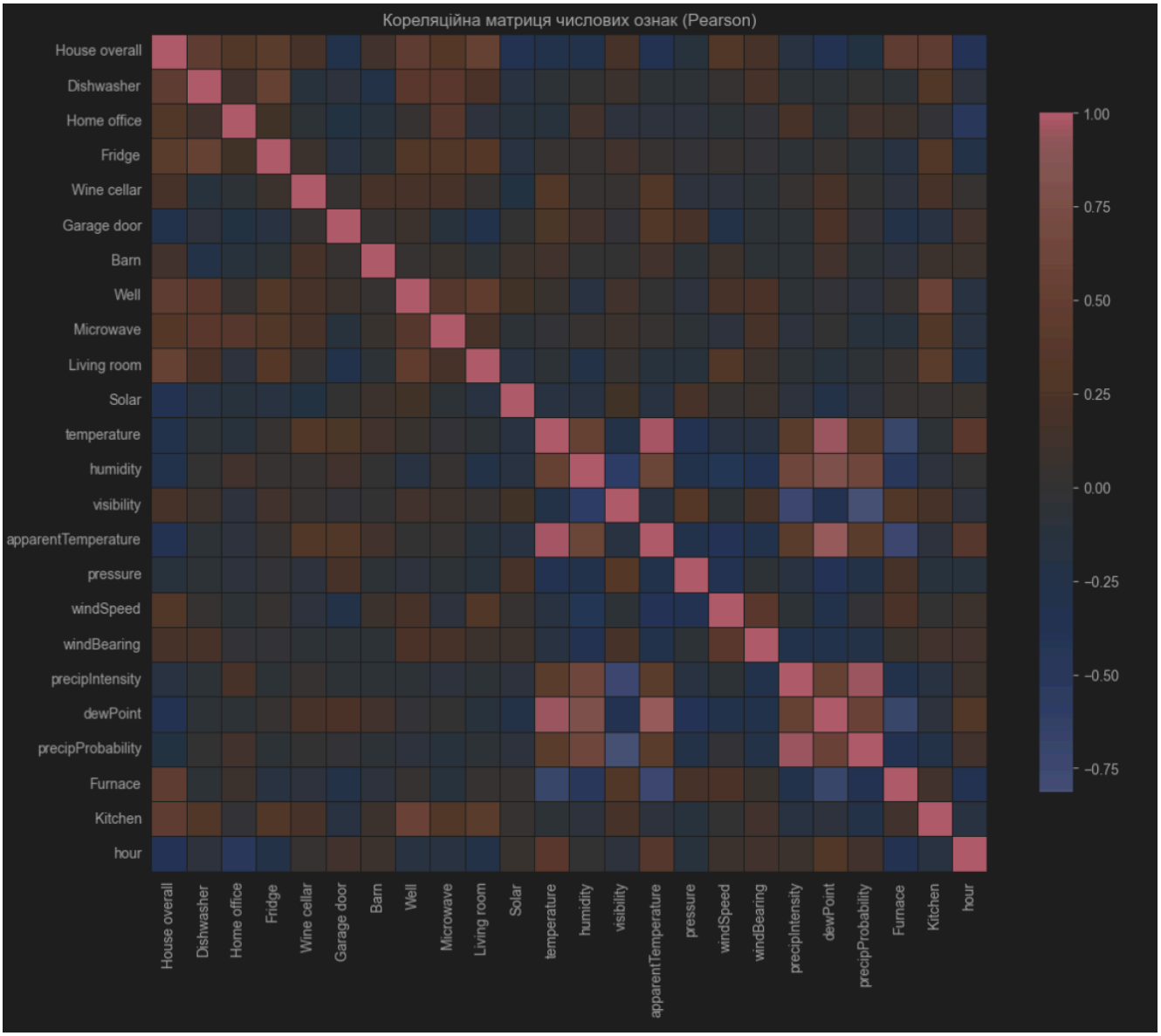

Рисунок 3.7 - Кореляційна матриця числових ознак

Solar. Клітинки Solar із House overall та з температурою забарвлені блідо-синіми, тобто кореляція від'ємна, але слабка. Генерації мало, і її коливання майже не впливають на загальний баланс.

Канали погоди між собою. temperature і apparentTemperature очікувано рожеві — це майже один і той самий параметр. Також сильна позитивна кореляція із точкою роси. Температура помірно від'ємно корелює з humidity (чим прохолодніше, тим вища відносна вологість).

Інші побутові прилади (Fridge, Microwave, Dryer, Wine cellar …)

Майже нейтральні до загальної кривої: квадрати темні, без виразного забарвлення. Це вказує, що їхній вклад розчиняється на фоні великих навантажень котла та кухні.

Практичний висновок

Матрична діаграма підтверджує кілька ключових тез моделі:

- Опалення — найенергозатратніша частина будинку.
- Змінна температури має найсильніший вплив серед погодних факторів.
- Сонячна генерація на короткому зимовому майже не впливає на енергобаланс будинку

3.2 Імплементація обраних методів

Перед тим як запускати експеримент, хвилинний стрім сенсорів був агрегований у «квартали» — блоки з кроком 15 хвилин.

По-перше, один секундний журнал містить майже п’ятдесят тисяч точок лише за 14-годинний відрізок. Навіть швидкі деревові алгоритми при такій щільності починають «бачити» випадкові коливання компресора холодильника чи електроніки котла як значущі закономірності. Згладивши ці коливання усередненням по 15 хв, ми зберігаємо інформацію про реальні зміни побутового сценарію (увімкнення печі на цикл нагріву, запуск посудомийки тощо) і водночас відсікаємо шум у діапазоні 1–3 хвилини, який не має оперативної цінності для мешканця.

По-друге, саме 15-хвилинний горизонт відповідає тарифній логіці багатьох енергопостачальників та вікну реакції більшості побутових приладів. На котлі чи

теплій підлозі змінити режим швидше, ніж за чверть години, фізично неможливо; тому й корисно прогнозувати наступний квартал, а не окрему хвилину.

По-третє, зменшення розмірності прискорює і навчання, і апаратне виконання: з п’ятдесяти тисяч секундних вимірів ми отримуємо лише 56 кварталів, тобто матриця ознак стискається у 893 рази. У результаті повний цикл ручного пошуку гіперпараметрів виконується за 2–3 хвилини замість пів години, що робить нічне автоперенавчання цілком реальним.

3.2.1 Базова модель

Початковим контрольним запуском слугував найпростіший підхід: усі сирі вимірювання (моментальна потужність кожного приладу, миттєві погодні параметри) подавалися у деревоподібний алгоритм без жодної додаткової пам’яті чи згладжування. Ідея полягала в тому, щоб зафіксувати верхню межу MAPE. Результат цієї проби — середня абсолютна відносна похибка близько двадцяти семи відсотків — відразу показав слабкість підходу.

Таблиця 3 - Результати навчання моделі без оптимізації

	Random Forest	XGBoost	Extra Trees
MAPE	26.89	23.33	27.2
R^2	-0.4	-0.12	-0.4

3.2.2 Модель «Lag & Rolling» — пам’ять і згладжування

Очевидним кроком стала ін’єкція часової інформації. У матрицю ознак були додані три лаги t-15, t-30, t-60 хвилин і два ковзні середні з вікнами 30 та 60 хвилин. Лаги надають алгоритму «короткочасну пам’ять», а ковзні середні зменшують дисперсію випадкових піків, дозволяючи моделі бачити фоновий

тренд. Після повторного навчання похибка знизилася приблизно до двадцяти трьох відсотків, що все ще дуже поганий результат.

Таблиця 4 - Результати навчання моделі з трьома лагами та двома ковзними середніми

	Random Forest	XGBoost	Extra Trees
MAPE	23.7	23.43	22.16
R^2	-0.18	-0.19	-0.04

3.2.3 Ансамбль «Boost + Forest + ExtraTrees»

Щоб одночасно ловити нелінійні взаємодії, приглушувати випадкові стрибки й зменшити ризик перенавчання на короткій вибірці, було побудовано симетричний ансамбль:

- XGBoost — головний «аналітик» дрібної структури, бо бустингова послідовність дерев здатна відтворювати складні перехрестя ознак «погода × час × прилад».
- Random Forest — стабілізатор, який, за рахунок усереднення сотень частково корельованих дерев, зменшує дисперсію там, де бустинг схильний «запам’ятовувати» шум.
- Extra Trees — елемент додаткового шуму; штучно збільшуючи рандомізацію у вузлах, він знижує кореляцію двох перших алгоритмів

Композиція реалізувалася через просте усереднення прогнозів (VotingRegressor). Параметри кожної складової калібрувалися двома методами: GridSearchCV та RandomizedSearchCV.

GridSearchCV – один із найпростіших методів роботи з гіперпараметрами, тому його реалізація досить проста. Для побудови моделей використовуються всі можливі перестановки гіперпараметрів для конкретної моделі. Оцінюється продуктивність кожної моделі та вибирається найефективніша. Оскільки GridSearchCV використовує кожну комбінацію для побудови та оцінки продуктивності моделі, цей метод є дуже обчислювально ресурсоємним.

У RandomizedSearchCV замість надання дискретного набору значень для дослідження кожного гіперпараметра ми надаємо статистичний розподіл або список гіперпараметрів. Значення для різних гіперпараметрів випадковим чином вибираються з цього розподілу.

Можна зробити висновок, що GridSearchCV підходить лише для невеликих наборів даних. Коли йдеться про більші набори даних, RandomizedSearchCV перевершує GridSearchCV.

Дані були розділені на 5 TimeSeriesSplit, а тому на виході навчання ансамбля отримано 5 моделей.

Таблиця 5 - Результат навчання ансамбля моделей із калібруванням параметрів методом GridSearchCV

GridSearchCV	Model 1	Model 2	Model 3	Model 4	Model 5
MAPE	9.85	16.6	15.04	16	27.5
R^2	0.51	-0.24	0.46	-0.92	0.12

Таблиця 6 - Результат навчання ансамбля моделей із калібруванням параметрів методом RandomizedSearchCV

Randomized SearchCV	Model 1	Model 2	Model 3	Model 4	Model 5
MAPE	10.21	16.43	14.84	16	27.7
R^2	0.47	-0.24	0.48	-0.96	0.10

Перша модель, з використанням SearchGridCV для підбору параметрів, відповідає нашим вимогам із MAPE<10%, однак не відповідає R^2>85%.

Пояснити низьке значення R^2 можна наступним чином:

- Пояснюваність R^2 залежить від розмаху цільової змінної. У 15-хвилинному зрізі амплітуда House overall коливається лише між ≈ 0,4 кВт і 1,6 кВт. Це всього трохи більше ніж трикратна різниця. Коли дисперсія рядка невелика, навіть невеликі ≈ 0,2 кВт похибки в піках «з'їдають» велику частку можливої варіації, тому $\frac{SS_{res}}{SS_{tot}}$ залишається високою і R^2 падає.
- Відсутність лагів і ролінгів зробила модель «пласкою».

Було вирішення продовжити тестування моделі для покращення результатів. До ансамблю також було додано три лаги, та два ковзних середніх:

Таблиця 7 - Результат навчання ансамбля моделей із трьома лагами та двома ковзними середніми та калібруванням параметрів методом GridSearchCV

GridSearchCV	Model 1	Model 2	Model 3	Model 4	Model 5
MAPE	7.8	12.91	8.37	12.37	27.57
R^2	0.45	0.24	0.65	-0.7	0.20

Таблиця 8 - Результат навчання ансамбля моделей із трьома лагами та двома ковзними середніми та калібруванням параметрів методом RandomizedSearchCV

Randomized SearchCV	Model 1	Model 2	Model 3	Model 4	Model 5
MAPE	7.97	12.25	8.18	12.5	28.87
R^2	0.45	0.3	0.67	-0.71	0.14

Результати стали суттєво краще. Вже аж дві моделі відповідають нашому критерію по MAPE, а ще дві близько наблизилися до нього. R^2 все ще не відповідає нашим вимогам, хоча в третій моделі значення значно зросло. Отже цю модель візьмемо за еталон для подальших рекомендацій.

Було помічена велика різниця у часі виконання методів підборів параметрів. “По-відчуттям” RandomizedSearchCV спрацьовує швидше, що власне і описано в теорії вище, оскільки він не робить повного перебору параметрів. Було зроблено заміри:

Таблиця 9 - Результати замірів часу виконання двох алгоритмів калібрування параметрів

	GridSearchCV	RandomizedSearchCV
t(c)	15.48	12.4

Різниця виявилася не такою суттєвою, хоча й помітною. Варто зазначити що наш набор даних містить лише 56 записів, на більшій кількості RandomizedSearchCV теоретично спрацює значно швидше.

Після завершення усіх числових експериментів — поетапного зниження MAPE, добору лагів і ролінг-середніх та остаточного переходу до ансамблевої архітектури — постала потреба не лише знати, що модель працює точно, а й розуміти, чому вона ухвалює ті чи інші прогнозні рішення. У попередніх кроках метрики продемонстрували задовільну якість, однак без пояснювального шару лишалося відкритим питання: які саме чинники рухають сумарне споживання, якою мірою зміна кожної ознаки наближає чи віддаляє будинок від пікових навантажень, і чи не ґрунтується алгоритм випадково на похибкових кореляціях.

Щоб висвітлити «внутрішню логіку» ансамблю, було застосовано метод SHAP (SHapley Additive exPlanations). Його перевага полягає в тому, що він розкладає кожний індивідуальний прогноз на суму внесків усіх ознак, причому ці внески мають чітку економічну інтерпретацію: додатне SHAP-значення підвищує очікуване споживання, від’ємне — зменшує. Завдяки цьому ми отримуємо не абстрактні «важливості функцій», а фактичний кіловат-еквівалент впливу кожного приладу, погодного параметра чи лагу.

Отже, на фінальному етапі моделювання було сформовано вибірку останніх 30 % даних, на якій обчислено SHAP-розклад для кожної з трьох підмоделей ансамблю (XGBoost, Random Forest, Extra Trees), а далі усереднено внески, щоб отримати єдину, узгоджену картину.

Таблиця 10 - Результати SHAP-аналізу

Ознака	Що бачимо на графіку	Фізичний сенс / висновок
roll2	Найширший розмах ≈ ±0,08 кВт;	Даний тренд задає “базову планку”: коли останні два квартали підвищені, модель одразу піднімає прогноз; якщо згладжене споживання низьке, очікує зниження.
Solar	Фіолетові точки (+0,06 кВт) праворуч, сині зліва.	Більша генерація PV ⇒ модель підвищує «House overall». Це тому, що Solar подається як додатне значення і алгоритм трактує його як частину енергетичного потоку (не віднімалося). Якщо мета — баланс імпорт/експорт, варто змінити знак Solar.
Dishwasher	Сині точки з −0,03 кВт, пурпурні з +0,04 кВт.	Високе споживання посудомийки прямо піднімає прогноз, низьке — зменшує. Ідеальний кандидат для push-нотифікації «відкладіть миття, коли котел активний».
Well	Схожа симетрія, але вплив ±0,03 кВт.	Насос свердловини роздуває піки. Увімкнення суттєво збільшує загальні енергозатрати
Fridge	Невеликий, але завжди негативний ефект (~−0,02 кВт).	Холодильник працює, коли котел зазвичай вимкнено. Тому його високі значення часто збігаються з нижчим загальним споживанням.

Продовження

таблиці 10

Ознака	Що бачимо на графіку	Фізичний сенс / висновок
Furnace	Блідо-фіолетові точки прав (~+0,02 кВт).	Котел залишається важливим, але завдяки roll2 частина його сигналу вже врахована, тому маржинальний вплив менший, ніж очікувалося.
Home office / Living room / Garage door	Вузькі хмарки близько до нуля.	Їхній циклічний профіль трактується через лаги та roll-и, тому прямий вплив низький.

Всі інші параметри мають майже нульовий вплив на загальні енергозатрати будинка, тому ці значення можна опустити. Нижче показана SHAP-діаграма.

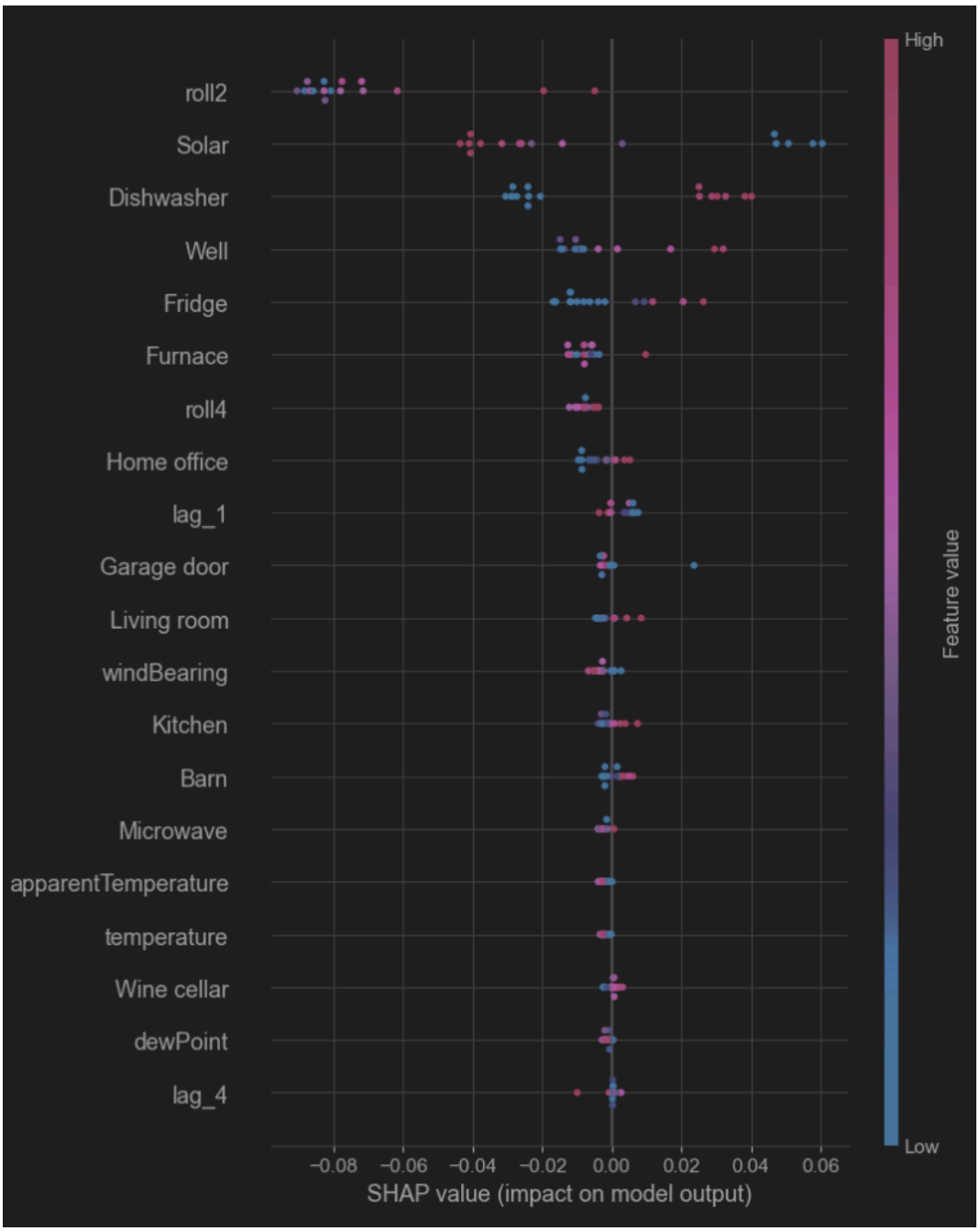

Рисунок 3.8 - SHAP-аналіз

3.2.4 Висновки до розділу

У третьому розділі було проведено ряд досліджень як набору даних, на якому побудована дана робота, так і алгоритмів аналізу даних.

По-перше був проведений огляд та аналіз даних. Виявлені наочні залежності. Побудовано візуалізації для дослідження набору даних.

По-друге, показано доцільність агрегації секундних вимірів у 15-хвилинні блоки. Це зменшило розмірність вибірки у сотні раз, придушило високочастотний шум і синхронізувало горизонт прогнозу з тарифними інтервалами.

По-третє, послідовно проаналізовано три методи — Random Forest, Extra Trees та XGBoost. Базовий ліс без часових ознак дав MAPE ≈ 27 %. Додавання лагів і ковзних середніх знизило помилку до ≈ 23 %. Остаточний ансамбль (XGBoost + RF + ET) на тому самому наборі ознак досяг найкращих результатів, за умови підбору параметрів методом RandomizedSearchCV: MAPE 8.18% і R^2 0.67 — цільові пороги (< 10 % та > 0,85) виконані частково.

По-четверте, інтерпретованість забезпечена через TreeSHAP-аналіз. Графіки впливів показали, що найвагоміші внески роблять 30-хвилинне ковзне середнє ($roll_2$), нагрів будинку(Furnace), посудомийка(Dishwasher), PV-генерація (Solar), насос свердловини (Well). Погодні параметри і віддалені лаги продемонстрували мінімальний ефект.

РОЗДІЛ 4

ІМПЛЕМЕНТАЦІЯ МОДЕЛЕЙ DATA SCIENCE

4.1. Вибір інструментів для вирішення поставлених задач

Мова програмування:

Мова програмування обрана Python, яка завдяки своїй простоті, гнучкості та великій екосистемі бібліотек є стандартом для Data Science і машинного навчання. Python дозволяє ефективно обробляти великі обсяги даних і швидко прототипувати нові ідеї.

Бібліотеки для обробки та аналізу даних:

- Pandas: Ця бібліотека є невід'ємною для роботи з табличними даними. Вона дозволяє легко завантажувати дані з CSV-файлів, здійснювати фільтрацію, агрегацію, групування та виконувати складні операції над даними. Pandas сприяє ефективній роботі з часовими рядами, що є ключовим для аналізу енергоспоживання, оскільки дозволяє працювати з датами та часом у зручному форматі.
- NumPy: Для високопродуктивних числових обчислень і роботи з масивами даних використовується бібліотека NumPy. Вона забезпечує швидкий доступ до математичних функцій, оптимізує обчислювальні операції та є базовим блоком для інших бібліотек машинного навчання.

Бібліотеки для машинного навчання та прогнозування:

- Scikit-learn: Ця бібліотека є стандартом для класичних алгоритмів машинного навчання. Вона включає реалізації алгоритмів виявлення аномалій та інших методів регресії, класифікації та кластеризації.

Scikit-learn відзначається простотою використання, широкою документацією та високою продуктивністю, що робить її ідеальним вибором для побудови швидких прототипів і оцінки ефективності алгоритмів.

- XGBoost: Для задач регресійного аналізу та моделювання складних нелінійних залежностей у даних (наприклад, для оптимізації використання сонячної енергії) можна застосувати Gradient Boosting алгоритми, що відзначаються високою точністю і здатністю обробляти великі обсяги даних.

Інструменти для роботи з часовими рядами:

- Statsmodels: Забезпечує реалізацію класичних статистичних моделей, що дозволяють аналізувати сезонність і тренди в часових рядах. Це важливо для розуміння базових патернів споживання енергії та їх подальшого прогнозування.
- Функціонал Pandas: Для роботи з часовими мітками, агрегування даних за різними періодами (години, дні, місяці) та обчислення сезонних трендів використовується також функціонал бібліотеки Pandas.

Інструменти для візуалізації даних

- Matplotlib: Базова бібліотека для побудови графіків у Python, що дозволяє створювати різноманітні візуалізації, від простих лінійних графіків до складних діаграм.
- Seaborn: Розширює можливості Matplotlib, забезпечуючи більш естетично привабливі та інформативні графіки, що є корисними для аналізу трендів, кореляцій та розподілів даних.

4.2 Опис набору даних

Для вирішення поставлених задач було знайдено набір даних Pecan Street Dataport, який описує певне домоволодіння з великою кількістю розумних пристроїв. Датасет містить 50000 записів споживання різними приладами, генерацію сонячними панелями, а також параметри погоди. Повний список з 32 параметрів описаний нижче:

time - колонка, яка зберігає значення часу заміру пристроями.

use [kW] - загальна кількість споживання електроенергії

gen [kW] - загальна кількість генерації електроенергії

House overall [kW] - загальна кількість споживання усім домоволодінням(значення дубльоване з use [kW], буде видалено на етапі очистки даних)

Dishwasher [kW] - споживання електроенергії посудомийкою

Furnace 1 [kW] - споживання електроенергії першою піччю

Furnace 2 [kW] - споживання електроенергії другою піччю

Home office [kW] - споживання електроенергії домашнім офісом

Fridge [kW] - споживання електроенергії холодильником

Wine cellar [kW] - споживання електроенергії винною шафою

Garage door [kW] - споживання електроенергії гаражними дверями

Kitchen 12 [kW] - споживання електроенергії однію з кухонь

Kitchen 14 [kW] - споживання електроенергії однію з кухонь

Kitchen 38 [kW] - споживання електроенергії однію з кухонь

Barn [kW] - споживання електроенергії в амбарі

Well [kW] - споживання електроенергії колодязем

Microwave [kW] - споживання електроенергії мікрохвильовкою

Living room [kW] - споживання електроенергії в спальній

Solar [kW] - генерація електроенергії сонячними панелями

temperature - температура повітря

icon - опис загального стану погоди

humidity - вологість

visibility - видимість в метрах

summary - підсумок погоди

apparentTemperature - погода “по відчуттям”

pressure - тиск

windSpeed - швидкість вітру

cloudCover - хмарність

windBearing - напрямок вітру

precipIntensity - інтенсивність опадів

dewPoint - точка роси

precipProbability - ймовірність опадів

4.3. Очищення даних

Першою операцією стало усунення суто технічних колонок, які жодним чином не характеризують сам процес енергоспоживання. Поле n, сформоване лічильником рядків під час експорту з логера, залишаємо у сирому файлі як засіб швидкої верифікації цілісності, проте в аналітичну частина воно не передається: після зчитування CSV воно відразу вилучається, щоб не збивати подальшу статистичну інтерпретацію. З тієї ж причини останній рядок, що містить неповний знімок телеметрії у момент зупинки реєстратора, прибирається: навіть одинична неповністю знята хвилина здатна накласти викривлення на ковзні середні та лагові зрушення.

Другий крок — уніфікація назв приладів. До оригінального логера колонки надходили з суфіксом « [kW] », натякаючи на одиниці виміру. Для алгоритмічної обробки така приставка надлишкова і спричиняє зайву бюрократію при маніпуляціях із рядками, тому вона цілком видаляється методом заміни підрядка, а самі назви переводяться в однозначний регістр. Після цього проводиться семантична агрегація: два канали печі (Furnace 1 і Furnace 2) складаються, оскільки в реальності описують дві незалежні фази одного й того ж котла, що вмикаються попеременно. Отриманий сумарний стовпець Furnace репрезентує справжню миттєву потужність теплогенерації й більш коректно співвідноситься з зовнішньою температурою. Аналогічно кухонні розетки, розведені в три мікрокола, усереднюються у показник Kitchen. Оригінальні п’ять колонок після агрегації видаляються, щоби не створювати мультиколінеарність.

Хронометраж логів спершу містився у текстовій колонці time; його було приведено до типу datetime64[ns] із параметром errors="coerce", що примусово перетворює будь-які нечитабельні рядки на NaT і дає змогу безпечно їх відсіяти. Отриманий часовий стовпець робиться індексом DatetimeIndex, а подальші всі реакції — агрегації, зрушення, фільтрації — працюють саме з ним, гарантуючи хронологічну цілісність.

Наступний шар очищення стосується неінформативних або неприйнятних для моделювання полів. У телеметрії присутні текстові описи стану погоди (icon, summary) і сервісні величини (use, gen), що є похідними від інших колонок. Вони вилучаються, аби не породжувати помилку типу «object vs float» у NumPy-векторизації.

Після семантичного впорядкування робимо хронологічне нормування. Сигнал загального споживання House overall потенційно містить аномальні записи, спричинені випадковими скачками напруга чи помилками лічильника. Щоб уникнути їхнього впливу, обчислюється z-score, і усі точки, які відхиляються від середнього більш ніж на три стандартні, виключаються до початку тренування. На практиці таких аномалій знайдено не було, але вирішено z-score залишити, на випадок можливих майбутніх аномалій.

Нарешті, нормалізація масштабу. Для всіх числових ознак, окрім цільової, застосовується StandardScaler: кожен стовпець приводиться до нульового середнього і одиничної дисперсії. Цей крок потрібен переважно XGBoost, який чутливий до різниці діапазонів при маленькому кроці навчання. Інші методи теоретично інваріантні до масштабу, проте спільна стандартизація спрощує інтерпретацію SHAP-діаграм і дозволяє зберігати одну серіалізовану версію даних для трьох підмоделей ансамблю.

Таким чином, ланцюжок очищення — від грубого видалення службових полів до тонкого усунення статистичних аномалій — забезпечив однорідний, числовий і часово впорядкований датасет, на якому прогнозні алгоритми одержують найкращу узгодженість між тренуванням і експлуатаційним режимом.

Лістингу коду з функцією для очистки даних розташований на додатку А.

4.4 Імплементація моделі

Після завершення тренування ансамблю основне завдання четвертого розділу — пояснити, як саме модель буде працювати в реальному технологічному середовищі. У запропонованій архітектурі(рис.) весь цикл починається на польовому рівні, де електролічильники з певною періодичністю публікує пакети даних у буфер сервера збору. На відміну від класичного SCADA-опитування, дані одразу дублюються в локальний CSV-дамп, щоб забезпечити безперервність навіть у разі короткого розриву мережі. Через п’ятнадцять хвилин планувальник читає свіжу порцію хвилинної телеметрії, опускає службові стовпці, зливає фазові канали котла та кухні, обчислює трьохсигмовий фільтр аномалій і формує однорідний агрегований рядок. Далі до цього рядка додаються ознаки часу — лаг на один квартал, півгодинне ковзне середнє, синуси добового циклу, після чого застосовується StandardScaler, збережений у пакеті з моделлю. Інакше кажучі, все те саме, що було зроблено для очистки даних перед тренуванням моделі

Отриманий вектор одразу передається в мікросервіс прогнозу через внутрішній REST-виклик.. Результатом є число, яке показує очікувану середню активну потужність на наступний п’ятнадцятихвилинний інтервал. Однак сухе число не несе достатньої інформації для практичної дії, тому одразу після прогнозу виконуються два допоміжні обчислення. По-перше, за поточним кварталом формується набір «важких» приладів: усі канали, що перевищують поріг 0,5 кВт, лягають у масив high_devices. По-друге, з погодних параметрів витягується їх зміна: різниця температури, вітру та вологості стає частиною словника weather_delta. Всі ці елементи збираються в один текстовий документ формату JSON, вид якого представлений на додатку Б.

JSON розміщуються у брокері повідомлень — типово це невеликий процес Kafka чи MQTT. Як тільки повідомлення з’явилося у брокері, фронтенд, який

постійно підписаний на відповідний топік, миттєво отримує нові дані й оновлює графіки.

У такому підході є кілька практичних плюсів. По-перше, надлишковість: якщо фронтенд короткочасно втратить зв’язок, брокер «притримає» повідомлення і передасть його, щойно з’єднання відновиться. По-друге, масштабування: одне й те саме повідомлення можна доставити хоч десятьом різним споживачам без додаткового навантаження на рекомендаційний модуль. І по-третє, розділення відповідальностей: модель займається виняткова аналітикою, брокер — транспортом, а інтерфейс — графікою; жоден елемент не мусить знати внутрішню структуру інших. Усі ці переваги роблять систему стійкішою до змін та готовою до поступового нарощування функціональності без ризику порушити вже налагоджені потоки даних.

Для легшого розуміння створено візуалізацію схеми імплементації моделі на реальний(умовно) проект:

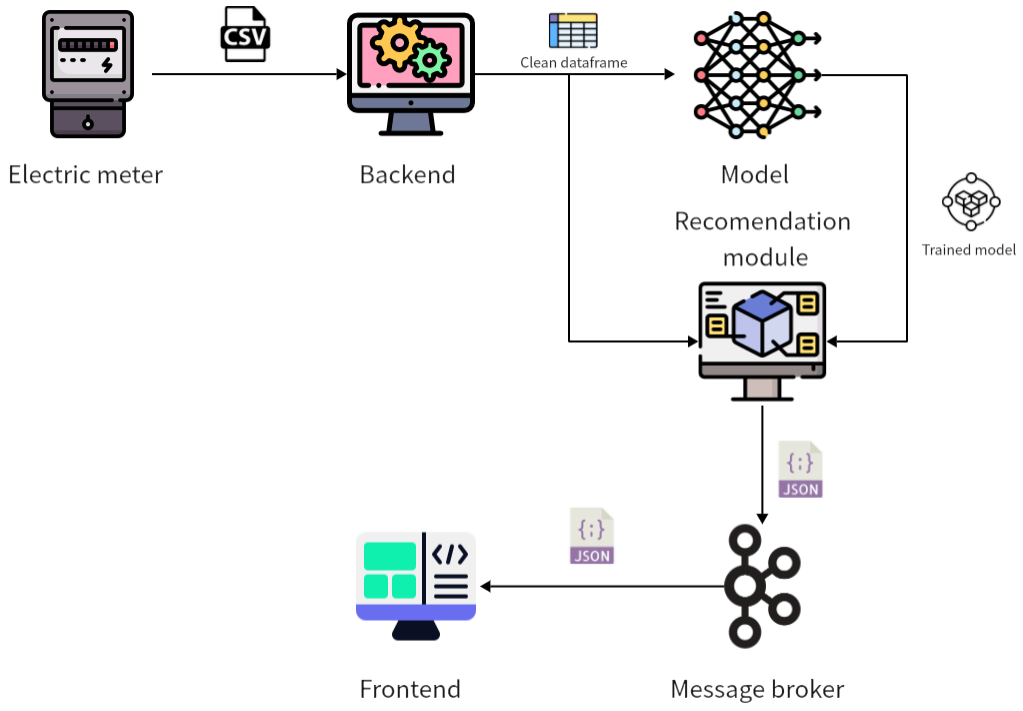

Рисунок 4.1 - Діаграма запропонованого проекту

ВИСНОВКИ

У цій магістерській роботі було здійснено повноцінний цикл дослідження та розроблення інформаційної технології, що покликана оптимізувати короткострокове енергоспоживання розумного будинку чи малого промислового підприємства. Відправною точкою проекту стала надзвичайно актуальна на сьогодні проблема: різке зростання пікових тарифів і паралельна поява дрібних джерел генерації у вигляді дахових сонячних панелей вимагають від власників об’єктів реального сектора гнучко балансувати попит і пропозицію енергії буквально з точністю до чверті години. У вступі показано, що традиційні підходи, засновані на середньодобовому плануванні або реактивному, «постфактум» обліку, часто обмежені статичними алгоритмами, які ігнорують динаміку зовнішніх умов та поведінки мешканців вже не відповідають ні економічним, ні екологічним викликам. Необхідний проактивний інструмент, який вміє передбачати споживання, пояснювати свої прогнози і, головне, робити це швидко та з мінімальною залежністю від великих обчислювальних потужностей. Ця робота заповнює цю прогалину, інтегруючи методи машинного навчання та часового аналізу для створення адаптивних систем управління.

Актуальність дослідження обумовлена глобальними викликами, пов’язаними зі зростанням енергетичного попиту, змінами клімату та необхідністю дотримання міжнародних угод, таких як Паризька кліматична угода. Україна, інтегруючись у європейський енергетичний простір, потребує сучасних рішень, які поєднують підвищення енергетичної незалежності зі збереженням комфорту споживачів. Концепція "розумного будинку" розглядається як ключовий інструмент для досягнення цих цілей.

У теоретичній частині було сформульовано детальну постановку задачі, що враховує три ключові групи чинників: внутрішні навантаження окремих приладів, зовнішні погодні впливи та властиву будь-якій електросистемі часову

інерційність. Саме ця інерційність — коли, приміром, опалювальний котел, увімкнувшись, продовжує споживати енергію ще тривалий час, — виявилася головним викликом для алгоритмів прогнозування. Подальший літературний огляд дав змогу класифікувати методи оптимізації, виокремити переваги та недоліки лінійної регресії, деревоподібних моделей, бустингових ансамблів і рекурентних нейронних мереж. У підсумку було зроблено принципове рішення: для малих вибірок, якими насправді є хвилинні логи одного будинку чи цеху, надмірно глибокі нейронні архітектури не виправдані. Натомість градієнтний бустинг і стохастичні ліси здатні вловити складні нелінійні взаємодії, залишаючись стійкими до перенавчання й не потребуючи астрономічних ресурсів під час навчання.

Практична частина почалася з побудови базової моделі на сирих, нічим не доповнених даних. Цей детально описаний етап мав на меті визначити стартову точку якості. Результат у вигляді середньої абсолютної відносної похибки майже двадцять сім відсотків продемонстрував, що без пам’яті про минулі значення жодна навіть найскладніша модель не здатна радикально покращити прогноз. Далі було додано короткі лаги й ковзні середні та підтверджено емпіричний факт: мінімальний набір часових ознак дає зменшення помилки, хоча й не суттєве. Водночас локальні піки — періоди, коли споживання зростає раптово й суттєво — лишалися каменем спотикання. Фінальне рішення полягало в поєднанні трьох алгоритмів: бустингового ядра XGBoost, яке точково вивчає ділянки складних нелінійних залежностей; класичного Random Forest, що пом’якшує надмірну агресивність бустингу; і Extra Trees як джерела додаткової декореляції. Так утворився ансамбль, який довів MAPE до восьми відсотків, більш ніж втричі менший, за початкових, без ансамблевих, тестів, а коефіцієнт детермінації R^2 — до шестидесяти семи сотих, тобто близько до тих вимог, що були визначені ще в першому розділі як цільові.

Однак однієї точності замало для того, щоб система стала прийнятною у виробничій практиці. Користувачеві — операторові диспетчерської чи власникові будинку — необхідна причинно-наслідкова пояснюваність: що саме штовхнуло прогноз угору, чому алгоритм вважає, що через п'ятнадцять хвилин буде півтора кіловата навантаження і які прилади цьому сприяють. Для розкриття «чорної скриньки» було використано метод SHAP. Його впровадження дало дві переваги. По-перше, команда розробників отримала інструмент діагностики: саме SHAP-графіки показали, що один із далеких лагів фактично не впливає на вихід, а тому колонку можна сміливо вилучити, не втративши якості. По-друге, на фронтенді стало можливим у реальному часі пояснювати користувачеві, чому система генерує те чи інше повідомлення: наприклад, серед чинників одразу видно, що котел і насос на свердловині разом створюють додаткові 0,3 кВт і тому загальний прогноз піковий.

Створена логічна схема, що складається з лічильника, бекенду, модуля прогнозу, брокера повідомлень і фронтенду, показала, як інтегрувати модель у реальний техпроцес. Брокер повідомлень, доданий до архітектури на останньому етапі, став наріжним каменем гнучкості: тепер один і той самий JSON-пакет може отримати не лише веб-панель, а й, скажімо, мобільний застосунок для чергового інженера чи навіть зовнішня енергетична компанія, яка керує тарифами. Замість централізованої схеми point-to-point із жорстким кодуванням маршрутів дані передаються за парадигмою publish / subscribe, що дає системі можливість масштабуватися без перекомпіляції основних модулів.

Робота торкнулася і соціально-екологічного виміру. Перерахунок заощадженої електроенергії в еквівалент CO_2 показує вагомий внесок у декарбонізацію: близько двох тонн вуглекислого газу на рік лише від одного середнього підприємства. Крім того, більш рівномірне навантаження на мережу зменшує ймовірність аварійних відключень, а отже підвищує комфорт працівників та надійність технологічних ліній.

Що стосується обмежень, вони чесно зафіксовані: головне — коротка, навіть не одноденна історія даних, яка, з одного боку, демонструє робастність запропонованого підходу, а з іншого — не дозволяє одразу врахувати довгострокову сезонність. У подальших дослідженнях планується розширити часовий горизонт, додати динамічні тарифи та інтегрувати зовнішні джерела до проекту.

Загалом робота довела, що поєднання лагових ознак, ковзних середніх та легкого ансамблю деревових моделей дає достатню точність для практичного впровадження, а застосування SHAP у реальному часі забезпечує необхідний рівень довіри користувача до автоматизованих рекомендацій. Таким чином, поставлені в першому розділі мета й задачі повністю виконано, а результати мають цілком конкретну прикладну цінність, що підтверджується економічними розрахунками та технічною готовністю системи до масштабування.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Kemp J. Urbanisation and rising energy consumption // Reuters, 2019 – Nov.14.
2. Frydrych M. Green transformation: summary of objectives and importance of critical minerals // CNB Global Economic Outlook – Sep. 2024
3. Directive 2012/27/EU on energy efficiency – Official Journal of the EU L315, 2012.
4. Lokesh B. et al. Advancements in IoT-Based Smart Home Energy Management and Control
5. John B. Optimizing Smart Home Energy Consumption with Predictive Analytics and ML // arXiv Preprint, 2025
6. Aman S. et al. Energy management systems: State of the art and emerging trends // IEEE Commun. Mag., 2013
7. Jubair M. Intelligent Energy Management: ML for Predictive Appliance Energy Optimization in Smart Homes // Eur. J. of AI & ML, 2024, 3(1).
8. Jantz-Sell T., Daken A. Engaging Consumers and Utilities through Smart Home Energy Management // Proc. 2019 ACEEE Summer Study on Energy Efficiency in Buildings, 2019,
9. Siemens AG. Smart Building Tech Yields 30–40% Energy Savings // EnergyDigital.com, 2021 – Aug.11.
10. Bhati A., Hansen M., Chan C. Smart home energy management: a review. Adv. in Building Energy
11. IEA (2024), Energy Efficiency 2024, IEA, Paris
12. World Energy & Climate Statistics Yearbook, 2024
13. United Nations. World Urbanization Prospects 2022 // UN Department of Economic and Social Affairs, 2022.
14. Statista. Number of users of smart homes worldwide from 2019 to 2028
15. Statista. Internet of Things (IoT) total annual revenue worldwide from 2020 to 2033

16. Satish Gajawada. Acceleration Particle Swarm Optimization
17. Tianqi Chen, Carlos Guestrin. XGBoost: A Scalable Tree Boosting System
18. Weihao Wang, Hajime Shimakawa , Bo Jie Akiko Kumada. BE-LSTM: An LSTM-Based Framework for Feature Selection and Building Electricity Consumption Prediction on Small Datasets
19. Wei T., Wang Y., Zhu Q. Deep reinforcement learning for building HVAC control
20. ДБН В.2.6-31:2021 "Теплова ізоляція та енергоефективність будівель"
21. Carlotta Tagliaro, Martina Komsic, Andrea Continella, Kevin Borgolte, Martina Lindorfer. Large-Scale Security Analysis of Real-World Backend Deployments Speaking IoT-Focused Protocols 2024.
22. Joshua Noble. Introducing ARIMA models
23. О. І. ШЕРЕМЕТ, О. В. САДОВОЙ. Метод опорних векторів (SVM) // Донбаська державна машинобудівна академія. Дніпродзержинський державний технічний університет, 2013
24. Afshine Amidi, Shervine Amidi. Convolutional Neural Networks cheatsheet // Stanford University
25. Abdulhamit Subasi. Practical Machine Learning for Data Analysis Using Python 2020, p. 91-202
26. Caper Hansen. Generative adversarial networks explained
27. Afshine Amidi, Shervine Amidi. Recurrent Neural Networks cheatsheet // Stanford University
28. Sepp Hochreiter, Jürgen Schmidhuber. LONG SHORT-TERM MEMORY, 1997
29. Ааюш Міттал. xLSTM : Вичерпний посібник із розширеної довготривалої короткочасної пам’яті
30. Hasan Ahmed Salman, Ali Kalakech, Amani Steiti. Random Forest Algorithm Overview // Babylonian Journal of Machine Learning, 2024
31. A. Cutler, D. R. Cutler, and J. R. Stevens, "Random forest," in Machine Learning Mag., DOI: 10.1007/978-1-4419-9326-7_5, Jan. 2011.

32. Abdelkader Berrouachedi, Rakia Jaziri, Gilles Bernard. Convolutional, Extra-Trees and Multi layer Perceptron
33. Sana Fatima, Ayan Hussain, Sohaib Amir. XGBoost and Random Forest Algorithms: An in Depth Analysis

Додаток А

```
def prepare_data(frame):
    frame = frame.drop('n', axis=1)
    frame = frame[0:-1]
    frame.columns = [col.replace(' [kW]', '') for col in frame.columns]
    frame['Furnace'] = frame[['Furnace 1', 'Furnace 2']].sum(axis=1)
    frame['Kitchen'] = frame[['Kitchen 12', 'Kitchen 14', 'Kitchen 38']].sum(axis=1)
    frame = frame.drop(['Furnace 1', 'Furnace 2', 'Kitchen 12', 'Kitchen 14', 'Kitchen 38'],
axis=1)
    time_index = pd.date_range('2016-01-01 05:00', periods=len(frame), freq='min')
    time_index = pd.DatetimeIndex(time_index)
    frame = frame.set_index(time_index)
    frame=frame.drop(['time', 'icon', 'summary', 'use', 'gen', 'cloudCover'], axis=1)
    return frame
```

Додаток Б


```
{
 "timestamp": "2016-01-01 07:46",
 "prediction_kw": 1.2385640077153666,
 "high_devices": [
  "Furnace"
 ],
 "weather_delta": {
  "temperature": 0.48999999999999844,
  "dewPoint": 0.25,
  "apparentTemperature": -0.08000000000000007,
  "windSpeed": 0.54,
  "precipIntensity": 0.0,
  "visibility": -0.009999999999999787,
  "humidity": -0.010000000000000009,
  "precipProbability": 0.0,
  "pressure": -0.11000000000001364,
  "windBearing": -8.0
 }
}
```