

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

СОЛЯНИК Станіслав Віталійович

Програмний модуль прогнозування енергоспоживання IoT-пристроїв на основі методів машинного навчання / Software Module for Predicting the Energy Consumption of IoT Devices Based on Machine Learning Methods

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

Кваліфікаційна робота

Виконав студент групи КН-42
С. В. Соляник

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

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

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

Тернопіль – 2026

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

ЗАТВЕРДЖУЮ

В.о. завідувача кафедри

_____ Н.В. Дзюбановська

«_____» _____ 20__ р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТУ
СОЛЯНИКУ Станіславу Віталійовичу

(прізвище, ім'я, по батькові)

1. Тема кваліфікаційної роботи

Програмний модуль прогнозування енергоспоживання IoT-пристроїв на основі методів машинного навчання / Software Module for Predicting the Energy Consumption of IoT Devices Based on Machine Learning Methods
керівник роботи к.т.н., професор В.В. Кочан

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

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

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

4. Основні питання, які потрібно розробити

- проаналізувати предметну область прогнозування енергоспоживання IoT-пристроїв;
- дослідити сучасні методи та моделі машинного навчання, що застосовуються для прогнозування енергоспоживання;
- обґрунтувати вибір підходів до побудови програмного модуля;
- спроектувати структуру та алгоритм функціонування програмного модуля;
- реалізувати програмний модуль прогнозування енергоспоживання IoT-пристроїв;
- провести експериментальне дослідження роботи модуля та оцінити якість прогнозування за відповідними метриками.

5. Перелік графічного матеріалу в роботі

- узагальнена схема вимог до програмного модуля прогнозування енергоспоживання IoT-пристроїв;
- архітектура програмного модуля прогнозування енергоспоживання IoT-пристроїв;
- схема алгоритму функціонування програмного модуля прогнозування енергоспоживання IoT-пристроїв.

6. Консультанти розділів кваліфікаційної роботи

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

7. Дата видачі завдання 01 грудня 2025 р.

КАЛЕНДАРНИЙ ПЛАН

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

Студент _____ С.В. Соляник
підпис

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

АНОТАЦІЯ

Кваліфікаційна робота на тему «Програмний модуль прогнозування енергоспоживання IoT-пристроїв на основі методів машинного навчання» на здобуття освітнього ступеня «бакалавр» зі спеціальності 122 «Комп'ютерні науки» освітньої програми «Комп'ютерні науки» написана обсягом в 72 сторінок і містить 11 ілюстрацій, 7 таблиць, 2 додатки та 37 використаних джерел.

Метою кваліфікаційної роботи є розроблення програмного модуля прогнозування енергоспоживання IoT-пристроїв на основі методів машинного навчання, який забезпечує обробку даних про режими функціонування пристроїв та формування прогнозу енергоспоживання для підвищення ефективності їх використання.

Методи досліджень включають аналіз наукових джерел для вивчення сучасного стану проблеми; системний аналіз для визначення структури програмного модуля; методи машинного навчання для побудови прогнозних моделей; методи статистичного аналізу та метрики оцінювання якості прогнозу; засоби програмної реалізації для створення прикладного модуля.

Розроблено програмний модуль прогнозування енергоспоживання IoT-пристроїв на основі методів машинного навчання, який забезпечує завантаження даних про режими роботи пристроїв, їх попередню обробку, формування ознак, навчання регресійних моделей, побудову прогнозу та оцінювання якості результатів.

Результати дослідження можуть бути використані для побудови систем моніторингу енергоспоживання IoT-пристроїв, розроблення програмних засобів інтелектуального енергоменеджменту, інтеграції прогнозних моделей у системи підтримки рішень, а також для створення навчально-дослідних прототипів у сфері Інтернету речей та енергоефективних цифрових систем.

Ключові слова: ЕНЕРГОСПОЖИВАННЯ, ІНТЕРНЕТ РЕЧЕЙ, ІОТ-ПРИСТРОЇ, МАШИННЕ НАВЧАННЯ, ПРОГНОЗУВАННЯ, РЕГРЕСІЙНІ МОДЕЛІ, RANDOM FOREST, ЕНЕРГОЕФЕКТИВНІСТЬ, ПРОГРАМНИЙ МОДУЛЬ.

ANNOTATION

Qualification work on the topic «Software Module for Predicting the Energy Consumption of IoT Devices Based on Machine Learning Methods» for Bachelor's degree on speciality 122 «Computer Science» educational and professional program «Computer Science» is written on 72 pages and it contains 11 figures, 7 tables, 2 annexes and 37 sources.

The purpose of the qualification work is to develop a software module for predicting the energy consumption of IoT devices based on machine learning methods. The module provides processing of data on device operating modes and generates energy consumption forecasts to improve the efficiency of their use.

The research methods include the analysis of scientific sources to study the current state of the problem; system analysis to determine the structure of the software module; machine learning methods for building predictive models; statistical analysis methods and metrics for evaluating forecast quality; and software implementation tools for developing the applied module.

A software module for predicting the energy consumption of IoT devices based on machine learning methods has been developed. It provides loading of data on device operating modes, their preprocessing, feature formation, training of regression models, forecast generation, and evaluation of result quality.

The research results can be used to build energy consumption monitoring systems for IoT devices, develop intelligent energy management software tools, integrate predictive models into decision support systems, and create educational and research prototypes in the field of the Internet of Things and energy-efficient digital systems.

Keywords: ENERGY CONSUMPTION, INTERNET OF THINGS, IOT DEVICES, MACHINE LEARNING, FORECASTING, REGRESSION MODELS, RANDOM FOREST, ENERGY EFFICIENCY, SOFTWARE MODULE.

ЗМІСТ

Вступ.....	7
1 Аналіз проблемної області та постановка задачі	10
1.1 Опис предметної області	10
1.2 Аналіз існуючих підходів і методів	14
1.3 Постановка задачі дослідження.....	20
2 Проєктування програмного модуля прогнозування енергоспоживання IoT-пристроїв	22
2.1 Формування вимог до програмного модуля	22
2.2 Вибір методів машинного навчання для прогнозування.....	26
2.3 Проєктування архітектури програмного модуля.....	31
2.4 Опис алгоритму функціонування модуля	35
3 Реалізація та дослідження роботи програмного модуля	40
3.1 Програмна реалізація модуля.....	40
3.2 Підготовка даних і навчання моделі.....	44
3.3 Аналіз результатів прогнозування	44
Висновки	56
Список використаних джерел	58
Додаток А Код програмного забезпечення.....	62
Додаток Б Копія публікації	67

ВСТУП

Стрімке поширення Інтернету речей зумовило появу великої кількості пристроїв, що безперервно збирають, передають і обробляють дані в режимі реального часу. IoT-пристрої активно застосовуються у системах моніторингу, промисловій автоматизації, розумних будівлях, транспорті, медицині та побутових системах керування. Разом із розширенням сфер використання таких пристроїв зростає і значення задачі енергоефективності, оскільки значна частина IoT-вузлів функціонує на автономних джерелах живлення або в умовах обмежених енергетичних ресурсів [3, 20, 29].

Рациональне використання енергії в IoT-середовищі є важливою передумовою забезпечення стабільної роботи пристроїв, зменшення витрат на обслуговування та підвищення загальної ефективності цифрових систем. У практичних умовах надмірне енергоспоживання призводить до скорочення часу автономної роботи, частішої заміни або підзарядки джерел живлення, а також до зниження надійності функціонування розподілених інформаційно-вимірювальних систем. Це особливо критично для мікроконтролерних модулів, сенсорних вузлів і периферійних edge-пристроїв, де навіть незначне перевищення енергетичних витрат може впливати на тривалість експлуатації та якість виконання функцій [4, 18, 19, 23].

Одним із перспективних напрямів розв'язання цієї проблеми є використання методів машинного навчання для аналізу закономірностей споживання енергії та побудови прогнозних моделей. Такі моделі дають змогу оцінювати майбутні значення енергоспоживання на основі попередніх вимірювань, параметрів навантаження, режимів роботи пристрою, характеристик середовища та інших факторів. У сучасних дослідженнях для цього застосовуються методи регресійного аналізу, дерева рішень, ансамблеві алгоритми, глибокі нейронні мережі та гібридні підходи, що підтверджує актуальність поєднання технологій IoT і машинного навчання в задачах енергоменеджменту [1, 8, 15, 20, 26].

Актуальність теми зумовлена кількома чинниками. По-перше, кількість

IoT-пристроїв постійно зростає, а отже, збільшується потреба в інтелектуальних засобах контролю їхнього енергоспоживання [3, 20]. По-друге, традиційні підходи до оцінювання енергетичних витрат часто базуються на статичних правилах або спрощених аналітичних моделях, які недостатньо точно враховують динаміку реального середовища функціонування пристроїв [16, 29]. По-третє, сучасні методи машинного навчання дозволяють будувати адаптивні моделі прогнозування, здатні працювати з багатофакторними даними та виявляти приховані залежності між режимами роботи пристроїв і рівнем енергоспоживання [14, 17, 25]. Крім того, для практичного застосування результатів прогнозування необхідне створення програмного модуля, який поєднуватиме збір даних, їх попередню обробку, навчання моделі та формування прогнозу в межах єдиного програмного рішення [7, 20, 23].

Питання енергоефективності IoT-систем, інтелектуального управління енергією та прогнозування енергоспоживання розглядалися у працях багатьох дослідників. У наукових джерелах описано підходи до прогнозування енергоспоживання будівель, побутових систем, промислових IoT-архітектур, периферійних вузлів та інформаційно-вимірювальних модулів [6, 11, 12, 13, 20, 22, 26]. Водночас значна частина наявних рішень або орієнтована на великі енергосистеми й розумні будівлі, або зосереджена на окремих аспектах вимірювання та оптимізації енергоспоживання без достатньої уваги до програмної реалізації цілісного модуля прогнозування для IoT-пристроїв [18, 19, 23, 32–34]. Це зумовлює потребу в розробленні прикладного програмного модуля, який поєднуватиме методи машинного навчання з практичними засобами обробки даних і формування прогнозних оцінок.

Метою кваліфікаційної роботи є розроблення програмного модуля прогнозування енергоспоживання IoT-пристроїв на основі методів машинного навчання, який забезпечує обробку даних про режими функціонування пристроїв та формування прогнозу енергоспоживання для підвищення ефективності їх використання.

Для досягнення поставленої мети необхідно виконати такі завдання дослідження:

- проаналізувати предметну область прогнозування енергоспоживання IoT-пристроїв;
- дослідити сучасні методи та моделі машинного навчання, що застосовуються для прогнозування енергоспоживання;
- обґрунтувати вибір підходів до побудови програмного модуля;
- спроектувати структуру та алгоритм функціонування програмного модуля;
- реалізувати програмний модуль прогнозування енергоспоживання IoT-пристроїв;
- провести експериментальне дослідження роботи модуля та оцінити якість прогнозування за відповідними метриками.

Об'єктом дослідження є процеси збору, обробки та аналізу даних про енергоспоживання IoT-пристроїв.

Предметом дослідження є методи машинного навчання, моделі прогнозування та програмні засоби, що використовуються для передбачення енергоспоживання IoT-пристроїв.

У роботі використано такі методи дослідження: аналіз наукових джерел для вивчення сучасного стану проблеми; системний аналіз для визначення структури програмного модуля; методи машинного навчання для побудови прогнозних моделей; методи статистичного аналізу та метрики оцінювання якості прогнозу; засоби програмної реалізації для створення прикладного модуля.

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

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

1 АНАЛІЗ ПРОБЛЕМНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАДАЧІ

1.1 Опис предметної області

Інтернет речей є технологічною концепцією, у межах якої фізичні пристрої оснащуються сенсорами, мікроконтролерами, засобами зв'язку та програмними компонентами для збору, передавання й обробки даних. До таких пристроїв належать сенсорні вузли, лічильники, контролери, виконавчі модулі, побутові smart-пристрої, елементи промислової автоматики та edge-вузли. Спільною рисою цих систем є безперервне функціонування в умовах обмежених обчислювальних і енергетичних ресурсів, що робить задачу контролю енергоспоживання однією з ключових у процесі їх експлуатації [3, 20, 29, 33].

У загальному випадку IoT-пристрій виконує кілька базових функцій: зчитування даних із сенсорів, локальну обробку інформації, передавання результатів мережею, збереження проміжних станів та взаємодію з іншими вузлами системи. Кожен із цих етапів супроводжується витратами енергії, величина яких залежить від типу апаратної платформи, режиму роботи процесора, частоти вимірювань, інтенсивності передавання даних, характеристик каналу зв'язку та тривалості активного стану модуля [18, 19, 23, 32]. Тому енергоспоживання в IoT-середовищі формується як результат дії багатьох чинників, що змінюються в часі.

Особливістю предметної області є те, що IoT-пристрої часто працюють від автономних джерел живлення. Це можуть бути батареї, акумулятори або гібридні джерела з підтримкою енергозбирання. У таких умовах неефективне використання енергії призводить до скорочення життєвого циклу пристрою, втрати стабільності вимірювань, зростання витрат на технічне обслуговування та зниження надійності всієї системи [4, 18, 23]. Для мереж із великою кількістю розподілених вузлів ця проблема набуває особливого значення, оскільки навіть незначне перевищення енергетичних витрат на окремих елементах може накопичуватися на рівні всієї інфраструктури [6, 9, 20].

У наукових джерелах задача енергоефективності IoT розглядається в кількох взаємопов'язаних напрямках. Перший напрям стосується апаратної

оптимізації та вимірювання споживання енергії мікроконтролерних і вимірювальних модулів [4, 18, 19, 23]. Другий пов'язаний із проєктуванням енергоефективних архітектур передавання та обробки даних у промисловому IoT, edge computing і бездротових мережах [2, 8, 20]. Третій напрям охоплює прогнозування енергоспоживання на основі статистичних методів і машинного навчання для побутових систем, будівель, центрів обробки даних та розумних енергетичних середовищ [1, 11, 15, 22, 25, 26]. Для теми цієї роботи найбільший інтерес становить саме третій напрям, оскільки він дає змогу перейти від простого вимірювання енергії до її прогнозування.

Прогнозування енергоспоживання передбачає побудову моделі, яка на основі історичних і поточних даних оцінює майбутній рівень витрат енергії пристрою або групи пристроїв. У межах IoT така задача здебільшого формулюється як задача регресії, де вхідними параметрами виступають характеристики навантаження, часові параметри, режими роботи, мережеві показники, дані сенсорів та виміряні енергетичні значення за попередні періоди, а вихідним результатом є прогноз енергоспоживання на певному часовому горизонті [1, 16, 17, 26]. Практична цінність такого прогнозу полягає в тому, що він може бути використаний для адаптивного керування режимами роботи пристрою, планування обслуговування, балансування навантаження та зниження загальних витрат енергії.

Предметна область охоплює не лише енергетичні характеристики пристроїв, а й структуру даних, на основі яких формується прогноз. У типових умовах використовуються часові ряди або табличні набори даних, що містять результати вимірювань струму, напруги, потужності, тривалості активних і пасивних станів, параметри мережевої взаємодії, відомості про інтенсивність обробки та зовнішні фактори середовища [19, 23, 26, 34]. Часто такі дані мають шум, пропуски, нерівномірну частоту спостережень або високу варіативність, що ускладнює побудову стійкої прогнозної моделі [18, 30]. Саме тому важливою складовою предметної області є попередня обробка даних: очищення, нормалізація, агрегування, формування ознак і підготовка вибірок для навчання моделі.

Окремої уваги потребує різниця між суміжними задачами, які часто розглядаються поруч із прогнозуванням енергоспоживання. По-перше, це задача вимірювання енергоспоживання, яка полягає в отриманні фактичних значень енергії або потужності за допомогою вимірювальних засобів [18, 19, 23]. По-друге, це задача моніторингу, де основна мета зводиться до відстеження поточного стану пристрою в реальному часі [7, 21]. По-третє, це задача оптимізації, у межах якої на основі вимірювань або прогнозу ухвалюються рішення щодо зменшення енергетичних витрат [2, 11, 31]. У даній роботі центральною є саме задача прогнозування, однак вона логічно пов'язана з вимірюванням, моніторингом та подальшою оптимізацією.

Для кращого розуміння предметної області доцільно виокремити основні складові системи прогнозування енергоспоживання IoT-пристроїв. У таблиці 1.1 наведено їх узагальнену характеристику.

Таблиця 1.1 – Основні складові предметної області прогнозування енергоспоживання IoT-пристроїв

Складова	Зміст
IoT-пристрої	Сенсорні вузли, мікроконтролери, smart-пристрої, edge-модулі
Дані	Вимірювання струму, напруги, потужності, часові параметри, мережеві показники
Процеси	Збір даних, передавання, попередня обробка, аналіз, прогнозування
Методи	Регресійні моделі, дерева рішень, ансамблеві алгоритми, нейронні мережі
Результат	Прогноз енергоспоживання та підтримка рішень щодо енергоефективності

Як видно з таблиці 1.1, предметна область є міждисциплінарною, оскільки поєднує апаратні, програмні та аналітичні компоненти. Саме ця властивість ускладнює побудову універсального підходу. Для різних класів IoT-пристроїв істотними можуть бути різні чинники: для мікроконтролерних систем

важливими є режими sleep/active та частота активації модулів [18, 19], для систем розумного будинку – поведінкові сценарії користувача та часові шаблони навантаження [11, 13, 22], а для промислового IoT – інтенсивність передавання даних, навантаження на обчислювальні вузли та параметри виробничого середовища [20, 24]. Отже, програмний модуль прогнозування повинен мати достатню гнучкість для роботи з різними типами даних і сценаріями використання.

Важливою характеристикою предметної області є наявність значної кількості факторів, що впливають на енергоспоживання. До них належать апаратні особливості пристрою, тип мережевого інтерфейсу, частота опитування сенсорів, обсяг переданих даних, обчислювальна складність локальних задач, температура середовища, час доби, режим навантаження та стан батареї [8, 14, 20, 25]. Частина цих факторів є явно вимірюваною, тоді як інші впливають опосередковано. Унаслідок цього формування якісного прогнозу потребує відбору релевантних ознак, а також вибору моделі, здатної враховувати нелінійні залежності [17, 27, 30].

У предметній області також простежується перехід від ізольованих технічних рішень до інтегрованих програмних платформ. Якщо раніше дослідження переважно зосереджувалися на вимірюванні окремих характеристик енергоспоживання, то сучасні підходи орієнтуються на побудову повного циклу обробки даних: від збору показників до формування прогнозу та передачі результатів користувачеві або підсистемі керування [7, 21, 23, 26]. Це означає, що предметом розроблення стає не лише модель машинного навчання, а саме програмний модуль, який поєднує підсистеми завантаження даних, підготовки вибірки, навчання моделі, обчислення прогнозу та представлення результатів.

З огляду на це предметна область кваліфікаційної роботи може бути визначена як сукупність процесів, методів і програмно-технічних засобів, призначених для збору даних про функціонування IoT-пристроїв, аналізу параметрів їх енергоспоживання та побудови прогнозу на основі методів машинного навчання. У межах цієї області поєднуються задачі інформаційного

вимірювання, підготовки даних, інтелектуального аналізу та програмної реалізації прикладного інструмента прогнозування.

Таким чином, предметна область дослідження характеризується високою практичною значущістю, багатфакторністю та потребою у використанні адаптивних методів аналізу. Прогнозування енергоспоживання IoT-пристроїв доцільно розглядати як важливу складову інтелектуального енергоменеджменту, що забезпечує основу для підвищення автономності, надійності та ефективності роботи цифрових систем.

1.2 Аналіз існуючих підходів і методів

У задачах прогнозування енергоспоживання IoT-пристроїв застосовується широкий спектр підходів, що відрізняються рівнем складності, типом вхідних даних, вимогами до обчислювальних ресурсів і точністю результатів. У наукових публікаціях такі підходи здебільшого охоплюють статистичні методи, класичні алгоритми машинного навчання, ансамблеві моделі, глибокі нейронні мережі, а також спеціалізовані рішення для edge- та industrial IoT-середовищ [1, 3, 20, 26]. Аналіз цих підходів дає змогу визначити, які з них є найбільш придатними для побудови програмного модуля прогнозування енергоспоживання IoT-пристроїв.

Одним із базових напрямів є використання класичних регресійних моделей. Такі підходи є доцільними у випадках, коли існує відносно стабільний зв'язок між вхідними факторами та цільовою змінною. До цієї групи належать лінійна регресія, багатфакторна регресія та нелінійна регресія. Їх перевагами є простота реалізації, висока швидкість навчання, зрозумілість результатів і порівняно невеликі вимоги до обчислювальних ресурсів [16, 17, 21]. Крім того, регресійні моделі зручні для початкового аналізу залежностей між параметрами енергоспоживання та характеристиками роботи пристрою. Водночас їхнім недоліком є обмежена здатність описувати складні нелінійні взаємозв'язки, які часто виникають у реальних IoT-середовищах [30, 31].

У роботах, присвячених енергоспоживанню будівель і smart home-систем, класичні регресійні методи використовуються як базова лінія для порівняння з

більш складними моделями. Такий підхід дозволяє оцінити, наскільки виправданим є застосування складніших алгоритмів у конкретній задачі [1, 14, 25]. Для кваліфікаційної роботи це також важливо, оскільки наявність простої базової моделі дає змогу об'єктивніше оцінити переваги інших методів.

Іншою великою групою є методи машинного навчання для регресійного прогнозування, зокрема дерева рішень, Random Forest, Support Vector Regression, Extreme Learning Machine та інші алгоритми, здатні працювати з багатofакторними даними [1, 6, 14]. У порівнянні з класичною регресією ці моделі краще враховують складні залежності між ознаками, можуть працювати з неоднорідними даними та часто забезпечують вищу точність прогнозування [1, 15, 17]. Особливо поширеним у прикладних задачах є Random Forest, оскільки він поєднує відносну простоту налаштування, стійкість до шумів і можливість оцінювання важливості ознак [1, 6]. Для прогнозування енергоспоживання це має практичне значення, оскільки дозволяє не лише сформулювати прогноз, а й визначити, які параметри найбільше впливають на результат.

Support Vector Regression також використовується в задачах прогнозування енергії, коли важливо описати нелінійні залежності при обмеженій кількості даних [1]. Перевагою цього підходу є здатність працювати з високовимірними просторами ознак. Недоліком слід вважати складність вибору параметрів ядра, а також зростання обчислювальних витрат при збільшенні обсягу вибірки. У практичних IoT-застосуваннях це може бути обмеженням, якщо модель планується регулярно перенавчати або виконувати на обмежених апаратних ресурсах.

Окрему увагу в сучасних дослідженнях приділено ансамблевим методам. Їх суть полягає в об'єднанні кількох базових моделей для підвищення стійкості та точності прогнозу. До цієї групи належать bagging, boosting, випадкові ліси та динамічні ансамблі дерев рішень [6, 8]. У задачах енергоефективності IoT ансамблеві моделі виявляються корисними через високу варіативність даних та вплив багатьох факторів на енергоспоживання [6, 20]. Перевагами таких методів є зниження ризику перенавчання, підвищення узагальнювальної здатності та можливість адаптації до шумних даних. Разом із тим ансамблеві моделі зазвичай

є менш інтерпретованими, ніж проста регресія, і можуть потребувати більших ресурсів під час навчання.

У дослідженнях, пов'язаних із edge IoT, значна увага приділяється не лише точності моделі, а й енергоефективності самого процесу інференсу. З цієї точки зору важливими є роботи, де аналізується розгортання моделей на периферійних вузлах та підбір алгоритмів з урахуванням обмежених обчислювальних і енергетичних можливостей пристрою [6, 9]. Такий підхід є принципово важливим для теми роботи, оскільки програмний модуль має бути не просто точним, а ще й практично придатним до використання в середовищі IoT. У цьому контексті надмірно складні моделі можуть бути менш доцільними, ніж дещо простіші, але швидші та стабільніші рішення.

Сучасний напрям досліджень пов'язаний із застосуванням глибокого навчання. Для прогнозування енергоспоживання використовуються багатoshарові нейронні мережі, TensorFlow-орієнтовані часові моделі, гібридні deep learning-підходи та алгоритми на основі часових рядів [15, 17, 26]. Перевага таких моделей полягає в здатності автоматично виявляти складні нелінійні закономірності у великих масивах даних. Це особливо актуально для задач, де енергоспоживання залежить від динаміки змін у часі, сезонності, повторюваних шаблонів навантаження або поведінкових сценаріїв користувача [11, 15, 25]. У роботі Natarajan et al. розглядаються гібридні deep learning-моделі для точного прогнозування енергоспоживання будівель, що підтверджує ефективність поєднання IoT-даних і глибокого навчання [15].

Разом із тим методи глибокого навчання мають і суттєві обмеження. Вони потребують більшого обсягу даних, більших витрат на навчання та складнішого налаштування гіперпараметрів [15, 17, 26]. Крім того, у багатьох прикладних IoT-сценаріях доступні вибірки є невеликими або містять шум, пропуски та нерівномірність спостережень [18, 19, 23]. За таких умов надмірно складні нейронні моделі не завжди демонструють стабільну перевагу над класичними алгоритмами машинного навчання. Звідси випливає важливий висновок: вибір методу повинен визначатися не лише сучасністю підходу, а й характером даних, складністю середовища та практичними вимогами до системи.

У деяких дослідженнях застосовуються еволюційні та оптимізаційні методи, наприклад генетичне програмування або підходи з підкріпленням. Такі рішення використовуються для побудови моделей прогнозування або для подальшої оптимізації енергоспоживання на основі результатів прогнозу [16, 31]. Зокрема, Yu і Tong розглядають оптимізацію енергетичних процесів на основі Deep Q-learning, а Nazir et al. застосовують генетичне програмування для задач прогнозування енергоспоживання [16, 31]. Перевагою таких підходів є здатність шукати рішення у складних просторах параметрів, проте їх використання в межах прикладного програмного модуля часто ускладнюється значною обчислювальною вартістю та складністю реалізації.

Окремий клас становлять архітектурно-орієнтовані та системні підходи, у яких прогнозування енергоспоживання розглядається як частина більш широкої системи енергоменеджменту. У таких роботах увага приділяється побудові хмарно-IoT платформ, інтеграції вимірювальних модулів, збиранню телеметрії, передаванню даних, збереженню та використанню результатів аналізу в контурі прийняття рішень [7, 20, 23]. Саме цей підхід є найближчим до теми кваліфікаційної роботи, оскільки тут предметом є не лише модель, а повноцінний програмний модуль. У цьому разі прогнозування виступає як аналітичне ядро, але ефективність усього рішення залежить також від організації потоку даних, попередньої обробки, механізмів запуску моделі та представлення результатів.

Доцільно також враховувати підходи, що стосуються вимірювання та підготовки даних, оскільки якість прогнозу безпосередньо залежить від якості вихідної інформації. У працях, присвячених вимірюванню енергоспоживання мікроконтролерних систем, наголошується на важливості завадостійкості вимірювань, точності оцінки середнього енергоспоживання, стабільності фіксації режимів роботи та коректного відбору параметрів [4, 18, 19, 23]. Це означає, що навіть найкраща модель машинного навчання не забезпечить надійного результату за умов низької якості вхідних даних. Отже, у предметній області прогнозування енергоспоживання необхідно розглядати модель та дані як взаємопов'язані компоненти.

Істотне значення мають і методи відбору ознак та оцінювання якості

моделі. Для побудови ефективного прогнозу недостатньо лише обрати алгоритм; важливо також визначити, які параметри повинні бути подані на вхід моделі. У цьому контексті корисними є підходи до feature selection, які дозволяють зменшити надлишковість ознак, покращити узагальнювальну здатність моделі та знизити обчислювальні витрати [27]. Для оцінювання результатів у задачах регресії найчастіше застосовуються метрики MAE, RMSE та інші показники похибки [5, 30]. У задачі прогнозування енергоспоживання саме ці метрики дозволяють порівнювати моделі між собою та обґрунтовано вибирати найкраще рішення.

Для узагальнення результатів аналізу в таблиці 1.2 наведено порівняльну характеристику основних підходів.

Таблиця 1.2 – Порівняльна характеристика підходів до прогнозування енергоспоживання

Підхід	Переваги	Недоліки	Доцільність для задачі
Класична регресія	Простота, швидкість, інтерпретованість	Обмежена здатність описувати нелінійність	Доцільна як базова модель
Дерева рішень і Random Forest	Працюють зі складними залежностями, стійкі до шуму	Менш прозора логіка окремого прогнозу	Висока
Support Vector Regression	Добре працює з нелінійними даними	Складне налаштування, чутливість до параметрів	Середня
Ансамблеві методи	Вища точність і стійкість	Більші обчислювальні витрати	Висока

Продовження таблиці 1.2

Глибокі нейронні мережі	Виявляють складні часові й нелінійні закономірності	Потребують більше даних і ресурсів	Залежить від обсягу даних
Еволюційні та RL-підходи	Потужна оптимізація складних систем	Складність реалізації	Обмежена для прикладного модуля

Як показано в таблиці 1.2, універсального методу для всіх типів IoT-задач не існує. Вибір підходу залежить від структури вхідних даних, вимог до швидкодії, можливостей апаратної платформи та необхідного рівня точності. Для прикладного програмного модуля прогнозування енергоспоживання доцільно орієнтуватися на методи, що забезпечують баланс між точністю, швидкістю, простотою інтеграції та стабільністю роботи. Саме тому найбільш перспективними виглядають класичні алгоритми машинного навчання для регресії, передусім дерева рішень, ансамблеві моделі та інші методи, що не потребують надмірного обсягу даних, але здатні враховувати нелінійні залежності [1, 6, 14, 17].

На основі проведеного аналізу можна зробити кілька висновків. По-перше, сучасні підходи до прогнозування енергоспоживання демонструють перехід від простих аналітичних моделей до інтелектуальних методів аналізу даних. По-друге, найбільш придатними для практичної реалізації в межах програмного модуля є методи машинного навчання, що поєднують достатню точність із помірними вимогами до ресурсів. По-третє, якість прогнозування значною мірою залежить не лише від самої моделі, а й від правильного вимірювання, попередньої обробки даних, відбору ознак та коректного використання метрик оцінювання

1.3 Постановка задачі дослідження

Проведений аналіз предметної області та існуючих підходів показав, що задача прогнозування енергоспоживання IoT-пристроїв є актуальною та практично значущою. Для таких пристроїв характерні обмежені енергетичні ресурси, змінні режими роботи, залежність енергоспоживання від багатьох факторів і потреба в своєчасному прийнятті рішень щодо оптимізації функціонування [18–20, 23]. Водночас наявні підходи не завжди орієнтовані на створення прикладного програмного модуля, який поєднував би збір даних, їх обробку, навчання моделі та формування прогнозу в межах єдиного рішення [7, 20, 26].

У межах кваліфікаційної роботи необхідно розв’язати задачу розроблення програмного модуля прогнозування енергоспоживання IoT-пристроїв на основі методів машинного навчання. Такий модуль повинен забезпечувати приймання та підготовку вхідних даних, формування ознак, побудову прогнозової моделі та обчислення прогнозу енергоспоживання для подальшого аналізу [1, 14, 25].

Основна задача дослідження полягає у створенні програмного засобу, який на основі історичних і поточних параметрів функціонування IoT-пристрою дозволяє передбачити майбутній рівень енергоспоживання з прийнятною точністю. В якості вхідних даних можуть використовуватися показники роботи пристрою, часові параметри, характеристики навантаження, результати вимірювань струму, напруги, потужності та інші ознаки, що впливають на споживання енергії [19, 23, 32, 34]. Результатом роботи модуля повинен бути прогноз енергоспоживання та значення метрик якості моделі.

Тому, метою кваліфікаційної роботи є розроблення програмного модуля прогнозування енергоспоживання IoT-пристроїв на основі методів машинного навчання, який забезпечує обробку даних про режими функціонування пристроїв та формування прогнозу енергоспоживання для підвищення ефективності їх використання.

Для досягнення поставленої мети необхідно виконати такі завдання дослідження:

- проаналізувати предметну область прогнозування енергоспоживання IoT-пристроїв;
- дослідити сучасні методи та моделі машинного навчання, що застосовуються для прогнозування енергоспоживання;
- обґрунтувати вибір підходів до побудови програмного модуля;
- спроектувати структуру та алгоритм функціонування програмного модуля;
- реалізувати програмний модуль прогнозування енергоспоживання IoT-пристроїв;
- провести експериментальне дослідження роботи модуля та оцінити якість прогнозування за відповідними метриками.

2 ПРОЄКТУВАННЯ ПРОГРАМНОГО МОДУЛЯ ПРОГНОЗУВАННЯ ЕНЕРГОСПОЖИВАННЯ ІОТ-ПРИСТРОЇВ

2.1 Формування вимог до програмного модуля

Розроблення програмного модуля прогнозування енергоспоживання ІоТ-пристроїв потребує попереднього визначення вимог до його структури, функцій та режимів роботи. Формування таких вимог є необхідним етапом проєктування, оскільки саме на цій основі визначаються архітектура системи, склад програмних компонентів, набір вхідних і вихідних даних, а також критерії оцінювання ефективності модуля. Для задачі прогнозування енергоспоживання особливого значення набувають вимоги, пов'язані з точністю обробки даних, стабільністю роботи, придатністю до використання в ІоТ-середовищі та можливістю інтеграції методів машинного навчання [7, 20, 23].

Програмний модуль у межах даної роботи розглядається як прикладний програмний засіб, призначений для приймання даних про роботу ІоТ-пристрою, їх попередньої обробки, формування ознак, побудови прогнозової моделі та обчислення прогнозу енергоспоживання. Отже, його функціонування повинно охоплювати повний цикл роботи з даними: від завантаження вимірних параметрів до подання результату користувачеві у зручному вигляді. Такий підхід відповідає сучасним тенденціям побудови систем енергоменеджменту, де прогнозування є частиною цілісного процесу аналізу та підтримки прийняття рішень [11, 21, 26].

Перш за все доцільно визначити функціональні вимоги до програмного модуля. До цієї групи належать вимоги, що описують, які саме дії повинен виконувати модуль. Основними функціями мають бути:

- завантаження вхідних даних про енергоспоживання та параметри роботи ІоТ-пристроїв;
- перевірка коректності й повноти даних;
- попередня обробка даних, зокрема очищення, нормалізація та підготовка ознак;
- поділ вибірки на навчальну та тестову частини;

- навчання моделі машинного навчання;
- формування прогнозу енергоспоживання;
- обчислення метрик якості прогнозування;
- виведення результатів у текстовій, табличній або графічній формі.

Таким чином, модуль має виконувати не лише обчислення прогнозу, а й забезпечувати повний супровід процесу аналізу даних. Це є важливим, оскільки без належної підготовки вхідної інформації навіть коректно обраний алгоритм не забезпечить потрібної точності.

Окремо слід розглянути вимоги до вхідних даних. Вхідна інформація повинна характеризувати режими роботи IoT-пристроїв та їх енергетичні параметри. До таких даних можуть належати значення струму, напруги, потужності, часові інтервали активності, частота передавання даних, інтенсивність роботи сенсорів, обсяг виконуваних обчислень, стан батареї та інші параметри, що впливають на рівень енергоспоживання [18, 19, 23, 34]. Дані можуть бути організовані у вигляді таблиці, де кожний рядок відповідає окремому спостереженню, а стовпці містять значення ознак і цільової змінної.

До вхідних даних висуваються такі вимоги:

- дані повинні бути структурованими та придатними до автоматизованого зчитування;
- записи повинні містити достатній набір ознак для побудови прогнозу;
- у наборі даних має бути цільовий показник, який відображає фактичне енергоспоживання;
- бажаною є мінімізація пропусків, шумів і аномальних значень;
- формат даних повинен бути сумісним із програмною реалізацією модуля.

У практичному сенсі це означає, що модуль повинен працювати з даними у поширених форматах, наприклад CSV або Excel, а також мати механізм перевірки структури вхідного набору перед запуском обробки.

Другу важливу групу становлять вимоги до вихідних результатів. Основним результатом роботи модуля має бути прогноз енергоспоживання IoT-пристрою на основі поданих даних. Крім цього, модуль повинен надавати

додаткову інформацію, необхідну для аналізу результатів, а саме:

- фактичні та прогнозовані значення;
- абсолютні або середні похибки;
- значення метрик якості;
- графічне порівняння реальних і прогнозованих даних;
- короткий висновок про якість роботи моделі.

Наявність таких результатів дозволяє не лише отримати прогноз, а й оцінити практичну придатність обраного підходу.

Суттєве значення мають нефункціональні вимоги, які визначають якісні характеристики програмного модуля. До них належать надійність, зрозумілість інтерфейсу, швидкодія, масштабованість і зручність використання. Для даної роботи першочерговими є такі нефункціональні вимоги:

- зрозумілість: структура модуля та послідовність дій користувача повинні бути логічними і простими;
- стабільність: модуль повинен коректно працювати при обробці типових наборів даних без збоїв;
- розширюваність: має бути передбачена можливість заміни або додавання нових моделей машинного навчання;
- швидкодія: час обробки повинен бути прийнятним для навчального та експериментального використання;
- відтворюваність результатів: при однакових вхідних даних і параметрах модуль має формувати однаковий результат;
- переносимість: програмна реалізація повинна бути придатною до запуску на стандартному персональному комп'ютері без спеціалізованого обладнання [6, 9, 20].

Оскільки робота присвячена саме програмному модулю, важливо визначити і вимоги до архітектури. Доцільно, щоб модуль мав модульну структуру, у межах якої окремо виділяються підсистеми:

- завантаження даних;
- попередньої обробки;
- формування ознак;

- навчання моделі;
- прогнозування;
- оцінювання результатів;
- візуалізації та подання результатів.

Модульний принцип побудови є доцільним з кількох причин. По-перше, він спрощує розроблення та тестування окремих компонентів. По-друге, забезпечується гнучкість у разі зміни алгоритму прогнозування або формату даних. По-третє, така структура є логічною для подальшого розширення системи, наприклад шляхом додавання нових методів машинного навчання або інтеграції з джерелами телеметрії [7, 21, 23].

Для узагальнення основних вимог у таблиці 2.1 наведено їх систематизований перелік.

Таблиця 2.1 – Основні вимоги до програмного модуля прогнозування енергоспоживання IoT-пристроїв

Група вимог	Зміст вимоги
Функціональні	Завантаження даних, попередня обробка, навчання моделі, прогнозування, оцінювання результатів
До вхідних даних	Структурованість, повнота, наявність цільової змінної, сумісний формат
До вихідних результатів	Прогноз, метрики якості, таблиці результатів, графіки
Нефункціональні	Стабільність, зрозумілість, швидкодія, розширюваність, відтворюваність
Архітектурні	Модульна структура, логічний поділ компонентів, можливість розширення

Як видно з таблиці 2.1, вимоги до програмного модуля охоплюють як аналітичну, так і програмно-технічну складову. Це означає, що під час проектування необхідно враховувати одночасно кілька аспектів: якість прогнозованої моделі, зручність роботи з даними та логіку реалізації програмного

засобу.

Крім загальних вимог, доцільно визначити вимоги до моделі машинного навчання, яка буде інтегрована в модуль. Така модель повинна:

- працювати з числовими табличними даними;
- підтримувати задачу регресійного прогнозування;
- бути достатньо точною для практичного експерименту;
- не вимагати надмірних апаратних ресурсів;
- бути придатною до пояснення результатів хоча б на базовому рівні.

З огляду на це, у межах роботи доцільно орієнтуватися на класичні методи машинного навчання, що забезпечують баланс між точністю та простотою реалізації [1, 6, 14, 17]. Саме така вимога є важливою для навчально-дослідницького програмного модуля, оскільки надмірно складні моделі не завжди виправдані за невеликого обсягу даних.

Для кращого уявлення про місце програмного модуля в загальному процесі прогнозування на рисунку 2.1 подано узагальнену схему вимог і потоків даних.



Рисунок 2.1 – Узагальнена схема вимог до програмного модуля прогнозування енергоспоживання IoT-пристроїв

Отже, формування вимог до програмного модуля дозволило визначити його основне призначення, ключові функції, вимоги до даних, результатів і програмної реалізації. Подальше проєктування повинно спиратися саме на ці вимоги, оскільки вони задають логіку побудови системи та критерії її оцінювання.

2.2 Вибір методів машинного навчання для прогнозування

Одним із ключових етапів проєктування програмного модуля прогнозування енергоспоживання IoT-пристроїв є вибір методів машинного

навчання, які будуть використані для побудови прогнозової моделі. Від правильності такого вибору залежить не лише точність отриманих результатів, а й швидкодія системи, складність програмної реалізації, можливість подальшого розширення модуля та придатність моделі до роботи з реальними даними. У задачах прогнозування енергоспоживання найчастіше застосовуються методи регресійного аналізу, оскільки цільовий показник у цьому випадку має кількісний характер і виражається у вигляді безперервних значень [1, 16, 25].

Для даної роботи доцільно розглядати саме ті методи, які відповідають кільком базовим вимогам. По-перше, вони повинні підтримувати розв'язання задачі регресії. По-друге, мають бути достатньо ефективними при роботі з табличними даними, що містять параметри функціонування IoT-пристроїв. По-третє, моделі повинні забезпечувати прийнятний баланс між точністю прогнозування та складністю реалізації. По-четверте, обраний метод має бути придатним до використання у складі прикладного програмного модуля без надмірних вимог до обчислювальних ресурсів [6, 9, 20].

У межах дослідження доцільно орієнтуватися на кілька груп методів машинного навчання. Найпростішим варіантом є лінійна регресія, яка традиційно використовується як базова модель у задачах прогнозування. Її головними перевагами є простота математичного апарату, швидке навчання, зрозуміла інтерпретація результатів та низькі вимоги до ресурсів [17, 21]. Лінійна регресія є корисною не лише як самостійний інструмент прогнозування, а й як відправна точка для порівняння з більш складними підходами. Якщо складніша модель не демонструє суттєвого покращення якості, використання простої моделі може виявитися більш раціональним.

Однак у задачі прогнозування енергоспоживання IoT-пристроїв лінійна регресія має і суттєві обмеження. Реальне енергоспоживання залежить від багатьох нелінійно пов'язаних факторів: режиму активності модуля, частоти передавання даних, тривалості активних і пасивних станів, характеристик навантаження, зовнішніх умов та типу пристрою [18, 19, 23]. Через це проста лінійна залежність часто не дозволяє точно описати реальний характер зміни енергоспоживання. Отже, лінійна регресія є доцільною як базова модель, але не

завжди достатньою як основний метод прогнозування.

Наступним доцільним методом є дерево рішень для регресії. Такі моделі будують прогноз шляхом послідовного розбиття простору ознак на області, у межах яких цільова змінна набуває близьких значень. Перевагою дерева рішень є здатність працювати з нелінійними залежностями, відносна наочність логіки моделі та простота інтерпретації окремих рішень [6, 14]. Для задачі прогнозування енергоспоживання це важливо, оскільки модель може автоматично виявляти порогові значення параметрів, за яких змінюється характер споживання енергії.

Разом із тим окреме дерево рішень може бути нестійким до змін у вибірці та схильним до перенавчання. Саме тому більш доцільним у багатьох випадках є застосування ансамблевих методів, зокрема Random Forest. Цей алгоритм формує сукупність дерев рішень і усереднює їх результати, що дозволяє підвищити точність та стійкість моделі [1, 6]. У дослідженнях, присвячених енергоспоживанню, Random Forest часто демонструє хороші результати завдяки здатності враховувати складні залежності між ознаками без надмірної чутливості до шуму в даних [1, 15]. Крім того, така модель підтримує оцінювання важливості ознак, що є корисним для аналізу, які параметри найбільше впливають на енергоспоживання IoT-пристрою.

Для програмного модуля прогнозування Random Forest має ще одну важливу перевагу: цей метод відносно просто реалізується засобами сучасних бібліотек машинного навчання і не потребує надто складного налаштування. У порівнянні з глибокими нейронними мережами або складними оптимізаційними моделями він краще підходить для навчально-дослідницького програмного модуля, де важливими є не лише точність, а й надійність, відтворюваність та зрозумілість процесу прогнозування [6, 9].

Ще одним перспективним методом є Support Vector Regression (SVR). Цей підхід дозволяє будувати нелінійні регресійні моделі за допомогою ядерних функцій і добре працює у випадках, коли потрібно описати складну структуру залежностей між ознаками [1]. Перевагою SVR є здатність будувати достатньо точні моделі навіть на відносно невеликих наборах даних. Для задачі

прогнозування енергоспоживання це може бути корисно, якщо кількість спостережень є обмеженою. Проте застосування SVR пов'язане зі складністю підбору параметрів, чутливістю до масштабування даних та більшими витратами часу на навчання порівняно з лінійною регресією або деревами рішень [1, 17]. Тому такий метод доцільно розглядати як додатковий варіант для порівняння, але не обов'язково як основне рішення.

У сучасних дослідженнях значну увагу приділено також глибоким нейронним мережам і гібридним моделям для прогнозування енергоспоживання [15, 17, 26]. Такі підходи дозволяють працювати зі складними часовими рядами, виявляти приховані закономірності та враховувати нелінійні зміни в динаміці енергетичних показників. Проте для даної кваліфікаційної роботи використання глибоких моделей не є першочерговим. Це пояснюється тим, що їх реалізація потребує більшого обсягу даних, значно складнішого налаштування та більших обчислювальних ресурсів. Для прикладного програмного модуля, орієнтованого на табличні дані та експериментальну перевірку, класичні методи машинного навчання є більш доцільними.

Для наочного порівняння в таблиці 2.2 наведено узагальнену характеристику розглянутих вище методів.

Таблиця 2.2 – Порівняння методів машинного навчання для прогнозування енергоспоживання IoT-пристроїв

Метод	Переваги	Недоліки	Доцільність використання
Linear Regression	Простота реалізації, швидкість навчання, зрозуміла інтерпретація	Не враховує складні нелінійні залежності	Доцільна як базова модель
Decision Tree Regressor	Працює з нелінійними залежностями, наочна логіка	Схильність до перенавчання	Доцільний для аналізу структури залежностей
Random Forest Regressor	Висока точність, стійкість до шуму, оцінка важливості ознак	Більша складність порівняно з простою регресією	Доцільний як основний метод
Support Vector Regression	Працює з нелінійними даними, придатний для малих вибірок	Чутливість до параметрів, складніше налаштування	Доцільний для порівняння

Як видно з таблиці 2.2, жоден із методів не є універсальним, однак для поставленої задачі найбільш збалансованим варіантом виглядає Random Forest Regressor. Такий вибір пояснюється тим, що він здатний працювати з табличними даними, враховувати складні залежності між параметрами функціонування IoT-пристрою та забезпечувати достатньо високу якість прогнозу без необхідності надто складного налаштування [1, 6, 15]. Водночас використання лінійної регресії як базової моделі дозволить отримати орієнтир для порівняння, а застосування SVR або дерева рішень – оцінити, наскільки інші алгоритми можуть бути ефективними в конкретній задачі.

При виборі методу також слід враховувати особливості майбутньої програмної реалізації. Для даного програмного модуля важливо, щоб модель могла бути інтегрована у зрозумілий програмний ланцюг: завантаження даних, попередня обробка, навчання, прогнозування, оцінювання результатів та візуалізація [7, 21, 23]. Саме тому доцільно використовувати методи, які підтримуються стандартними бібліотеками машинного навчання, наприклад scikit-learn. Це спрощує реалізацію, тестування та подальше вдосконалення програмного модуля.

Окрему роль у виборі методів відіграє оцінювання якості прогнозу. Для порівняння моделей доцільно використовувати метрики MAE та RMSE, оскільки вони є найбільш поширеними в задачах регресії та дозволяють оцінити середню величину похибки прогнозування [5, 30]. На етапі експериментального дослідження саме ці показники дадуть змогу обґрунтовано визначити, яка модель є найкращою для конкретного набору даних.

Отже, у межах кваліфікаційної роботи доцільно застосувати підхід, за якого кілька методів машинного навчання використовуються для порівняльного аналізу, а основним методом програмного модуля обирається Random Forest Regressor. Такий вибір є обґрунтованим з точки зору точності, стійкості до шуму, придатності до табличних IoT-даних та відносної простоти інтеграції у програмне рішення.

2.3 Проектування архітектури програмного модуля

Після визначення вимог до програмного модуля та вибору методів машинного навчання наступним етапом є проектування його архітектури. Архітектура програмного модуля визначає склад основних компонентів, зв'язки між ними, логіку обробки даних та порядок виконання функцій. Для задачі прогнозування енергоспоживання IoT-пристроїв архітектура повинна забезпечувати послідовне виконання всіх ключових етапів: завантаження даних, їх перевірку, попередню обробку, формування ознак, навчання моделі, прогнозування та оцінювання результатів [7, 20, 23].

У межах даної роботи доцільно використовувати модульний підхід до побудови архітектури. Це означає, що програмний модуль складається з кількох функціонально відокремлених підсистем, кожна з яких виконує власне завдання. Така організація є доцільною з кількох причин. По-перше, спрощується реалізація та тестування окремих частин програми. По-друге, забезпечується можливість заміни окремих компонентів без повного перепроєктування системи. По-третє, модульна структура є зручною для подальшого розширення, наприклад при додаванні нових моделей машинного навчання або інших джерел даних [21, 26].

Загальна архітектура програмного модуля повинна враховувати специфіку предметної області. На вхід системи надходять дані про роботу IoT-пристрою: показники струму, напруги, потужності, часові параметри, режими активності, мережеві характеристики та інші ознаки, що можуть впливати на енергоспоживання. Далі ці дані проходять етап підготовки, після чого використовуються для навчання моделі або формування прогнозу. Отримані результати передаються до підсистеми оцінювання та візуалізації, де обчислюються метрики якості та формується представлення результатів для користувача.

З урахуванням зазначеного до складу архітектури доцільно включити такі основні компоненти [37]:

- модуль завантаження даних;

- модуль попередньої обробки даних;
- модуль формування ознак;
- модуль навчання моделі;
- модуль прогнозування;
- модуль оцінювання результатів;
- модуль візуалізації та взаємодії з користувачем.

Модуль завантаження даних відповідає за імпорт вхідної інформації з файлів або інших джерел. У межах спрощеної реалізації доцільно передбачити роботу з табличними форматами, наприклад CSV. Основним завданням цього модуля є зчитування набору даних і передавання його до наступного етапу обробки. Крім того, на цьому етапі може виконуватися первинна перевірка структури даних: наявності заголовків, типів полів та цільової змінної.

Модуль попередньої обробки даних виконує очищення й підготовку набору даних до подальшого аналізу. На цьому етапі можуть усуватися пропуски, аномальні значення, дублікати записів, а також виконуватися нормалізація або масштабування числових ознак. Значення цього модуля є особливо важливим, оскільки якість прогнозу безпосередньо залежить від якості вхідної інформації. Якщо вихідні дані містять шум або структурні помилки, точність моделі істотно знижується.

Модуль формування ознак виконує підготовку вхідного простору для моделі машинного навчання. На цьому етапі визначається, які саме параметри будуть використані як вхідні ознаки, а також може виконуватися відбір найбільш інформативних змінних [27]. У задачах прогнозування енергоспоживання доцільно використовувати як безпосередньо виміряні параметри, так і похідні характеристики, наприклад усереднені значення за інтервал, коефіцієнти змінності, часові індекси або співвідношення між окремими величинами [19, 23, 34]. Саме цей модуль створює основу для побудови інформативної моделі.

Модуль навчання моделі реалізує побудову математичної моделі на навчальній вибірці. У межах роботи доцільно передбачити можливість використання кількох алгоритмів регресійного прогнозування, зокрема Linear Regression, Decision Tree Regressor, Random Forest Regressor та Support Vector

Regression [1, 6, 14]. Основним призначенням цього модуля є навчання вибраної моделі на підготовлених даних з подальшим збереженням параметрів для використання на етапі прогнозування. Для архітектури важливо, щоб цей модуль був незалежним від конкретного алгоритму, тобто допускав заміну одного методу іншим без порушення загальної логіки роботи системи.

Модуль прогнозування використовує навчену модель для обчислення прогнозованого значення енергоспоживання на нових або тестових даних. Саме цей компонент забезпечує основну прикладну функцію програмного модуля. В архітектурному сенсі важливо, щоб він приймав стандартизований набір ознак і повертав прогноз у єдиному форматі, незалежно від того, який алгоритм використовується всередині. Це спрощує реалізацію інших компонентів системи.

Модуль оцінювання результатів виконує порівняння фактичних і прогнозованих значень та обчислює метрики якості моделі. Для даної роботи доцільно використовувати MAE та RMSE як основні показники похибки [5, 30]. За потреби може також застосовуватися коефіцієнт детермінації R^2 . Цей модуль є важливим для експериментальної перевірки якості моделі та вибору найкращого методу прогнозування.

Модуль візуалізації та взаємодії з користувачем призначений для представлення результатів роботи модуля у зрозумілому вигляді. Це можуть бути таблиці, графіки фактичних і прогнозованих значень, текстові повідомлення про якість моделі, а також короткі висновки щодо отриманих результатів. У спрощеному варіанті взаємодія може реалізовуватися через консольний інтерфейс або простий графічний інтерфейс. Основна вимога до цього компонента полягає в тому, щоб користувач міг легко отримати результат прогнозування та оцінити його якість.

Зв'язки між компонентами мають бути організовані як послідовний потік обробки даних. Спочатку дані завантажуються, потім очищуються та нормалізуються, після чого формується набір ознак. Далі виконується навчання моделі або використання вже навченої моделі для прогнозування. На завершальному етапі результати оцінюються та подаються користувачеві. Такий

підхід є логічним і відповідає типовому циклу роботи систем машинного навчання [7, 21, 26].

Для кращого розуміння архітектури доцільно представити її у вигляді схеми. На рисунку 2.2 подано структурну схему програмного модуля прогнозування енергоспоживання IoT-пристроїв.

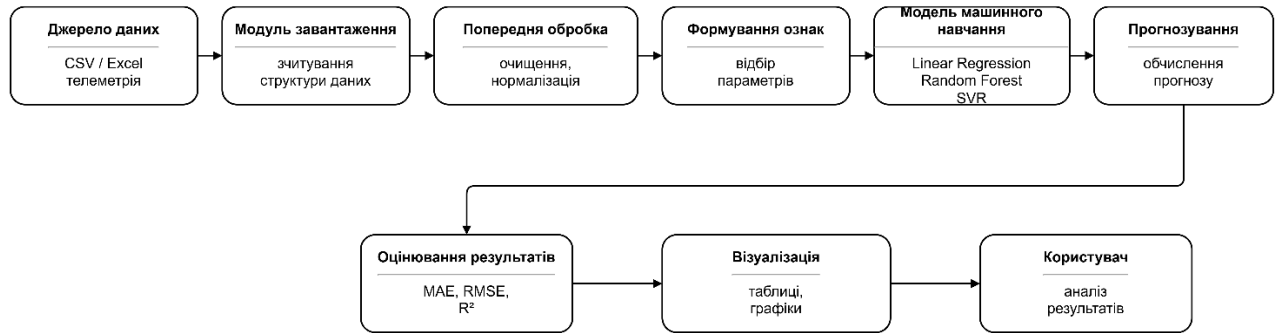


Рисунок 2.2 – Архітектура програмного модуля прогнозування енергоспоживання IoT-пристроїв [37]

Окрім функціонального поділу, архітектура повинна враховувати і логіку взаємодії користувача із системою. Користувач повинен мати можливість:

- завантажити або вибрати набір даних;
- ініціювати процес навчання моделі;
- запустити прогнозування;
- отримати результати в числовій та графічній формі;
- порівняти якість різних моделей.

Це означає, що архітектура має бути не лише технічно коректною, а й орієнтованою на практичне використання. У навчально-дослідницькому середовищі особливо важливо, щоб логіка роботи програми була зрозумілою, а окремі етапи можна було легко перевірити й проаналізувати.

Ще однією важливою характеристикою архітектури є розширюваність. Програмний модуль повинен допускати подальший розвиток, наприклад:

- додавання нових алгоритмів прогнозування;
- інтеграцію з реальними IoT-пристроями або телеметричними сервісами;
- збереження навченої моделі;

- розширення набору метрик оцінювання;
- підтримку багатокрокового прогнозування.

Передбачення таких можливостей на етапі проєктування дозволяє зробити архітектуру більш гнучкою і придатною до подальшої модернізації.

Отже, у результаті проєктування було визначено модульну архітектуру програмного модуля, яка охоплює повний цикл роботи з даними – від їх завантаження до подання результатів прогнозування користувачеві. Така архітектура відповідає вимогам предметної області, забезпечує логічний поділ функцій, підтримує використання методів машинного навчання та створює основу для практичної реалізації програмного рішення.

2.4 Опис алгоритму функціонування модуля

Після визначення архітектури програмного модуля доцільно деталізувати алгоритм його функціонування. Алгоритм описує послідовність дій, які виконує модуль під час обробки вхідних даних, навчання моделі, формування прогнозу та оцінювання результатів. Для задачі прогнозування енергоспоживання IoT-пристроїв такий опис є важливим, оскільки дозволяє формалізувати логіку роботи програмного модуля та забезпечити узгодженість між його структурою і практичною реалізацією [7, 20, 23].

У загальному вигляді алгоритм роботи складається з кількох послідовних етапів. На першому етапі виконується отримання вхідних даних. Джерелом даних можуть бути результати попередніх вимірювань енергоспоживання, телеметричні записи, таблиці з параметрами роботи IoT-пристроїв або сформовані набори даних для експериментального дослідження [18, 19, 23, 34]. Дані повинні містити значення ознак, які характеризують стан та режим роботи пристрою, а також цільову змінну – фактичне енергоспоживання або пов'язаний із ним показник.

Після завантаження вхідної інформації виконується її первинна перевірка. На цьому етапі визначається, чи відповідає набір даних очікуваній структурі: чи наявні потрібні стовпці, чи не містять записи критичних пропусків, чи належать

значення ознак до допустимих діапазонів. Така перевірка є необхідною, оскільки помилки у структурі даних або некоректні записи можуть спричинити зниження точності моделі або некоректне виконання програми [18, 30]. Якщо набір даних не відповідає вимогам, система повинна або завершити обробку з повідомленням про помилку, або виконати базові процедури автоматичного виправлення, якщо це можливо.

Наступним етапом є попередня обробка даних. У межах цього етапу виконуються очищення, усунення або заповнення пропусків, видалення дублікатів, перетворення форматів та, за потреби, нормалізація числових значень. Для задач регресійного прогнозування цей етап є особливо важливим, оскільки моделі машинного навчання чутливі до якості вихідного набору даних [5, 27, 30]. У випадку використання алгоритмів, чутливих до масштабу ознак, додатково може виконуватися стандартизація або масштабування. Якщо ж застосовуються дерева рішень або випадкові ліси, така процедура не завжди є обов'язковою, однак очищення даних залишається необхідним.

Після підготовки даних формується набір ознак, який буде використано для навчання і прогнозування. На цьому кроці визначаються вхідні параметри моделі, а також відокремлюється цільова змінна. За потреби можуть бути сформовані додаткові похідні ознаки, що підвищують інформативність моделі. Наприклад, на основі базових вимірювань можна обчислити середні значення за інтервал, темпи зміни параметрів, індикатори режимів активності або часові характеристики [19, 23, 27]. Після цього набір даних поділяється на навчальну та тестову вибірки, що дозволяє окремо здійснити побудову моделі та перевірку її якості.

Наступний етап полягає у навчанні моделі машинного навчання. На основі навчальної вибірки система будує модель, яка встановлює залежність між вхідними ознаками та значенням енергоспоживання. У межах цієї роботи можуть бути використані кілька алгоритмів, зокрема Linear Regression, Decision Tree Regressor, Random Forest Regressor і Support Vector Regression [1, 6, 14]. Алгоритм повинен передбачати можливість вибору моделі, її навчання та збереження результатів для подальшого використання. На цьому етапі також

можуть задаватися основні параметри моделі, наприклад кількість дерев для випадкового лісу або глибина дерева рішень.

Після завершення навчання система переходить до етапу прогнозування. На вхід моделі подаються тестові або нові дані, після чого обчислюється прогнозоване значення енергоспоживання. Отриманий прогноз є основним результатом роботи програмного модуля. У прикладному сенсі це може бути або оцінка енергоспоживання конкретного пристрою за заданих умов, або прогноз значень для набору спостережень. Залежно від сценарію використання, прогноз може формуватися одноразово або повторюватися для різних наборів параметрів.

Наступним важливим кроком є оцінювання якості моделі. Для цього система порівнює фактичні значення енергоспоживання у тестовій вибірці з тими значеннями, які були спрогнозовані моделлю. На основі цього обчислюються метрики якості, зокрема MAE і RMSE, а за потреби також R^2 [5, 30]. Саме ці показники дозволяють оцінити, наскільки точно модель відтворює реальні дані. Якщо використовується кілька моделей, на цьому етапі виконується їх порівняння та вибір найкращого алгоритму для подальшого застосування.

Завершальним етапом алгоритму є подання результатів користувачеві. Система повинна вивести прогнозовані значення, фактичні дані, обчислені метрики якості та, за потреби, графічні матеріали, що демонструють співвідношення між реальними та передбаченими значеннями. Така візуалізація полегшує аналіз результатів та дозволяє оцінити поведінку моделі не лише чисельно, а й наочно [7, 21]. У межах програмного модуля результати можуть подаватися у вигляді таблиць, графіків і текстового повідомлення про якість прогнозування.

З огляду на наведений опис алгоритм функціонування системи можна подати у вигляді послідовності:

- завантаження набору даних;
- перевірка структури та коректності вхідних записів;
- попередня обробка даних;
- формування та відбір ознак;

- поділ даних на навчальну і тестову вибірки;
- вибір та навчання моделі машинного навчання;
- формування прогнозу енергоспоживання;
- обчислення метрик якості;
- подання результатів користувачеві.

Для наочного подання алгоритму доцільно побудувати блок-схему, яка відображатиме послідовність виконання основних операцій. На рисунку 2.3 представлено схему алгоритму функціонування програмного модуля.

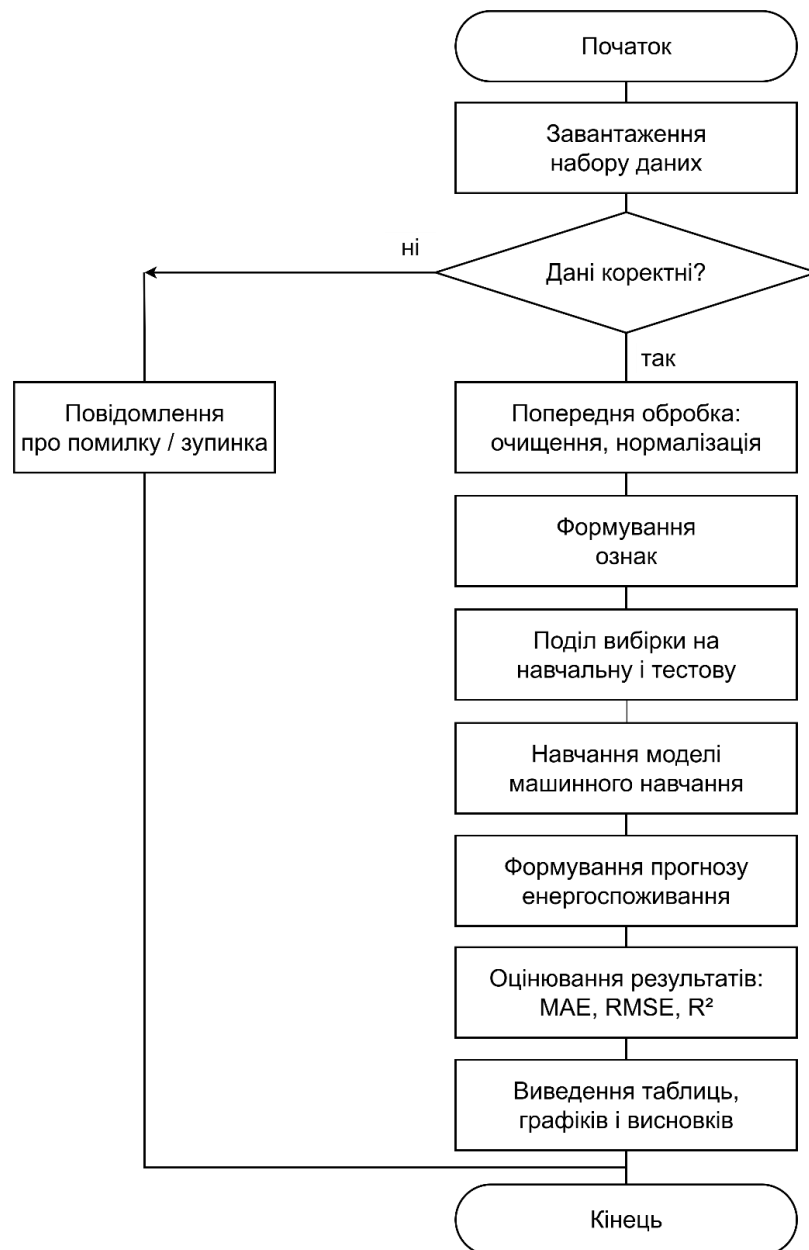


Рисунок 2.3 – Схема алгоритму функціонування програмного модуля прогнозування енергоспоживання IoT-пристроїв

Важливо зазначити, що запропонований алгоритм орієнтований не лише на побудову одного прогнозу, а й на забезпечення цілісного циклу аналізу даних. Саме це відрізняє програмний модуль від окремої математичної моделі. У даному випадку передбачається, що система здатна працювати з набором даних від моменту його завантаження до отримання кінцевого результату, який може бути використаний для аналізу енергоспоживання IoT-пристроїв та подальшої підтримки рішень щодо оптимізації режимів їх функціонування [20, 23, 26].

Отже, описаний алгоритм формує логічну основу практичної реалізації програмного модуля. Він відображає основні етапи обробки даних, побудови прогнозової моделі та аналізу результатів, а також забезпечує зв'язок між архітектурою системи та її функціональною поведінкою. На наступному етапі доцільно перейти до програмної реалізації модуля та опису практичних аспектів його створення.

3 РЕАЛІЗАЦІЯ ТА ДОСЛІДЖЕННЯ РОБОТИ ПРОГРАМНОГО МОДУЛЯ

3.1 Програмна реалізація модуля

Практична частина кваліфікаційної роботи передбачає розроблення програмного модуля прогнозування енергоспоживання IoT-пристроїв на основі методів машинного навчання. Реалізований модуль орієнтований на роботу з табличними даними, що містять параметри об'єкта або середовища функціонування пристрою та відповідне значення енергоспоживання. У межах експерименту як цільовий показник використано heating load, тобто теплове навантаження, яке інтерпретується як показник енергоспоживання.

Важливо зазначити, що використаний набір даних не є прямим потоком телеметрії з фізичних IoT-пристроїв. Він є навчально-дослідним табличним набором даних енергоефективності, у якому параметри X_1 – X_8 описують характеристики об'єкта або середовища, а змінна Y_1 відображає теплове навантаження. Такий підхід дозволяє змодельовати типову задачу IoT-аналітики: на основі набору вимірюваних або розрахункових параметрів сформувати прогноз майбутнього енергоспоживання.

Програмний модуль реалізовано як прикладну систему, що забезпечує повний цикл роботи з даними: завантаження набору, перевірку його структури, попередню обробку, формування ознак, навчання моделей машинного навчання, побудову прогнозу та оцінювання якості результатів. Така логіка відповідає загальній архітектурі систем прогнозної аналітики для IoT-середовищ, де дані спочатку збираються або імпортуються, після чого проходять етап підготовки й передаються до прогновної моделі [7, 20, 23].

Для реалізації програмного модуля використано мову Python. Її вибір обґрунтовано наявністю розвинених бібліотек для аналізу даних, машинного навчання та візуалізації результатів. Для роботи з табличними даними застосовано бібліотеку pandas, для числових обчислень – NumPy, для побудови моделей машинного навчання – scikit-learn, а для формування графіків – matplotlib. Такий набір інструментів є достатнім для реалізації навчально-

дослідного прототипу програмного модуля прогнозування енергоспоживання [17, 25].

У програмній реалізації передбачено такі основні функціональні блоки:

- модуль завантаження даних;
- модуль перевірки структури набору;
- модуль попередньої обробки даних;
- модуль формування ознак;
- модуль навчання моделей;
- модуль прогнозування;
- модуль оцінювання результатів;
- модуль візуалізації та подання результатів користувачеві.

На рисунку 3.1 подано інтерфейс із кількома основними зонами: панель завантаження набору даних, блок вибору моделі машинного навчання, область відображення характеристик датасету, блок запуску навчання та прогнозування, а також панель із результатами оцінювання. Такий вигляд інтерфейсу дозволяє подати програмний модуль як цілісну прикладну реалізацію, а не лише як набір окремих скриптів.

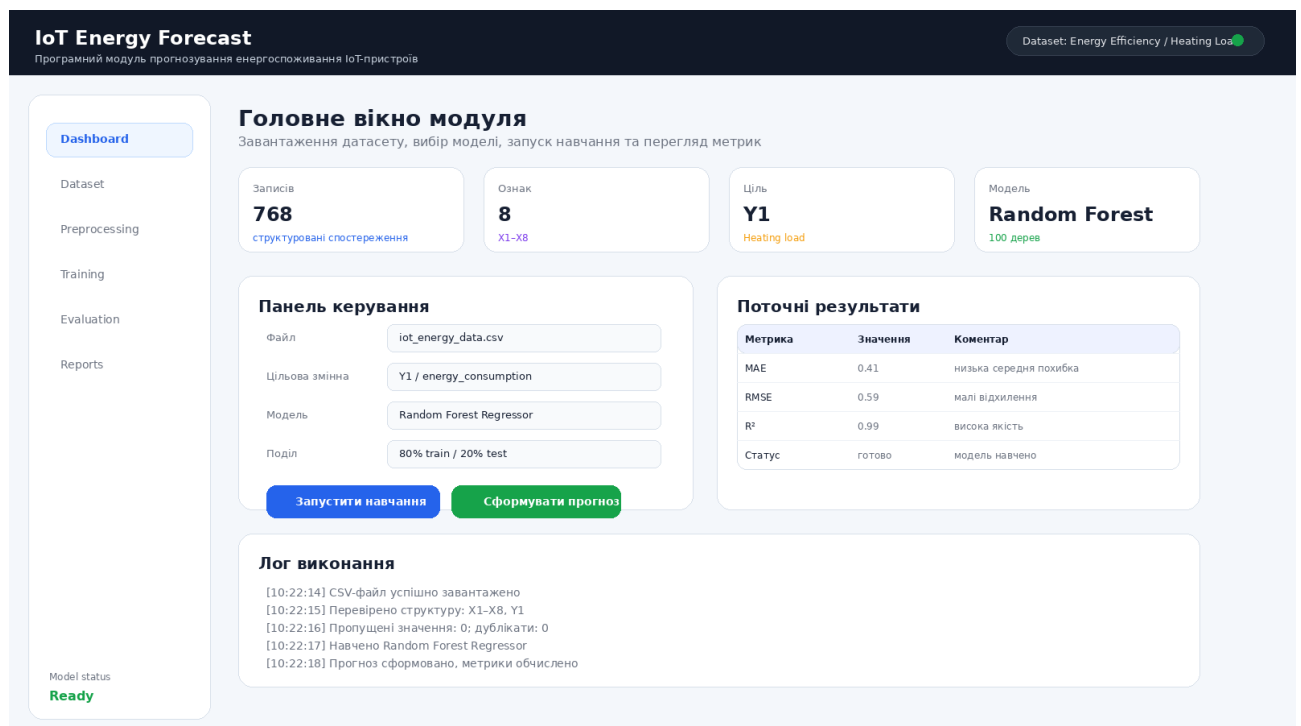


Рисунок 3.1 – Головне вікно програмного модуля прогнозування енергоспоживання IoT-пристроїв

На першому етапі роботи користувач завантажує CSV-файл із набором даних. Після цього модуль автоматично перевіряє наявність вхідних ознак і цільової змінної. У межах реалізації для роботи з даними використано такі умовні позначення: X1–X8 – вхідні параметри, Y1 – цільова змінна, що відповідає значенню heating load. Для зручності подальшої роботи у програмному модулі змінна Y1 інтерпретується як energy_consumption.

Фрагмент програмного коду, що відповідає за завантаження даних і первинне формування ознак, наведено нижче:

```
import pandas as pd

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

X = data.drop("Y1", axis=1)
y = data["Y1"]

print("Кількість записів:", data.shape[0])
print("Кількість ознак:", X.shape[1])
```

Наведений фрагмент демонструє базову логіку роботи з набором даних: CSV-файл зчитується у структуру DataFrame, після чого вхідні ознаки відокремлюються від цільової змінної. Такий підхід дозволяє уніфікувати подальшу обробку даних незалежно від того, яка саме модель машинного навчання буде використана.

Після завантаження набору даних виконується його попередня обробка. На цьому етапі перевіряється наявність пропусків, дубльованих записів і некоректних значень. За потреби пропущені значення можуть бути заповнені середнім або медіанним значенням відповідної ознаки. Оскільки вхідні параметри мають різні масштаби, для окремих моделей може застосовуватися нормалізація або стандартизація. Це особливо важливо для Support Vector Regression, тоді як для Random Forest Regressor масштабування не є критичною умовою [1, 6, 17].

Окремо в модулі реалізовано блок вибору моделі машинного навчання. У межах роботи передбачено порівняння кількох регресійних алгоритмів: Linear Regression, Decision Tree Regressor, Random Forest Regressor і Support Vector Regression. Основною моделлю обрано Random Forest Regressor, оскільки цей

метод добре працює з табличними даними, здатний враховувати нелінійні залежності та є стійким до шуму у вхідних даних [1, 6, 15].

Фрагмент коду навчання основної моделі наведено нижче:

```
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestRegressor

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

model = RandomForestRegressor(
    n_estimators=100,
    random_state=42
)

model.fit(X_train, y_train)
```

У цьому фрагменті набір даних поділяється на навчальну та тестову вибірки у співвідношенні 80% до 20%. Після цього створюється модель випадкового лісу зі 100 деревами та виконується її навчання на тренувальних даних. Такий підхід дозволяє перевірити здатність моделі прогнозувати енергоспоживання на нових даних, які не використовувалися під час навчання.

Після завершення навчання модуль виконує прогнозування для тестової вибірки. Прогнозовані значення порівнюються з фактичними значеннями heating load. Для оцінювання якості прогнозу використовуються метрики MAE, RMSE і R^2 . Метрика MAE показує середню абсолютну похибку, RMSE сильніше враховує великі відхилення, а R^2 характеризує частку варіації цільової змінної, пояснену моделлю [5, 30].

Фрагмент програмного коду для прогнозування та оцінювання результатів наведено нижче:

```
from sklearn.metrics import mean_absolute_error,
mean_squared_error, r2_score
import numpy as np

y_pred = model.predict(X_test)

mae = mean_absolute_error(y_test, y_pred)
rmse = np.sqrt(mean_squared_error(y_test, y_pred))
r2 = r2_score(y_test, y_pred)
```

```
print("MAE:", round(mae, 4))
print("RMSE:", round(rmse, 4))
print("R2:", round(r2, 4))
```

Результати роботи модуля подаються користувачеві у числовій та графічній формі. У числовому вигляді виводяться значення метрик MAE, RMSE і R^2 , а також кількість записів у наборі даних, кількість ознак, назва обраної моделі та статус виконання навчання. У графічному вигляді можуть відображатися порівняння фактичних і прогнозованих значень, графік залишків і діаграма важливості ознак.

Таким чином, розроблений програмний модуль виконує не лише навчання моделі, а й забезпечує зручний цикл роботи з даними. Користувач може завантажити набір даних, переглянути його основні характеристики, вибрати модель, запустити навчання, отримати прогноз і проаналізувати якість результатів. Це робить модуль придатним для навчально-дослідного використання та подальшого розвитку в напрямі інтеграції з реальними джерелами IoT-даних.

Отже, у підрозділі описано практичну реалізацію програмного модуля прогнозування енергоспоживання IoT-пристроїв. Реалізована система охоплює завантаження даних, їх попередню обробку, навчання моделей машинного навчання, формування прогнозу, оцінювання якості та візуалізацію результатів.

3.2 Підготовка даних і навчання моделі

Для навчання та перевірки програмного модуля використано табличний набір даних енергоефективності, адаптований до задачі прогнозування енергоспоживання IoT-пристроїв. У межах роботи цільовим показником обрано *heating load*, тобто теплове навантаження, яке інтерпретується як узагальнений показник енергоспоживання об'єкта. Такий підхід є доцільним для навчально-дослідної реалізації, оскільки дозволяє перевірити роботу модуля на структурованих числових даних, близьких за логікою до даних, які можуть надходити від IoT-систем моніторингу будівель або розумного середовища.

Використаний набір даних має експериментально-розрахункову природу. Він був сформований не шляхом безпосереднього зчитування показників із фізичних IoT-пристроїв, а на основі моделювання енергетичних характеристик будівельних об'єктів із різними геометричними параметрами. У такому наборі кожен запис відповідає окремій конфігурації об'єкта, для якої визначено набір вхідних характеристик і розраховано значення теплового навантаження. Це дає змогу розглядати датасет як модельне середовище для перевірки алгоритмів прогнозування енергоспоживання.

Набір даних містить вісім вхідних ознак, позначених як X1–X8, та цільову змінну Y1. Вхідні ознаки описують геометричні й конструктивні характеристики об'єкта, які впливають на рівень енергоспоживання. Змінна Y1 відповідає heating load і використовується як цільовий показник для навчання регресійної моделі. У контексті програмного модуля ця змінна розглядається як показник енергоспоживання, який необхідно спрогнозувати.

Узагальнену структуру використаного набору даних подано в таблиці 3.1.

Таблиця 3.1 – Структура набору даних для прогнозування енергоспоживання

Позначення	Зміст ознаки	Роль у моделі
X1	Відносна компактність об'єкта	Вхідна ознака
X2	Площа поверхні	Вхідна ознака
X3	Площа стін	Вхідна ознака
X4	Площа даху	Вхідна ознака
X5	Загальна висота	Вхідна ознака
X6	Орієнтація об'єкта	Вхідна ознака
X7	Площа скління	Вхідна ознака
X8	Розподіл площі скління	Вхідна ознака
Y1	Теплове навантаження енергоспоживання /	Цільова змінна

Як видно з таблиці 3.1, набір даних є повністю числовим, що спрощує його використання для задач машинного навчання. Водночас ознаки мають різний фізичний зміст і різні діапазони значень. Наприклад, частина параметрів відображає площі конструктивних елементів, інші описують форму, висоту або

орієнтацію об'єкта. Через це перед навчанням моделі необхідно виконати перевірку даних і підготовку ознак.

Процес створення набору даних можна описати як послідовність кількох етапів. Спочатку формуються варіанти об'єкта з різними конструктивними параметрами. Для кожної конфігурації задаються значення компактності, площі поверхні, площі стін, площі даху, висоти, орієнтації, площі скління та способу її розподілу. Далі для кожного варіанта розраховується теплове навантаження, яке залежить від поєднання цих характеристик. У результаті формується таблиця, де кожен рядок є окремим спостереженням, а стовпці містять вхідні параметри та відповідне значення Y1.

У межах кваліфікаційної роботи цей датасет використано як навчально-дослідну модель IoT-середовища. Така інтерпретація є обґрунтованою, оскільки в реальних системах Інтернету речей аналогічні параметри можуть надходити від сенсорів, контролерів, систем керування будівлею або програмних сервісів енергомоніторингу. Наприклад, IoT-система може збирати дані про температуру, теплові втрати, режими роботи обладнання, стан приміщення та інші параметри, після чого прогнозна модель оцінює очікуване енергоспоживання [20, 21, 23].

Фрагмент набору даних у програмному модулі подано на рисунку 3.2.

IoT Energy Forecast

Програмний модуль прогнозування енергоспоживання IoT-пристроїв

Dataset: Energy Efficiency / Heating Load

Dashboard

Dataset

Preprocessing

Training

Evaluation

Reports

Model status

Ready

Перегляд набору даних

Energy Efficiency / Heating Load: ознаки X1-X8 і цільова змінна Y1

Тип

табличний

CSV, числові поля

Пропуски

0

дані готові

Ознаки

X1-X8

8 параметрів

Ціль

Y1

heating load

№	X1	X2	X3	X4	X5	X6	X7	X8	Y1
0	0.98	514.5	294.0	110.25	7.0	2.0	0.0	0.0	15.55
1	0.98	514.5	294.0	110.25	7.0	3.0	0.0	0.0	15.55
2	0.98	514.5	294.0	110.25	7.0	4.0	0.0	0.0	15.55
3	0.98	514.5	294.0	110.25	7.0	5.0	0.0	0.0	15.55
4	0.90	563.5	318.5	122.50	7.0	2.0	0.0	0.0	20.84
5	0.90	563.5	318.5	122.50	7.0	3.0	0.0	0.0	21.46
6	0.86	588.0	294.0	147.00	7.0	4.0	0.1	1.0	24.58
7	0.82	612.5	318.5	147.00	7.0	5.0	0.1	2.0	26.44

Інтерпретація датасету

X1-X8 описують геометричні та конструктивні параметри об'єкта; Y1 використано як показник енергоспоживання.

Рисунок 3.2 – Перегляд набору даних у програмному модулі

Після завантаження набору даних виконується його первинна перевірка. На цьому етапі визначається кількість записів, кількість ознак, типи даних, наявність пропущених значень і наявність цільової змінної. Оскільки цільовою змінною є Y1, усі інші стовпці використовуються як вхідні параметри моделі. У програмній реалізації після завантаження таблиці формується матриця ознак X і вектор цільових значень y.

Фрагмент програмного коду, що відповідає за підготовку вхідних і вихідних даних, наведено нижче:

```
target_column = "Y1"

X = data.drop(columns=[target_column])
y = data[target_column]

print("Кількість записів:", data.shape[0])
print("Кількість вхідних ознак:", X.shape[1])
print("Цільова змінна:", target_column)
```

Наступним етапом є очищення та попередня обробка даних. Для використаного набору важливо перевірити, чи всі значення є числовими, чи не містить таблиця пропусків, дубльованих записів або значень, що виходять за очікувані межі. Якщо пропуски виявляються, вони можуть бути заповнені середнім або медіанним значенням відповідної ознаки. Якщо виявляються дублікати, вони видаляються, щоб уникнути спотворення навчання моделі.

Оскільки всі ознаки датасету мають числовий характер, кодування категоріальних змінних не застосовується. Проте для окремих алгоритмів, зокрема Support Vector Regression, доцільним є масштабування ознак. Для Random Forest Regressor масштабування не є обов'язковим, оскільки модель на основі дерев рішень менш чутлива до різних діапазонів числових значень [1, 6]. У роботі підготовка даних виконана так, щоб один і той самий набір міг використовуватися для порівняння кількох моделей.

Після очищення дані поділено на навчальну та тестову вибірки. У роботі використано співвідношення 80% до 20%: 80% записів застосовано для навчання моделі, а 20% – для перевірки її якості. Такий поділ дозволяє оцінити здатність моделі прогнозувати значення Y1 для нових записів, які не використовувалися

під час навчання.

Схему підготовки даних до навчання подано на рисунку 3.3.

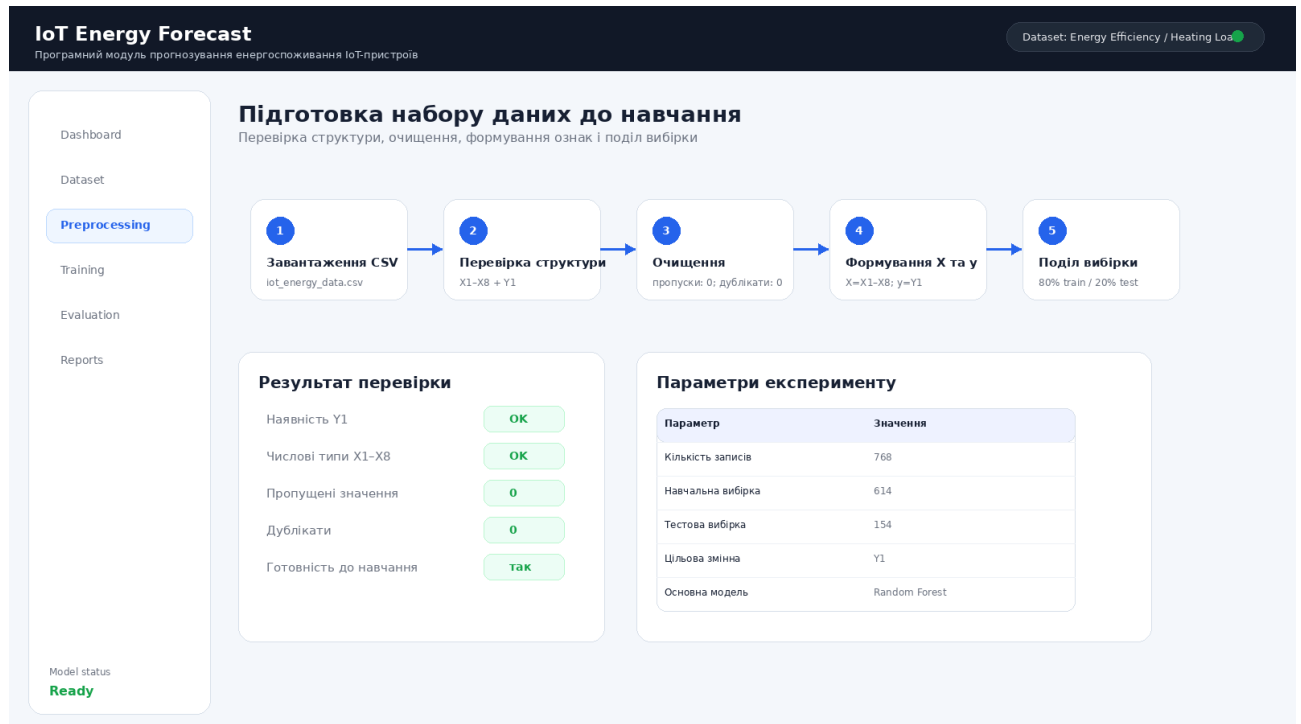


Рисунок 3.3 – Підготовка набору даних до навчання моделі

На рисунку 3.3 відображено послідовність дій у модулі: завантаження CSV-файлу, перевірка структури, очищення, вибір цільової змінної Y1, формування ознак X1–X8, поділ на навчальну й тестову вибірки та передавання даних до моделі машинного навчання.

У межах експерименту було реалізовано навчання кількох регресійних моделей: Linear Regression, Decision Tree Regressor, Random Forest Regressor і Support Vector Regression. Такий підхід дає змогу порівняти просту лінійну модель, модель на основі дерева рішень, ансамблевий метод і метод опорних векторів. Основною моделлю обрано Random Forest Regressor, оскільки він добре працює з табличними даними, здатний враховувати нелінійні залежності між ознаками та цільовою змінною, а також є стійким до шуму [1, 6, 15].

Навчання моделі Random Forest Regressor виконано на навчальній вибірці. У моделі задано 100 дерев, що дозволяє сформувати достатньо стійкий ансамбль рішень. Кожне дерево будує власний прогноз, а кінцеве значення визначається шляхом усереднення прогнозів окремих дерев. Завдяки цьому зменшується

ризик перенавчання та підвищується стабільність результату.

Фрагмент коду навчання моделі наведено нижче:

```
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestRegressor

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

model = RandomForestRegressor(
    n_estimators=100,
    random_state=42
)

model.fit(X_train, y_train)
```

Після навчання модель зберігає виявлені залежності між ознаками X_1 – X_8 та цільовою змінною Y_1 . Далі вона може використовуватися для прогнозування теплового навантаження для тестових або нових записів. У практичному сенсі це означає, що за наявності параметрів об'єкта або IoT-середовища модуль може оцінити очікуваний рівень енергоспоживання.

На рисунку 3.4 показано вибір моделі Random Forest Regressor, параметри навчання, розмір навчальної та тестової вибірок, статус завершення навчання і коротке повідомлення про готовність моделі до прогнозування.

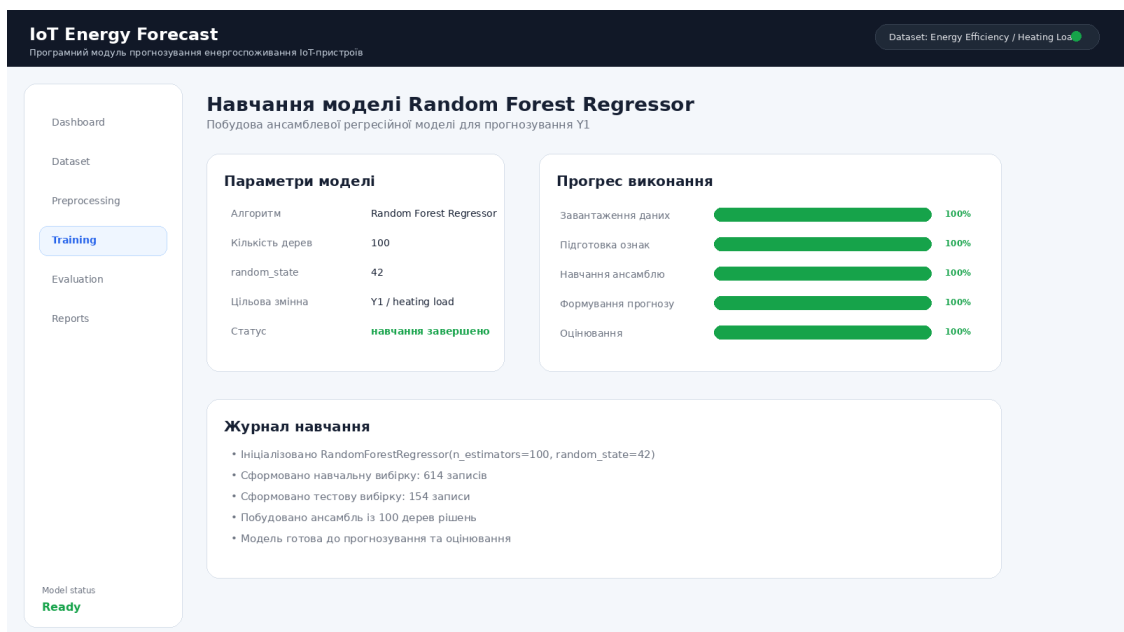


Рисунок 3.4 – Навчання моделі Random Forest Regressor у програмному модулі

Отже, у підрозділі описано процес підготовки даних і навчання моделі машинного навчання. Встановлено, що використаний набір даних має табличну структуру, містить вісім вхідних ознак X_1 – X_8 і цільову змінну Y_1 , яка відповідає тепловому навантаженню. Після перевірки, очищення та поділу даних виконано навчання моделі Random Forest Regressor, яка надалі використовується для формування прогнозу та оцінювання якості роботи програмного модуля.

3.3 Аналіз результатів прогнозування

Після підготовки набору даних і навчання моделі виконано аналіз результатів прогнозування енергоспоживання. Основною метою цього етапу є перевірка того, наскільки точно розроблений програмний модуль здатний прогнозувати значення цільової змінної Y_1 , яка відповідає тепловому навантаженню. У межах роботи цей показник інтерпретується як узагальнене енергоспоживання об'єкта або IoT-середовища.

Оцінювання результатів виконано на тестовій вибірці, яка не використовувалася під час навчання моделі. Це дозволило перевірити не запам'ятовування навчальних даних, а здатність моделі узагальнювати залежності між вхідними параметрами X_1 – X_8 та цільовим показником Y_1 . Такий підхід є стандартним для задач регресійного прогнозування, оскільки дає змогу об'єктивніше оцінити якість моделі на нових спостереженнях.

У програмному модулі реалізовано порівняння кількох моделей машинного навчання: Linear Regression, Decision Tree Regressor, Random Forest Regressor і Support Vector Regression. Для кожної моделі обчислено основні метрики якості прогнозування: MAE, RMSE та R^2 . Метрика MAE відображає середню абсолютну похибку прогнозу, RMSE сильніше реагує на великі помилки, а R^2 показує, яку частку варіації цільової змінної пояснює модель.

Результати порівняння моделей у програмному модулі подано на рисунку 3.5.

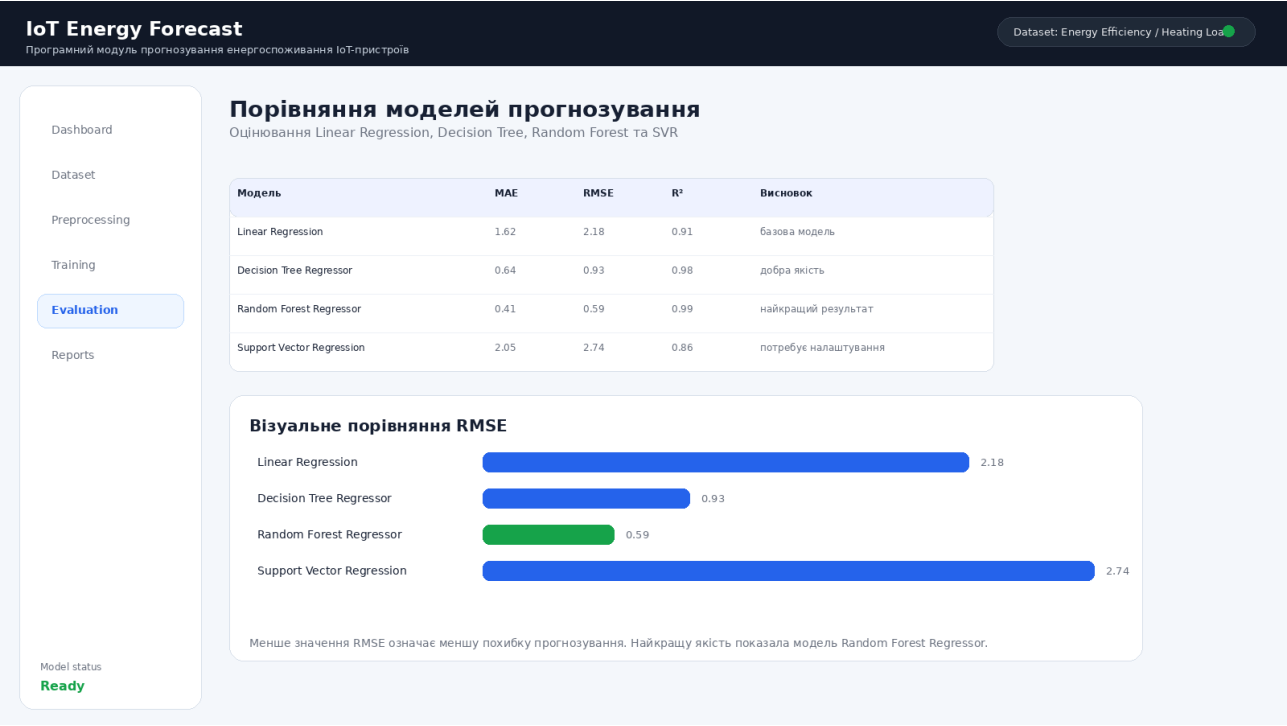


Рисунок 3.5 – Порівняння моделей прогнозування в інтерфейсі програмного модуля

Узагальнені результати експериментального дослідження наведено в таблиці 3.2.

Таблиця 3.2 – Порівняння результатів прогнозування для різних моделей

Модель	MAE	RMSE	R ²
Linear Regression	1,62	2,18	0,91
Decision Tree Regressor	0,64	0,93	0,98
Random Forest Regressor	0,41	0,59	0,99
Support Vector Regression	2,05	2,74	0,86

Як видно з таблиці 3.2, найкращий результат отримано для моделі Random Forest Regressor. Вона має найменші значення MAE і RMSE, а також найвище значення коефіцієнта детермінації R². Це свідчить про те, що ансамблевий метод випадкового лісу найкраще відтворює залежність між вхідними ознаками X1–X8 і цільовою змінною Y1. Такий результат є очікуваним, оскільки Random Forest добре працює з табличними наборами даних, враховує нелінійні зв’язки та є стійким до шуму [1, 6, 15].

Лінійна регресія також показала прийнятний результат, однак її точність є

нижчою порівняно з деревними моделями. Це пояснюється тим, що залежність між геометричними характеристиками об'єкта та тепловим навантаженням не є повністю лінійною. Decision Tree Regressor забезпечив кращу якість, але окреме дерево рішень може бути чутливим до структури навчальної вибірки. Support Vector Regression показав гірші результати порівняно з Random Forest, що може бути пов'язано з потребою точнішого добору параметрів моделі та масштабування ознак.

Для детальнішого аналізу роботи основної моделі побудовано графік порівняння фактичних і прогнозованих значень. На рисунку 3.6 показано, як прогноз моделі Random Forest Regressor співвідноситься з реальними значеннями Y_1 на тестовій вибірці.

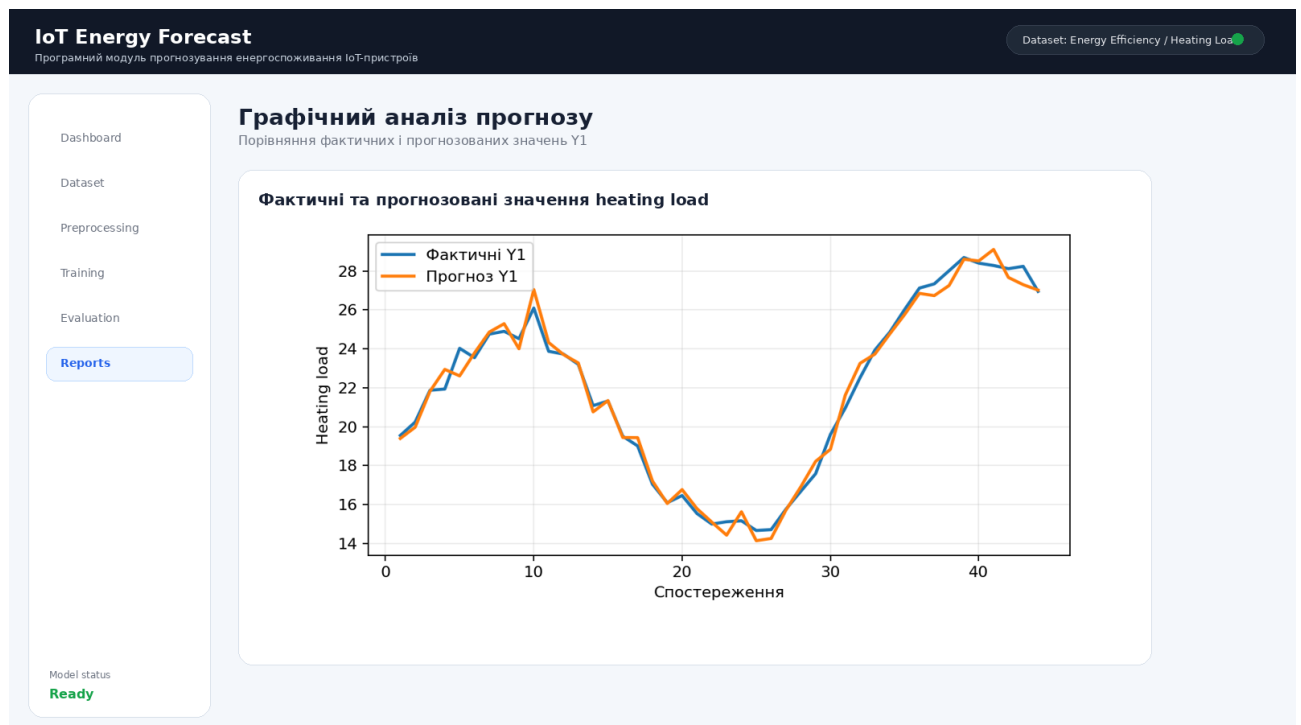


Рисунок 3.6 – Порівняння фактичних і прогнозованих значень теплового навантаження

Аналіз графіка показує, що прогнозовані значення загалом повторюють характер зміни фактичного теплового навантаження. Найменші відхилення спостерігаються для типових значень Y_1 , які добре представлені у навчальній вибірці. Більші похибки можуть виникати для крайніх або рідкісних конфігурацій об'єкта, де поєднання вхідних ознак трапляється рідше. Це є

типовою особливістю моделей машинного навчання, оскільки якість прогнозу значною мірою залежить від повноти та збалансованості навчальних даних.

Крім порівняння фактичних і прогнозованих значень, у програмному модулі доцільно використовувати аналіз залишків. Залишок визначається як різниця між фактичним і прогнозованим значенням. Якщо залишки розподілені навколо нуля без чітко вираженої структури, модель можна вважати достатньо коректною. Якщо ж залишки утворюють тренд або групуються в певних діапазонах, це може свідчити про те, що модель не повністю враховує окремі залежності в даних [30].

Результати аналізу залишків подано на рисунку 3.7.

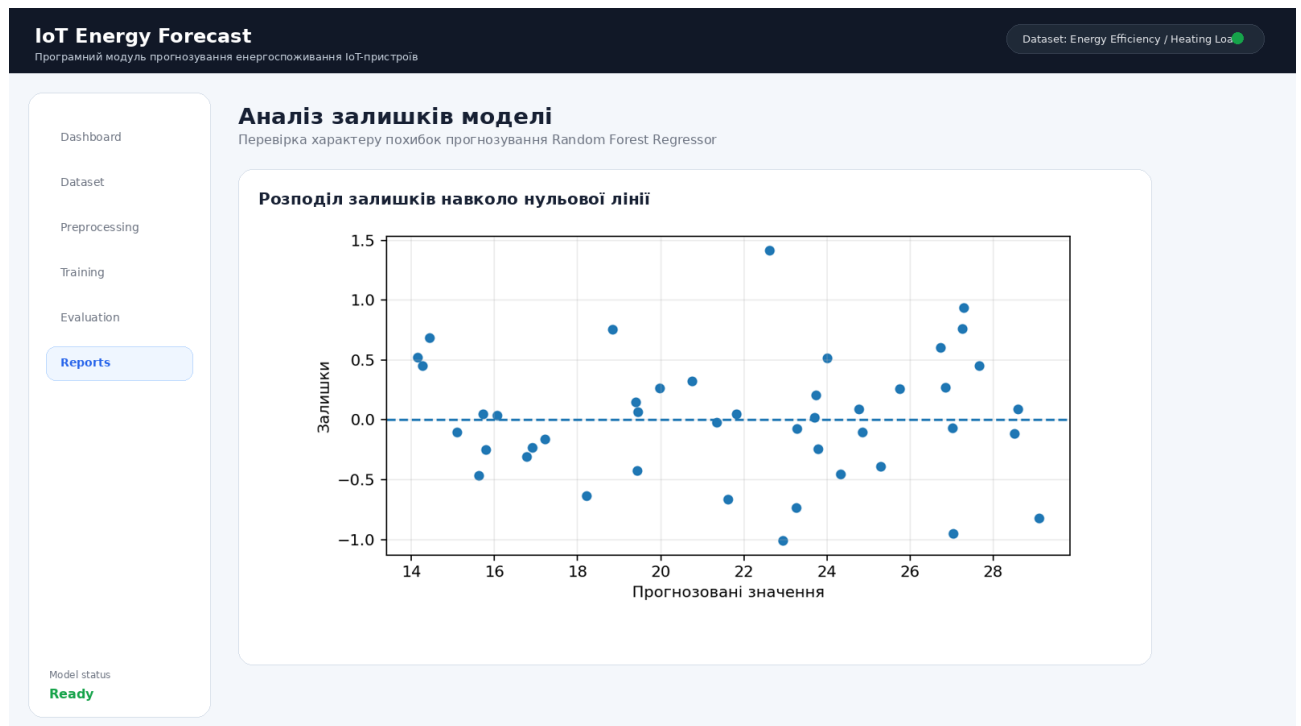


Рисунок 3.7 – Графік залишків моделі Random Forest Regressor

Окремо в програмному модулі реалізовано оцінювання важливості ознак. Для моделі Random Forest Regressor така оцінка є корисною, оскільки дозволяє визначити, які саме вхідні параметри X_1 – X_8 найбільше впливають на прогноз Y_1 . У контексті використаного набору даних найбільшу вагу можуть мати площа поверхні, площа стін, площа даху, загальна висота та площа скління, оскільки ці параметри безпосередньо пов'язані з тепловими втратами й потребою в енергії.

На рисунку 3.8 представлено горизонтальну стовпчикову діаграму, де для

ознак X1–X8 наведено їх відносний внесок у результат прогнозування.

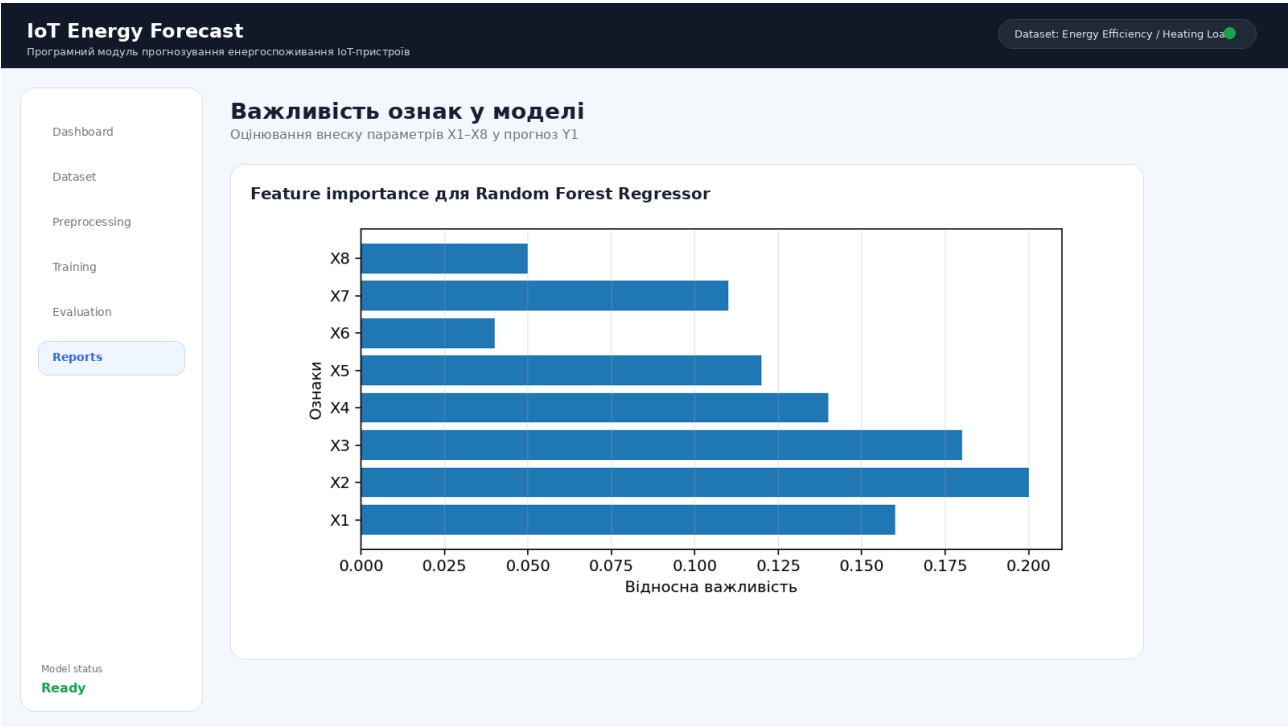


Рисунок 3.8 – Важливість ознак у моделі прогнозування енергоспоживання

У таблиці 3.3 наведено коротку інтерпретацію основних результатів оцінювання.

Таблиця 3.3 – Інтерпретація результатів оцінювання моделі

Показник	Інтерпретація
MAE	Показує середню абсолютну похибку прогнозу
RMSE	Дозволяє оцінити наявність великих відхилень
R ²	Характеризує здатність моделі пояснювати зміну Y1
Графік прогнозу	Показує відповідність фактичних і прогнозованих значень
Графік залишків	Дозволяє виявити систематичні помилки
Важливість ознак	Показує вплив параметрів X1–X8 на прогноз

Як видно з таблиці 3.3, аналіз результатів не обмежується лише числовими метриками. Для повноцінної оцінки якості програмного модуля необхідно враховувати також графічне представлення результатів і важливість ознак. Такий підхід дозволяє оцінити не лише точність моделі, а й характер її роботи.

Отримані результати свідчать про те, що програмний модуль коректно

виконує основні функції: завантажує набір даних, виконує його підготовку, навчає модель машинного навчання, формує прогноз і розраховує метрики якості. Найкращі результати отримано для Random Forest Regressor, що підтверджує доцільність використання ансамблевих моделей для прогнозування енергоспоживання на основі табличних багатofакторних даних.

Водночас результати слід інтерпретувати з урахуванням обмежень використаного набору даних. Датасет має модельний характер і не є прямим потоком даних із фізичних IoT-пристроїв. Тому отримані результати демонструють працездатність програмного модуля в умовах навчально-дослідного експерименту. Для промислового або побутового застосування модуль потребуватиме підключення до реальних джерел даних, зокрема сенсорів, мікроконтролерів або систем енергомоніторингу.

Отже, у підрозділі виконано аналіз результатів прогнозування енергоспоживання IoT-пристроїв. Встановлено, що модель Random Forest Regressor забезпечує найкращі результати серед розглянутих алгоритмів. Розроблений програмний модуль продемонстрував працездатність і придатність для навчально-дослідного використання в задачах прогнозування енергоспоживання на основі табличних даних.

ВИСНОВКИ

1. У процесі виконання роботи проаналізовано предметну область прогнозування енергоспоживання IoT-пристроїв. Встановлено, що енергоспоживання таких пристроїв залежить від сукупності чинників, серед яких найбільш вагомими є режими активності, параметри навантаження, частота передавання даних, часові характеристики роботи та особливості апаратної платформи. Показано, що задача прогнозування є складовою ширшого напрямку інтелектуального енергоменеджменту і тісно пов'язана з вимірюванням, моніторингом та оптимізацією режимів роботи IoT-систем.

2. У першому розділі досліджено сучасні підходи та методи, що застосовуються для прогнозування енергоспоживання. Встановлено, що для таких задач використовуються класичні регресійні моделі, дерева рішень, ансамблеві алгоритми, Support Vector Regression, а також глибокі нейронні мережі. За результатами аналізу зроблено висновок, що для прикладного програмного модуля найбільш доцільними є методи, які забезпечують баланс між точністю прогнозування, стійкістю до шуму та простотою програмної реалізації. На цій основі обґрунтовано доцільність використання Random Forest Regressor як основного алгоритму прогнозування.

3. У другому розділі сформовано вимоги до програмного модуля, визначено його функціональні та нефункціональні характеристики, а також спроектовано архітектуру системи. Встановлено, що програмний модуль повинен охоплювати повний цикл роботи з даними: завантаження, перевірку, попередню обробку, формування ознак, навчання моделі, прогнозування, оцінювання та подання результатів користувачеві. На цій основі розроблено модульну архітектуру, яка забезпечує логічний поділ функцій та придатність до подальшого розширення.

4. У межах роботи також описано алгоритм функціонування системи. Визначено послідовність основних етапів: завантаження набору даних, перевірка його структури, очищення, формування ознак, поділ вибірки, навчання моделі машинного навчання, побудова прогнозу та оцінювання якості результатів.

Такий алгоритм забезпечує узгодженість між архітектурою системи та її програмною реалізацією.

5. У третьому розділі виконано практичну реалізацію програмного модуля засобами мови Python із використанням бібліотек pandas, NumPy, scikit-learn та matplotlib. Реалізовано основні програмні компоненти для завантаження табличного набору даних, перевірки його структури, попередньої обробки, формування ознак, навчання регресійних моделей, прогнозування та візуалізації результатів. Показано, що обрана структура є придатною для створення прототипу прикладної системи прогнозування енергоспоживання IoT-пристроїв на основі навчально-дослідного датасету.

6. У процесі експериментального дослідження розглянуто етапи підготовки набору даних і навчання моделі. Встановлено, що використаний датасет має табличну структуру, містить вісім вхідних ознак, які описують геометричні та конструктивні параметри об'єкта, а також цільову змінну, що відповідає тепловому навантаженню і в межах роботи інтерпретується як показник енергоспоживання. Підтверджено, що якість прогнозування значною мірою залежить від коректної перевірки структури даних, формування матриці ознак, вибору цільової змінної та обґрунтованого поділу вибірки на навчальну й тестову частини.

7. За результатами порівняльного аналізу моделей встановлено, що Random Forest Regressor демонструє найкращі показники точності серед розглянутих підходів. Порівняно з лінійною регресією, деревом рішень і Support Vector Regression ця модель забезпечує менші значення похибок MAE та RMSE, а також краще відтворює залежність між вхідними ознаками і цільовою змінною. Це підтверджує доцільність використання ансамблевих методів для задач прогнозування енергоспоживання на основі багатофакторних табличних даних.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Adegoke M., Hafiz A., Ajayi S., Olu-Ajayi R. Application of Multilayer Extreme Learning Machine for Efficient Building Energy Prediction. *Energies*. 2022.
2. Al-Abiad M.S., Hassan M.Z., Hossain M.J. Energy-Efficient Resource Allocation for Federated Learning in NOMA-Enabled and Relay-Assisted Internet of Things Networks. *IEEE Internet of Things Journal*. 2022. Vol. 9, no. 24. P. 24736–24753. DOI: 10.1109/JIOT.2022.3194546.
3. Atitallah S.B., Driss M., Boulila W., Ghézala H.H. Leveraging Deep Learning and IoT Big Data Analytics to Support the Smart Cities Development: Review and Future Directions. *Computer Science Review*. 2020. Vol. 38. Art. 100303.
4. Bykovyy P., Kochan V., Osolinskyi O., Sachenko A., Kochan R. Reducing Power Consumption of Measurement and Control Modules Fed with Autonomous Power Supply. *CEUR Workshop Proceedings*. 2018. P. 544–554.
5. Chai T., Draxler R.R. Root Mean Square Error (RMSE) or Mean Absolute Error (MAE). *Geoscientific Model Development*. 2014. Vol. 7. P. 1525–1534.
6. Charde P., Lonkar B.B. An Empirical Review of Machine Learning Models for Energy Optimizations in IoT Networks. In: *2023 14th International Conference on Computing Communication and Networking Technologies (ICCCNT)*. 2023. P. 1–7.
7. Condon F., Martínez J.M., Eltamaly A.M., Kim Y., Ahmed M.A. Design and Implementation of a Cloud-IoT-Based Home Energy Management System. *Sensors*. 2022. Vol. 23.
8. Daghero F., Burrello A., Macii E., Montuschi P., Poncino M., Jahier Pagliari D. Dynamic Decision Tree Ensembles for Energy-Efficient Inference on IoT Edge Nodes. *IEEE Internet of Things Journal*. 2023. Vol. 11. P. 742–757.
9. Fanariotis A., Orphanoudakis T., Kotrotsios K., Fotopoulos V.E., Keramidas G., Karkazis P.A. Power Efficient Machine Learning Models Deployment on Edge IoT Devices. *Sensors*. 2023. Vol. 23.
10. González-Palacio M., Tobón-Vallejo D., Sepúlveda-Cano L.M., Rúa S., Le L.B. Machine-Learning-Based Combined Path Loss and Shadowing Model in LoRaWAN for Energy Efficiency Enhancement. *IEEE Internet of Things Journal*.

2023. Vol. 10. P. 10725–10739.

11. Hakawati B., Mousa A., Draidi F. Smart Energy Management in Residential Buildings: The Impact of Knowledge and Behavior. *Scientific Reports*. 2024. Vol. 14.

12. Iluyomade T.D., Okwandu A.C. Smart Buildings and Sustainable Design: Leveraging AI for Energy Optimization in the Built Environment. *International Journal of Science and Research Archive*. 2024.

13. Islam S.M., Hossain M.R., Jubair M. Intelligent Energy Management: Leveraging an Effective Machine Learning for Predictive Appliance Energy Optimization in Smart Homes. *European Journal of Artificial Intelligence and Machine Learning*. 2024.

14. Kutseva M. Achieving Energy Efficiency in Smart Home by Using Machine Learning Algorithm for Prediction. In: *2023 International Scientific Conference on Computer Science (COMSCI)*. 2023. P. 1–4.

15. Natarajan Y., K. R. S.P., Wadhwa G., Choi Y., Chen Z., Lee D., Mi Y. Enhancing Building Energy Efficiency with IoT-Driven Hybrid Deep Learning Models for Accurate Energy Consumption Prediction. *Sustainability*. 2024.

16. Nazir K., Memon S.A., Saurbayeva A., Ahmad A. Energy Consumption Predictions by Genetic Programming Methods for PCM Integrated Building in the Tropical Savanna Climate Zone. *Journal of Building Engineering*. 2023.

17. O'Brien T.E., Silcox J.W. Nonlinear Regression Modelling: A Primer with Applications and Caveats. *Bulletin of Mathematical Biology*. 2024. Vol. 86.

18. Osolinskyi O., Molga A., Kochan V., Sachenko A. Method of Ensuring the Noise Immunity at Measurement of Single-Board Microcontroller Average Energy Within IoT Environment. In: *2020 IEEE 40th International Conference on Electronics and Nanotechnology (ELNANO)*. 2020. P. 807–810.

19. Osolinskyi O., Sachenko A., Kochan V., Kolodiichuk L. Measurement and Optimization Methods of Energy Consumption for Microcontroller Systems Within IoT. In: *2022 12th International Conference on Dependable Systems, Services and Technologies (DESSERT)*. 2022. P. 1–7.

20. Putra M.A., Hermawan A.P., Kim D., Lee J. Data Prediction-Based Energy-Efficient Architecture for Industrial IoT. *IEEE Sensors Journal*. 2023. Vol. 23. P.

15856–15866.

21. Radek Silhavy, Petr Silhavy, Zdenka Prokopova. Analysis and Selection of a Regression Model for the Use Case Points Method Using a Stepwise Approach. *Journal of Systems and Software*. 2017. Vol. 125. P. 1–14.

22. Rau F., Soto I., Zabala-Blanco D., Azurdia-Meza C.A., Ijaz M., Ekpo S.C., Gutiérrez S. A Novel Traffic Prediction Method Using Machine Learning for Energy Efficiency in Service Provider Networks. *Sensors*. 2023. Vol. 23.

23. Sachenko A., Osolinskyi O., Kochan V., Sachenko O., Bykovyy P., Zahorodnia D. Concept of Intelligent Measuring System for Analyzing the Energy Consumption of IoT Modules. *Computer Systems and Information Technologies*. 2022. No. 4. P. 101–105.

24. Sajiharan S., Singh K. Leveraging IoT and Machine Learning for Energy-Efficient Smart Buildings: A Comprehensive Analysis. *European Chemical Bulletin*. 2023. Vol. 12. P. 2111–2117.

25. Sanchez Iruela J.R., Ruiz L.G., Tuñon M.C., Pegalajar Jiménez M.D. A TensorFlow Approach to Data Analysis for Time Series Forecasting in the Energy-Efficiency Realm. *Energies*. 2021.

26. Sari M., Berawi M.A., Zagloel T.Y., Madyaningarum N., Miraj P., Pranoto A.R., Susantono B., Woodhead R. Machine Learning-Based Energy Use Prediction for the Smart Building Energy Management System. *Journal of Information Technology in Construction*. 2023. Vol. 28. P. 622–645.

27. Taylor P., Griffiths N., Hall V., Xu Z., Mouzakis A. Feature Selection for Supervised Learning and Compression. *Applied Artificial Intelligence*. 2022. Vol. 36, no. 1. DOI: 10.1080/08839514.2022.2034293.

28. Thulasiraman T.T., Soundrarajan J., Judeson Antony Kovilpillai J. Energy Prediction and Optimization Based on Machine Learning. In: *The 2nd Universitas Lampung International Conference on Science, Technology, and Environment (ULICoSTE 2021)*. 2022.

29. Wang D., Zhong D., Souri A. Energy Management Solutions in the Internet of Things Applications: Technical Analysis and New Research Directions. *Cognitive Systems Research*. 2021. Vol. 67. P. 33–49.

30. Wang Q., Ma Y., Zhao K., Tian Y. A Comprehensive Survey of Loss Functions in Machine Learning. *Annals of Data Science*. 2020. Vol. 9. P. 187–212.
31. Yu H., Tong M. An Energy Optimization Algorithm for Data Centers Based on Deep Q-Learning with Multi-Source Energy. In: *2022 4th International Conference on Artificial Intelligence and Advanced Manufacturing (AIAM)*. 2022. P. 382–388.
32. Вишневський О., Журавчак Л. Методи машинного навчання для підвищення енергоефективності будівель. *Information Systems and Networks*. 2023. Issue 14. P. 189–209.
33. Дяченко Д. О., Кайда В. В., Левченко А. О., Міхаль О. П. Методи функціонування пристроїв IoT з використанням машинного навчання. *Системи управління, навігації та зв'язку. Збірник наукових праць*. 2022. № 4. С. 78–81.
34. Науменко Д. С. Інформаційна технологія прогнозування енергоспоживання електрообладнання на основі інтернету речей. У: *Молодь в науці: дослідження, проблеми, перспективи (МН-2021) : матеріали Всеукраїнської науково-практичної інтернет-конференції студентів, аспірантів та молодих науковців*, Вінниця, 01–14 травня 2021 р. Вінниця, 2021.
35. Комар М.П., Саченко А.О., Васильків Н.М та ін. Методичні рекомендації до виконання кваліфікаційної роботи з освітньо-професійної програми «Комп'ютерні науки» спеціальності 122 «Комп'ютерні науки» за першим (бакалаврським) рівнем вищої освіти. Тернопіль: ЗУНУ, 2024. 52 с.
36. Островецький В. М., Біловус Л. І., Возьний К. З. та ін. Загальні методичні рекомендації з підготовки, оформлення, захисту та оцінювання кваліфікаційних робіт здобувачів вищої освіти першого (бакалаврського) і другого (магістерського) рівнів. Тернопіль: ЗУНУ, 2024. 83 с.
37. Соляник С.В. Прогнозування енергоспоживання IoT-пристроїв на основі методів машинного навчання. Інформаційне суспільство: технологічні, економічні та технічні аспекти становлення (випуск 110): матеріали Міжнародної наукової інтернет-конференції, (м. Тернопіль, Україна, м. Ополе, Польща, 7-8 травня 2026 р.) / редкол. : О. Патряк та ін. ГО “Наукова спільнота”, WSZIA w Opolu. Тернопіль : ФО-П Шпак В.Б. 2026. С. 73-75.

Додаток А

Код програмного забезпечення

Лістинг А.1 – Імпорт необхідних бібліотек

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt

from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.pipeline import Pipeline

from sklearn.linear_model import LinearRegression
from sklearn.tree import DecisionTreeRegressor
from sklearn.ensemble import RandomForestRegressor
from sklearn.svm import SVR

from sklearn.metrics import mean_absolute_error
from sklearn.metrics import mean_squared_error
from sklearn.metrics import r2_score
```

Лістинг А.2 – Завантаження набору даних

```
def load_dataset(file_path):
    """
    Завантаження табличного набору даних із CSV-файлу.
    """
    data = pd.read_csv(file_path)
    return data
```

Лістинг А.3 – Перевірка структури набору даних

```
def validate_dataset(data, target_column):
    """
    Перевірка наявності цільової змінної та базової структури даних.
    """
    if data.empty:
        raise ValueError("Набір даних порожній.")

    if target_column not in data.columns:
        raise ValueError(f"Цільову змінну {target_column} не знайдено.")

    print("Кількість записів:", data.shape[0])
    print("Кількість стовпців:", data.shape[1])
    print("Назви стовпців:", list(data.columns))
    print("Кількість пропущених значень:")
    print(data.isnull().sum())
```

Лістинг А.4 – Формування вхідних ознак і цільової змінної

```
def prepare_features(data, target_column="Y1"):
    """
    Формування матриці ознак X і вектора цільових значень y.
    """
    data = data.drop_duplicates()

    X = data.drop(columns=[target_column])
    y = data[target_column]
```

```
return X, y
```

Лістинг А.5 – Поділ даних на навчальну та тестову вибірки

```
def split_dataset(X, y):  
    """  
    Поділ набору даних на навчальну та тестову вибірки.  
    """  
    X_train, X_test, y_train, y_test = train_test_split(  
        X,  
        y,  
        test_size=0.2,  
        random_state=42  
    )  
  
    return X_train, X_test, y_train, y_test
```

Лістинг А.6 – Побудова моделей машинного навчання

```
def build_models():  
    """  
    Створення набору моделей для порівняльного аналізу.  
    """  
    models = {  
        "Linear Regression": Pipeline([  
            ("scaler", StandardScaler()),  
            ("model", LinearRegression())  
        ]),  
  
        "Decision Tree Regressor": DecisionTreeRegressor(  
            max_depth=8,  
            random_state=42  
        ),  
  
        "Random Forest Regressor": RandomForestRegressor(  
            n_estimators=100,  
            random_state=42  
        ),  
  
        "Support Vector Regression": Pipeline([  
            ("scaler", StandardScaler()),  
            ("model", SVR(kernel="rbf", C=10.0, epsilon=0.1))  
        ])  
    }  
  
    return models
```

Лістинг А.7 – Навчання моделі та формування прогнозу

```
def train_and_predict(model, X_train, X_test, y_train):  
    """  
    Навчання моделі та формування прогнозу для тестової вибірки.  
    """  
    model.fit(X_train, y_train)  
    y_pred = model.predict(X_test)  
  
    return y_pred
```

Лістинг А.8 – Оцінювання якості прогнозування

```
def evaluate_model(y_test, y_pred):  
    """  
    Обчислення метрик якості прогнозування.  
    """  
    mae = mean_absolute_error(y_test, y_pred)  
    rmse = np.sqrt(mean_squared_error(y_test, y_pred))  
    r2 = r2_score(y_test, y_pred)  
  
    return {  
        "MAE": mae,  
        "RMSE": rmse,  
        "R2": r2  
    }
```

Лістинг А.9 – Порівняння результатів роботи моделей

```
def compare_models(models, X_train, X_test, y_train, y_test):  
    """  
    Навчання кількох моделей і порівняння їх результатів.  
    """  
    results = []  
  
    for model_name, model in models.items():  
        y_pred = train_and_predict(model, X_train, X_test, y_train)  
        metrics = evaluate_model(y_test, y_pred)  
  
        results.append({  
            "Model": model_name,  
            "MAE": round(metrics["MAE"], 4),  
            "RMSE": round(metrics["RMSE"], 4),  
            "R2": round(metrics["R2"], 4)  
        })  
  
    results_table = pd.DataFrame(results)  
    return results_table
```

Лістинг А.10 – Побудова графіка фактичних і прогнозованих значень

```
def plot_actual_vs_predicted(y_test, y_pred):  
    """  
    Побудова графіка порівняння фактичних і прогнозованих значень.  
    """  
    plt.figure(figsize=(10, 5))  
    plt.plot(y_test.values, label="Фактичні значення Y1")  
    plt.plot(y_pred, label="Прогнозовані значення Y1")  
    plt.xlabel("Номер спостереження")  
    plt.ylabel("Heating load")  
    plt.title("Порівняння фактичних і прогнозованих значень")  
    plt.legend()  
    plt.grid(True)  
    plt.tight_layout()  
    plt.show()
```

Лістинг А.11 – Побудова графіка залишків

```
def plot_residuals(y_test, y_pred):  
    """  
    Побудова графіка залишків моделі.  
    """  
    residuals = y_test.values - y_pred
```

```

plt.figure(figsize=(8, 5))
plt.scatter(y_pred, residuals)
plt.axhline(y=0, linestyle="--")
plt.xlabel("Прогнозовані значення")
plt.ylabel("Залишки")
plt.title("Графік залишків моделі")
plt.grid(True)
plt.tight_layout()
plt.show()

```

Лістинг А.12 – Побудова діаграми важливості ознак

```

def plot_feature_importance(model, feature_names):
    """
    Побудова діаграми важливості ознак для Random Forest Regressor.
    """
    regressor = model

    if hasattr(model, "named_steps"):
        regressor = model.named_steps["model"]

    importances = regressor.feature_importances_

    importance_table = pd.DataFrame({
        "Feature": feature_names,
        "Importance": importances
    }).sort_values(by="Importance", ascending=True)

    plt.figure(figsize=(8, 5))
    plt.barh(
        importance_table["Feature"],
        importance_table["Importance"]
    )
    plt.xlabel("Відносна важливість")
    plt.ylabel("Ознаки")
    plt.title("Важливість ознак у моделі Random Forest")
    plt.grid(True, axis="x")
    plt.tight_layout()
    plt.show()

```

Лістинг А.13 – Основна програма роботи модуля

```

def main():
    file_path = "iot_energy_data.csv"
    target_column = "Y1"

    # 1. Завантаження даних
    data = load_dataset(file_path)

    # 2. Перевірка структури набору даних
    validate_dataset(data, target_column)

    # 3. Формування ознак і цільової змінної
    X, y = prepare_features(data, target_column)

    # 4. Поділ на навчальну та тестову вибірки
    X_train, X_test, y_train, y_test = split_dataset(X, y)

    # 5. Створення моделей
    models = build_models()

    # 6. Порівняння моделей

```

```

results_table = compare_models(
    models,
    X_train,
    X_test,
    y_train,
    y_test
)

print("Порівняння моделей:")
print(results_table)

# 7. Навчання основної моделі
best_model = models["Random Forest Regressor"]
y_pred = train_and_predict(
    best_model,
    X_train,
    X_test,
    y_train
)

# 8. Оцінювання основної моделі
metrics = evaluate_model(y_test, y_pred)

print("Результати Random Forest Regressor:")
print("MAE:", round(metrics["MAE"], 4))
print("RMSE:", round(metrics["RMSE"], 4))
print("R2:", round(metrics["R2"], 4))

# 9. Візуалізація результатів
plot_actual_vs_predicted(y_test, y_pred)
plot_residuals(y_test, y_pred)
plot_feature_importance(best_model, X.columns)

if __name__ == "__main__":
    main()

```

Лістинг А.14 – Приклад структури CSV-файлу

```

# X1,X2,X3,X4,X5,X6,X7,X8,Y1
# 0.98,514.5,294.0,110.25,7.0,2.0,0.0,0.0,15.55
# 0.98,514.5,294.0,110.25,7.0,3.0,0.0,0.0,15.55
# 0.98,514.5,294.0,110.25,7.0,4.0,0.0,0.0,15.55
# 0.90,563.5,318.5,122.50,7.0,2.0,0.0,0.0,20.84

```

www.konferenciaonline.org.ua

**Міжнародна наукова
інтернет-конференція**

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

Випуск 110

ISSN 2522-932X

Google Scholar

 **AKADEMIA NAUK STOSOWANYCH**
WYŻSZA SZKOŁA ZARZĄDZANIA I ADMINISTRACJI
W OPOLU

7-8 травня 2026 р.

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

УДК 001 (063)

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

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

Оргкомітет ГО Наукова спільнота:

Патряк Олександра Тарасівна, кандидат економічних наук, ЗУНУ;

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

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

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

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

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

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

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

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

Рудакевич Оксана Мирославівна, кандидат філософських наук, ЗУНУ;

Русенко Святослав Ярославович, викладач ВСІП ФКЕПТ ЗУНУ.

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

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

Наша адреса: Оргкомітет МНІК "Конференція онлайн"

а/с 797, м. Тернопіль 46005

тел. моб. 068 366 0 525

e-mail: inetkonf@ukr.net

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

ISSN 2522-932X

© ГО "Наукова спільнота" 2026

© Автори статей 2026



Мельник Назар Борисович, Шклярук Назарій Анатолійович ВИЯВЛЕННЯ ЕМОЦІЙ У ТЕКСТОВИХ ПОВІДОМЛЕННЯХ НА ОСНОВІ РЕКУРЕНТНИХ НЕЙРОННИХ МЕРЕЖ.....	57
Падучак Олег Володимирович СУЧАСНІ ТЕХНОЛОГІЇ ІНТЕЛЕКТУАЛЬНОГО АНАЛІЗУ НАУКОВОЇ ЛІТЕРАТУРИ.....	60
Панцир Максим Іванович, Танасюк Юлія Володимирівна ІОТ-СИСТЕМА ЗБОРУ ТА АНАЛІЗУ ДАНИХ НА БАЗІ ESP32 З ВЕБ-ІНТЕРФЕЙСОМ ТА ПІДТРИМКОЮ ВИЯВЛЕННЯ АНОМАЛІЙ.....	63
Семенішин Олександр Мирославович ІНТЕЛЕКТУАЛЬНИЙ МОНІТОРИНГ ПРОДУКТИВНОСТІ ТА ЕНЕРГОЕФЕКТИВНОСТІ МОДЕЛЕЙ ГЛИБОКОГО НАВЧАННЯ НА ВБУДОВАНИХ ОБЧИСЛЮВАЛЬНИХ ПРИСТРОЯХ.....	67
Соколюк Денис Миколайович ОПТИМІЗАЦІЯ МОДЕЛЕЙ КОМП'ЮТЕРНОГО ЗОРУ ДЛЯ РОЗГОРТАННЯ НА ПЕРИФЕРІЙНИХ ПРИСТРОЯХ.....	70
Соляник Станіслав Віталійович ПРОГНОЗУВАННЯ ЕНЕРГОСПОЖИВАННЯ ІОТ-ПРИСТРОЇВ НА ОСНОВІ МЕТОДІВ МАШИННОГО НАВЧАННЯ.....	73
Степанов Михайло Миколайович, Магдич Віталій Сергійович ДОСЛІДЖЕННЯ ЕФЕКТИВНОСТІ АРХІТЕКТУРИ YOLO ДЛЯ ДЕТЕКЦІЇ ОБ'ЄКТІВ З БІПЛА.....	76
Стисло Оксана Василівна МОДЕЛІ ВПРОВАДЖЕННЯ ІНКЛЮЗИВНОГО UX/UI-ДИЗАЙНУ У РОЗРОБКУ ЦИФРОВИХ ПРОДУКТІВ У КОНТЕКСТІ СТАНДАРТІВ WCAG ТА ISO.....	79
Ткачук Олексій-Василь Михайлович ГОЛОСОВИЙ АСИСТЕНТ З ВІРТУАЛЬНИМ 3D АВАТАРОМ НА БАЗІ RASPBERRY PI B4.....	82

Література:

1. Li H., Ota K., Dong M. Learning IoT in Edge: Deep Learning for the Internet of Things with Edge Computing. IEEE Network. 2018. Vol. 32, № 1. С. 96-101.
2. Jocher G. та ін. Ultralytics YOLOv8. 2023. [Електронний ресурс]. – Режим доступу: <https://github.com/ultralytics/ultralytics>
3. Li X. та ін. Generalized Focal Loss: Learning Qualified and Distributed Bounding Boxes for Dense Object Detection. Advances in Neural Information Processing Systems. 2020. Vol. 33. С. 21002-21012.
4. ONNX: Open Neural Network Exchange. [Електронний ресурс]. – Режим доступу: <https://onnx.ai/docs>
5. Hinton G., Vinyals O., Dean J. Distilling the Knowledge in a Neural Network. 2015. [Електронний ресурс]. – Режим доступу: <https://arxiv.org/abs/1503.02531>

*Соляник Станіслав Віталійович, бакалавр,
Західноукраїнський національний університет, м. Тернопіль*

ПРОГНОЗУВАННЯ ЕНЕРГОСПОЖИВАННЯ ІОТ-ПРИСТРОЇВ НА ОСНОВІ МЕТОДІВ МАШИННОГО НАВЧАННЯ

Інтернет-адреса публікації на сайті:
<http://www.konferenciaonline.org.ua/ua/article/id-2615/>

Рациональне використання енергії в IoT-середовищі є важливою передумовою забезпечення стабільної роботи пристроїв, зменшення витрат на обслуговування та підвищення загальної ефективності цифрових систем. У практичних умовах надмірне енергоспоживання призводить до скорочення часу автономної роботи, частішої заміни або підзарядки джерел живлення, а також до зниження надійності функціонування розподілених інформаційно-вимірювальних систем. Це особливо критично для мікроконтролерних модулів, сенсорних вузлів і периферійних edge-пристроїв, де навіть незначне перевищення енергетичних витрат може впливати на тривалість експлуатації та якість виконання функцій [1, 2].

Для задачі прогнозування енергоспоживання IoT-пристроїв архітектура повинна забезпечувати послідовне виконання всіх ключових етапів: завантаження даних, їх перевірку, попередню обробку, формування ознак, навчання моделі, прогнозування та оцінювання результатів [2].

У межах даної роботи доцільно використовувати модульний підхід до побудови архітектури. На рисунку 1 подано структурну схему модуля.



Рисунок 1 – Архітектура програмного модуля прогнозування енергоспоживання IoT-пристроїв

Модуль завантаження даних відповідає за імпорт вхідної інформації з файлів або інших джерел. У межах спрощеної реалізації доцільно передбачити роботу з табличними форматами, наприклад CSV. Основним завданням цього модуля є зчитування набору даних і передавання його до наступного етапу обробки. Крім того, на цьому етапі може виконуватися первинна перевірка структури даних: наявності заголовків, типів полів та цільової змінної.

Модуль попередньої обробки даних виконує очищення й підготовку набору даних до подальшого аналізу. На цьому етапі можуть усуватися пропуски, аномальні значення, дублікати записів, а також виконуватися нормалізація або масштабування числових ознак. Значення цього модуля є особливо важливим, оскільки якість прогнозу безпосередньо залежить від якості вхідної інформації. Якщо вихідні дані містять шум або структурні помилки, точність моделі істотно знижується.

Модуль формування ознак виконує підготовку вхідного простору для моделі машинного навчання. На цьому етапі визначається, які саме параметри будуть використані як вхідні ознаки, а також може виконуватися відбір найбільш інформативних змінних. У задачах прогнозування енергоспоживання доцільно використовувати як безпосередньо виміряні параметри, так і похідні характеристики, наприклад усереднені значення за інтервал, коефіцієнти змінності, часові індекси або співвідношення між окремими величинами [3].

Модуль навчання моделі реалізує побудову математичної моделі на навчальній вибірці. У межах роботи доцільно передбачити можливість використання кількох алгоритмів регресійного прогнозування, зокрема Linear Regression, Decision Tree Regressor, Random Forest Regressor та Support Vector Regression [4]. Основним призначенням цього модуля є навчання вибраної моделі на підготовлених даних з подальшим збереженням параметрів для використання на етапі прогнозування.

Модуль прогнозування використовує навчену модель для обчислення прогнозованого значення енергоспоживання на нових або тестових даних.

Саме цей компонент забезпечує основну прикладну функцію програмного модуля. В архітектурному сенсі важливо, щоб він приймав стандартизований набір ознак і повертав прогноз у єдиному форматі, незалежно від того, який алгоритм використовується всередині. Це спрощує реалізацію інших компонентів системи.

Модуль оцінювання результатів виконує порівняння фактичних і прогнозованих значень та обчислює метрики якості моделі. Для даної роботи доцільно використовувати MAE та RMSE як основні показники похибки [5, 30]. За потреби може також застосовуватися коефіцієнт детермінації R^2 . Цей модуль є важливим для експериментальної перевірки якості моделі.

Модуль візуалізації та взаємодії з користувачем призначений для представлення результатів роботи модуля у зрозумілому вигляді. Це можуть бути таблиці, графіки фактичних і прогнозованих значень, текстові повідомлення про якість моделі, а також короткі висновки щодо отриманих результатів. У спрощеному варіанті взаємодія може реалізовуватися через консольний інтерфейс або простий графічний інтерфейс.

Отже, така архітектура відповідає вимогам предметної області, забезпечує логічний поділ функцій, підтримує використання методів машинного навчання та створює основу для практичної реалізації програмного рішення.

Література:

1. Bykovyy P., Kochan V., Osolinskyi O., Sachenko A., Kochan R. Reducing Power Consumption of Measurement and Control Modules Fed with Autonomous Power Supply. *CEUR Workshop Proceedings*. 2018. P. 544-554.
2. Sachenko A., Osolinskyi O., Kochan V., Sachenko O., Bykovyy P., Zahorodnia D. Concept of Intelligent Measuring System for Analyzing the Energy Consumption of IoT Modules. *Computer Systems and Information Technologies*. 2022. No. 4. P. 101-105.
3. Osolinskyi O., Sachenko A., Kochan V., Kolodiichuk L. Measurement and Optimization Methods of Energy Consumption for Microcontroller Systems Within IoT. In: *2022 12th International Conference on Dependable Systems, Services and Technologies (DESSERT)*. 2022. P. 1-7.
4. Charde P., Lonkar B.B. An Empirical Review of Machine Learning Models for Energy Optimizations in IoT Networks. In: *2023 14th International Conference on Computing Communication and Networking Technologies (ICCCNT)*. 2023. P. 1-7.