

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»
Факультет прикладної математики
Кафедра системного програмування і спеціалізованих
комп'ютерних систем**

«До захисту допущено»
Завідувач кафедри
_____ Віталій РОМАНКЕВИЧ
«___» _____ 2025 р.

**Дипломний проєкт
на здобуття ступеня бакалавра
за освітньо-професійною програмою
«Системне програмування та спеціалізовані комп'ютерні системи»
спеціальності 123 «Комп'ютерна інженерія»
на тему: «Інформаційна система аналізу ефективності використання
енергії в побутових приладах»**

Виконала:
студентка 4 курсу, групи KB-11
Шевчук Ярослава Олегівна _____

Керівник:
ст. викладач каф. СПіСКС,
Радченко Костянтин Олександрович _____

Консультант з нормоконтролю:
доц.каф.СПіСКС, к.т.н.,
Клятченко Ярослав Михайлович _____

Рецензент:
ст. викл. каф. обчислювальної техніки ФІОТ,
Алещенко Олексій Вадимович _____

Засвідчую, що у цьому дипломному
проєкті немає запозичень з праць інших
авторів без відповідних посилань.
Студентка _____

Київ – 2025 року

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»**

ФАКУЛЬТЕТ ПРИКЛАДНОЇ МАТЕМАТИКИ

Кафедра системного програмування і спеціалізованих комп'ютерних систем

Рівень вищої освіти – перший (бакалаврський)

Освітньо-професійною програма

**«Системне програмування та спеціалізовані
комп'ютерні системи»**

Спеціальність 123 «Комп'ютерна інженерія»

ЗАТВЕРДЖУЮ

Завідувач кафедри

_____ Віталій РОМАНКЕВИЧ
(підпис)

“ ___ ” _____ 2024 р.

ЗАВДАННЯ

на дипломний проєкт студента

Шевчук Ярослави Олегівни

1. Тема проєкту «Інформаційна система аналізу ефективності використання енергії в побутових приладах»,

керівник проєкту ст. викладач каф. СПіСКС.

Радченко Костянтин Олександрович,

затверджені наказом по університету від «29» травня 2025 р. № 1808-С

2. Термін подання студентом проєкту _____

3. Вихідні дані до проєкту див. Технічне завдання

4. Зміст пояснювальної записки:

- 1) Аналіз проблеми ефективного енергоспоживання побутовими приладами
- 2) Математичне моделювання, аналіз методів прогнозування та вибір технологій
- 3) Вибір бібліотеки та проектування інформаційної системи
- 4) Реалізація системи та аналіз отриманих результатів

5. Перелік графічного матеріалу:

- 1) Схема бази даних системи
- 2) Алгоритм тренування моделі машинного навчання
- 3) Алгоритм моніторингу мережі побутових пристроїв

6. Консультанти розділів проєкту

| Розділ | Прізвище, ініціали та посада консультанта | Підпис, дата | |
|---------------|--|----------------|------------------|
| | | завдання видав | завдання прийняв |
| Нормоконтроль | Клятченко Я. М., доцент кафедри СПіСКС, к.т.н., доцент | | |

7. Дата видачі завдання «25» грудня 2024 р.

Календарний план

| № з/п | Назва етапів виконання дипломного проєкту | Термін виконання етапів проєкту | Примітка |
|-------|---|---------------------------------|----------|
| 1 | Вивчення літератури за тематикою роботи | 15.04.2025 | |
| 2 | Розроблення та узгодження технічного завдання | 27.04.2025 | |
| 3 | Аналіз існуючих програм для аналізу енергоспоживання | 28.04.2025 | |
| 4 | Розроблення структури інформаційної системи | 05.05.2025 | |
| 5 | Програмна реалізація системи на основі ML.NET | 15.05.2025 | |
| 6 | Розроблення інтерфейсу користувача у WinForms | 20.05.2025 | |
| 7 | Тестування моделі та функціоналу системи | 21.05.2025 | |
| 8 | Підготовка матеріалів першого розділу дипломного проєкту | 25.05.2025 | |
| 9 | Підготовка матеріалів другого розділу дипломного проєкту | 29.05.2025 | |
| 10 | Підготовка матеріалів третього розділу дипломного проєкту | 01.06.2025 | |
| 11 | Підготовка матеріалів четвертого розділу дипломного проєкту | 02.06.2025 | |
| 12 | Оформлення технічної документації та графічної частини | 03.06.2025 | |
| 13 | Попередній огляд матеріалів дипломної роботи на кафедрі | 04.06.2025 | |

Студент

Ярослава ШЕВЧУК

Керівник проєкту

Костянтин РАДЧЕНКО

АНОТАЦІЯ

Дипломний проєкт включає пояснювальну записку (85 с., 40 рис., 8табл., список використаної літератури з 31 найменування, 3 додатки).

Об'єкт розробки – інформаційна система аналізу ефективності використання енергії в побутових приладах.

Розроблена система дозволяє прогнозувати споживання електроенергії на основі вхідних параметрів (тип приладу, температура, сезон, кількість мешканців), аналізувати історичні дані та оцінювати потенційну економію.

У ході розробки:

- проведено аналіз особливостей енергоспоживання та актуальних підходів до його оцінювання;
- обґрунтовано доцільність застосування машинного навчання та обрано метод швидкого дерева;
- побудовано математичну модель прогнозування з урахуванням багатофакторного впливу;
- реалізовано програмну систему на мові C# з використанням ML.NET та архітектурою WinForms;
- проведено експериментальні дослідження, що підтвердили точність моделі ($R^2 = 0,7553$) та ефективність інтерактивного інтерфейсу для аналізу споживання.

Середовище розробки – Visual Studio 2022.

Ключові слова: енергоспоживання, машинне навчання, C#, прогнозування, побутові прилади.

ANNOTATION

The diploma project includes an explanatory note (85 p., 40 fig., 8 tables., list of used literature of 31 titles, 3 appendices).

The object of development is an information system for analyzing the efficiency of energy use in household appliances.

The developed system allows you to predict electricity consumption based on input parameters (device type, temperature, season, number of residents), analyze historical data, and estimate potential savings.

During the development:

- an analysis of the features of energy consumption and current approaches to its assessment was carried out;
- the feasibility of using machine learning was substantiated and the fast tree method was selected;
- a mathematical forecasting model was built taking into account multifactorial influence;
- a software system was implemented in C# using ML.NET and WinForms architecture;
- experimental studies were conducted that confirmed the accuracy of the model ($R^2 = 0.7553$) and the effectiveness of the interactive interface for consumption analysis.

Development environment – Visual Studio 2022.

Keywords: energy consumption, machine learning, C#, forecasting, home appliances.

| Поз. | Формат | ПОЗНАЧЕННЯ | НАЙМЕНУВАННЯ | Кількість аркушів | № прим. | Примітки |
|-------------|-----------------|--------------------|--|--|--|----------|
| | A4 | ІАЛЦ.045480.002 ТЗ | Інформаційна система аналізу ефективності використання енергії в побутових приладах. Технічне завдання | | | |
| | A4 | ІАЛЦ.045480.003 ТП | Інформаційна система аналізу ефективності використання енергії в побутових приладах. Відомість технічного проєкту | | | |
| | A4 | ІАЛЦ.045480.004 ПЗ | Інформаційна система аналізу ефективності використання енергії в побутових приладах. Пояснювальна записка | | | |
| | A4 | ІАЛЦ.045480.005 Д1 | Схема бази даних системи. Структурна схема | | | |
| | | | | ІАЛЦ.045480.001 ОА | | |
| Змін. | Арк. | № докум. | Підпис | Дата | | |
| Розробив | Шевчук Я. О. | | | | Літ. | Аркуш |
| Перевірив | Радченко К.О. | | | | | Аркушів |
| Консульт. | | | | | КПІ ім. Ігоря Сікорського, ФПМ КВ-11 | |
| Н. контроль | Клятченко Я.М. | | | | | |
| Зав. каф. | Романкевич В.О. | | | | | |
| | | | | Інформаційна система аналізу ефективності використання енергії в побутових приладах Опис альбому | | |

ЗМІСТ

| | |
|--|---|
| 1. НАЙМЕНУВАННЯ ТА ГАЛУЗЬ РОЗРОБКИ..... | 2 |
| 2. ПІДСТАВА ДЛЯ РОЗРОБКИ..... | 2 |
| 3. МЕТА І ПРИЗНАЧЕННЯ РОБОТИ..... | 2 |
| 4. ДЖЕРЕЛА РОЗРОБКИ..... | 2 |
| 5. ТЕХНІЧНІ ВИМОГИ..... | 3 |
| 5.1 Вимоги до програмного продукту, що розробляється | 3 |
| 5.2 Вимоги до апаратного забезпечення | 3 |
| 5.3 Вимоги до програмного та апаратного забезпечення користувача | 3 |
| 6. ЕТАПИ РОЗРОБКИ..... | 4 |

| | | | | | | | | | | |
|-------------------------|--------------------|------------------------|---------------------|--------------------|--|--|--|--------------------|---------------------|-----------------------|
| | | | | | <i>ІАЛЦ.045480.002 ТЗ</i> | | | | | |
| <i>Зм</i> | <i>Лист</i> | <i>№ докум.</i> | <i>Підп.</i> | <i>Дата</i> | Інформаційна система аналізу ефективності використання енергії в побутових приладах <i>Технічне завдання</i> | | | <i>Лім.</i> | <i>Аркуш</i> | <i>Аркушів</i> |
| <i>Розроб.</i> | Шевчук Я.О. | | | | | | | | | 4 |
| <i>Перев.</i> | Радченко К.О. | | | | | | | | | |
| <i>Н. контр.</i> | Клятченко Я. М. | | | | | | | | | |
| <i>Зав. Каф.</i> | Романкевич В. О. | | | | | | | | | |
| | | | | | КПІ ім. Ігоря Сікорського, ФПМ, КВ-11 | | | | | |

1 НАЙМЕНУВАННЯ ТА ГАЛУЗЬ РОЗРОБКИ

Назва розробки: «Інформаційна система аналізу ефективності використання енергії в побутових приладах».

Галузь застосування: енергоменеджмент, автоматизовані системи контролю побутового енергоспоживання, розумні будинки, аналітичні сервіси для оцінки енергоефективності.

2. ПІДСТАВА ДЛЯ РОЗРОБКИ

Підставою для розробки є завдання на дипломне проектування на здобуття другого (магістерського) рівня вищої освіти, затверджене кафедрою системного програмування і спеціалізованих комп'ютерних систем Національного технічного університету України «Київський політехнічний інститут імені Ігоря Сікорського».

3. МЕТА І ПРИЗНАЧЕННЯ РОБОТИ

Метою даного проєкту є створення легкої у використанні інформаційної системи, що дозволяє здійснювати прогнозування енергоспоживання побутових приладів з урахуванням зовнішніх та поведінкових факторів, а також виконувати аналітичну оцінку потенційної енергоефективності різних сценаріїв використання.

4. ДЖЕРЕЛА РОЗРОБКИ

Джерелом інформації є технічна та науково-технічна література з енергоменеджменту та машинного навчання, документація бібліотеки ML.NET, офіційна документація платформи Visual Studio, публікації у фахових періодичних виданнях, електронні ресурси й аналітичні огляди в мережі Інтернет.

| | | | | | | |
|-----------|-------------|-----------------|--------------|-------------|---------------------------|------|
| | | | | | ІАЛЦ.045480.002 ТЗ | Арк. |
| <i>Зм</i> | <i>Лист</i> | <i>№ докум.</i> | <i>Підп.</i> | <i>Дата</i> | | 2 |

5. ТЕХНІЧНІ ВИМОГИ

5.1 Вимоги до програмного продукту, що розробляється:

- сумісність із сучасними версіями операційної системи Windows (від Windows 10 і вище);
- реалізація прогнозування енергоспоживання на основі машинного навчання;
- підтримка імпорту вхідних даних з файлів CSV;
- автоматичне навчання моделі та виведення метрик якості (MAE, MSE, RMSE, R²);
- реалізація інтерфейсу на базі Windows Forms для введення параметрів та виведення результатів;
- зручність користування системою без необхідності спеціальної підготовки;
- локалізація інтерфейсу українською мовою.

5.2 Вимоги до апаратного забезпечення:

- центральний процесор не нижче Intel Core i3 10-го покоління або AMD Ryzen 3;
- обсяг оперативної пам'яті не менше 4 ГБ;
- вільне місце на жорсткому диску – не менше 200 МБ;
- наявність дисплея з мінімальною роздільною здатністю 1366×768 пікселів;
- підтримка .NET Runtime (версії не нижче .NET 4.7.2).

5.3 Вимоги до програмного та апаратного забезпечення користувача:

- Операційна система: Windows 10 / 11.

| | | | | | | |
|----|------|----------|-------|------|---------------------------|------|
| | | | | | ІАЛЦ.045480.002 ТЗ | Арк. |
| Зм | Лист | № докум. | Підп. | Дата | | 3 |

6. ЕТАПИ РОЗРОБКИ

| № з/п | Назва етапів виконання дипломного проєкту | Термін виконання етапів |
|-------|---|-------------------------|
| 1 | Вивчення літератури за тематикою роботи | 15.11.2024 |
| 2 | Розроблення та узгодження технічного завдання | 27.11.2024 |
| 3 | Аналіз існуючих програм для аналізу енергоспоживання | 28.12.2024 |
| 4 | Розроблення структури інформаційної системи | 05.02.2025 |
| 5 | Програмна реалізація системи на основі ML.NET | 15.03.2025 |
| 6 | Розроблення інтерфейсу користувача у WinForms | 20.04.2025 |
| 7 | Тестування моделі та функціоналу системи | 21.05.2025 |
| 8 | Підготовка матеріалів першого розділу дипломного проєкту | 25.05.2025 |
| 9 | Підготовка матеріалів другого розділу дипломного проєкту | 29.05.2025 |
| 10 | Підготовка матеріалів третього розділу дипломного проєкту | 01.06.2025 |
| 11 | Підготовка матеріалів четвертого розділу дипломного проєкту | 02.06.2025 |
| 12 | Оформлення технічної документації та графічної частини | 03.06.2025 |
| 13 | Попередній огляд матеріалів дипломної роботи на кафедрі | 04.06.2025 |

| Поз. | Формат | ПОЗНАЧЕННЯ | НАЙМЕНУВАННЯ | Кількість | № | Примітки |
|------|--------|--------------------|--|-----------|---|----------|
| | A4 | ІАЛЦ.045480.004 ПЗ | Інформаційна система аналізу ефективності використання енергії в побутових приладах. | | | |
| | | | Пояснювальна записка | | | |
| | A4 | ІАЛЦ.045480.005 Д1 | Схема бази даних системи. | | | |
| | | | Структурна схема | | | |
| | A4 | ІАЛЦ.045480.006 Д2 | Схема побудови моделі ML та її використання. | 1 | | |
| | | | Структурна схема | | | |
| | A4 | ІАЛЦ.045480.007 Д3 | Алгоритм тренування машинного навчання. | | | |
| | | | Блок-схема алгоритму | | | |
| | A4 | ІАЛЦ.045480.008 Д4 | Алгоритм моніторингу побутових пристроїв. | | | |
| | | | Блок-схема алгоритму | | | |
| | | Диск CD-ROM | Текст пояснювальної | | | |
| | | | з | | | |
| | | | Графічний матеріал | | | |

ІАЛЦ.045480.003 ТП

| Змін. | Арк. | № докум. | Підпис | Дата | | | | |
|-------------|------|-----------------|--------|------|--|--|-------|---------|
| Розробив | | Шевчук Я.О. | | | Інформаційна система аналізу ефективності використання енергії в побутових приладах Відомість технічного проєкту | Літ. | Аркуш | Аркушів |
| Перевірив | | Радченко К.О. | | | | | | |
| Консульт. | | | | | | | | |
| Н. контроль | | Клятченко Я.М. | | | | | | |
| Зав. каф. | | Романкевич В.О. | | | | | | |
| | | | | | | КПІ ім. Ігоря Сікорського, ФПМ КВ-11 | | |

Пояснювальна записка
до дипломного проєкту
на тему: «Інформаційна система аналізу ефективності
використання енергії в побутових приладах»

Київ – 2025

ЗМІСТ

| | |
|--|----|
| ПЕРЕЛІК СКОРОЧЕНЬ, УМОВНИХ ПОЗНАЧЕНЬ, ТЕРМІНІВ | 3 |
| ВСТУП | 4 |
| 1. АНАЛІЗ ПРОБЛЕМИ ЕФЕКТИВНОГО ЕНЕРГОСПОЖИВАННЯ ПОБУТОВИМИ ПРИЛАДАМИ..... | 5 |
| 1.1 Особливості та актуальність моніторингу енергоспоживання | 5 |
| 1.2 Методи оцінювання ефективності використання енергії | 8 |
| 1.3 Застосування машинного навчання для аналізу енергоспоживання | 11 |
| 1.4 Порівняльний аналіз розробленої системи із аналогами..... | 17 |
| 1.5 Постановка задачі | 26 |
| Висновки до розділу | 27 |
| 2 МАТЕМАТИЧНЕ МОДЕЛЮВАННЯ, АНАЛІЗ МЕТОДІВ ПРОГНОЗУВАННЯ ТА ВИБІР ТЕХНОЛОГІЙ | 29 |
| 2.1 Огляд та порівняння моделей машинного навчання для прогнозування | 29 |
| 2.2 Побудова математичної моделі ефективності | 34 |
| 2.3 Вибір технологій для розробки інформаційної системи..... | 40 |
| Висновки до розділу | 45 |
| 3 ВИБІР БІБЛІОТЕКИ ТА ПРОЕКТУВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ | 46 |
| 3.1 Аналіз та вибір бібліотеки машинного навчання | 46 |
| 3.2 Розробка алгоритмів прогнозування енергоспоживання..... | 49 |
| 3.3 Побудова архітектури інформаційної системи | 54 |
| Висновки до розділу | 60 |
| 4 ВИБІР БІБЛІОТЕКИ ТА ПРОЕКТУВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ | 61 |

| | | | | | | | | |
|-------------|------|-----------------|--------|------|--|------------------------|-------|---------|
| | | | | | ІАЛЦ.045480.004 ПЗ | | | |
| Змін. | Арк. | № докум. | Підпис | Дата | Інформаційна система аналізу ефективності використання енергії в побутових приладах Пояснювальня записка | Літ. | Аркуш | Аркушів |
| Розробив | | Шевчук Я.О. | | | | | 1 | 64 |
| Перевірив | | Радченко К.О. | | | | КПІ | | |
| Консульт. | | | | | | Ім. Ігоря Сікорського, | | |
| Н. контроль | | Клятченко Я. М. | | | | ФПМ КВ-11 | | |
| Зав. каф. | | Романкевич В.О. | | | | | | |

| | |
|---|----|
| 4.1 Реалізація моделі машинного навчання у програмному середовищі | 61 |
| 4.2 Формування та аналіз набору даних для навчання моделі | 73 |
| 4.3 Проведення експериментів та аналіз результатів прогнозування | 81 |
| Висновки до розділу | 85 |
| ВИСНОВКИ..... | 86 |
| СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ..... | 88 |
| ДОДАТКИ..... | 92 |

| | | | | |
|-------|------|----------|--------|------|
| | | | | |
| Змін. | Арк. | № докум. | Підпис | Дата |

ІАЛЦ.045480.004 ПЗ

ПЕРЕЛІК СКОРОЧЕНЬ, УМОВНИХ ПОЗНАЧЕНЬ, ТЕРМІНІВ

CRM (англ. Customer Relationship Management) – система управління взаємовідносинами з клієнтами.

CSV (англ. Comma-Separated Values) – формат текстових файлів для зберігання табличних даних, де значення розділяються комами.

EEI (англ. Energy Efficiency Index) – індекс енергоефективності; числова характеристика енергетичної доцільності пристрою.

IoT (англ. Internet of Things) – інтернет речей; концепція взаємодії фізичних пристроїв через мережу для збору та обміну даними.

MAE (англ. Mean Absolute Error) – середня абсолютна похибка; метрика оцінки точності регресійної моделі.

ML (англ. Machine Learning, МН) – машинне навчання; галузь штучного інтелекту, що займається розробкою алгоритмів, здатних навчатися на основі даних.

MSE (англ. Mean Squared Error) – середня квадратична похибка; метрика оцінки якості моделі, чутлива до великих відхилень.

ККД (коефіцієнт корисної дії) – відношення корисної енергії до загальної спожитої, виражене у відсотках.

МОВДР (метод опорних векторів для регресії) – алгоритм, що будує регресійну модель, мінімізуючи похибки прогнозування в межах допустимого порогу.

НБК (наївний байєсівський класифікатор) – статистичний метод класифікації, що базується на формулі Байєса з припущенням незалежності ознак.

ШД (швидке дерево) – метод машинного навчання, що реалізує градієнтний бустинг на основі ансамблю дерев рішень.

| | | | | | | |
|-------|------|----------|--------|------|--------------------|------|
| | | | | | ІАЛЦ.045480.004 ПЗ | Арк. |
| Змін. | Арк. | № докум. | Підпис | Дата | | 3 |

ВСТУП

У сучасних умовах енергетичної нестабільності, зростання тарифів на електроенергію та необхідності забезпечення сталого розвитку суспільства проблема раціонального використання енергоресурсів у побутовому секторі набуває особливої ваги. Побутове енергоспоживання становить суттєву частку загального навантаження на енергетичну інфраструктуру, при цьому значна частина енергії витрачається неефективно через недосконалу експлуатацію побутових приладів, відсутність інформованості користувачів або слабку регуляцію з боку державних інституцій [1].

Особливої гостроти ця проблема набула в Україні, де через воєнні дії, часткову руйнацію об'єктів критичної інфраструктури та обмеженість енергетичних ресурсів виникає потреба у підвищенні енергоефективності на всіх рівнях – від промислового виробництва до побуту. У цих умовах кожен споживач має потенціал відігравати важливу роль у стабілізації навантаження на енергосистему шляхом усвідомленого контролю за споживанням.

Світова практика доводить, що запровадження цифрових технологій аналізу даних є дієвим інструментом підвищення енергоефективності. Проте більшість побутових користувачів в Україні не мають доступу до інструментів, які дозволяють здійснювати персоналізовану оцінку ефективності роботи приладів на основі специфічних умов їх використання: сезону, температурного режиму, кількості мешканців тощо.

Відтак, виникає потреба у створенні спеціалізованих програмних рішень, які здатні забезпечити глибший аналіз показників енергоспоживання, виявляти нераціональне використання ресурсів та формувати прогнози для подальшої оптимізації. Залучення методів машинного навчання у таких системах дозволяє не лише підвищити точність оцінок, але й вивести на новий рівень процес прийняття рішень щодо керування побутовими енерговитратами.

| | | | | |
|-------|------|----------|--------|------|
| | | | | |
| Змін. | Арк. | № докум. | Підпис | Дата |

ІАЛЦ.045480.004 ПЗ

Арк.

4

АНАЛІЗ ПРОБЛЕМИ ЕФЕКТИВНОГО ЕНЕРГОСПОЖИВАННЯ ПОБУТОВИМИ ПРИЛАДАМИ

1.1 Особливості та актуальність моніторингу енергоспоживання

У сучасному світі питання енергоспоживання займає провідне місце серед глобальних викликів людства. Безперервне зростання попиту на електроенергію, зумовлене як урбанізацією, так і зростанням кількості побутової електротехніки, створює постійне навантаження на енергетичну систему держав. Особливо гостро ця проблема постає у країнах з високою енергоемністю економіки та обмеженим доступом до стабільних джерел енергоресурсів, серед яких нині перебуває й Україна.

Моніторинг енергоспоживання – це процес безперервного спостереження, збору, аналізу та інтерпретації даних щодо використання енергії з метою оцінювання її ефективності, виявлення відхилень та прийняття відповідних керуючих рішень [31]. Його роль не обмежується лише контролем. Навпаки, йдеться про глибоке аналітичне дослідження закономірностей використання енергії, виявлення неефективних сценаріїв споживання та формування прогнозів для оптимізації витрат.

Особливістю сучасного моніторингу енергоспоживання є необхідність його інтеграції з цифровими технологіями, зокрема засобами автоматизації, телеметрії, хмарних обчислень та штучного інтелекту. Традиційні методи обліку, що ґрунтуються лише на фактичних даних, не здатні надати своєчасну інформацію для прийняття рішень, особливо у динамічних середовищах, таких як побутові господарства з варіативними режимами споживання.

Загальносвітовою тенденцією є перехід до смарт-моніторингу – інтелектуальних систем, які здатні не тільки фіксувати споживання, а й аналізувати поведінкові патерни користувачів, враховувати сезонність, типи побутових приладів, кількість мешканців, кліматичні умови тощо. Саме такі фактори найбільше впливають на ефективність споживання енергії в домашніх умовах. Без їх врахування будь-які спроби підвищити енергоефективність

| | | | | |
|-------|------|----------|--------|------|
| | | | | |
| Змін. | Арк. | № докум. | Підпис | Дата |

ІАЛЦ.045480.004 ПЗ

Арк.

5

залишаються поверховими. З метою узагальнення основних чинників, що зумовлюють актуальність моніторингу енергоспоживання у сучасних умовах, на рис. 1.1 подано візуалізацію ключових напрямів, у межах яких ця проблема набуває особливого значення.



Рисунок 1.1 – Актуальність моніторингу енергоспоживання

Моніторинг споживання електроенергії є багатовимірним явищем, що охоплює економічні, екологічні, нормативні та технологічні аспекти [3]. Із зростанням цін на енергоносії домогосподарства все частіше зіштовхуються з необхідністю оптимізації витрат, що, в свою чергу, стимулює попит на засоби контролю енергоспоживання в реальному часі. Це дозволяє не лише своєчасно планувати бюджет, а й виявляти найбільш енерговитратні прилади, коригуючи поведінку користувачів.

Не менш важливим є екологічний вимір проблеми, який стосується як скорочення викидів CO₂ через зменшення загального енергоспоживання, так і усвідомленої участі у глобальних кліматичних ініціативах, зокрема Європейській зеленій угоді. Енергетична відповідальність, сформована на рівні кожного домогосподарства, сприяє досягненню загальнодержавних цілей у сфері сталого розвитку.

| | | | | |
|-------|------|----------|--------|------|
| | | | | |
| Змін. | Арк. | № докум. | Підпис | Дата |

Важливу роль відіграє і державне регулювання, яке проявляється через запровадження стандартів енергоефективності, таких як ENERGY STAR чи відповідні директиви ЄС. Завдяки ним забезпечується контроль за якістю побутових приладів і стимулюється модернізація обладнання. Контроль відповідності цим нормам стає додатковим мотиваційним чинником для громадян щодо впровадження засобів моніторингу.

Сучасний технологічний розвиток, зокрема стрімке поширення рішень у галузі IoT, автоматизації, використання систем на кшталт Smart Grid або Home Assistant, а також прогрес у сфері штучного інтелекту, відкриває нові горизонти для реалізації гнучких, адаптивних і самонавчальних систем моніторингу [17]. Використання алгоритмів прогнозування дозволяє не лише аналізувати вже спожиту енергію, а й моделювати майбутні сценарії, ґрунтуючись на поведінці користувача та зовнішніх факторах. Щоб повніше зрозуміти функціональні особливості сучасних систем енергомоніторингу в побуті, доцільно розглянути їхню структурну характеристику, подану на рис. 1.2.



Рисунок 1.2 – Особливостей моніторингу енергоспоживання приладів

На практиці моніторинг енергоспоживання приладів охоплює цілу низку технологічних, користувацьких і аналітичних функцій, які взаємопов'язані та

| | | | | |
|-------|------|----------|--------|------|
| Змін. | Арк. | № докум. | Підпис | Дата |
| | | | | |

спрямовані на підвищення енергоефективності. Насамперед він базується на вимірюванні параметрів електроспоживання в реальному часі за допомогою сучасних сенсорів, таких як PZEM-004T, Shelly EM чи TP-Link Tapo. Зібрані дані візуалізуються у вигляді графіків, таблиць, сповіщень і можуть негайно сигналізувати про перевищення граничних значень, виявлення несправностей або енергетичних витоків.

Інтерфейси сучасних систем, зокрема мобільні застосунки та вебпанелі, забезпечують користувачу віддалений доступ до інформації та можливість оперативного реагування на зміни. Це дозволяє не лише переглядати історію споживання, а й аналізувати індивідуальні звички, формуючи на їх основі персоналізовані рекомендації. Наприклад, система може запропонувати оптимальний графік вмикання чи вимикання приладу з урахуванням типового профілю споживання, пори року та часу доби.

Ключовою особливістю таких рішень є глибока інтеграція з системами домашньої автоматизації. Завдяки підтримці протоколів і платформ, таких як Google Home чи Apple HomeKit, можливим стає автоматичне вимкнення приладів у разі виявлення перевантаження, неефективного споживання чи тривалої бездіяльності. Така взаємодія з середовищем дозволяє не лише покращити показники енергоощадності, а й створити зручне і безпечне цифрове середовище для користувача.

Важливою є також функція оцінювання ефективності роботи окремих приладів, що дає змогу порівнювати їхній коефіцієнт корисної дії, визначати прилади з аномальним або надмірним енергоспоживанням, а отже – приймати обґрунтовані рішення щодо їх заміни чи оптимізації використання. Таким чином, сучасні інформаційні системи енергомоніторингу набувають ознак не лише засобу вимірювання, а й інтелектуального помічника, що здатен формувати прогностичні сценарії, реагувати на ризики та активно керувати споживанням у побутовому середовищі.

1.2 Методи оцінювання ефективності використання енергії

| | | | | |
|-------|------|----------|--------|------|
| | | | | |
| Змін. | Арк. | № докум. | Підпис | Дата |

ІАЛЦ.045480.004 ПЗ

Арк.

8

Раціональне використання енергетичних ресурсів у побуті є визначальним чинником сталого розвитку та економічної безпеки держави. З огляду на постійне зростання вартості електроенергії, потребу в енергозбереженні та екологічну відповідальність, питання оцінювання ефективності споживання енергії виходить за межі суто технічного аналізу і перетворюється на комплексну міждисциплінарну задачу [10]. Особливого значення вона набуває в умовах використання автоматизованих інформаційних систем, що дозволяють здійснювати аналіз енергоспоживання приладів у реальному часі з урахуванням численних зовнішніх факторів, таких як температура, сезонність, побутові звички тощо.

Сучасні підходи до оцінювання енергоефективності охоплюють як традиційні інженерні методи, так і інтелектуальні технології, зокрема алгоритми машинного навчання. Кожен з методів має свою специфіку застосування, рівень точності, потребу в обчислювальних ресурсах і ступінь інтеграції з користувацькими інтерфейсами. Для систематизації підходів їх доцільно подати у вигляді аналітичної табл. 1.1.

Таблиця 1.1 –Методи оцінювання ефективності використання енергії

| Назва методу | Сутність методу | Сфера застосування / Особливості |
|-----------------------------------|---|--|
| 1 | 2 | 3 |
| Порівняльний аналіз | Співставлення фактичного енергоспоживання з середньостатистичними або нормативними значеннями | Простота реалізації, використовується для виявлення приладів з аномальним споживанням |
| Розрахунок ККД або питомих витрат | Визначення відношення корисного ефекту до спожитої енергії | Актуально для нагрівальних, охолоджувальних, обігрівальних приладів; вимагає дані про кінцевий результат |

Продовження таблиці 1.1.

| 1 | 2 | 3 |
|--------------------------------------|--|--|
| Прогнозне моделювання | Застосування моделей машинного навчання для передбачення майбутніх рівнів споживання на основі історичних та поведінкових даних | Висока точність, динамічне оновлення моделі, потребує якісного навчального датасету |
| Індекс енергоефективності (EEI) | Комплексний показник, який об'єднує кілька параметрів: тривалість роботи, навантаження, середнє споживання | Використовується для класифікації приладів (A++, A+, тощо); зручно для маркування та простого порівняння |
| Виявлення аномалій на основі історії | Аналіз патернів споживання для виявлення відхилень від звичних режимів | Актуально для систем із довготривалим моніторингом; дозволяє виявляти потенційні несправності або витoki енергії |
| Балансувальний аналіз (Smart Grid) | Врахування взаємодії домогосподарства з енергосистемою для визначення пікових навантажень і пропозиції оптимального часу використання приладів | Потребує інтеграції з мережею Smart Grid, застосовується для розумного планування навантаження |

Наведена таблиця свідчить про розмаїття методів, які дозволяють оцінити ефективність використання електроенергії в побутовому середовищі. Одні з них ґрунтуються на статичних даних і нормативних порівняннях, інші – на поведінковому аналізі, машинному навчанні або телеметричних даних з інтелектуальних систем. Зокрема, моделі на базі машинного навчання мають здатність до адаптації та уточнення прогнозів у реальному часі, що робить їх особливо цінними для складних побутових сценаріїв з численними змінними.

Загалом, вибір конкретного методу визначається як технічними можливостями інформаційної системи, так і цільовим призначенням аналізу – чи йдеться про контроль, попередження перевитрат, діагностику несправностей або

персоналізовану оптимізацію. Комплексне використання кількох методів забезпечує найвищу точність оцінки й формує базу для прийняття обґрунтованих рішень у межах системи енергоефективного управління.

1.3 Застосування машинного навчання для аналізу енергоспоживання

У зв'язку з постійним ускладненням структури енергоспоживання та зростанням обсягів даних, пов'язаних із його вимірюванням, традиційні методи статистичного аналізу виявляються недостатньо гнучкими для виявлення прихованих закономірностей і формування точних прогнозів. У цьому контексті методи машинного навчання (ML – Machine Learning) посідають центральне місце в сучасних інформаційно-аналітичних системах, призначених для оцінювання, оптимізації та прогнозування енергоспоживання, зокрема в побутовому секторі.

Машинне навчання – це галузь штучного інтелекту, що забезпечує здатність комп'ютерних систем автоматично виявляти закономірності в даних і вдосконалювати свої прогнози без явного програмування на основі апріорних правил. На відміну від класичних підходів, ML-моделі не потребують суворої формалізації взаємозв'язків між змінними [8]. Замість цього, вони формують адаптивну модель на основі історичних прикладів, враховуючи численні фактори, що впливають на енергоспоживання: тип пристрою, температуру навколишнього середовища, сезон року, режим використання, кількість мешканців тощо. Врахування таких змінних дає змогу моделі адаптуватися до контексту споживання і надавати точні результати навіть за умов нестабільного або неповного середовища даних. На рис. 1.3 наведено схему, яка узагальнює ключові вектори реалізації інтелектуального аналізу даних у даній сфері.

| | | | | |
|-------|------|----------|--------|------|
| | | | | |
| Змін. | Арк. | № докум. | Підпис | Дата |



Рисунок 1.3 – Напрямки застосування машинного навчання

Машинне навчання в межах моніторингу енергоспоживання забезпечує не лише прогнозування величини споживання, а й реалізацію низки інтелектуальних функцій, орієнтованих на зменшення втрат, підвищення комфорту користувача та автоматизацію прийняття рішень. Насамперед система здійснює вимірювання енергії в реальному часі з використанням спеціалізованих сенсорів, таких як PZEM-004T, Shelly EM або TP-Link Tapo. Зібрані дані оперативно візуалізуються у вигляді графіків, діаграм і текстових сповіщень, що дає змогу користувачу відразу оцінити поточний стан споживання.

У разі виявлення підозрілих змін, таких як різке перевищення потужності або незвична активність приладу, система здатна сформувати сповіщення про потенційне перевантаження чи несправність. Такі сигнали можуть бути надіслані через мобільний застосунок або вебінтерфейс, що забезпечує повноцінний віддалений доступ до системи й оперативне реагування з боку користувача. У межах цих інтерфейсів реалізовано аналітику, яка базується на вивченні звичок користувача: періодичності ввімкнення пристроїв, тривалості їхньої роботи, часових шаблонів тощо.

| | | | | |
|-------|------|----------|--------|------|
| | | | | |
| Змін. | Арк. | № докум. | Підпис | Дата |

На основі зібраної історії споживання алгоритми машинного навчання можуть генерувати персоналізовані рекомендації – наприклад, щодо зміни графіка роботи певного приладу або доцільності його заміни. Це дозволяє не лише знизити рівень споживання, а й сформувати модель поведінки користувача, орієнтовану на сталий розвиток. Особливу цінність має також виявлення енергетичних витоків, які можуть виникати непомітно, але призводити до суттєвих втрат у довготривалій перспективі.

Іншою важливою складовою є автоматизація реакцій системи. Наприклад, у разі фіксації перевищення встановленого ліміту або зафіксованої неактивності в приміщенні, система може самостійно вимкнути відповідний прилад. Ця функція реалізується через інтеграцію з екосистемами розумного дому, такими як Google Home або Apple HomeKit. Таким чином, інтелектуальна система здатна не лише аналізувати споживання, а й активно впливати на нього, забезпечуючи оптимальну конфігурацію без участі людини.

Застосування машинного навчання у побутовому моніторингу енергії дає змогу сформувати замкнений цикл: збір → аналіз → прогноз → рекомендація → дія. Усе це виводить систему на якісно новий рівень керованості та енергетичної відповідальності, створюючи основу для побудови інтелектуального середовища споживання, яке самостійно адаптується до змінних умов і оптимізує витрати. Водночас центральною умовою побудови таких моделей є якісне, структуроване та репрезентативне джерело вхідних даних. Щоб реалізувати повноцінний аналітичний і прогнозний потенціал системи, модель має бути навчена на інформації, яка комплексно відображає фактори впливу на енергоспоживання.

На рис. 1.4 подано класифікацію основних джерел даних, які використовуються для навчання моделей машинного навчання у системах побутового енергомоніторингу.

| | | | | |
|-------|------|----------|--------|------|
| | | | | |
| Змін. | Арк. | № докум. | Підпис | Дата |



Рисунок 1.4 – Джерела даних для навчання моделей

Насамперед джерела даних умовно поділяються на внутрішні – отримані безпосередньо із сенсорів та пристроїв користувача, – і зовнішні, що надходять із публічних або спеціалізованих сервісів. Внутрішні дані включають вимірювання, зафіксовані розумними лічильниками, ІоТ-пристроями, сенсорами температури, вологості, руху та іншими модулями навколишнього середовища. Важливою є інформація про активність приладів у режимі реального часу, яка дозволяє будувати кореляційні зв'язки між умовами середовища та поведінкою споживача, зокрема за частотою вмикання або тривалістю роботи пристроїв.

Окремий блок даних становлять записи безпосереднього енергоспоживання. Ці дані, отримані з розумних лічильників, становлять основу для побудови регресійних моделей та індикаторів енергоефективності. Вони є найбільш надійним джерелом для оцінювання реального стану споживання і виявлення трендів у побутових умовах.

| | | | | |
|-------|------|----------|--------|------|
| | | | | |
| Змін. | Арк. | № докум. | Підпис | Дата |

Не менш важливою складовою є зовнішні джерела. До них належать погодні дані, наприклад, із сервісів OpenWeatherMap, які дозволяють моделі враховувати вплив температури, вологості, атмосферного тиску на режими роботи систем опалення чи кондиціонування. Інтеграція таких даних дає змогу значно підвищити точність прогнозу та адаптивність моделі до сезонних коливань [12]. Також застосовуються енергетичні тарифи – у випадку динамічного ціноутворення це дозволяє оптимізувати використання приладів у періоди найнижчої вартості електроенергії. Наприклад, рекомендація запуску пральної машини вночі або автоматичне вимкнення обігрівача в години найвищих цін ґрунтується саме на таких тарифних моделях. Однак, незважаючи на численні переваги машинного навчання в аналізі енергоспоживання, впровадження таких підходів супроводжується низкою викликів, які можуть впливати як на точність прогнозів, так і на практичну придатність систем у побутових умовах. Для систематизації основних обмежень і труднощів, характерних для застосування ML-моделей у цій сфері, на рис. 1.5 наведено узагальнене представлення ключових проблем і шляхів їх подолання.



Рисунок 1.5 – Виклики та обмеження машинного навчання

| | | | | |
|-------|------|----------|--------|------|
| Змін. | Арк. | № докум. | Підпис | Дата |
| | | | | |

Одним з найпоширеніших викликів є якість вхідних даних. Вона може страждати через наявність шумів, пропусків або нестабільних вимірювань, особливо в умовах роботи з недорогими сенсорами та домашніми IoT-пристроями. У таких випадках виникає потреба у попередній обробці даних, зокрема в нормалізації, фільтрації, інтерполяції та заповненні відсутніх значень. Ще однією проблемою є недостатній обсяг даних, що особливо характерно для нових користувачів або нестабільних середовищ. У таких випадках ефективним підходом виявляється трансферне навчання, коли модель попередньо тренується на великій загальній вибірці, а потім адаптується до конкретного сценарію.

Значну увагу також привертає питання інтерпретованості моделей. У практиці побутового моніторингу важливо не лише отримати результат, але й пояснити його користувачу. Так звані «чорні скриньки» (наприклад, глибокі нейронні мережі) мають обмежену прозорість і можуть викликати недовіру через складність пояснення внутрішньої логіки. Альтернативою є моделі, які краще піддаються інтерпретації, як-от дерева рішень або інструменти пояснення на кшталт SHAP, що дають змогу візуалізувати вплив кожної змінної на результат прогнозу.

Технічні обмеження також стосуються енергоефективності самих ML-алгоритмів. Іронічно, але моделі, що мають допомогти зменшити споживання енергії, самі можуть потребувати значних обчислювальних ресурсів. Це особливо критично при впровадженні в малопотужних пристроях або автономних системах. Витрати на обчислення можуть стати суттєвим бар'єром при масштабуванні рішень у середовищі домашніх мереж. Для цього розробляються оптимізовані моделі, такі як TinyML – спеціальні варіанти машинного навчання, призначені для мікроконтролерів і низькопотужних IoT-платформ, що дозволяє реалізувати локальну обробку без потреби у постійному підключенні до хмари.

Таким чином, ефективне використання машинного навчання в системах моніторингу енергоспоживання потребує глибокого розуміння як переваг, так і обмежень цих технологій. Застосування таких моделей повинне

| | | | | |
|-------|------|----------|--------|------|
| | | | | |
| Змін. | Арк. | № докум. | Підпис | Дата |

ІАЛЦ.045480.004 ПЗ

Арк.
16

супроводжуватись продуманою стратегією обробки даних, вибором інтерпретованих підходів, врахуванням ресурсних обмежень та адаптацією до умов конкретного користувача. Лише за таких умов інтелектуальні системи зможуть стати не лише технологічною новацією, а реальною користю для побутового енергоменеджменту.

1.4 Порівняльний аналіз розробленої системи із аналогами

У сучасних умовах підвищеного інтересу до енергоощадних технологій спостерігається активний розвиток програмного забезпечення, орієнтованого на моніторинг, аналіз та оптимізацію споживання електроенергії в побутовому секторі. Наявність численних комерційних та відкритих рішень у цій сфері свідчить про високу затребуваність відповідних інструментів серед кінцевих користувачів, а також про прагнення інтегрувати принципи сталого споживання в щоденну практику експлуатації електроприладів.



Рисунок 1.6 – Приклад інтерфейсу системи «The Sense Electricity» [20]

| | | | | |
|-------|------|----------|--------|------|
| | | | | |
| Змін. | Арк. | № докум. | Підпис | Дата |

The Sense Electricity – це програмно-апаратне рішення, призначене для детального моніторингу споживання електроенергії у побутових умовах [16]. Основне призначення системи полягає в наданні користувачу повної інформації про те, які саме прилади використовують електроенергію, коли саме вони працюють і скільки споживають у реальному часі. Завдяки цьому користувач може виявити енерговитратні пристрої, оптимізувати побутові звички та зменшити загальні витрати на електроенергію. На рис. 1.6 наведено приклад інтерфейсу користувача системи, який наочно демонструє структуру даних, що відображаються в реальному часі, та функції візуального аналізу споживання.

Архітектура системи складається з апаратного модуля, що встановлюється безпосередньо у щитову панель будинку і підключається до головної електричної лінії. Цей пристрій виконує функцію сенсора, зчитуючи електричні сигнали з високою частотою, а також передає дані через Wi-Fi на хмарну платформу. У хмарі здійснюється обробка даних із використанням алгоритмів машинного навчання, які дозволяють ідентифікувати окремі прилади за характером споживання. Користувач отримує доступ до аналітики через мобільний застосунок або вебінтерфейс, де в інтерактивному режимі відображається поточна і історична інформація, формується оцінка витрат, надсилаються сповіщення про виявлену активність або аномалії. Завдяки локальній обробці базових сигналів та централізованому інтелектуальному аналізу в хмарі, система забезпечує баланс між точністю, масштабованістю та простотою експлуатації.

Переваги застосунку *The Sense Electricity*:

- висока точність моніторингу завдяки обробці сигналів із частотою до мікросекундного рівня, що дозволяє розпізнавати споживання окремих приладів;
- зручний інтерфейс – мобільний застосунок надає користувачу наочну візуалізацію споживання в реальному часі та доступ до історичних даних;

| | | | | |
|-------|------|----------|--------|------|
| | | | | |
| Змін. | Арк. | № докум. | Підпис | Дата |

ІАЛЦ.045480.004 ПЗ

Арк.

18

- інтелектуальне розпізнавання приладів – система використовує машинне навчання для автоматичного іменування пристроїв за унікальними шаблонами навантаження;
- підтримка сповіщень та аналіз аномалій – користувач отримує повідомлення про незвичну активність, зростання споживання або нічне ввімкнення приладів.

Недоліки застосунку The Sense Electricity:

- висока вартість обладнання порівняно з простішими рішеннями на основі смарт-розеток або менш точних лічильників;
- орієнтація переважно на ринок США – складність встановлення в електромережах з іншими стандартами (наприклад, в Європі);
- не всі прилади ідентифікуються автоматично – деякі пристрої мають схожі сигнатури, і модель не завжди їх правильно класифікує;
- залежність від хмарної інфраструктури – обробка даних здійснюється на серверах компанії, що викликає питання щодо приватності та потребує стабільного інтернет-з'єднання.

Отже, Sense Electricity – це високотехнологічне рішення для домашнього енергомоніторингу, яке поєднує точність апаратних сенсорів із можливостями штучного інтелекту для глибокого аналізу споживання. Застосунок особливо корисний для користувачів, які прагнуть не лише контролювати витрати, а й розуміти, як саме їх побутова техніка взаємодіє з електромережею. Попри високу вартість і окремі технічні обмеження, система демонструє високу ефективність та інформативність, що робить її конкурентоспроможною на ринку розумного енергоменеджменту.

Home Assistant – це відкрите програмне середовище для автоматизації розумного дому, яке дозволяє об'єднати різноманітні пристрої, сервіси й датчики в єдину керовану екосистему (рис. 1.7). Основне призначення Home Assistant полягає в забезпеченні централізованого управління всіма аспектами домашньої автоматизації – від освітлення та безпеки до енергоспоживання й клімат-контролю [7]. У сфері енергоменеджменту система дозволяє користувачу

відстежувати споживання електроенергії окремими приладами, аналізувати динаміку витрат і формувати інтелектуальні сценарії для підвищення енергоефективності.

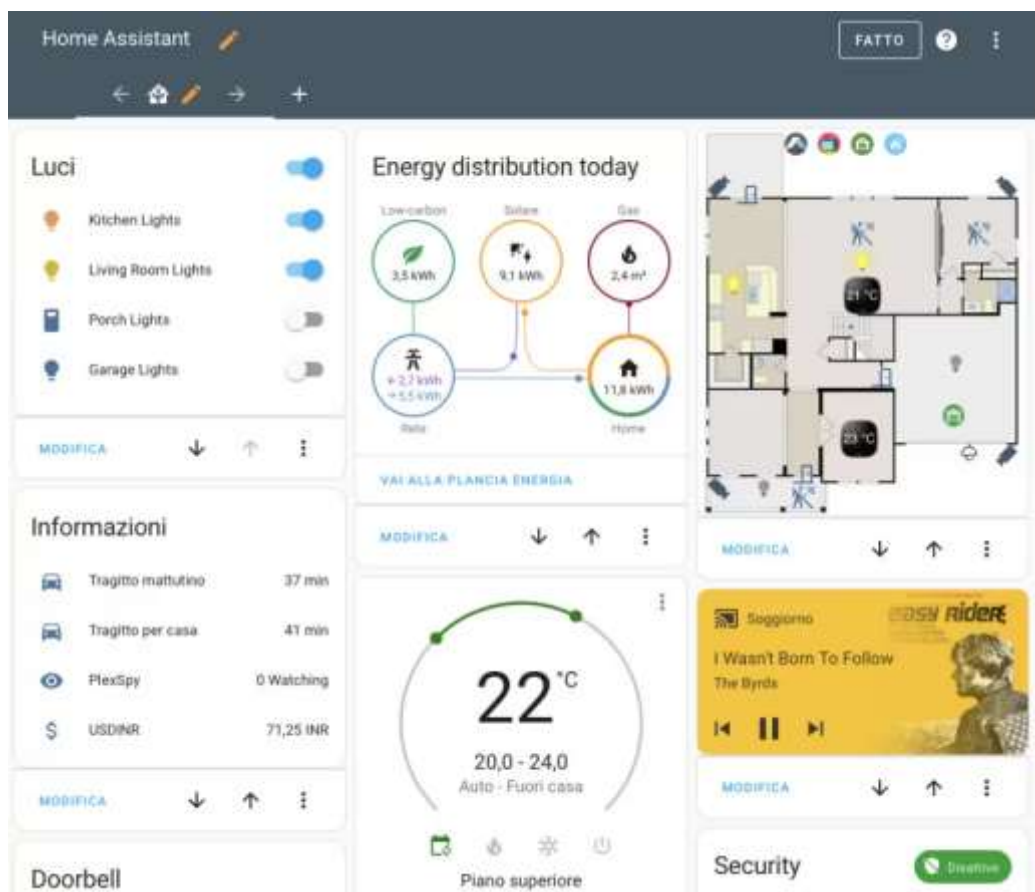


Рисунок 1.7 – Приклад інтерфейсу системи «Home Assistant» [5]

Архітектура Home Assistant побудована на модульному принципі. Серверна частина системи працює локально на пристрої користувача, як-от Raspberry Pi, міні-ПК або NAS-сервер, і забезпечує повну автономність без необхідності постійного підключення до хмарних сервісів. Компонентна система дозволяє підключати тисячі інтеграцій – від розумних лічильників і Wi-Fi-розеток до платформ типу Shelly, TP-Link, MQTT, Z-Wave або Zigbee. Візуалізація даних здійснюється через вбудований вебінтерфейс, у якому користувач може створювати власні панелі моніторингу, графіки та аналітичні дашборди. Завдяки наявності підтримки автоматизацій та сценаріїв, побудованих на умові «якщо–то», система реагує на зміни в середовищі або

| | | | | |
|-------|------|----------|--------|------|
| | | | | |
| Змін. | Арк. | № докум. | Підпис | Дата |

споживанні, вмикаючи чи вимикаючи пристрої відповідно до встановлених правил. Такий підхід забезпечує масштабованість, гнучкість і повний контроль над усіма енергетичними процесами в розумному домі.

Переваги застосунку Home Assistant:

- відкрита архітектура та локальна автономність – система працює без необхідності передачі даних у хмару, що підвищує безпеку та конфіденційність;
- висока гнучкість налаштування – підтримує понад 2000 інтеграцій, включаючи енергетичні сенсори, смарт-розетки та розподілені енергосистеми;
- можливість повної кастомізації інтерфейсу – користувач самостійно створює панелі, графіки та сценарії реагування;
- розширені сценарії автоматизації – система підтримує складну логіку керування приладами з урахуванням часу, подій, показників споживання, температури тощо.

Недоліки застосунку Home Assistant:

- високий поріг входу – для повноцінного налаштування потрібні технічні знання, особливо при роботі з YAML-файлами та локальними інтеграціями;
- відсутність офіційної техпідтримки – проект є спільноторієнтованим і вимагає самостійного пошуку рішень у документації або на форумах;
- часом складна інтеграція деяких пристроїв – особливо з новими або нестандартними API, що вимагає додаткових налаштувань;
- необхідність постійного оновлення – часті оновлення можуть порушувати стабільність роботи інтеграцій без належної адаптації конфігурацій.

Отже, Home Assistant – це потужна платформа для побудови інтелектуального середовища побутової автоматизації, яка особливо ефективна в задачах аналізу та керування енергоспоживанням. Її відкритість, гнучкість і підтримка локального зберігання даних роблять її ідеальним вибором для

| | | | | | | |
|-------|------|----------|--------|------|---------------------------|------|
| | | | | | ІАЛЦ.045480.004 ПЗ | Арк. |
| Змін. | Арк. | № докум. | Підпис | Дата | | 21 |

користувачів із технічним бекграундом, які прагнуть повного контролю над власною інфраструктурою. Незважаючи на складність первинного налаштування, система забезпечує функціональність, недосяжну для багатьох комерційних рішень.

Wattics – це хмарна аналітична платформа, призначена для моніторингу, аналізу та оптимізації енергоспоживання у комерційних, промислових і побутових середовищах (рис. 1.8). Її основне призначення полягає в наданні розширеної аналітики для прийняття енергетичних рішень на основі реальних даних [23]. Система дозволяє користувачам виявляти неефективності, контролювати витрати, формувати звіти для менеджменту та здійснювати енергетичне моделювання.



Рисунок 1.8 – Приклад інтерфейсу системи «Wattics» [24]

Архітектура Wattics побудована на принципі багаторівневої інтеграції. На нижньому рівні знаходяться пристрої збору даних: лічильники, датчики, контролери, які підключаються до шлюзових пристроїв, таких як DEXMA або відповідні API-інтерфейси. Дані зчитуються в реальному часі та передаються через захищений інтернет-канал до хмарної платформи Wattics. На серверному

рівні дані зберігаються, нормалізуються та аналізуються за допомогою вбудованих алгоритмів, зокрема методів машинного навчання. Користувач взаємодіє із системою через веб-інтерфейс, який включає модулі побудови графіків, індикаторів КРІ, візуалізацію трендів, порівняння об'єктів і генерацію автоматичних рекомендацій. Платформа також підтримує інтеграцію з іншими системами енергетичного обліку, бухгалтерського обліку та CRM. Така архітектура забезпечує масштабованість, високу доступність та можливість централізованого аналізу енерговитрат на будь-якому рівні складності.

Переваги застосунку Wattics:

- потужна аналітика та прогнозування – система використовує алгоритми машинного навчання для виявлення відхилень, прогнозу споживання та формування рекомендацій;
- підтримка великої кількості джерел даних – платформа інтегрується з різноманітними лічильниками, BMS-системами, SCADA та API, що дозволяє створювати централізовану енергетичну аналітику;
- гнучкий візуальний інтерфейс – користувачі можуть створювати власні дашборди, графіки, КРІ-панелі та звіти для прийняття управлінських рішень;
- підходить для масштабних застосувань – система підтримує моніторинг великої кількості об'єктів і підрозділів, що робить її придатною для енергоменеджменту підприємств та мереж.

Недоліки застосунку Wattics:

- висока вартість ліцензії – орієнтація на бізнес-сегмент передбачає підписку з платними тарифами, недоступними для індивідуального користувача;
- залежність від хмарної інфраструктури – обробка даних здійснюється виключно на сторонніх серверах, що може викликати питання безпеки та приватності;
- не оптимізована для домашнього використання;

| | | | | |
|-------|------|----------|--------|------|
| | | | | |
| Змін. | Арк. | № докум. | Підпис | Дата |

ІАЛЦ.045480.004 ПЗ

Арк.

23

– потребує спеціального обладнання або шлюзів – інтеграція не завжди можлива із звичайними домашніми пристроями без попередньої адаптації.

Отже, Wattics є професійним інструментом для аналітики енергоспоживання, який відзначається високою гнучкістю, широкими можливостями інтеграції та просунутими функціями візуалізації й прогнозування. Його доцільно використовувати в комерційному та промисловому контексті, де важлива точність, масштабованість та глибина аналізу. У побутовому середовищі застосунок може виявитися надлишково складним і дорогим, однак може бути релевантним для домогосподарств, що мають комплексну інфраструктуру з елементами Smart Grid або мікроенергетики.

З огляду на проведений аналіз трьох розглянутих рішень доцільно узагальнити їх ключові характеристики у табличній формі, що дозволить наочно порівняти функціональні можливості, рівень складності впровадження, цільову аудиторію та наявність інтелектуальних компонентів.

Такий підхід дає змогу обґрунтовано виділити напрями, у яких доцільно вдосконалити нове рішення. Узагальнені результати представлено в табл. 1.2.

Таблиця 1.2 – Порівняння існуючих програмних рішень

| Критерій | Sense Electricity | Home Assistant | Wattics |
|-------------------|-----------------------------------|--------------------------------------|--------------------------------------|
| 1 | 2 | 3 | 4 |
| Тип застосунку | Комерційний (апаратно-програмний) | Відкрите самостійне налаштування ПЗ, | SaaS-платформа для бізнесу |
| Цільова аудиторія | Побутові користувачі (США) | Технічно підготовлені користувачі | Комерційні та промислові користувачі |
| Джерело даних | Сенсори електрощитка | ІоТ-пристрої, локальні сенсори | Smart-лічильники, SCADA, API |

| | | | | |
|-------|------|----------|--------|------|
| | | | | |
| Змін. | Арк. | № докум. | Підпис | Дата |

ІАЛЦ.045480.004 ПЗ

Арк.

24

Продовження таблиці 1.2.

| 1 | 2 | 3 | 4 |
|------------------------|------------------------------------|-------------------------------------|---------------------------------------|
| Функціонал моніторингу | В реальному часі | В реальному часі, історія | В реальному часі, розширена аналітика |
| Інтерфейс користувача | Інтуїтивний, мобільний | Веб-інтерфейс з кастомізацією | Аналітична панель, графіки |
| Вартість | Висока (обладнання підписка) + | Безкоштовне (open source) | Платна підписка |
| Локальність обробки | Хмара | Повністю локальна | Хмара |
| Використання ШІ | Так (розпізнавання приладів) | Обмежено (при ручному налаштуванні) | Так (аналіз та прогнозування) |
| Підтримка рекомендацій | Так | Обмежено | Так |
| Простота впровадження | Висока для США, складна для Європи | Середня (потрібні тех. знання) | Низька для побутового користувача |
| Масштабованість | Обмежена | Висока | Висока |

З таблиці видно, що існуючі рішення або потребують значних технічних навичок для налаштування (Home Assistant), або є складними й недоступними для пересічного користувача через високу вартість або складну архітектуру (Sense, Wattics). Крім того, лише окремі системи частково реалізують механізми штучного інтелекту, переважно у вигляді вбудованих моделей або зовнішніх сервісів, без можливості гнучкого тренування на власних даних користувача. Жодне з рішень не забезпечує одночасно простоту використання, безкоштовність, локальність, та адаптованість під регіональні умови, що й визначає доцільність розробки нового застосунку.

| | | | | |
|-------|------|----------|--------|------|
| | | | | |
| Змін. | Арк. | № докум. | Підпис | Дата |

ІАЛЦ.045480.004 ПЗ

Арк.

25

1.5 Постановка задачі

Метою даного проєкту є розробка інформаційної системи для прогнозування енергоспоживання побутових приладів із використанням алгоритмів машинного навчання, що дозволить здійснювати оцінку ефективності їх використання на основі заданих користувачем параметрів. Система повинна забезпечити інструмент для попереднього аналізу споживання електроенергії за умов, наближених до реальних, з урахуванням впливу типу пристрою, температури, сезону та кількості мешканців у домогосподарстві.

Задача, що вирішується системою:

Побудова автоматизованої програмної системи, яка, ґрунтуючись на історичних даних про енергоспоживання та відповідних умовах використання побутових приладів, дає змогу прогнозувати майбутнє значення споживання для нових комбінацій параметрів.

Вхідні дані:

- CSV-файл з історичними даними про споживання енергії;
- параметри прогнозу: тип пристрою, температура, сезон, кількість мешканців.

Вихідні дані:

- прогнозоване значення енергоспоживання;
- числові метрики точності моделі (MAE, MSE, R^2);
- повідомлення про успішність навчання та тестування моделі.

Функціональні вимоги:

- зчитування CSV-файлу з історичними даними;
- попередня обробка даних, включаючи перетворення категоріальних змінних;
- побудова та тренування регресійної моделі машинного навчання;
- автоматичний поділ вибірки на навчальну та тестову частини;
- відображення результатів прогнозу в інтерфейсі WinForms;
- обчислення та виведення ключових метрик точності моделі;
- можливість введення нових даних для прогнозування вручну;

| | | | | | | |
|-------|------|----------|--------|------|--------------------|------|
| | | | | | ІАЛЦ.045480.004 ПЗ | Арк. |
| Змін. | Арк. | № докум. | Підпис | Дата | | 26 |

- імітаційний запуск прогнозування на випадкових записах із бази для перевірки моделі в умовах, наближених до реального моніторингу.

Нефункціональні вимоги:

- забезпечення зручного, інтуїтивно зрозумілого графічного інтерфейсу користувача;
- мінімальні вимоги до ресурсів ПК для запуску (використання .NET);
- стабільна робота при неповних або частково відфільтрованих даних;
- підтримка одноразового локального запуску без складного встановлення;
- коректна обробка введених значень, перевірка діапазонів і типів даних.

Обмеження задачі:

- використання одного алгоритму машинного навчання;
- джерелом даних виступають лише CSV-файли з попередньо підготовленими прикладами;
- система працює в автономному режимі без інтеграції з фізичними сенсорами чи пристроями.

Результатом розробки є застосунок, що дозволяє користувачеві заздалегідь оцінити очікуване споживання електроенергії для заданого набору побутових умов. Отримані значення можуть бути використані для внутрішньої оцінки ефективності пристроїв, виявлення потенційних перевитрат або як допоміжна інформація для побутового планування енерговитрат.

Висновки до розділу

У рамках даного розділу було проведено всебічний аналіз проблеми ефективного енергоспоживання побутовими приладами з позиції сучасних технологічних, методичних і програмних підходів. Визначено актуальність систем моніторингу енергоспоживання в умовах глобального зростання навантаження на енергосистеми, проаналізовано особливості збору, обробки та інтерпретації даних у домашньому середовищі. Розглянуто сучасні методи оцінювання ефективності використання енергії, включаючи індекси

| | | | | | | |
|-------|------|----------|--------|------|--------------------|------|
| | | | | | ІАЛЦ.045480.004 ПЗ | Арк. |
| Змін. | Арк. | № докум. | Підпис | Дата | | 27 |

енергоефективності, питомі витрати, виявлення аномалій та підходи прогнозного моделювання. Показано, що класичні методи часто поступаються за гнучкістю та точністю інтелектуальним системам, що використовують машинне навчання.

Окрема увага приділена аналізу потенціалу застосування алгоритмів машинного навчання для моделювання споживання енергії. Визначено ключові напрями використання ML-моделей, охарактеризовано джерела даних для їх навчання, а також систематизовано обмеження, з якими стикаються розробники подібних рішень. На основі цього сформовано перелік вимог до майбутньої системи, зокрема: обмеження джерел даних CSV-формату, регресійний характер задачі, необхідність простої реалізації в середовищі WinForms та використання моделі машинного навчання для прогнозування. Крім того, проведено критичний огляд сучасних програмних продуктів – Sense Electricity, Home Assistant та Wattics – з виділенням їхніх переваг і недоліків. Це дозволило обґрунтувати потребу у розробці нового програмного рішення, що поєднує інтелектуальну обробку даних з простотою впровадження, безкоштовністю та мінімальними вимогами до користувача.

Сформульована постановка задачі надає чітке уявлення про цільове призначення системи, вхідні та вихідні параметри, а також функціональні й нефункціональні вимоги до її реалізації. Отримані результати слугують концептуальною основою для математичного моделювання процесу прогнозування енергоспоживання та обґрунтування вибору відповідного методу машинного навчання, що буде детально розглянуто в наступному розділі.

| | | | | |
|-------|------|----------|--------|------|
| | | | | |
| Змін. | Арк. | № докум. | Підпис | Дата |

МАТЕМАТИЧНЕ МОДЕЛЮВАННЯ, АНАЛІЗ МЕТОДІВ ПРОГНОЗУВАННЯ ТА ВИБІР ТЕХНОЛОГІЙ

2.1 Огляд та порівняння моделей машинного навчання для прогнозування

Задача прогнозування енергоспоживання побутових приладів належить до категорії складних багатофакторних аналітичних задач, розв'язання яких потребує глибокого урахування динамічних характеристик споживання, що залежать від широкого спектра чинників. Зокрема, значний вплив мають нерівномірність розподілу навантаження впродовж доби, сезонна мінливість, типологічні особливості електроприладів, умови їх експлуатації, а також соціо-поведінкові звички користувачів. Така багатовимірність та варіативність даних унеможлиблює використання статичних або лінійних підходів, вимагаючи застосування адаптивних методів обробки інформації.

Враховуючи вищезазначене, особливої актуальності набувають методи машинного навчання, які здатні автоматично виявляти приховані закономірності у складних масивах даних та формувати прогностичні моделі з високим рівнем узагальнення. Серед таких підходів найкращі результати демонструють ансамблеві методи, що поєднують вихідні оцінки декількох базових моделей у єдину зважену відповідь. Завдяки такому підходу досягається підвищення точності передбачень, зниження ризику переадаптації та забезпечення стійкості моделі до випадкових флуктуацій у даних, що є особливо важливим у сфері енергетичного прогнозування.

Одним із найрезультативніших представників цієї групи є метод швидкого дерева (ШД), що реалізує механізм градієнтного підсилення [13]. Сутність методу полягає у поетапному навчанні послідовності дерев рішень, кожне з яких уточнює прогноз шляхом корекції залишкової похибки попередніх. Завдяки цьому поступово досягається зменшення сумарної похибки моделі, що дозволяє отримати високу точність навіть за наявності складної структури даних або окремих шумів [26]. На рис. 2.1 відображено узагальнену схему роботи методу

| | | | | |
|-------|------|----------|--------|------|
| | | | | |
| Змін. | Арк. | № докум. | Підпис | Дата |

ІАЛЦ.045480.004 ПЗ

Арк.

29

ШД, яка ілюструє поетапний процес формування остаточного ансамблевого прогнозу на основі результатів слабких моделей.

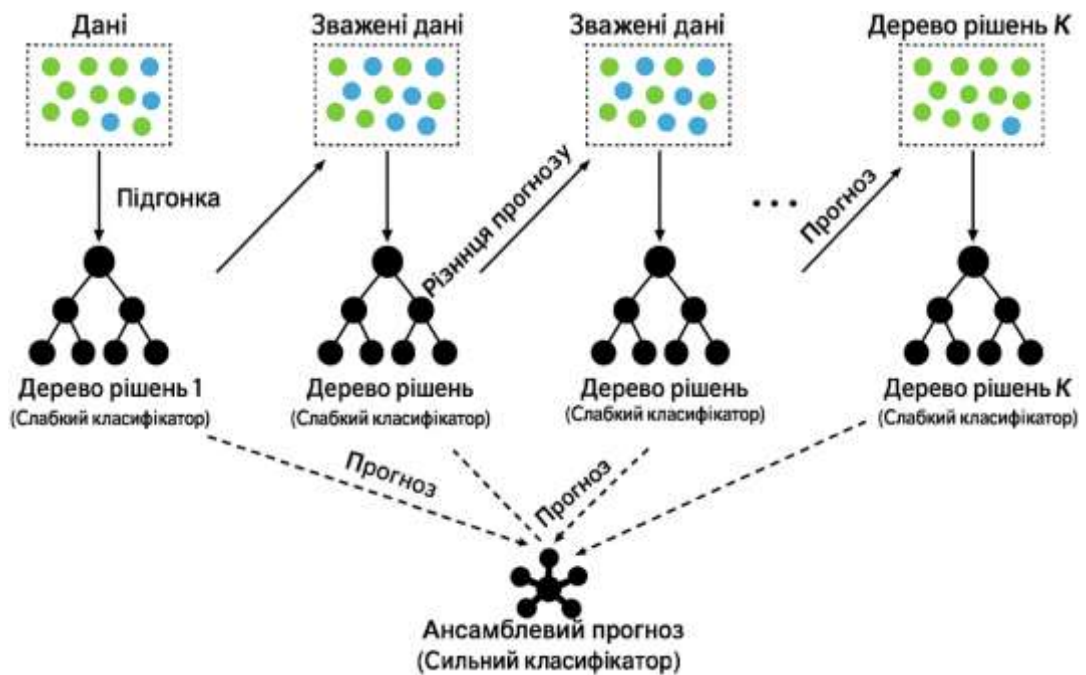


Рисунок 2.1 – Схема роботи методу ШД

Під час ітеративної процедури навчання кожне нове дерево рішень створюється з урахуванням помилок попереднього. Зважування прикладів дозволяє сфокусувати увагу наступної моделі на тих випадках, де прогноз виявився неточним. Після завершення усіх ітерацій відбувається об'єднання результатів окремих дерев у єдиний узагальнений прогноз. Така стратегія дозволяє суттєво підвищити точність моделі, зберігаючи при цьому помірну обчислювальну складність, що робить метод швидкого дерева ефективним інструментом у задачах регресійного прогнозування побутового енергоспоживання.

Серед методів, що ефективно вирішують задачі регресії в умовах обмеженого обсягу даних або підвищених вимог до точності, окрему увагу заслуговує метод опорних векторів для регресії (Support Vector Regression, МОВДР) [25]. Цей підхід ґрунтується на ідеї побудови гіперплощини у багатовимірному просторі ознак, яка наближає залежність між вхідними

| | | | | |
|-------|------|----------|--------|------|
| Змін. | Арк. | № докум. | Підпис | Дата |
| | | | | |

параметрами та прогнозованим значенням із максимально допустимим відхиленням. Основна мета – знайти таку функцію, яка забезпечує не лише високу точність наближення, але й узагальнюючу здатність, тобто стійкість до перенавчання [18].

Принцип роботи МОВДР реалізується через перетворення вхідного простору у вищовимірний простір ознак за допомогою ядрових функцій (наприклад, радіальної базисної функції), після чого здійснюється пошук такої моделі, яка мінімізує похибку у межах заданої епсилон-трубки. Важливою властивістю цього методу є те, що він базується лише на частині навчальних прикладів – так званих опорних векторах, які розташовані на межі зони допуску або за її межами, що дозволяє моделі зосередитися на найбільш значущих даних.

На рис. 2.2 зображено структурну схему роботи моделі підтримки векторів у формі мережі, що включає вхідний шар ознак, прихований шар ядрових перетворень та вихідний шар з підсумовуванням з урахуванням ваг і зсуву. Кожен вузол прихованого шару реалізує функцію подібності між вхідним вектором і певним опорним прикладом, після чого відбувається лінійна комбінація отриманих значень із ваговими коефіцієнтами та додаванням зсуву для формування кінцевого прогнозу.

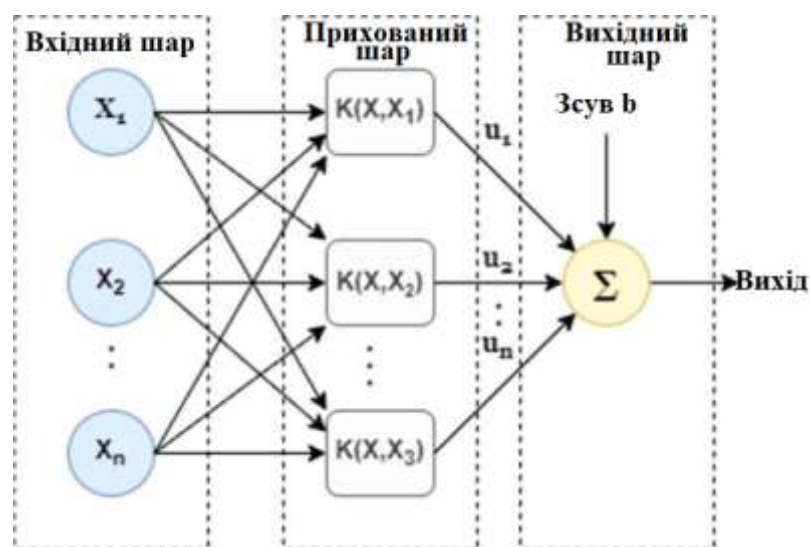


Рисунок 2.2 – Схема роботи методу ОВ

| | | | | |
|-------|------|----------|--------|------|
| | | | | |
| Змін. | Арк. | № докум. | Підпис | Дата |

Дана архітектура дозволяє моделі точно наближати складні нелінійні залежності навіть при невеликій кількості навчальних прикладів. Метод опорних векторів демонструє особливо добрі результати в умовах високої розмірності простору ознак і обмеженого рівня шуму, що робить його придатним для задач побутового прогнозування, де якість даних не завжди є ідеальною, а обсяг вибірки обмежений.

Наївний байєсівський класифікатор (НБК) належить до класу простих, проте результативних методів, які можуть бути застосовані у задачах аналізу споживання енергії та виявлення характерних режимів функціонування приладів [3]. Його математична основа ґрунтується на теоремі Байєса, яка дозволяє обчислювати ймовірність належності об'єкта до певного класу з урахуванням його ознак за припущенням їх статистичної незалежності.

Метод демонструє конкурентні результати при роботі з великими обсягами структурованих або категоріальних даних [4]. У задачах енергетичного аналізу наївний байєсівський класифікатор доцільно використовувати як базову модель для класифікації споживчих профілів, виявлення типових сценаріїв використання приладів або попереднього сегментування користувачів.

На рис. 2.3 схематично показано принцип дії НБК. Зліва представлено сукупність об'єктів, що мають різні ознаки (наприклад, форма, колір, розмір), які подаються на вхід класифікатору.

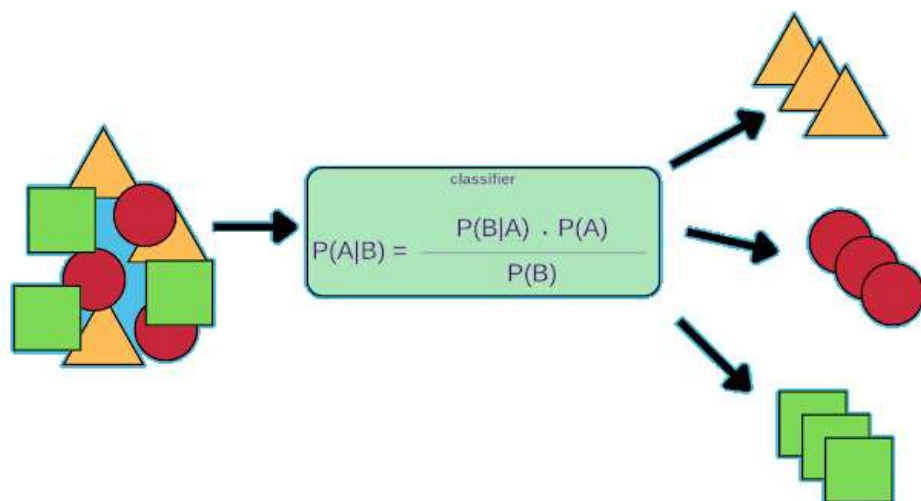


Рисунок 2.3 – Схема роботи методу НБК

| | | | | |
|-------|------|----------|--------|------|
| | | | | |
| Змін. | Арк. | № докум. | Підпис | Дата |

У середині моделі виконується обчислення ймовірностей на основі теореми Байєса, після чого кожен об'єкт відноситься до одного з можливих класів згідно з найвищою апостеріорною ймовірністю. Вихідним результатом є розділення множини об'єктів за класами, що дозволяє здійснити швидку, інтерпретовану та статистично обґрунтовану класифікацію.

Попри свою простоту, метод наївного Байєса часто використовується як базовий орієнтир у задачах машинного навчання і є надійним інструментом у випадках, коли потрібна висока швидкодія та стійкість до часткової відсутності даних.

Таблиця 2.1 – Порівняльна характеристика розглянутих методів

| Характеристика | Швидке дерево | Метод опорних векторів | Наївний байєсівський класифікатор |
|---------------------------------------|---------------------------------|------------------------------------|-----------------------------------|
| Точність регресійного прогнозу | Висока | Висока на малих вибірках | Помірна |
| Стійкість до шумів у даних | Висока | Помірна | Низька |
| Підтримка змішаних типів ознак | Так (кількісні й категоріальні) | Лише числові (або через кодування) | Категоріальні переважно |
| Інтерпретованість моделі | Середня | Низька | Висока |
| Потреба в налаштуванні параметрів | Помірна | Висока | Низька |
| Швидкодія при прогнозуванні | Висока | Помірна | Висока |
| Придатність для задач регресії | Так | Так | Обмежена |
| Адаптивність до великого обсягу даних | Так | Обмежена | Так |

Враховуючи специфіку задачі прогнозування побутового енергоспоживання, доцільно здійснити порівняльний аналіз методів машинного навчання, які розглядались вище з метою визначення їхніх ключових властивостей, сильних сторін та обмежень. Порівняльна табл. 2.1 дозволяє систематизувати отриману інформацію.

Вибір методу швидкого дерева для реалізації системи прогнозування енергоспоживання зумовлений його здатністю ефективно працювати з даними змішаного типу, забезпечуючи високу точність навіть у разі наявності шумів або неповноти вибірки. Метод демонструє хорошу масштабованість і стабільну швидкодію, що особливо важливо при інтеграції в середовище з обмеженими обчислювальними ресурсами. На відміну від методу опорних векторів, він не потребує складної оптимізації параметрів і краще адаптується до великого обсягу даних. У порівнянні з наївним байєсівським підходом, ШД забезпечує суттєво вищу точність регресійного прогнозу та кращу здатність моделювати складні залежності між параметрами.

2.2 Побудова математичної моделі ефективності

Оцінювання ефективності енергоспоживання побутових приладів є критично важливим етапом при розробці аналітичних рішень у сфері енергетичного моніторингу та управління. Для побудови обґрунтованих прогнозів і виявлення нераціональних сценаріїв використання необхідно перейти від суто емпіричних спостережень до формалізованого математичного опису, який дозволяє узагальнювати поведінку системи та робити обчислювальні висновки.

Першим кроком у цьому процесі є визначення структури вхідних даних, на основі яких буде здійснюватися навчання моделі. Необхідно окреслити, які змінні характеризують як сам прилад, так і умови його експлуатації, та яким чином за цими змінними можна передбачити рівень енергоспоживання. Формалізація цих зв'язків забезпечує основу для подальшого застосування

| | | | | |
|-------|------|----------|--------|------|
| | | | | |
| Змін. | Арк. | № докум. | Підпис | Дата |

ІАЛЦ.045480.004 ПЗ

Арк.

34

методів машинного навчання й дозволяє оцінити ефективність приладу в різних сценаріях.

Нехай у розпорядженні є вибірка емпіричних даних, кожен запис якої описує побутовий електричний прилад, контекст його використання та відповідне значення енергоспоживання. Така вибірка формалізується у вигляді множини:

$$D = \{(x_i, y_i)\}_{i=1}^N, \quad (2.1)$$

де $i = 1, 2, \dots, N$ – індекс спостереження; $x_i \in R^d$ – вектор ознак, що характеризує i -те спостереження; $y_i \in R$ – відповідне числове значення споживання енергії (у кВт·год).

Цільовою є задача побудови регресійної функції $f(x)$, яка для довільного нового вектора ознак x_{new} дозволить отримати оцінку $\hat{y}_{new} = f(x_{new})$, яка наближено відтворює реальне споживання енергії.

Оскільки вхідні дані містять як числові, так і категоріальні характеристики, необхідною є їх попередня обробка. Після кодування категорійних ознак (тип приладу, сезон) вектор ознак x_i набуває такого вигляду:

$$x_i = (a_{i1}, \dots, a_{ik}, t_i, s_{i1}, \dots, s_{il}, h_i)^T, \quad (2.2)$$

де $a_{i1}, \dots, a_{ik} \in \{0, 1\}$ – елементи one-hot-кодування типу приладу (наприклад, «мікрохвильова піч», «холодильник» тощо); $t_i \in R$ – температура навколишнього середовища, виміряна в момент використання приладу; $s_{i1}, \dots, s_{il} \in \{0, 1\}$ – one-hot-представлення сезону («зима», «літо» тощо); $h_i \in R$ – чисельність домогосподарства, тобто кількість осіб, які мешкають у житловому приміщенні.

Це перетворення дозволяє застосовувати методи машинного навчання, зокрема ШД, без втрати семантики вхідних параметрів.

Метод швидкого дерева реалізує ітеративне побудування ансамблю регресійних дерев, кожне з яких навчено на залишках попередніх передбачень. Це означає, що на кожній ітерації модель уточнює попередній прогноз, мінімізуючи залишкову похибку. Формально, прогноз на M -тій ітерації

| | | | | |
|-------|------|----------|--------|------|
| | | | | |
| Змін. | Арк. | № докум. | Підпис | Дата |

визначається як сума прогнозів усіх попередніх дерев з урахуванням коефіцієнта навчання:

$$\hat{y}_i^{(M)} = \sum_{m=1}^M v \cdot f_m(x_i), \quad (2.3)$$

де $f_m(x_i)$ – передбачення m -го дерева для i -го об'єкта; $v \in (0,1]$ – коефіцієнт навчання (learning rate), який контролює швидкість наближення до оптимального розв'язку; M – кількість дерев в ансамблі.

Таким чином, замість одноразового побудування складної моделі, ШД формує послідовність простіших моделей (дерев), що поетапно вдосконалюють якість прогнозу.

Кожне дерево в ансамблі можна інтерпретувати як розбиття простору ознак на L_m диз'юнктних областей, що відповідають листям дерева. Для кожної такої області передбачення моделі є сталою величиною:

$$f_m(x) = \sum_{l=1}^{L_m} w_{ml} \cdot 1(x \in S_{ml}), \quad (2.4)$$

де L_m – кількість листів у m -му дереві; w_{ml} – фіксоване передбачене значення у листі l ; S_{ml} – підпростір ознак, що відповідає l -му листу; $1(\cdot)$ – індикаторна функція, яка дорівнює 1, якщо x належить області S_{ml} , інакше 0.

Це означає, що дерево виконує поетапне бінарне розбиття простору R^d , і кожне передбачення зводиться до вибору однієї з фіксованих відповідей залежно від потрапляння об'єкта до відповідного листа.

Усі дерева у складі ансамблю спрямовані на мінімізацію сукупної похибки моделі. В класичному варіанті метод ШД націлений на зменшення середньої квадратичної помилки (MSE), що на M -тій ітерації визначається як:

$$L^{(M)} = \sum_{i=1}^N (y_i - \hat{y}_i^{(M)})^2, \quad (2.5)$$

де y_i – фактичне значення споживання; $\hat{y}_i^{(M)}$ – передбачене значення після побудови M дерев; N – кількість спостережень у навчальній вибірці.

Ця функція виступає як мета оптимізації: кожне нове дерево, що додається до ансамблю, будується з метою зменшити значення цієї суми похибок. Такий підхід забезпечує поступове уточнення моделі та її адаптацію до закономірностей, що містяться у навчальних даних.

| | | | | |
|-------|------|----------|--------|------|
| | | | | |
| Змін. | Арк. | № докум. | Підпис | Дата |

Після побудови $m - 1$ дерев, модель вже має певний проміжний прогноз $\hat{y}_i^{(m-1)}$. Для того щоб m -те дерево покращило цей прогноз, воно навчається не на первинних мітках y_i , а на так званих резидуалах (залишках), які в контексті методу ШД є аналітичними градієнтами функції помилки. У випадку квадратичної похибки цей залишок для кожного об'єкта має вигляд:

$$r_i^{(m)} = \frac{\partial L}{\partial \hat{y}_i} \Big|_{\hat{y}_i = \hat{y}_i^{(m-1)}} = 2 \left(\hat{y}_i - \hat{y}_i^{(m-1)} \right), \quad (2.6)$$

де $r_i^{(m)}$ – орієнтовна величина помилки для i -го об'єкта на поточній ітерації.

Ці значення залишків виступають у ролі нових міток для побудови наступного дерева, тобто $f_m(\cdot)$ апроксимує саме $r_i^{(m)}$.

Під час побудови дерева важливо визначити, яке з можливих розбиттів простору ознак покращує пояснювальну здатність дерева. Для цього застосовується критерій інформаційного приросту, який у регресійному варіанті виражається через зміну дисперсії резидуалів:

$$\Delta Q(q, \theta) = Q(S) - Q(S_L) - Q(S_R), \quad (2.7)$$

де q – ознака, за якою виконується поділ; θ – поріг поділу; S – поточний вузол (підмножина об'єктів); S_L, S_R , – лівий та правий дочірні вузли після поділу.

Іншими словами, вимірюється наскільки поділ зменшує сумарну дисперсію залишків – тобто наскільки «чистіші» залишки стали в обох частинах.

Щоб обчислити $Q(S)$, тобто дисперсію залишків у підмножині S , використовується наступне співвідношення:

$$Q(S) = \frac{1}{|S|} \sum_{i \in S} \left(r_i^{(m)} - \bar{r}_S \right)^2, \quad \bar{r}_S = \frac{1}{|S|} \sum_{i \in S} r_i^{(m)}, \quad (2.8)$$

де $Q(S)$ – середньоквадратичне відхилення залишків у межах вузла; \bar{r}_S – середнє значення залишків у поточній підмножині.

Дане визначення дисперсії, яке дозволяє кількісно оцінити якість поділу: чим менше розкид залишків у вузлі, тим більш передбачуване дерево.

Після того, як вузол дерева було сформовано та всі об'єкти потрапили до певного листа S_{ml} , потрібно визначити сталу величину w_{ml} , яку цей лист буде

повертати при передбаченні. Для цього використовується середнє значення залишків у відповідному підпросторі:

$$w_{ml} = \frac{1}{|S_{ml}|} \sum_{i \in S_{ml}} r_i^{(m)}. \quad (2.9)$$

Таким чином, дерево не просто класифікує об'єкти, а уточнює поточний стан моделі на основі аналізу залишків.

Після побудови усіх дерев модель набуває завершеної форми. Остаточний прогноз на i -му прикладі записується у вигляді повної суми по всіх деревах ансамблю:

$$\hat{y}_i = \sum_{m=1}^M v \cdot f_m(x_i), \quad (2.10)$$

де \hat{y}_i – фінальне передбачення для i -го запису; решта позначень залишаються аналогічними до формули 2.3.

Це дозволяє зрозуміти структуру ансамблю як послідовного коригування первинного прогнозу за рахунок слабких моделей (невеликих дерев), кожна з яких навчається на залишках попередніх.

Після завершення процесу навчання моделі, тобто побудови ансамблю з M дерев, виникає потреба у кількісному оцінюванні її продуктивності. Це дозволяє не лише перевірити здатність моделі до апроксимації навчальної вибірки, але й проаналізувати її здатність до узагальнення на незалежних тестових даних.

У класичних регресійних задачах найчастіше використовують три базові метрики: середню абсолютну похибку (MAE), корінь середньоквадратичної похибки (RMSE) та коефіцієнт детермінації (R^2).

Ця метрика MAE обчислює середню величину абсолютного відхилення між фактичними та передбаченими значеннями споживання енергії:

$$\text{MAE} = \frac{1}{T} \sum_{i \in T} |y_i - \hat{y}_i|, \quad (2.11)$$

де T – тестова вибірка; y_i – справжнє значення енергоспоживання; \hat{y}_i – прогноз моделі.

Цей показник особливо цінний тоді, коли важливо мінімізувати середню фактичну похибку без акценту на великі відхилення.

На відміну від MAE, RMSE посилено штрафує великі відхилення, оскільки базується на квадраті різниці:

$$\text{RMSE} = \sqrt{\frac{1}{|T|} \sum_{i \in T} (y_i - \hat{y}_i)^2}. \quad (2.12)$$

RMSE має таку саму розмірність, як і цільова змінна (тобто кВт·год), і тому добре інтерпретується в контексті практичних допусків до похибки в побутовому середовищі.

Додатково до абсолютних метрик часто використовують відносну характеристику, яка показує, яку частку дисперсії залежної змінної пояснює модель:

$$R^2 = 1 - \frac{\sum_{i \in T} (y_i - \hat{y}_i)^2}{\frac{1}{|T|} \sum_{i \in T} (y_i - \bar{y}_T)^2}. \quad (2.13)$$

де $\bar{y}_T = \frac{1}{|T|} \sum_{i \in T} y_i$ – середнє значення енергоспоживання в тестовій вибірці.

Значення R^2 , близьке до 1, вказує на високий рівень відповідності моделі фактичним даним, що є критично важливим при використанні прогнозів для подальшого енергоменеджменту.

Поza межами формального прогнозування, побудована модель може використовуватись як аналітичний інструмент для порівняння енергоспоживання за різних сценаріїв експлуатації побутових пристроїв. Для цього вводимо поняття абсолютної та відносної економії енергії, які можуть бути оцінені на основі змінених вхідних даних (наприклад, при зміні температури, сезону або типу приладу).

Абсолютна економія:

$$\Delta E = \hat{y}_{base} - \hat{y}_{opt}, \quad (2.14)$$

де \hat{y}_{base} – прогноз споживання за поточних умов; \hat{y}_{opt} – прогноз для енергозберігаючого сценарію (наприклад, менш енергоємний прилад).

Відносна економія у відсотках:

$$\eta = \frac{\Delta E}{\hat{y}_{base}} \cdot 100\%. \quad (2.15)$$

| | | | | |
|-------|------|----------|--------|------|
| | | | | |
| Змін. | Арк. | № докум. | Підпис | Дата |

Дані оцінки не лише дозволяють виявити прилади з підвищеним енергоспоживанням, а й надають користувачу аргументовані рекомендації щодо потенційної оптимізації режимів їх експлуатації.

Метод швидкого дерева у задачах прогнозування енергоспоживання побутових пристроїв зможе демонструвати високу придатність завдяки своїй здатності моделювати складні, нелінійні взаємозв'язки між множиною ознак (тип приладу, температура, сезонність, чисельність домогосподарства) та кількісною цільовою змінною.

Побудована математична модель забезпечує не лише точне чисельне передбачення споживання, але й пропонує природну інтерпретацію впливу кожної змінної завдяки структурі дерева. Ансамблевий характер алгоритму дозволяє зменшити перенавчання, а набір метрик дає змогу оцінити як загальну якість моделі, так і її прикладну придатність. Включення формалізованих показників економії енергії (формули 2.14–2.15) трансформує модель із діагностичного інструменту у засіб підтримки прийняття рішень щодо підвищення побутової енергоефективності.

2.3 Вибір технологій для розробки інформаційної системи

Розробка сучасної інформаційної системи передбачає залучення відповідного інструментарію, який забезпечить не лише реалізацію функціональних вимог, а й гнучкість, масштабованість та сумісність із існуючими технологічними платформами. На етапі вибору мов програмування та середовищ розробки необхідно враховувати специфіку предметної області, обсяг оброблюваних даних, тип взаємодії з користувачем, а також можливості інтеграції з аналітичними або штучно-інтелектуальними модулями.

Python є інтерпретованою мовою програмування високого рівня, яка характеризується простим та зрозумілим синтаксисом, що сприяє швидкій розробці прототипів і наукових обчислень [11]. Завдяки розгалуженій екосистемі бібліотек (таких як NumPy, Pandas, TensorFlow, Scikit-learn), Python є одним із найпоширеніших інструментів у сфері машинного навчання та обробки даних.

| | | | | |
|-------|------|----------|--------|------|
| | | | | |
| Змін. | Арк. | № докум. | Підпис | Дата |

Його використання доцільне в задачах аналітики, автоматизації процесів та побудови серверних частин інформаційних систем.

Java є універсальною мовою програмування з підтримкою принципу "написано один раз – працює всюди", що забезпечується віртуальною машиною Java (JVM) [28]. Вона має сувору об'єктно-орієнтовану парадигму, що сприяє модульності, масштабованості та повторному використанню коду, особливо в складних багаторівневих системах. Java широко застосовується у фінансових, телекомунікаційних та банківських програмних рішеннях, а також у розробці вебсервісів і мобільних додатків (Android).

C# є сучасною об'єктно-орієнтованою мовою програмування, розробленою корпорацією Microsoft для роботи на платформі .NET [14]. Вона поєднує у собі простоту використання з високою продуктивністю, що робить її зручною для створення як клієнтських застосунків (з графічним інтерфейсом), так і серверних систем. Мова активно використовується в розробці бізнес-орієнтованого програмного забезпечення, десктопних додатків (Windows Forms, WPF), а також вебплатформ через ASP.NET.

З огляду на різноманітність можливостей розглянутих мов програмування, доцільним є їх порівняння за низкою ключових характеристик, що мають практичне значення при виборі технологій для реалізації інформаційної системи. Зокрема, буде враховано як якісні, так і кількісні показники, серед яких: продуктивність виконання, підтримка типізації, наявність інструментів для побудови графічного інтерфейсу, ступінь інтеграції з корпоративними платформами, а також відповідність до стандартів безпеки та стабільності у виробничих середовищах.

У табл. 2.2 наведено узагальнене порівняння мов Python, Java та C# за основними технічними та експлуатаційними параметрами.

Виходячи з порівняльного аналізу, мова програмування C# є доцільним вибором для розробки інформаційної системи завдяки високій продуктивності, повній інтеграції з операційною системою Windows та розвиненим інструментам для створення графічного інтерфейсу. Вбудована підтримка асинхронного

програмування та паралельних обчислень забезпечує ефективну реалізацію задач, пов'язаних із обробкою великих обсягів даних або взаємодією з користувачем у режимі реального часу. Крім того, платформа .NET, на якій базується C#, гарантує стабільність, масштабованість та відповідність вимогам промислових розробок.

Таблиця 2.2 – Порівняння мов програмування

| Характеристика | Python | Java | C# |
|---|-------------------------------------|----------------------------|--|
| Синтаксична простота | Висока | Середня | Середня |
| Продуктивність виконання (операцій/с) | ~1.2 млн | ~5–7 млн | ~8–10 млн |
| Підтримка статичної типізації | Низька (динамічна типізація) | Висока | Висока |
| Наявність вбудованих GUI-інструментів | Обмежена (через сторонні пакети) | Помірна (JavaFX, Swing) | Розвинена (WinForms, WPF) |
| Інтеграція з Windows-середовищем | Обмежена | Середня | Повна |
| Наявність засобів для паралельних обчислень | Базова підтримка (через бібліотеки) | Підтримується через потоки | Вбудована підтримка (async/await, TPL) |
| Стабільність у виробничих системах | Висока (аналітичні задачі) | Висока | Висока |
| Рівень безпеки при розгортанні | Середній | Високий | Високий |

Наявність широкого спектра бібліотек, а також підтримка сучасних стандартів безпеки сприяє розробці надійного та підтримуваного програмного забезпечення. Сукупність цих факторів визначає C# як оптимальну мову для

| | | | | |
|-------|------|----------|--------|------|
| Змін. | Арк. | № докум. | Підпис | Дата |
|-------|------|----------|--------|------|

створення функціонально повноцінної та технологічно ефективної інформаційної системи.

Після визначення мови програмування C# наступним етапом є вибір середовища розробки, яке забезпечить ефективну підтримку життєвого циклу проєкту – від написання коду до тестування, налагодження та розгортання. Серед найпоширеніших інструментів для розробки на C# розглядаються Visual Studio 2022, Visual Studio Code та JetBrains Rider. Кожне з цих середовищ має власні переваги залежно від типу застосунку, вимог до інтеграції, продуктивності та зручності користувача.

Visual Studio 2022 є повнофункціональним середовищем розробки, спеціалізованим під платформу .NET, що забезпечує глибоку інтеграцію з інструментами Microsoft [21]. Воно підтримує розробку графічних застосунків, API, баз даних та має потужні засоби налагодження і профілювання продуктивності. Завдяки широким можливостям розширення, підтримці Git, Azure та автоматизованого тестування, Visual Studio 2022 вважається стандартом для корпоративної розробки на C#.

Visual Studio Code – це легкий, кросплатформний редактор з підтримкою C# через розширення, зокрема C# for OmniSharp [22]. Хоча він не містить повного функціоналу традиційної IDE, проте є дуже зручним для роботи з невеликими проєктами, мікросервісами або редагуванням конфігураційних файлів. Його ключовими перевагами є швидкодія, мінімалістичний інтерфейс та широка підтримка плагінів.

JetBrains Rider є потужним комерційним середовищем, що поєднує можливості ReSharper і IntelliJ-платформи, забезпечуючи зручну навігацію, розумний аналіз коду та інтеграцію з різними типами проєктів на .NET [15]. Rider підтримує не лише C#, а й інші мови, що робить його актуальним для мультиплатформних і гібридних проєктів. Середовище має високу стабільність, хорошу інтеграцію з системами контролю версій та підтримує розробку як під Windows, так і під Linux або macOS.

| | | | | |
|-------|------|----------|--------|------|
| | | | | |
| Змін. | Арк. | № докум. | Підпис | Дата |

ІАЛЦ.045480.004 ПЗ

Арк.
43

З метою обґрунтованого вибору оптимального інструменту для розробки системи доцільно здійснити порівняльний аналіз середовищ, що підтримують розробку на мові С# У табл. 2.3 наведено порівняння розглянутих середовищ розробки за основними характеристиками.

Таблиця 2.3 – Порівняння середовищ розробки для С#-застосунків

| Характеристика | Visual Studio 2022 | Visual Studio Code | JetBrains Rider |
|--|--------------------------|-----------------------------|-----------------|
| Повнота підтримки .NET-платформи | Повна | Часткова (через розширення) | Повна |
| Засоби побудови графічного інтерфейсу | Розширені | Відсутні | Обмежені |
| Налагодження і профілювання | Вбудовані, розширені | Обмежені (через плагіни) | Добрі |
| Інтеграція з Azure та іншими сервісами | Глибока | Базова | Базова |
| Підтримка тестування та CI/CD | Повна (вбудовані засоби) | Через сторонні інструменти | Часткова |
| Стабільність при роботі з великими рішеннями | Висока | Середня | Висока |
| Можливість роботи з БД (ORM, SQL) | Розширена інтеграція | Через розширення | Висока |

Обрання середовища Visual Studio 2022 зумовлено його широкою функціональністю, яка повністю охоплює потреби повноцінної розробки С#-застосунків на платформі .NET. Середовище забезпечує інтегровану підтримку створення графічного інтерфейсу, роботу з базами даних, засоби налагодження, тестування та профілювання, що суттєво прискорює розробку та підвищує якість програмного продукту. Завдяки глибокій інтеграції з хмарними сервісами Azure, системами керування версіями та CI/CD-пайплайнами, Visual Studio 2022 дозволяє організувати весь цикл розробки в межах єдиної

платформи. Високий рівень стабільності при роботі з великими рішеннями, розширюваність за рахунок вбудованих модулів і плагінів, а також офіційна підтримка з боку Microsoft роблять це середовище придатним для реалізації складних і масштабованих інформаційних систем.

Висновки до розділу

У рамках даного розділу було проведено всебічне дослідження, спрямоване на формалізацію задачі прогнозування енергоспоживання побутових пристроїв, аналіз ефективності методів машинного навчання та обґрунтований вибір технологій для реалізації інформаційної системи. Здійснено порівняння низки популярних алгоритмів класифікації, зокрема методу опорних векторів, наївного байєсівського класифікатора та швидкого дерева, на основі чого прийнято рішення використовувати швидке дерево як основну модель прогнозування завдяки його високій швидкодії та узагальнювальній здатності. Побудовано формалізовану математичну модель, яка описує взаємозв'язки між характеристиками пристроїв, умовами їх використання та фактичним рівнем споживаної енергії, що дозволяє забезпечити точне передбачення типових сценаріїв споживання.

На основі аналізу було обґрунтовано вибір C# як технологічної основи для реалізації системи, з огляду на її високу продуктивність, підтримку типізації, засоби побудови графічного інтерфейсу та інтеграцію з інфраструктурою Windows. Проведено також порівняння середовищ розробки, у результаті чого було обрано Visual Studio 2022 як найбільш функціонально насичену платформу, здатну забезпечити повний життєвий цикл розробки програмного забезпечення. Отримані в цьому розділі результати створюють міцне підґрунтя для наступного етапу – вибору відповідної бібліотеки для машинного навчання та проектування архітектури інформаційної системи з урахуванням сформованих технічних і функціональних вимог

| | | | | |
|-------|------|----------|--------|------|
| | | | | |
| Змін. | Арк. | № докум. | Підпис | Дата |

3 ВИБІР БІБЛІОТЕКИ ТА ПРОЕКТУВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ

3.1 Аналіз та вибір бібліотеки машинного навчання

Розробка системи прогнозування енергоспоживання побутових пристроїв передбачає інтеграцію алгоритмів машинного навчання, здатних обробляти різномірні параметри та формувати точні прогнози на основі історичних даних. Для забезпечення ефективності реалізації обраної моделі швидкого дерева в межах платформи .NET необхідно здійснити вибір відповідної бібліотеки машинного навчання, яка забезпечить підтримку необхідних функцій, сумісність із C# та можливість гнучкого розширення в подальшому.

Особливу увагу слід приділити продуктивності обробки даних, зручності побудови конвеєра навчання та прогнозування, а також наявності документації та прикладних інструментів для інтеграції в архітектуру програмного забезпечення. Серед бібліотек, доступних для середовища .NET, актуальними для розгляду є ML.NET, Accord.NET та Infer.NET. Кожна з них реалізує власний підхід до роботи з моделями, має певну спрямованість щодо типів задач і рівня підтримки алгоритмів, що й зумовлює необхідність подальшого аналізу їх можливостей.

ML.NET – це кросплатформенна бібліотека машинного навчання з відкритим кодом, розроблена компанією Microsoft. Вона дозволяє створювати, навчати та використовувати моделі машинного навчання безпосередньо у застосунках, написаних мовою C# або іншими .NET-орієнтованими мовами [9]. Основною перевагою ML.NET є те, що вона дозволяє працювати з моделями штучного інтелекту без потреби переходу на Python чи інші сторонні платформи, що особливо важливо при розробці корпоративних або вбудованих систем на базі .NET.

Бібліотека підтримує широкий спектр алгоритмів для класифікації, регресії, кластеризації, виявлення аномалій, прогнозування серій, а також роботу з текстовими та табличними даними. Особливістю є можливість створення

трансформаційного конвеєра (pipeline), в якому поєднуються попередня обробка, перетворення ознак і навчання моделі в одному узгодженому процесі.

Accord.NET – це потужна .NET-бібліотека з відкритим кодом, призначена для вирішення задач машинного навчання, статистичного аналізу, обробки зображень, обробки сигналів та чисельних обчислень [2]. Вона розроблена як розширення до AForge.NET, однак значно перевершує її за функціональністю, охоплюючи широкий спектр алгоритмів та інструментів для роботи з даними.

Цей інструмент орієнтований на дослідницькі, інженерні та прикладні завдання, пов'язані з аналізом даних, прогнозуванням, розпізнаванням образів, побудовою моделей і статистичною інтерпретацією результатів. Accord.NET включає реалізації найпоширеніших алгоритмів: від регресійних моделей і класифікаторів (SVM, дерев рішень, наївного Байєса) до методів зниження розмірності (PCA, LDA) та кластеризації (k-середніх, EM, DBSCAN).

Infer.NET – це бібліотека для побудови моделей ймовірнісного програмування, розроблена Microsoft Research [27]. Вона дозволяє створювати складні байєсівські моделі, в яких зв'язки між змінними задаються у вигляді математичних залежностей, а обчислення базуються на апроксимації повної ймовірнісної моделі.

На відміну від класичних підходів до машинного навчання, де модель навчається на основі фіксованої структури, Infer.NET дозволяє явно описати причинно-наслідкові залежності між параметрами. Це забезпечує глибший контроль над тим, як саме формується прогноз і як враховується невизначеність у даних. Система автоматично виконує виведення – тобто обчислення апостеріорних розподілів за заданою структурою моделі.

Розглянуті вище бібліотеки мають свої переваги у різних аспектах. ML.NET вирізняється високим ступенем готовності до промислового застосування, чітко структурованою архітектурою побудови моделей та вбудованими засобами для оцінювання якості. Accord.NET, у свою чергу, орієнтована на дослідницькі та інженерні задачі з високою гнучкістю в ручному налаштуванні алгоритмів і математичній обробці. Infer.NET, як інструмент

| | | | | |
|-------|------|----------|--------|------|
| | | | | |
| Змін. | Арк. | № докум. | Підпис | Дата |

ймовірнісного програмування, забезпечує виражену гнучкість у моделюванні причинно-наслідкових структур та роботи з неповними даними. З метою систематизації можливостей, на табл. 3.1 подано порівняльний аналіз зазначених бібліотек за основними характеристиками, що є релевантними при розробці прикладної аналітичної системи прогнозування енергоспоживання.

Таблиця 3.1 – Порівняння бібліотек машинного навчання для .NET

| Характеристика | ML.NET | Accord.NET | Infer.NET |
|--|-------------------------------|-------------------------------|----------------------------|
| Автоматична побудова конвеєра | Так | Обмежено | Відсутня |
| Обробка категоріальних ознак | Так (вбудований One-Hot) | Потрібна ручна реалізація | Через змінні розподілу |
| Інтеграція у WinForms/WPF | Повна, із підтримкою .NET 6/7 | Повна | потребує адаптації |
| Серійна інференція (PredictEngine) | Так | Частково | Не передбачено прямо |
| Підтримка збереження моделей | Так (власний формат .zip) | Частково (через серіалізацію) | Не передбачено як стандарт |
| Гнучкість математичного моделювання | Середня | Висока | Дуже висока |
| Підтримка ймовірнісних залежностей | Ні | Частково | Так |
| Автоматична побудова конвеєра (pipeline) | Так | Обмежено | Відсутня |
| Обробка категоріальних ознак | Так (вбудований One-Hot) | Потрібна ручна реалізація | Через змінні розподілу |

Вибір ML.NET для реалізації системи прогнозування енергоспоживання зумовлений поєднанням практичної зручності та технічної потужності цієї бібліотеки. Вона забезпечує повноцінну інтеграцію з .NET-інфраструктурою, що дає змогу легко вбудовувати моделі у WinForms-застосунки без потреби зовнішніх залежностей. Завдяки наявності вбудованих механізмів попередньої обробки даних, побудови трансформаційного конвеєра та оцінювання якості моделі, розробник отримує контрольований і формалізований процес машинного навчання. Окрім того, можливість збереження моделей у готовому до використання форматі й пряме використання інструменту PredictEngine дозволяє виконувати прогнозування у реальному часі без додаткових обчислювальних витрат. Усе це робить ML.NET придатним інструментом для побудови продуктивної, надійної та підтримуваної системи інтелектуального аналізу енергоспоживання.

3.2 Розробка алгоритмів прогнозування енергоспоживання

Прогнозування енергоспоживання побутових пристроїв вимагає формального опису та реалізації алгоритмів машинного навчання, що здатні враховувати низку факторів, зокрема тип пристрою, умови його експлуатації та характеристики домогосподарства. У розробленій системі було застосовано підхід, заснований на побудові регресійної моделі за допомогою бібліотеки ML.NET, яка забезпечує повний цикл машинного навчання: від завантаження даних і попередньої обробки до навчання моделі та виконання прогнозів.

Оскільки побудова моделі є центральним етапом, від якого залежить якість і практична застосовність усієї системи, розробка відповідного алгоритму здійснювалася з урахуванням вимог до надійності, інтерпретованості та зручності повторного використання. Алгоритм охоплює не лише технічні кроки тренування, а й перевірку вхідних даних, логування результатів та формування користувацького інтерфейсу взаємодії з моделлю.

У процесі побудови системи прогнозування енергоспоживання побутових пристроїв було реалізовано послідовність кроків, що відповідають загальній

| | | | | |
|-------|------|----------|--------|------|
| | | | | |
| Змін. | Арк. | № докум. | Підпис | Дата |

ІАЛЦ.045480.004 ПЗ

Арк.

49

структурі машинного навчання, зображеній на рис. 3.1. Відправною точкою є надходження вхідних даних, які підлягають підготовці шляхом очищення, усунення пропусків та аномалій, а також їхньої інтеграції та агрегування у єдиний формат. На цьому етапі формується цілісна й придатна до аналізу база, що слугує підґрунтям для подальших трансформацій.

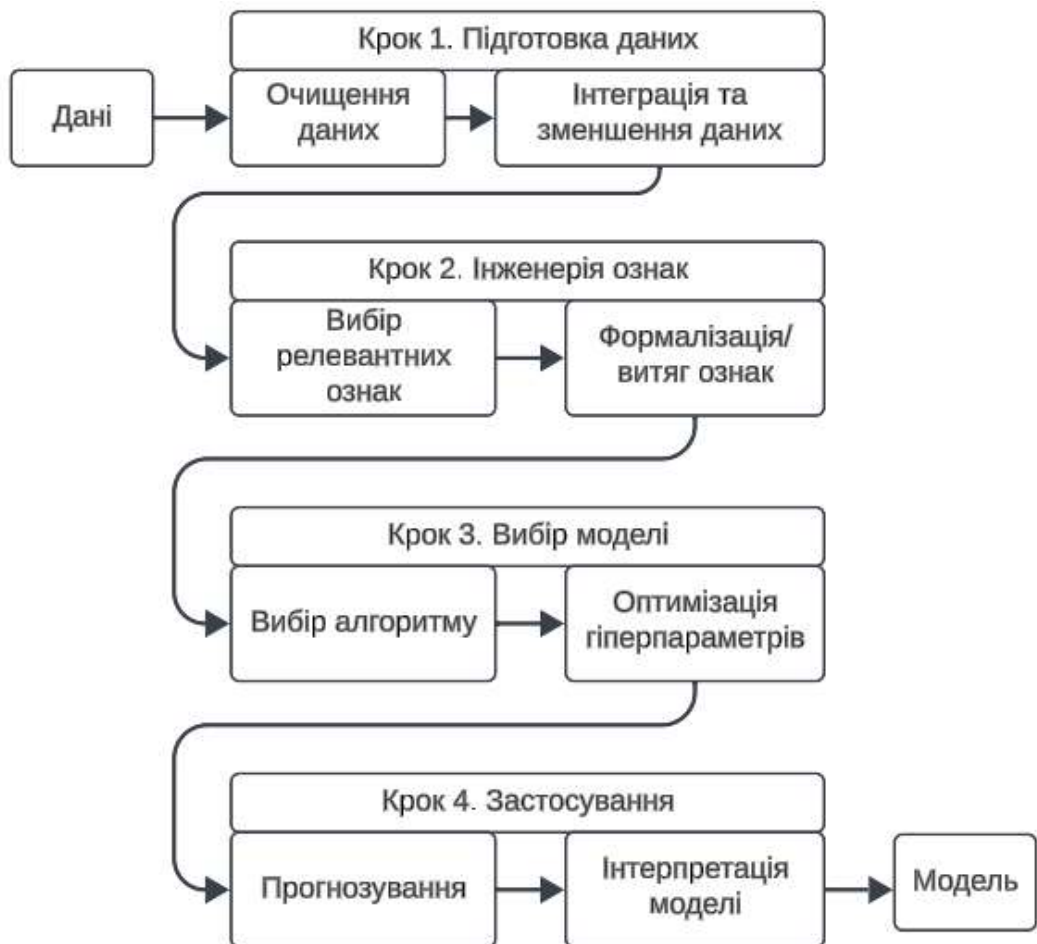


Рисунок 3.1 – Загальна схема побудови моделей МН та їх використання

Після підготовки даних здійснюється інженерія ознак, що включає відбір інформативних характеристик і їх формалізацію у формат, придатний для обробки алгоритмом. Це сприяє зменшенню розмірності та покращенню збіжності моделі. Далі відбувається побудова моделі: обирається регресійний алгоритм і налаштовуються його гіперпараметри для досягнення високої точності. Після навчання модель використовується для прогнозування на нових

| | | | | |
|-------|------|----------|--------|------|
| | | | | |
| Змін. | Арк. | № докум. | Підпис | Дата |

даних, а також для пояснення результатів, що дає змогу оцінити вплив окремих чинників на рівень споживання.

На рис. 3.2 подано блок-схему алгоритму тренування моделі машинного навчання, який реалізовано у рамках системи. Алгоритм включає послідовність дій, що виконуються у процесі створення моделі прогнозування на основі обраного набору даних.

Алгоритм починається з ініціалізації, після чого користувач здійснює вибір файлу тренувальних даних. На цьому етапі здійснюється перевірка коректності структури та вмісту даних. Якщо файл не відповідає вимогам (наприклад, відсутні обов'язкові поля, помилковий формат числових значень або некоректне кодування), система автоматично переходить до гілки помилки, де виводиться повідомлення про проблему.

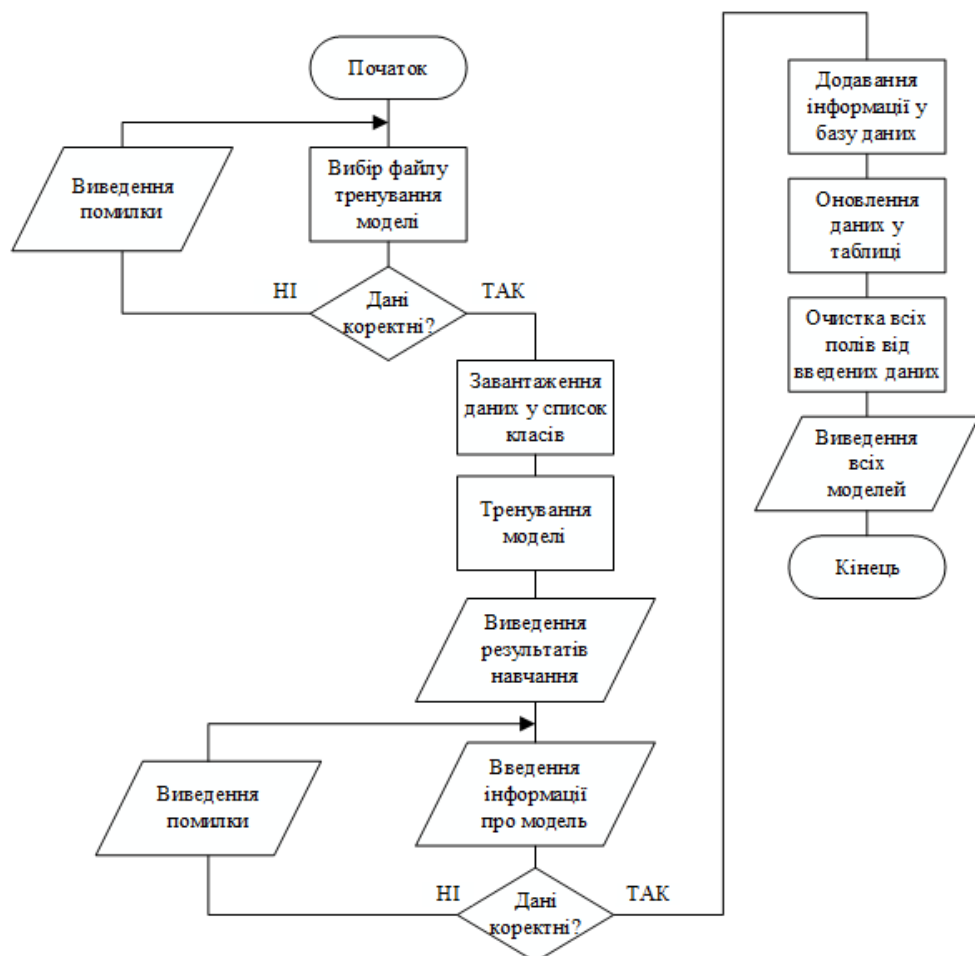


Рисунок 3.2 – Тренування моделі машинного навчання

| | | | | |
|-------|------|----------|--------|------|
| | | | | |
| Змін. | Арк. | № докум. | Підпис | Дата |

У випадку позитивної перевірки дані завантажуються до структури, з якою працює система – зокрема, у список класів, що відповідає вхідній структурі типу EnergyData. Далі відбувається процес тренування моделі з використанням методу швидкого дерева, після чого результати навчання – зокрема метрики точності, помилки та коефіцієнт детермінації – виводяться для аналізу.

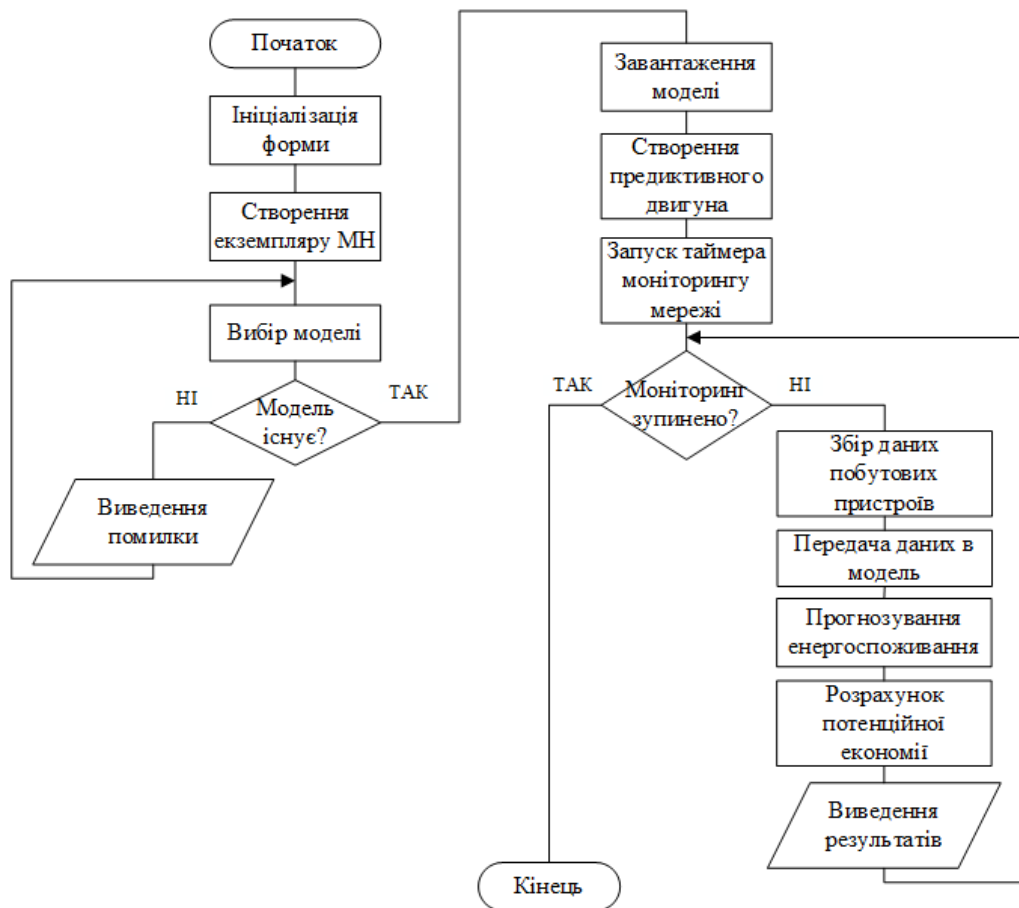


Рисунок 3.3 – Моніторинг мережі побутових пристроїв

Після цього користувачу пропонується внести інформацію про модель у базу даних, що дозволяє зберігати параметри навчання, використовувані файли та ключові характеристики моделі для подальшого використання. Як і на попередньому етапі, виконується перевірка правильності введеної інформації. У разі виявлення помилки – виконується повернення до користувача з відповідним повідомленням, і лише за умови коректного введення система переходить до завершального блоку, що включає: додавання запису в базу даних, оновлення

| | | | | |
|-------|------|----------|--------|------|
| Змін. | Арк. | № докум. | Підпис | Дата |
| | | | | |

виведених таблиць, очищення полів введення, виведення наявних моделей та завершення операції.

На рис. 3.3 зображено блок-схему алгоритму моніторингу побутових пристроїв, що забезпечує автоматичний запуск прогнозування в умовах реального часу.

Алгоритм починається з ініціалізації графічної форми, після чого створюється екземпляр системи машинного навчання і здійснюється вибір моделі, яку буде використано для подальших прогнозів. На цьому етапі відбувається перевірка наявності моделі у сховищі, і в разі її відсутності користувач отримує відповідне повідомлення про помилку. Якщо модель знайдено, система переходить до її завантаження та створення предиктивного двигуна – внутрішнього об'єкта, який відповідає за застосування моделі до вхідних даних. Після цього активується таймер моніторингу, що виконує регулярні перевірки й обчислення.

У кожному циклі перевіряється стан моніторингу. Якщо його зупинено (наприклад, вручну користувачем), алгоритм завершується. Якщо ж моніторинг активний, система переходить до збору даних з побутових пристроїв – це можуть бути як реальні сенсорні дані, так і симульовані параметри (тип пристрою, температура, сезон, кількість мешканців). Зібрані дані передаються у модель для здійснення прогнозу споживання енергії.

Окрім самого прогнозу, система додатково виконує розрахунок потенційної економії енергії, моделюючи альтернативні сценарії експлуатації (наприклад, заміну приладу на енергоощадний аналог або зміну режиму його використання). На завершальному етапі відбувається виведення результатів прогнозування у графічному інтерфейсі, що дозволяє користувачеві візуально оцінити як поточний рівень споживання, так і можливості для його оптимізації.

| | | | | |
|-------|------|----------|--------|------|
| | | | | |
| Змін. | Арк. | № докум. | Підпис | Дата |

3.3 Побудова архітектури інформаційної системи

Архітектура інформаційної системи визначає її функціональність, забезпечує розподіл відповідальностей між компонентами та підтримує масштабованість і зручність супроводу.

У рамках роботи реалізовано трьохрівневу архітектуру, що включає: рівень представлення, рівень бізнес-логіки та рівень доступу до даних [29]. Такий поділ дозволяє чітко структурувати код, зменшити зв'язність між модулями та спростити оновлення й тестування. Трирівнева модель забезпечує ізоляцію інтерфейсу від логіки обробки, дозволяє незалежно розробляти та тестувати модулі машинного навчання, а також спрощує зміну джерел або структури зберігання даних [11].

На рис. 3.4 показано діаграму класів рівня даних, яка відображає структуру компонентів для роботи з базою, збереження результатів і логування, організованих у вигляді провайдерів даних.

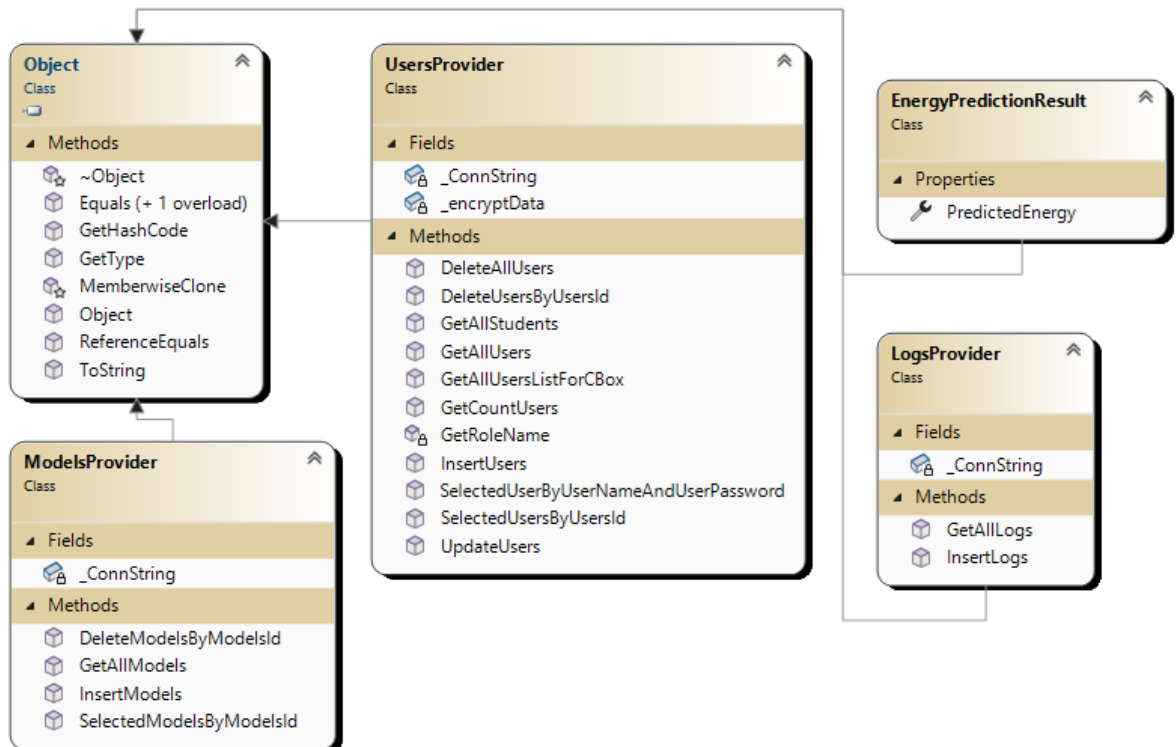


Рисунок 3.4 – Діаграма класів рівня даних

| | | | | |
|-------|------|----------|--------|------|
| Змін. | Арк. | № докум. | Підпис | Дата |
| | | | | |

Діаграма класів рівня даних складається із:

- класу `UsersProvider`, який відповідає за роботу з користувачами системи. Він містить поля для зберігання рядка підключення до БД (`_ConnectionString`) та об'єкта шифрування (`_encryptData`). Серед реалізованих методів – видалення користувачів, вибірка користувачів для списків та таблиць, авторизація, отримання ролі користувача та оновлення даних;
- класу `ModelsProvider`, який відповідає за збереження та отримання моделей машинного навчання. Його функціональність охоплює видалення моделей за ідентифікатором, отримання всіх моделей та вибірку окремих моделей. Клас також містить поле `_ConnectionString` для доступу до джерела даних;
- класу `LogsProvider`, що призначений для ведення журналу дій у системі. Він забезпечує вставку нових записів та отримання всіх логів. Зберігає поле `_ConnectionString`, яке використовується для підключення до відповідної таблиці журналів подій у БД;
- класу `EnergyPredictionResult`, який є моделлю результату прогнозування. Він має одну властивість – `PredictedEnergy`, що містить числове значення передбаченого споживання електроенергії. Клас використовується на етапі виклику методу `Predict` та виводу результатів користувачу.

Перелічені провайдери реалізують ізольований доступ до даних у межах бізнес-логіки, а також спрощують супровід і розширення системи без зміни основної архітектури. Такий підхід дозволяє дотримуватись принципів SOLID і зменшити ступінь зв'язності між компонентами системи.

У межах побудови трьохрівневої архітектури розробленої системи важливе місце займає рівень користувацького інтерфейсу, який забезпечує інтерактивну взаємодію між кінцевим користувачем та внутрішніми компонентами системи. Цей рівень реалізовано у вигляді набору Windows Forms-форм, кожна з яких виконує окрему логічну функцію – автентифікацію,

| | | | | |
|-------|------|----------|--------|------|
| | | | | |
| Змін. | Арк. | № докум. | Підпис | Дата |

ІАЛЦ.045480.004 ПЗ

Арк.
55

управління користувачами, запуск прогнозування, перегляд журналів тощо. Завдяки чіткому розмежуванню відповідальностей між формами вдалося досягти зручної, структурованої та масштабованої взаємодії у межах графічного інтерфейсу системи.

На рис. 3.5 зображено діаграму класів рівня користувацького інтерфейсу, яка ілюструє основні форми, їхні методи та взаємозв'язки. Кожен клас розширює базовий клас Form, успадковуючи необхідні властивості та події.

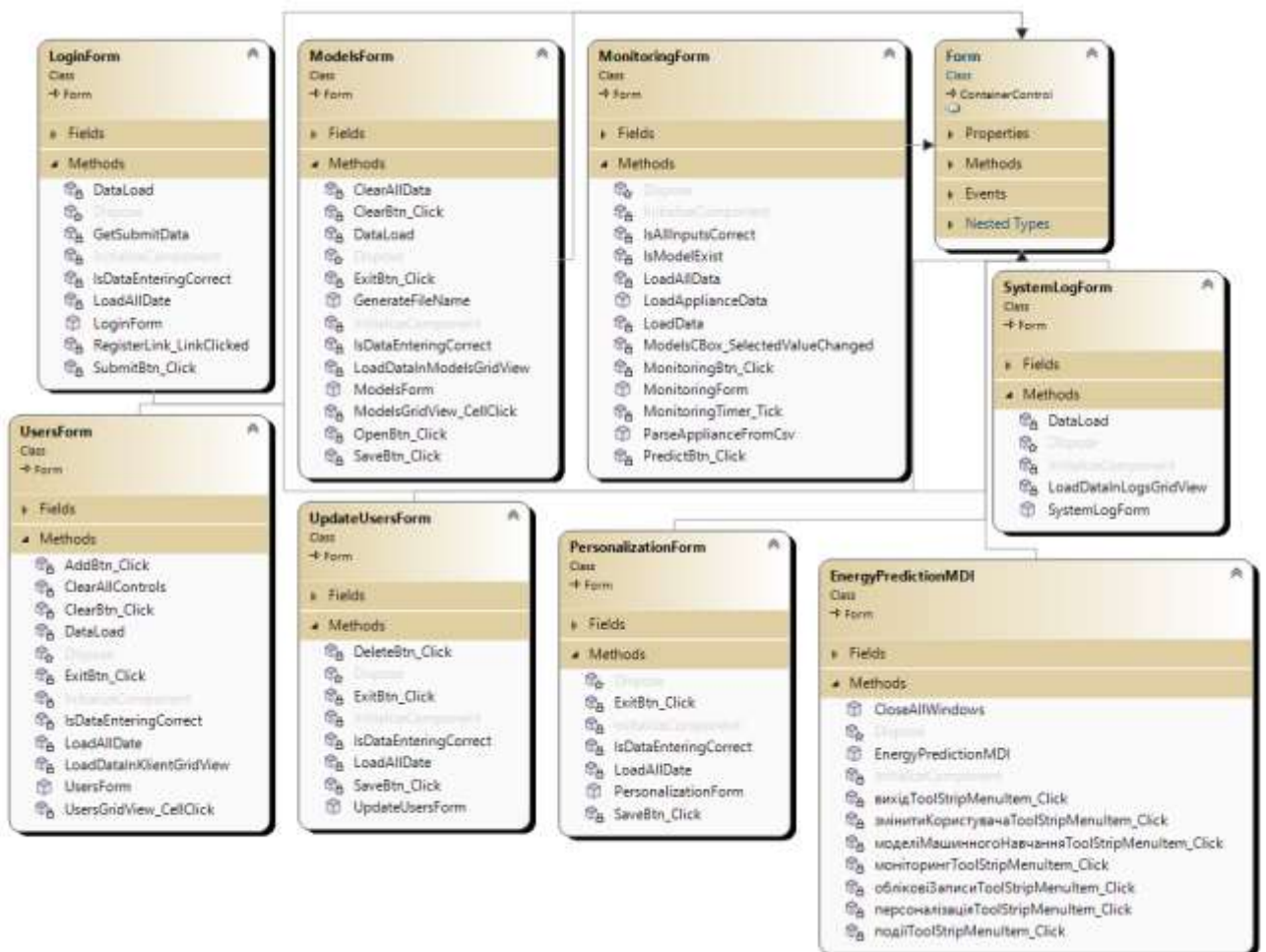


Рисунок 3.5 – Діаграма класів рівня інтерфейсу

Діаграма класів рівня інтерфейсу складається із:

- класу LoginForm, який відповідає за автентифікацію користувача. Містить методи для ініціалізації компонентів, обробки натискання

кнопки підтвердження, перевірки введених даних і переходу до реєстрації;

- класу UsersForm, який реалізує інтерфейс для перегляду, додавання та редагування користувачів. Він містить логіку обробки кнопок додавання, очищення, перевірки вхідних даних та завантаження інформації у таблицю користувачів;
- класу UpdateUsersForm, призначеного для редагування вже існуючих облікових записів користувачів. Він містить функціональність збереження, видалення, перевірки введених даних та завантаження інформації у поля форми;
- класу ModelsForm, який забезпечує взаємодію з моделями машинного навчання: завантаження, перегляд, збереження, очищення полів, генерацію імені моделі. Важливим є метод OpenBtn_Click, що ініціює процес тренування моделі;
- класу MonitoringForm, який виконує функції моніторингу енергоспоживання. Містить методи перевірки коректності введення, обробки натискання кнопки запуску прогнозування, запуску таймера моніторингу та обробки події таймера;
- класу SystemLogForm, що призначений для перегляду журналів подій. Реалізує завантаження логів у табличне подання через метод LoadDataInLogsGridView;
- класу PersonalizationForm, який реалізує функції персоналізації користувача. Забезпечує оновлення даних, збереження введених параметрів та перевірку їх коректності;
- класу EnergyPredictionMDI, який виступає як головна MDI-форма системи. Вона містить обробники для відкриття дочірніх вікон (моніторингу, журналів, моделей, редагування користувача тощо) через елементи меню, а також метод CloseAllWindows, що закриває всі дочірні форми.

| | | | | |
|-------|------|----------|--------|------|
| | | | | |
| Змін. | Арк. | № докум. | Підпис | Дата |

ІАЛЦ.045480.004 ПЗ

Арк.

57

Реалізований рівень користувацького інтерфейсу повністю охоплює функціональні потреби системи – від автентифікації та керування користувачами до виконання прогнозування й перегляду логів. Завдяки використанню ізольованих форм, кожна з яких виконує чітко визначене завдання, досягається висока структурованість, а поділ за функціональними ролями дозволяє легко масштабувати або змінювати інтерфейс без ризику порушення основної логіки системи. Такий підхід узгоджується з принципами трьохрівневої архітектури, де представлення даних реалізується як окремий самодостатній шар.

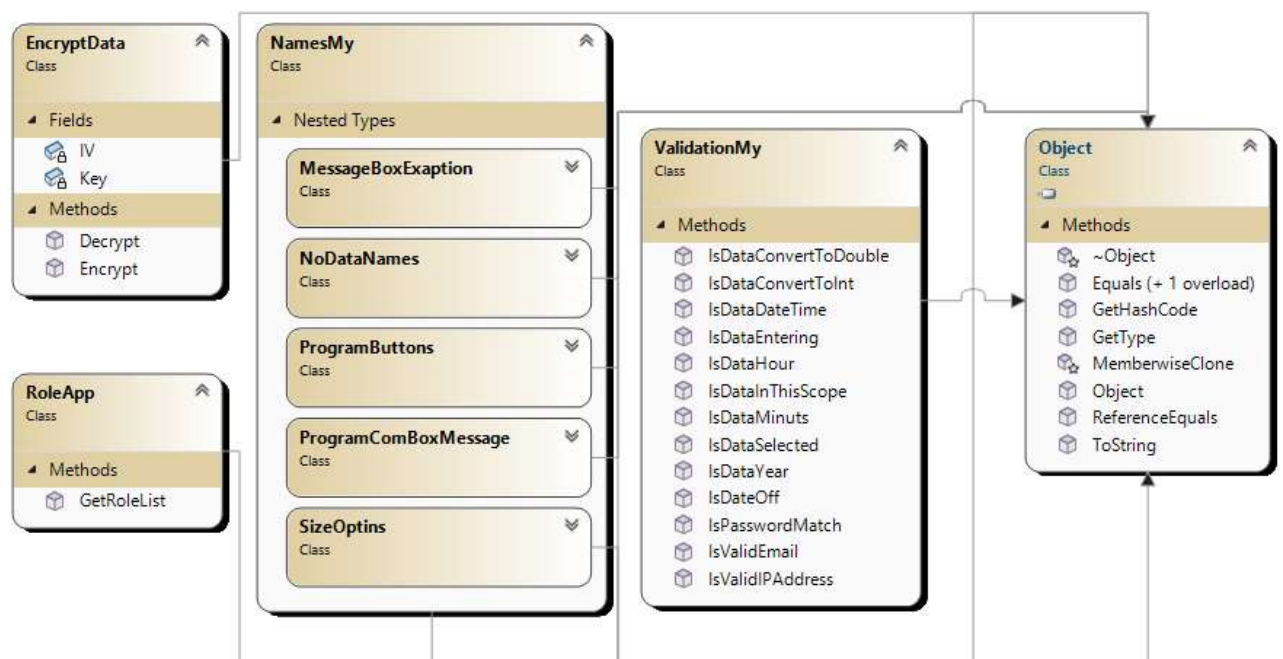


Рисунок 3.6 – Діаграма класів рівня бізнес-логіки

Рівень бізнес-логіки виконує роль проміжної ланки між інтерфейсом користувача та доступом до даних. Саме тут реалізуються ключові обчислення, перевірки, шифрування, валідація, а також формування службових повідомлень і керування логікою відповідей на дії користувача. Такий підхід забезпечує концентрацію усіх функціонально значущих обробок у єдиному шарі, що спрощує супровід, повторне використання компонентів і зменшує дублювання коду на рівні форм.

| | | | | |
|-------|------|----------|--------|------|
| | | | | |
| Змін. | Арк. | № докум. | Підпис | Дата |

На рис. 3.6 зображено діаграму класів рівня бізнес-логіки, яка демонструє набір службових класів, реалізованих у системі для підтримки ключових функціональних операцій. Кожен із представлених класів виконує свою окрему роль у логіці застосунку, не звертаючись безпосередньо до користувача чи до бази даних, а працюючи як незалежна логічна одиниця.

Діаграма класів рівня бізнес-логіки складається із:

- класу `EncryptData`, який забезпечує базову функціональність для шифрування та розшифрування даних. Він містить поля `IV` (ініціалізаційний вектор) та `Key` (ключ шифрування), що використовуються в алгоритмах захисту інформації. Реалізовані методи `Encrypt` і `Decrypt` дають змогу зашифрувати та дешифрувати текстові дані, наприклад, паролі чи інші конфіденційні параметри, що зберігаються в базі;
- класу `ValidationMy`, що виконує функції валідації введених користувачем даних. Він містить набір методів для перевірки типів, логічної коректності введення та інших умов. Цей клас активно використовується в інтерфейсному шарі для перевірки форм перед обробкою чи передачею інформації;
- класу `NamesMy`, який об'єднує вкладені типи із системними іменами, повідомленнями та назвами елементів. Зокрема, він містить внутрішні класи тощо. Цей клас слугує як централізоване сховище фіксованих ідентифікаторів, назв, стандартних повідомлень або текстових шаблонів, що використовуються у формах та повідомленнях;
- класу `RoleApp`, який забезпечує доступ до ролей користувачів. Його метод `GetRoleList` дозволяє отримати перелік доступних ролей для реєстрації чи авторизації, що є важливою частиною авторизаційної логіки застосунку.

Представлений рівень бізнес-логіки системи зосереджує в собі найважливіші перевірки, шифрування, структури повідомлень та логіку обробки даних, що дозволяє досягти високої рівня інкапсуляції функціональності. Класи

| | | | | |
|-------|------|----------|--------|------|
| | | | | |
| Змін. | Арк. | № докум. | Підпис | Дата |

ІАЛЦ.045480.004 ПЗ

Арк.

59

не залежать від інтерфейсного або дата-шару, що відповідає принципам слабого зв'язку і забезпечує повторне використання компонентів у різних частинах системи. Завдяки такій структурі забезпечується гнучкість, прозорість і логічна цілісність бізнес-операцій, що виконуються в системі.

Висновки до розділу

У рамках даного розділу було здійснено цілеспрямований аналіз інструментальних засобів машинного навчання, розроблено ключові алгоритми прогнозування та сформовано архітектурну модель інформаційної системи. На основі порівняльної оцінки бібліотек ML.NET, Accord.NET та Infer.NET було обґрунтовано доцільність використання саме ML.NET як стабільного інструменту для інтеграції алгоритмів регресійного прогнозування в середовище WinForms-застосунку. У ході розробки алгоритмів реалізовано дві основні процедури: тренування моделі на основі вхідних даних та періодичне прогнозування в режимі моніторингу побутових пристроїв із подальшим обчисленням потенційної енергоефективності. З метою забезпечення структурованості системи та можливості її масштабування побудовано трьохрівневу архітектуру, що охоплює рівні представлення, бізнес-логіки та доступу до даних. Для кожного з рівнів розроблено відповідну діаграму класів, здійснено їх функціональне описання, що дозволяє забезпечити чітке розмежування відповідальностей між компонентами системи. Отримані результати є основою для переходу до наступного етапу, що включає безпосередню реалізацію інформаційної системи, програмну інтеграцію компонентів, а також аналіз точності та ефективності прогнозів.

| | | | | |
|-------|------|----------|--------|------|
| | | | | |
| Змін. | Арк. | № докум. | Підпис | Дата |

ІАЛЦ.045480.004 ПЗ

Арк.
60

4 ВИБІР БІБЛІОТЕКИ ТА ПРОЕКТУВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ

4.1 Реалізація моделі машинного навчання у програмному середовищі

Розробка практичної моделі машинного навчання для прогнозування енергоспоживання в побутових умовах передбачає поетапну реалізацію процесу побудови, навчання та використання моделі у вибраному програмному середовищі. У межах даного проєкту було використано платформу ML.NET – фреймворк для побудови моделей машинного навчання у середовищі C#, який дозволяє ефективно інтегрувати інтелектуальні компоненти безпосередньо в структуру настільного застосунку. Враховуючи особливості задачі, що потребує прогнозування кількісного значення на основі кількох параметрів, реалізована модель виконує регресійний аналіз на основі методу швидкого дерева.

Першим кроком у побудові такої моделі є визначення вхідного та вихідного класів, які описують структуру даних, що надходять до моделі та повертаються з неї після прогнозування. На рис. 4.1 наведено структуру цих класів, яка демонструє відповідність між обраними параметрами та прогнозованим значенням.

```
public class EnergyData {
    15 references
    [LoadColumn(1)] public string ApplianceType { get; set; }
    7 references
    [LoadColumn(2)] public float EnergyConsumption { get; set; }
    14 references
    [LoadColumn(5)] public float Temperature { get; set; }
    14 references
    [LoadColumn(6)] public string Season { get; set; }
    14 references
    [LoadColumn(7)] public float HouseholdSize { get; set; }
}

// Клас результату прогнозу
4 references
public class EnergyPredictionResult {
    [ColumnName("Score")]
    13 references
    public float PredictedEnergy { get; set; }
}
```

Рисунок 4.1 – Вхідний та вихідний класи моделі машинного навчання

| | | | | |
|-------|------|----------|--------|------|
| | | | | |
| Змін. | Арк. | № докум. | Підпис | Дата |

У вхідному класі EnergyData описуються всі ознаки, що використовуються для навчання моделі. До них належать: тип пристрою, що представлений у вигляді текстового значення; температура довкілля як числова безперервна змінна; сезон – як категоріальна характеристика, що відображає річний період використання; кількість осіб у домогосподарстві; та основна цільова змінна – фактичне значення енергоспоживання приладу. Для кожної з ознак за допомогою атрибута вказано індекс стовпця у CSV-файлі, з якого буде зчитано відповідні значення. Вихідний клас EnergyPredictionResult містить єдине поле PredictedEnergy, яке представляє числовий результат роботи регресійної моделі. Значення передається з виходу моделі за допомогою атрибута «Score», що є стандартною нотацією в ML.NET для відображення прогнозованого значення.

Поданий на рис. 4.2 фрагмент коду реалізує обробку події натискання на кнопку відкриття файлу у графічному інтерфейсі застосунку.

```
private void OpenBtn_Click(object sender, EventArgs e) {  
    // Створення діалогового вікна для відкриття файлу  
    OpenFileDialog openFileDialog = new OpenFileDialog {  
        Filter = "Text files (*.csv)|*.csv|All files (*.*)|*.*",  
        FilterIndex = 2,  
        RestoreDirectory = true  
    };  
  
    if (openFileDialog.ShowDialog() == DialogResult.OK) {  
        _Path = openFileDialog.FileName;  
        FileNameTextBox.Text = openFileDialog.FileName;  
    }  
}
```

Рисунок 4.2 – Обробка події натискання на кнопку відкриття файлу

При активації події створюється стандартне діалогове вікно для вибору файлу з файлової системи користувача. Діалог обмежений фільтром, що передбачає відображення лише CSV-файлів або всіх типів файлів, при цьому формат CSV задається як пріоритетний. Якщо користувач підтверджує вибір, обраний шлях до файлу зберігається у змінну _Path, а також відображається в текстовому полі FileNameTextBox, що дозволяє користувачу візуально переконатись у правильності завантаженого джерела. Така реалізація забезпечує

| | | | | |
|-------|------|----------|--------|------|
| | | | | |
| Змін. | Арк. | № докум. | Підпис | Дата |

зручне підключення вхідних даних до моделі машинного навчання без потреби вручну вводити шлях до файлу.

У фрагменті коду на рис. 4.3 здійснюється ініціалізація середовища машинного навчання шляхом створення об'єкта `MLContext`, який виступає як головний контейнер для усіх операцій, пов'язаних із побудовою, тренуванням і використанням моделей у `ML.NET`. Для забезпечення відтворюваності результатів задається фіксоване значення початкового зерна генератора випадкових чисел – `seed: 1`. Одразу після цього розпочинається засікання часу виконання операцій за допомогою об'єкта `Stopwatch`, що дозволяє у подальшому виміряти тривалість процесу навчання або прогнозування моделі. Такий підхід є корисним для оцінки продуктивності системи.

```
// Ініціалізація MLContext
mlContext = new MLContext(seed: 1);
// Засікання часу
Stopwatch stopwatch = new Stopwatch();
stopwatch.Start();
```

Рисунок 4.3 – Ініціалізація середовища машинного навчання

На рис. 4.4 приведено код завантаження вхідних даних з CSV-файлу у форматі, що відповідає структурі класу `EnergyData`. Для цього використовується метод `LoadFromTextFile`, який зчитує дані з файлу за шляхом, заданим змінною `_Path`, враховуючи наявність заголовку та використовуючи кому як роздільник.

```
_fullData = mlContext.Data.LoadFromTextFile<EnergyData>(
    path: _Path,
    hasHeader: true,
    separatorChar: ',');
ReportTextBox.Text = "Завантаження даних\r\n";
Application.DoEvents();
// Загальна статистика
var dataEnumerable =
    mlContext.Data.CreateEnumerable<EnergyData>(_fullData, reuseRowObject: false).ToList();
int totalCount = dataEnumerable.Count;
float avgEnergy = dataEnumerable.Average(d => d.EnergyConsumption);
float minEnergy = dataEnumerable.Min(d => d.EnergyConsumption);
float maxEnergy = dataEnumerable.Max(d => d.EnergyConsumption);
ReportTextBox.Text += ("Статистика вхідних даних:\r\n");
ReportTextBox.Text += (" - Загальна кількість взірців: {totalCount}\r\n");
ReportTextBox.Text += (" - Середнє значення споживання енергії: {avgEnergy:F2} кВт·г\r\n");
ReportTextBox.Text += (" - Мінімальне значення: {minEnergy:F2} кВт·г\r\n");
ReportTextBox.Text += (" - Максимальне значення: {maxEnergy:F2} кВт·г\r\n");
```

Рисунок 4.4 – Завантаження вхідних даних та формування статистики

| | | | | | | |
|-------|------|----------|--------|------|--------------------|------|
| | | | | | ІАЛЦ.045480.004 ПЗ | Арк. |
| Змін. | Арк. | № докум. | Підпис | Дата | | 63 |

Після успішного завантаження у текстовому полі ReportTVox виводиться повідомлення про початок обробки. Далі здійснюється перетворення всієї вибірки в перелік об'єктів для обчислення загальної статистики: кількість записів, середнє, мінімальне та максимальне значення енергоспоживання. Результати статистичного аналізу формуються та виводяться у вигляді текстового звіту, що дозволяє користувачу одразу оцінити базові характеристики вибірки перед тренуванням моделі. Це також дає змогу виявити потенційні аномалії або некоректно зчитані дані на ранньому етапі роботи програми.

У наведеному на рис. 4.5 фрагменті коду реалізовано два ключові етапи підготовки даних до навчання моделі: розділення вибірки та побудову трансформацій. Спочатку вхідна вибірка `_fullData` розбивається на навчальний та тестовий піднабори у співвідношенні 80/20 за допомогою методу `TrainTestSplit`, що дозволяє оцінювати якість моделі на незалежних даних. Навчальна частина зберігається у змінній `trainData`, а тестова – у `testData`.

```
var trainTestSplit =  
  mlContext.Data.TrainTestSplit(_fullData, testFraction: 0.2);  
var trainData = trainTestSplit.TrainSet;  
var testData = trainTestSplit.TestSet;  
  
// 3. Побудова трансформацій  
var dataPipeline = mlContext.Transforms.CopyColumns("Label", "EnergyConsumption")  
  .Append(mlContext.Transforms.Categorical.OneHotEncoding("ApplianceTypeEncoded", "ApplianceType"))  
  .Append(mlContext.Transforms.Categorical.OneHotEncoding("SeasonEncoded", "Season"))  
  .Append(mlContext.Transforms.Concatenate("Features", "ApplianceTypeEncoded",  
    "Temperature", "SeasonEncoded", "HouseholdSize"));
```

Рисунок 4.5 – Розділення даних та побудова трансформацій

Після цього формується ланцюжок трансформацій, який буде застосовано до даних перед тренуванням моделі. На першому кроці колонка `EnergyConsumption` копіюється до нової колонки `Label`, яка виступає цільовою змінною для регресійної моделі. Далі виконуються операції перетворення категоріальних змінних: тип пристрою (`ApplianceType`) і сезон (`Season`) кодуються у вигляді векторів з допомогою `one-hot` кодування, результати зберігаються у нових ознаках `ApplianceTypeEncoded` та `SeasonEncoded`. На завершальному етапі всі ознаки об'єднуються в єдиний вектор ознак під назвою

Features, що включає як закодовані категоріальні, так і числові змінні – температуру та розмір домогосподарства. Такий підхід забезпечує приведення всіх вхідних параметрів до єдиного узгодженого формату, придатного для машинного навчання.

На рис. 4.6 наведено код створення та навчання моделі машинного навчання. На основі раніше побудованого ланцюжка трансформацій даних `dataPipeline` додається регресійний метод `FastTree`, що використовується як фінальний навчальний компонент. Об'єкт `trainer` створюється за замовчуванням із використанням типових параметрів, після чого об'єднується з пайплайном трансформацій у повноцінну модельний ланцюг.

```
var trainer = mlContext.Regression.Trainers.FastTree();
var pipeline = dataPipeline.Append(trainer);
Console.WriteLine("\nНавчання моделі швидкого дерева...\r\n");
model = pipeline.Fit(trainData);
stopwatch.Stop(); // Зупинка таймера

// Виведення часу навчання
ReportTextBox.Text += ($"\nЧас навчання моделі:" +
    $" {stopwatch.Elapsed.TotalSeconds:F2} секунд\r\n");
```

Рисунок 4.6 – Створення і навчання моделі

Процес тренування запускається за допомогою методу `Fit`, де як вхідні дані використовується навчальна вибірка `trainData`. Після завершення навчання відбувається зупинка таймера, запущеного на початковому етапі, і виводиться тривалість навчання у секундах у текстовому полі `ReportTextBox`. Це дозволяє оцінити обчислювальну ефективність моделі, що є важливим критерієм при її подальшому використанні в інтерактивному застосунку.

Фрагмент коду на рис. 4.7 відповідає за етап оцінювання якості натренованої моделі на незалежній тестовій вибірці.

| | | | | |
|-------|------|----------|--------|------|
| | | | | |
| Змін. | Арк. | № докум. | Підпис | Дата |

```

var predictions = model.Transform(testData);
var metrics =
    mlContext.Regression.Evaluate(predictions, labelColumnName: "EnergyConsumption");

ReportTextBox.Text += ("\nМетрики моделі (тестова вибірка):\r\n");
ReportTextBox.Text += ($" - MAE:    {metrics.MeanAbsoluteError:F4}\r\n");
ReportTextBox.Text += ($" - MSE:    {metrics.MeanSquaredError:F4}\r\n");
ReportTextBox.Text += ($" - RMSE:   {metrics.RootMeanSquaredError:F4}\r\n");
ReportTextBox.Text += ($" - R²:     {metrics.RSquared:F4}\r\n");

```

Рисунок 4.7 – Оцінка моделі на тестовій вибірці та виведення результату

На початку за допомогою методу Transform виконується застосування моделі до тестових даних, у результаті чого формується набір прогнозованих значень. Отримані передбачення передаються до функції Evaluate, яка обчислює стандартні регресійні метрики точності: середню абсолютну похибку (MAE), середню квадратичну похибку (MSE) та її корінь (RMSE).

У фрагменті коду на рис. 4.8 реалізується завершальний етап перевірки роботи моделі, який поєднує аналіз точності прогнозів на тестовій вибірці з демонстрацією індивідуальних передбачень для нових прикладів.

```

var actualData =
    mlContext.Data.CreateEnumerable<EnergyData>(testData, reuseRowObject: false).ToList();
var predictedData =
    mlContext.Data.CreateEnumerable<EnergyPredictionResult>(predictions,
        reuseRowObject: false).ToList();
// 8. Прогнозування на тестових прикладах вручну
var predEngine = mlContext.Model.CreatePredictionEngine<EnergyData,
    EnergyPredictionResult>(model);
var samples = new[] {
    new EnergyData { ApplianceType = "Fridge", Temperature = 5,
        Season = "Winter", HouseholdSize = 2 },
    new EnergyData { ApplianceType = "AirConditioning", Temperature = 38,
        Season = "Summer", HouseholdSize = 3 },
    new EnergyData { ApplianceType = "Microwave", Temperature = 15,
        Season = "Spring", HouseholdSize = 4 }
};
ReportTextBox.Text += ("\r\nПрогнози на контрольних прикладах:\r\n");
foreach (var sample in samples) {
    var prediction = predEngine.Predict(sample);
    ReportTextBox.Text += ($" {sample.ApplianceType}, {sample.Season}, " +
        $" {sample.Temperature}°C, {sample.HouseholdSize} " +
        $"ociб => {prediction.PredictedEnergy:F2} кВт·г\r\n");
}

```

Рисунок 4.8 – Аналіз точності прогнозів на тестовій вибірці та вивід результату

На початку з тестової вибірки видобуваються реальні значення (actualData) та відповідні передбачення моделі (predictedData), що дозволяє здійснити порівняння фактичного та прогнозованого споживання для подальшого аналізу похибок. Хоча класичної матриці помилок у регресії не існує, ця операція умовно виконує її функцію, надаючи змогу оцінити точність передбачень для кожного конкретного прикладу.

Далі створюється механізм індивідуального прогнозування – об'єкт PredictionEngine, що дає змогу виконувати передбачення для окремих записів у режимі реального часу. Формується масив нових контрольних прикладів, які імітують типові сценарії використання побутових приладів за різних сезонних і температурних умов. Для кожного такого прикладу обчислюється прогноз енергоспоживання, після чого результати виводяться у текстовому полі звіту. Це дозволяє не лише оцінити якість моделі на окремих прикладах, а й продемонструвати її здатність до адаптації до нових вхідних даних. Такий підхід є особливо корисним для візуального підтвердження працездатності системи та її потенціалу до інтеграції в практичне середовище користувача.

У методі на рис. 4.9 реалізується процес збереження натренованої моделі машинного навчання у файл для подальшого повторного використання без потреби в повторному навчанні. Перед виконанням операції здійснюється перевірка правильності введених користувачем даних за допомогою методу IsDataEnteringCorrect. У разі позитивного результату формується шлях до майбутнього файлу, ім'я якого генерується динамічно та додається до спеціального підкаталогу \teach\ у локальному каталозі застосунку. Далі в базу даних заноситься запис про нову модель через відповідний провайдер _ModelsProvider, зберігається сама модель у форматі .zip із використанням методу Save, де також передається схема даних.

| | | | | |
|-------|------|----------|--------|------|
| | | | | |
| Змін. | Арк. | № докум. | Підпис | Дата |

```

private void SaveBtn_Click(object sender, EventArgs e) {
    if (IsDataEnteringCorrect()) {
        //Зберігання моделі
        string pathName = @"\\teach\" + GenerateFileName() + ".zip";
        string localProj =
            System.IO.Path.GetDirectoryName(System.Reflection.Assembly.GetExecutingAssembly().Location);
        _ModelsProvider.InsertModels(ModelsNamesTBox.Text, pathName);
        mlContext.Model.Save(model, _fullData.Schema, localProj + pathName);
        ClearAllData();
        _LogsProvider.InsertLogs(LoginForm.CurrentUser.UsersId,
            "Було навчено модель " +
            ModelsNamesTBox.Text, DateTime.Now);
        MessageBox.Show("Дані успішно збережено!");
    }
}
}

```

Рисунок 4.9 – Процес збереження натренованої моделі

Після успішного збереження викликається метод очищення інтерфейсу `ClearAllData`, а також логування дій користувача через `_LogsProvider`, що дозволяє фіксувати факт створення та збереження нової моделі з прив'язкою до поточного користувача та часу. На завершення відображається повідомлення про успішне завершення операції, що слугує зворотним зв'язком для користувача й підтверджує, що модель готова до подальшого використання.

Метод `LoadAllData` відповідає за початкове завантаження всіх необхідних даних та ініціалізацію елементів інтерфейсу при відкритті форми або зміні контексту роботи (рис. 4.10).

```

private void LoadAllData() {
    ApplianceTypeCBox.SelectedIndex = 0;
    SeasonCBox.SelectedIndex = 0;
    _ModelsList = _ModelsProvider.GetAllModels();
    ModelsCBox.DataSource = _ModelsList;
    ModelsCBox.ValueMember = "ModelsId";
    ModelsCBox.DisplayMember = "ModelsName";
    _IsModelsLoad = true;
    _ApplianceData = LoadApplianceData(filePath);
    ModelsCBox_SelectedValueChanged(ModelsCBox, EventArgs.Empty);
}
}

```

Рисунок 4.10 – Початкове завантаження всіх необхідних даних

У першу чергу встановлюються стандартні значення для комбінованих списків вибору типу пристрою та сезону – обидва отримують нульовий індекс,

| | | | | |
|-------|------|----------|--------|------|
| | | | | |
| Змін. | Арк. | № докум. | Підпис | Дата |

тобто перший елемент у списку. Далі здійснюється отримання повного списку наявних моделей машинного навчання з бази даних за допомогою `_ModelsProvider`, після чого цей список прив'язується до випадального списку `ModelsCBox`.

Встановлюються ключові властивості відображення: `ValueMember` відповідає за унікальний ідентифікатор моделі, а `DisplayMember` – за її назву, яка показується користувачу. Змінна `_IsModelsLoad` активується для сигналізації про завершення процесу завантаження, що може використовуватись для контролю подальших подій. Окремо виконується завантаження додаткових даних щодо побутових пристроїв із вказаного файлу, після чого вручну викликається метод обробки події зміни вибраної моделі `ModelsCBox_SelectedValueChanged` для відображення інформації про поточну модель одразу після завантаження форми. Така логіка забезпечує повну готовність інтерфейсу до взаємодії без необхідності додаткових дій з боку користувача.

Метод `LoadData` реалізує завантаження раніше збереженої моделі машинного навчання з локального шляху, який формується на основі кореневого каталогу застосунку та переданого відносного шляху `filePath` (рис. 4.11).

```
private void LoadData(string filePath) {
    string localPath = Application.StartupPath + filePath;
    DataViewSchema modelSchema;
    try {
        // Завантаження моделі
        ITransformer model =
            mlContext.Model.Load(localPath, out modelSchema);
        predictionEngine =
            mlContext.Model.CreatePredictionEngine<EnergyData,
                EnergyPredictionResult>(model);
    } catch (Exception ex) {
        MessageBox.Show("Помилка при завантаженні моделі: " + ex.Message,
            "Помилка", MessageBoxButton.OK, MessageBoxIcon.Error);
    }
}
```

Рисунок 4.11 – Завантаження раніше збереженої моделі машинного навчання

Після формування повного шляху виконується спроба зчитування моделі з файлу за допомогою засобів ML.NET. У результаті завантаження створюється об'єкт `ITransformer`, який містить збережену модель, та витягується її схема даних `modelSchema`.

Після цього на основі завантаженої моделі створюється механізм прогнозування `PredictionEngine`, який дозволяє виконувати передбачення для окремих об'єктів класу `EnergyData` та повертати результат у форматі `EnergyPredictionResult`. У разі виникнення будь-якої помилки під час завантаження, зокрема при відсутності файлу, його пошкодженні або несумісності з даними, виводиться відповідне повідомлення у вікні з попередженням. Це забезпечує контрольований обробник виключень та покращує стабільність взаємодії з користувачем.

Метод `PredictBtn_Click` виконує обробку події натискання кнопки прогнозування у графічному інтерфейсі (рис. 4.12). На початку здійснюється перевірка коректності введених даних за допомогою методу `IsAllInputsCorrect`, а також перевіряється наявність завантаженої моделі через `IsModelExist`. Якщо обидві умови виконуються, відбувається формування об'єкта `EnergyData`, який слугує вхідними даними для моделі.

```
private void PredictBtn_Click(object sender, EventArgs e) {  
    // Перевірка коректності введення  
    if (IsAllInputsCorrect() && IsModelExist()) {  
        // Формування об'єкта з введених даних  
        EnergyData testApplianceData = new EnergyData {  
            ApplianceType = ApplianceTypeCBox.Text.Trim(),  
            Temperature = float.Parse(TemperatureTBox.Text),  
            Season = SeasonCBox.Text.Trim(),  
            HouseholdSize = float.Parse(HouseholdSizeTBox.Text)  
        };  
    }  
};
```

Рисунок 4.12 – Обробка події натискання кнопки прогнозування

Значення параметрів, таких як тип пристрою, температура, сезон та кількість осіб у домогосподарстві, зчитуються з відповідних елементів

| | | | | |
|-------|------|----------|--------|------|
| | | | | |
| Змін. | Арк. | № докум. | Підпис | Дата |

інтерфейсу, очищуються від зайвих пробілів, а числові значення парсяться у формат float.

У фрагменті коду на рис. 4.13 реалізується виведення введених користувачем даних та прогнозування енергоспоживання.

```
var inputInfo = new StringBuilder();
inputInfo.AppendLine("Введені дані для прогнозу:");
inputInfo.AppendLine($"Тип приладу: {testApplianceData.ApplianceType}");
inputInfo.AppendLine($"Температура: {testApplianceData.Temperature} °C");
inputInfo.AppendLine($"Сезон: {testApplianceData.Season}");
inputInfo.AppendLine($"Розмір домогосподарства: {testApplianceData.HouseholdSize}");
// Основний прогноз
var predictionBase = predictionEngine.Predict(testApplianceData);
```

Рисунок 4.13 – Виведення даних та прогнозування енергоспоживання

Для цього створюється об'єкт `StringBuilder`, у який послідовно додаються текстові рядки з інформацією про параметри приладу: його тип, температура доквілля, сезон використання та кількість осіб у домогосподарстві. Такий підхід дозволяє сформувати зручне форматване представлення вхідних даних, що може бути використане у звіті або для перевірки правильності введення. Після цього створений об'єкт `testApplianceData` передається до механізму прогнозування `predictionEngine`, який виконує обчислення передбаченого значення споживання електроенергії на основі натренованої моделі. Результат прогнозу зберігається у змінній `predictionBase` для подальшого виведення або аналізу.

Фрагмент коду на рис. 4.14 реалізує розрахунок потенційної економії енергії при переході від поточних параметрів використання побутового приладу до певного оптимізованого сценарію.

```
var predictionOpt =
    predictionEngine.Predict(optimizedData);
// Обчислення економії
float deltaE = predictionBase.PredictedEnergy -
    predictionOpt.PredictedEnergy;
float eta = (predictionBase.PredictedEnergy > 0) ?
    (deltaE / predictionBase.PredictedEnergy) * 100.0f : 0.0f;
```

Рисунок 4.14 – Розрахунок потенційної економії енергії

На початку здійснюється прогнозування споживання енергії для альтернативного варіанту параметрів, що зберігаються в об'єкті `optimizedData`, із використанням тієї ж самої моделі машинного навчання. Результат передбачення зберігається у змінній `predictionOpt`. Далі обчислюється абсолютна різниця між базовим прогнозом споживання та оптимізованим варіантом, що відображає зменшення обсягу енергії при зміні умов експлуатації. Після цього визначається відносна економія у відсотках шляхом ділення цієї різниці на базове значення споживання.

У фрагменті коду на рис. 4.15 реалізується формування структурованого текстового звіту про результати прогнозування та оцінку потенційної економії енергії.

```
var predictionInfo = new StringBuilder();
predictionInfo.AppendLine("\r\n--- Результат прогнозування ---");
predictionInfo.AppendLine($"Прогнозоване споживання: " +
    $"{predictionBase.PredictedEnergy:F2} кВт·г");
predictionInfo.AppendLine("\r\n--- Оптимізаційний сценарій ---");
predictionInfo.AppendLine($"Прилад: {optimizedData.ApplianceType}");
predictionInfo.AppendLine($"Прогнозоване споживання: " +
    $"{predictionOpt.PredictedEnergy:F2} кВт·г");
predictionInfo.AppendLine("\r\n--- Потенційна економія ---");
predictionInfo.AppendLine($"Абсолютна економія: {deltaE:F2} кВт·г");
predictionInfo.AppendLine($"Відносна економія: {eta:F2} %");
```

Рисунок 4.15 – Формування звіту про результати прогнозування та оцінка потенційної економії енергії

За допомогою об'єкта `StringBuilder` послідовно конструюється текст, який містить три логічні частини. У першій частині виводиться прогнозоване значення енергоспоживання для вихідного набору параметрів користувача. Далі описується альтернативний варіант – оптимізаційний сценарій – із зазначенням типу приладу та прогнозованого рівня споживання в цих умовах. Завершальна частина містить інформацію про абсолютну та відносну економію, що дозволяє оцінити ефективність зміни параметрів. Така структура звіту забезпечує зручне сприйняття результатів аналізу та дозволяє користувачу швидко оцінити доцільність застосування альтернативного режиму експлуатації.

4.2 Формування та аналіз набору даних для навчання моделі

Побудова якісної та стійкої моделі машинного навчання значною мірою залежить від репрезентативності й адекватності тренувального набору даних. Для розв'язання задачі прогнозування енергоспоживання побутових приладів було використано структурований датасет Smart Home Energy Consumption, який містить 100 000 записів про використання електроенергії в умовах розумного дому [19]. Цей набір даних забезпечує комплексне охоплення ключових характеристик споживання: типу приладу, часу й дати використання, зовнішньої температури, сезону та кількості мешканців у домогосподарстві. Така структура дозволяє моделі враховувати як індивідуальні, так і зовнішні фактори, що впливають на енерговитрати.

Таблиця 4.1 – Опис атрибутів тренувального набору даних

| № | Назва атрибута | Опис |
|---|--------------------------|---|
| 1 | Home ID | Ідентифікатор домогосподарства (анонімізований) |
| 2 | Appliance Type | Тип побутового приладу (наприклад, Fridge, Oven, Heater) |
| 3 | Energy Consumption (kWh) | Кількість спожитої енергії в кіловат-годинах (цільова змінна) |
| 4 | Time | Час споживання електроенергії у форматі 24-годинної шкали |
| 5 | Date | Дата реєстрації споживання |
| 6 | Outdoor Temperature (°C) | Температура навколишнього середовища на момент використання приладу |
| 7 | Season | Сезон року, в якому зафіксоване споживання (Winter, Summer тощо) |
| 8 | Household Size | Кількість мешканців у домогосподарстві |

| | | | | |
|-------|------|----------|--------|------|
| | | | | |
| Змін. | Арк. | № докум. | Підпис | Дата |

Особливістю вибраного набору є його придатність до застосування в задачах регресійного прогнозування, а також у більш складних сценаріях кластеризації або оптимізації енергоспоживання.

Табл. 4.1 узагальнює атрибутивний склад навчального набору даних, що використовується в моделі.

Наведений набір ознак забезпечує достатню інформативність для виявлення закономірностей у структурі енергоспоживання. Значення енергоспоживання обрано як цільову змінну для регресійного аналізу, тоді як інші атрибути використовуються як предиктори. Поєднання часових, кліматичних та поведінкових факторів створює сприятливі умови для побудови моделі, здатної до точного прогнозування з урахуванням реальних побутових умов. Для розуміння особливостей формування споживання в побутовому середовищі доцільним є окремий розгляд розподілу енергоспоживання за типами приладів. На рис. 4.16 представлено агреговану статистику сумарного споживання електроенергії для кожного типу приладу, зафіксованого у вибірці.

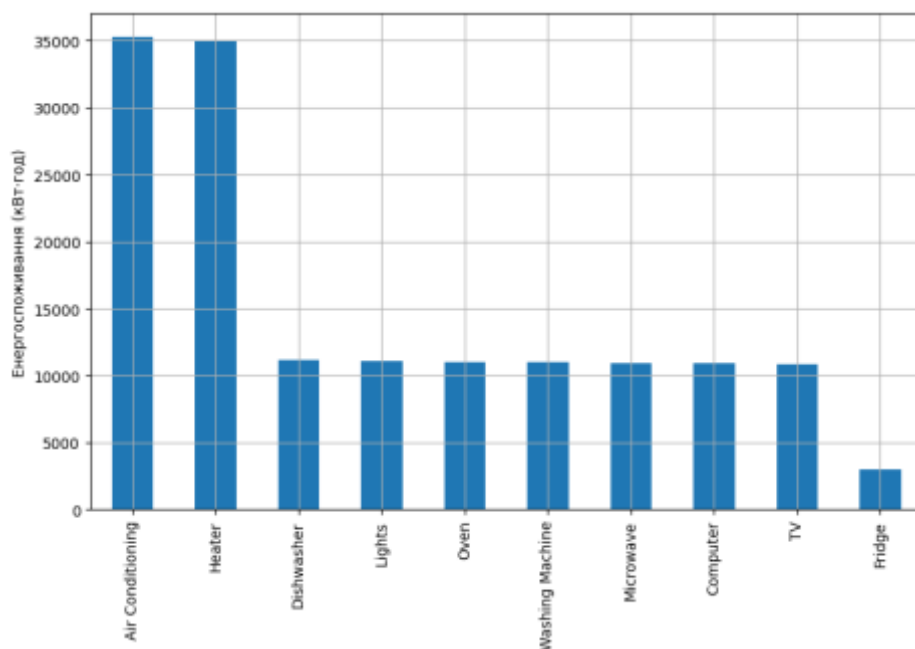


Рисунок 4.16 – Аналіз споживання енергії за типами приладів

Як видно з графіку, найбільше енергоспоживання припадає на кондиціонери та обігрівачі, що є очікуваним результатом з огляду на їхню

| | | | | |
|-------|------|----------|--------|------|
| Змін. | Арк. | № докум. | Підпис | Дата |
| | | | | |

потужність і тривалість використання в екстремальних сезонних умовах. Усі інші прилади – зокрема посудомийна машина, освітлення, духовка, пральна машина, мікрохвильова піч, комп'ютер, телевізор – демонструють подібні рівні навантаження, значно нижчі за згадані кліматичні системи. Найменше споживання зафіксовано у холодильників, що, попри постійну роботу, мають відносно стабільне і низьке енергоспоживання через циклічний характер увімкнення. Такий розподіл підтверджує доцільність врахування типу приладу як однієї з базових предикторних змінних у побудові моделі машинного навчання.

Аналіз залежності споживання електроенергії від кількості мешканців у домогосподарстві дає змогу оцінити вплив соціального фактора на побутову енергоінтенсивність. Така оцінка важлива для побудови більш гнучких моделей прогнозування, здатних адаптуватися до відмінностей у стилі життя та рівні використання приладів. На рис. 4.17 показано зміну середнього споживання енергії у кіловат-годинах залежно від кількості осіб, що проживають у помешканні.

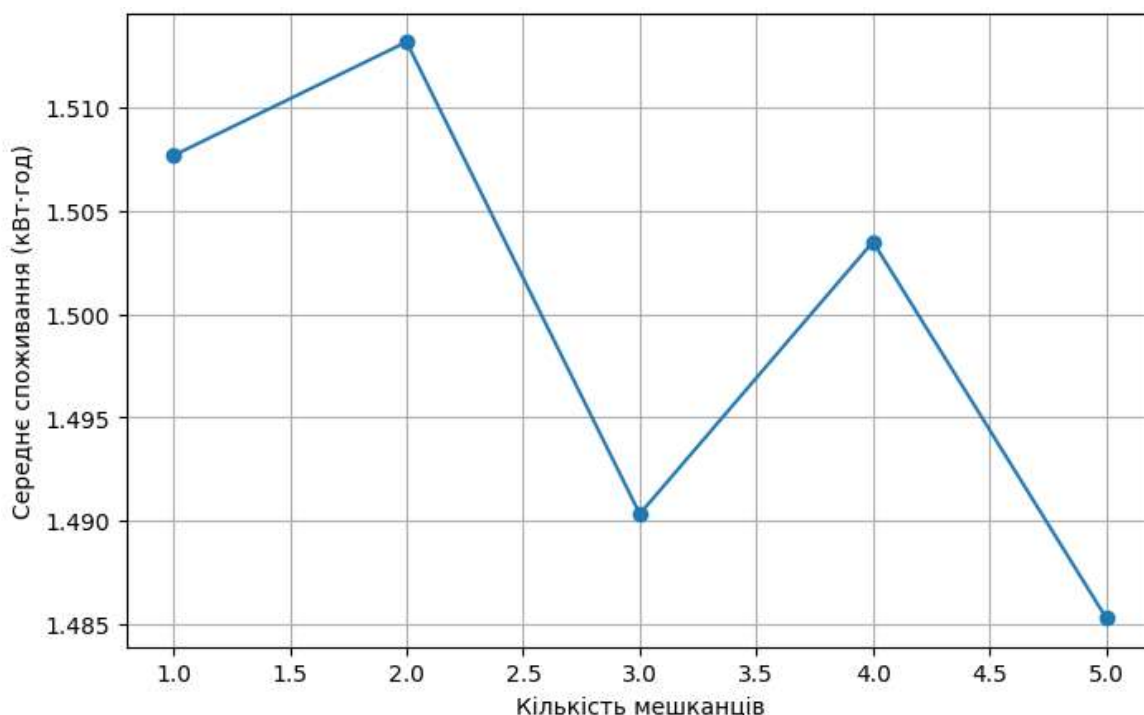


Рисунок 4.17 – Залежність енергоспоживання від розміру домогосподарства

| | | | | |
|-------|------|----------|--------|------|
| | | | | |
| Змін. | Арк. | № докум. | Підпис | Дата |

З графіка видно, що залежність не є лінійною. Найвищий рівень споживання спостерігається у домогосподарствах із двома мешканцями, тоді як із збільшенням чисельності до трьох осіб споживання дещо знижується, а далі коливається в межах вузького діапазону. При п'яти мешканцях відзначається найнижче середнє споживання, що може свідчити про ефект поділу використання приладів або вищу енергоефективність при колективному побутовому користуванні. Це підкреслює доцільність включення кількості мешканців до набору ознак моделі, хоча сам вплив має бути додатково проаналізований з урахуванням типу приладу та режимів експлуатації.

Рис. 4.18 ілюструє теплову карту кореляцій між основними числовими атрибутами тренувального набору даних: споживанням енергії, зовнішньою температурою та розміром домогосподарства. Такий тип візуалізації дозволяє швидко оцінити наявність лінійних залежностей між ознаками, що може бути корисним як для попереднього аналізу, так і для прийняття рішень щодо вибору методів трансформації даних або регуляризації.

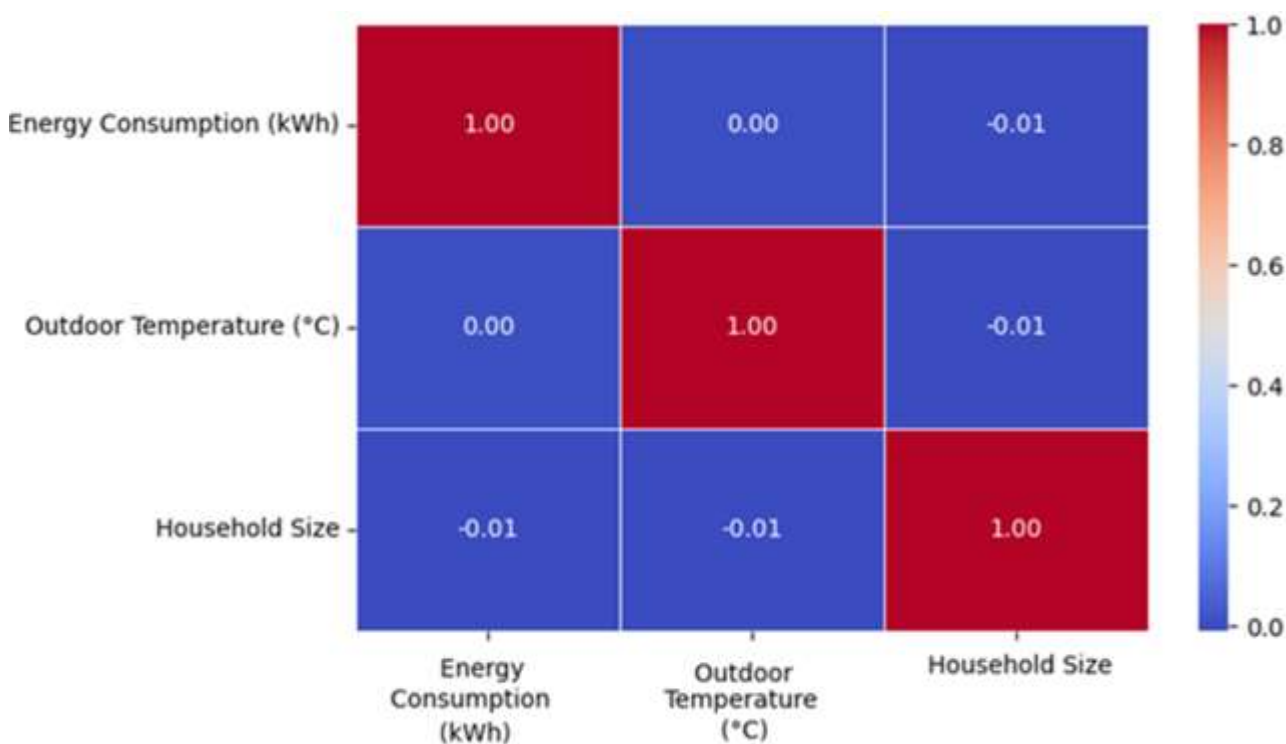


Рисунок 4.18 – Теплова карта кореляцій

Із рисунку видно, що усі пари змінних мають надзвичайно низькі або майже нульові коефіцієнти кореляції. Значення між споживанням енергії та температурою, а також між споживанням і кількістю мешканців не перевищують абсолютне значення 0.01, що свідчить про відсутність лінійного зв'язку між цими характеристиками. Такий результат не суперечить попереднім графікам, де спостерігались слабкі тренди або коливання, що могли мати нелінійний характер. Водночас це підтверджує доцільність використання методів машинного навчання, здатних моделювати складні, невидимі для класичної кореляції взаємозв'язки між ознаками.

На рис. 4.19 зображено розподіл енергоспоживання побутових приладів у різні пори року за допомогою діаграми розмаху, яка дозволяє оцінити медіанні значення, міжквартильний інтервал та наявність потенційних викидів. Такий підхід дає змогу порівняти сезонну варіативність у структурі споживання електроенергії без спрощення до середніх значень, відображаючи реальну дисперсію даних.

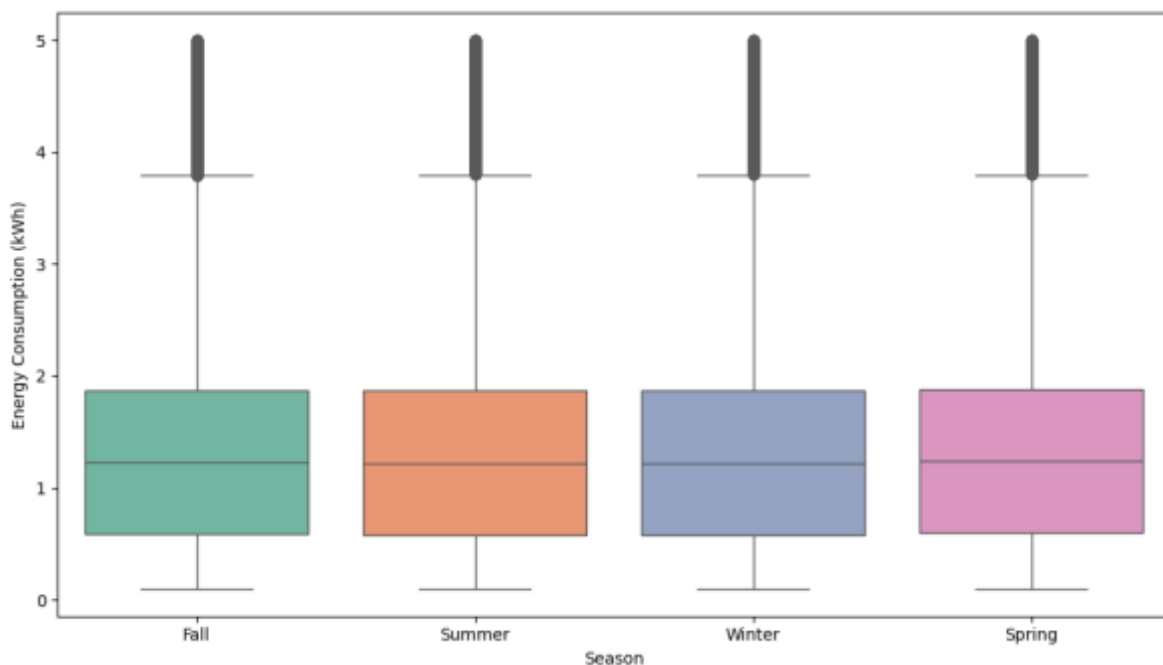


Рисунок 4.19 – Розподіл енергоспоживання за сезонами

| | | | | |
|-------|------|----------|--------|------|
| | | | | |
| Змін. | Арк. | № докум. | Підпис | Дата |

Аналіз показує, що енергоспоживання має найбільше зміщення вгору в зимовий період, що узгоджується з інтенсивною роботою обігрівачів та інших теплових приладів. Весна також демонструє вищу медіану споживання порівняно з осінню та літом, хоча рівень розкиду є дещо вужчим. Літній та осінній періоди характеризуються схожою структурою, з нижчими медіанами, що може бути пов'язано зі зменшенням потреб у кліматичному регулюванні або специфікою використання сезонних пристроїв. Значна кількість викидів у кожному сезоні свідчить про наявність окремих сценаріїв інтенсивного використання, що обумовлює необхідність подальшого моделювання на основі детальних контекстних параметрів.

Рис. 4.20 ілюструє залежність між енергоспоживанням побутових приладів і температурою зовнішнього середовища, з урахуванням сезонного чинника. Кожна точка на графіку представляє окреме спостереження, при цьому колір відповідає сезону, що дозволяє простежити контекстуальні відмінності у споживанні за температурними діапазонами. Така візуалізація забезпечує багатовимірне уявлення про взаємодію кліматичних і поведінкових факторів.

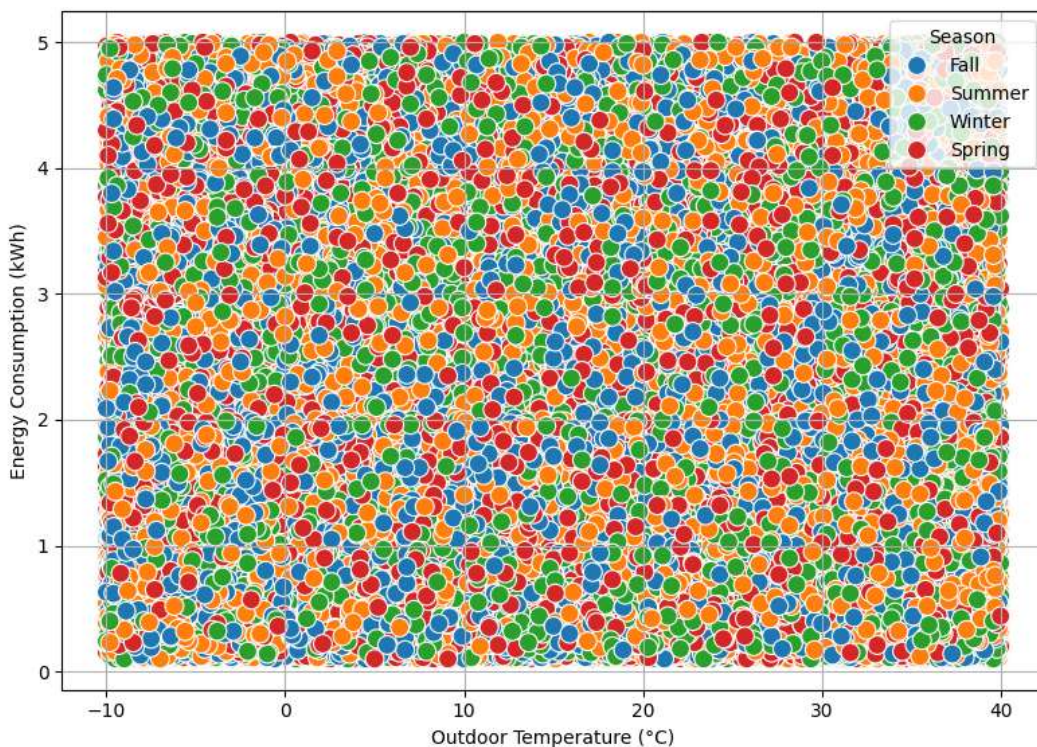


Рисунок 4.20 – Залежність енергоспоживання від зовнішньої температури

| | | | | |
|-------|------|----------|--------|------|
| | | | | |
| Змін. | Арк. | № докум. | Підпис | Дата |

Графік засвідчує відсутність чітко вираженої глобальної кореляції між температурою та енергоспоживанням, однак помітна широка варіативність значень у крайніх температурних зонах. Зокрема, при дуже низьких або дуже високих температурах спостерігається більша щільність точок із підвищеним енергоспоживанням, що може вказувати на інтенсивне використання кліматичних систем у зимовий та літній періоди. Водночас у помірних температурних межах навантаження більш рівномірне, а кольорове розмаїття підтверджує перехресну присутність сезонів в одному температурному інтервалі. Цей розподіл свідчить про складну природу впливу зовнішньої температури на побутове енергоспоживання і підкреслює необхідність застосування моделей, здатних урахувувати нелінійні взаємозв'язки.

На рис. 4.21 зображено порівняння середнього енергоспоживання побутових приладів за всією вибіркою, що дозволяє оцінити відносну енергоінтенсивність кожного з типів обладнання. Побудова графіка у горизонтальній орієнтації полегшує візуальне зіставлення показників між приладами та акцентує увагу на тих, що мають найвище середнє навантаження.

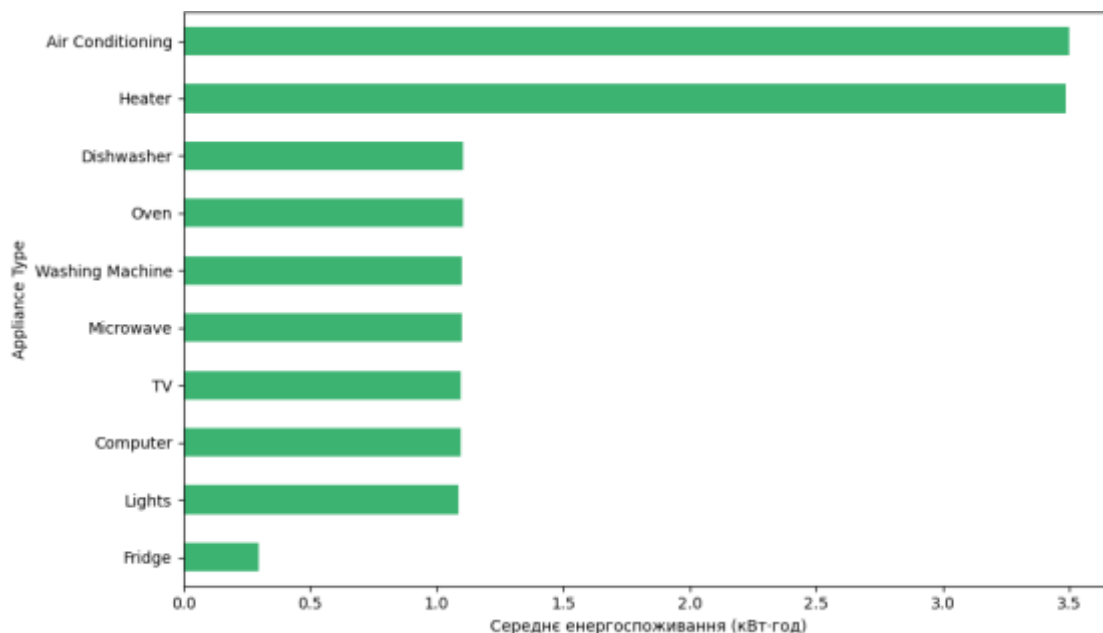


Рисунок 4.21 – Середнє енергоспоживання за типами приладів

| | | | | |
|-------|------|----------|--------|------|
| | | | | |
| Змін. | Арк. | № докум. | Підпис | Дата |

Найвищі значення фіксуються для кондиціонерів і обігрівачів, що вкотре підтверджує їхню домінуючу роль у структурі споживання, пов'язану з температурною регуляцією в оселі. Решта приладів – посудомийна машина, духовка, пральна машина, мікрохвильова піч, телевізор, комп'ютер і освітлення – мають схожі середні значення, які перебувають у межах одного діапазону. Найменше енергоспоживання притаманне холодильникам, що узгоджується з їхньою конструктивною особливістю – стабільна робота з короткими циклами активності. Цей аналіз дозволяє ідентифікувати категорії приладів, які мають найбільший потенціал для оптимізації або впровадження енергоощадних сценаріїв експлуатації.

Рис. 4.22 ілюструє залежність середнього енергоспоживання окремих типів побутових приладів від кількості мешканців у домогосподарстві. Така теплова карта дозволяє комплексно оцінити, як соціальний чинник впливає на навантаження різних категорій пристроїв, і виявити наявність стабільних або варіативних шаблонів використання.

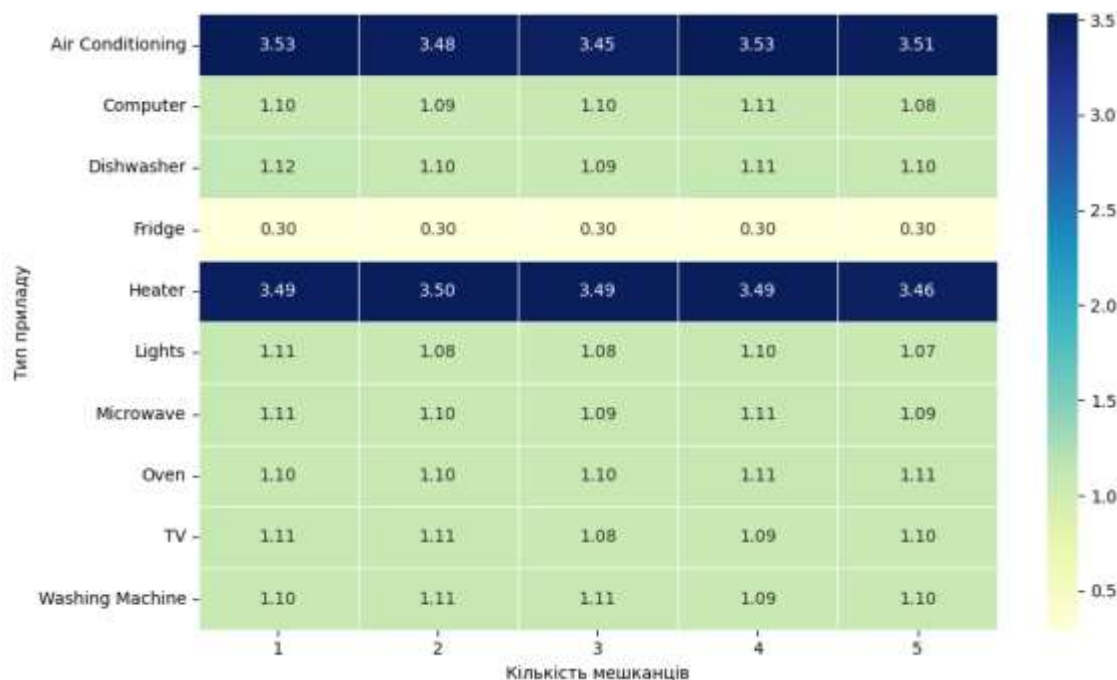


Рисунок 4.22 – Середнє енергоспоживання за розміром домогосподарства

Найвищі показники традиційно демонструють кондиціонери та обігрівачі – їхнє середнє споживання перевищує 3.4 кВт·г незалежно від кількості мешканців, що свідчить про стабільне, інтенсивне використання, орієнтоване більше на сезонні умови, ніж на чисельність користувачів. У випадку інших приладів – зокрема комп’ютерів, телевізорів, мікрохвильових печей чи освітлення – середні значення коливаються в межах вузького діапазону (переважно 1.08–1.12 кВт·г) і не демонструють істотної залежності від розміру домогосподарства. Виняток становить холодильник, споживання якого залишається фіксованим на рівні 0.30 кВт·г для всіх груп, що вказує на його автономний режим роботи. Такий розподіл підкреслює, що кількість мешканців має опосередкований вплив на більшість побутових приладів, а основні зміни у споживанні пов’язані з характером самого пристрою.

4.3 Проведення експериментів та аналіз результатів прогнозування

Після завершення етапів формування вибірки, побудови моделі та реалізації функціоналу в програмному середовищі було здійснено серію експериментів, спрямованих на оцінку точності прогнозування енергоспоживання побутових приладів.

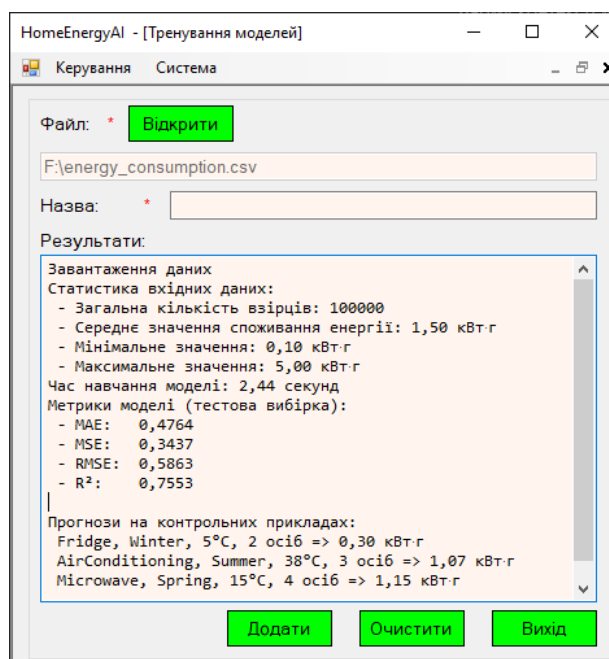


Рисунок 4.23 – Результати навчання моделі

| | | | | |
|-------|------|----------|--------|------|
| | | | | |
| Змін. | Арк. | № докум. | Підпис | Дата |

Оцінювання якості виконувалось на окремій тестовій вибірці, що не використовувалась у процесі навчання, що дозволяє оцінити узагальнюючу здатність моделі. На рис. 4.23 подано інтерфейс застосунку з виведенням результатів прогнозування, метрик якості моделі.

Інтерфейс відображає ключові характеристики вхідних даних: загальна кількість записів – 100 000, середнє значення споживання становить 1,50 кВт·г, при мінімальному та максимальному значеннях 0,10 і 5,00 кВт·г відповідно. Час тренування моделі склав 2,44 секунди, що свідчить про високу обчислювальну ефективність. Оцінка точності на тестовій вибірці дала такі результати: MAE = 0,4764, MSE = 0,3437, RMSE = 0,5863, а коефіцієнт детермінації R^2 досягнув значення 0,7553. Це означає, що модель пояснює понад 75% варіації в даних, що є позитивним результатом для задачі регресії з багатофакторним впливом.

Окрім того, наведено приклади прогнозування для типових сценаріїв використання. Наприклад, для холодильника взимку при температурі 5 °C і двох мешканців передбачене споживання становить 0,30 кВт·г, що узгоджується з очікуваними значеннями для цього типу пристрою. Водночас кондиціонер улітку при 38 °C та трьох мешканців має прогноз 1,87 кВт·г, що також відображає вплив високої температури на навантаження системи охолодження. Це свідчить про те, що модель коректно враховує як сезонні, так і кліматичні чинники при формуванні прогнозу. Результати експериментів демонструють, що модель здатна працювати стабільно, швидко та з достатньою точністю, що відкриває можливості для її застосування в реальних умовах моніторингу побутового енергоспоживання.

На рис. 4.24 зображено роботу модуля моніторингу, який реалізує імітацію прогнозування споживання електроенергії в умовах змінних параметрів. Інтерфейс дозволяє користувачу ввести значення ключових характеристик побутової ситуації – тип приладу, температуру, сезон та розмір домогосподарства – після чого здійснюється обчислення прогнозованого рівня енергоспоживання, як за поточними, так і за умовно оптимізованими умовами.

| | | | | |
|-------|------|----------|--------|------|
| | | | | |
| Змін. | Арк. | № докум. | Підпис | Дата |

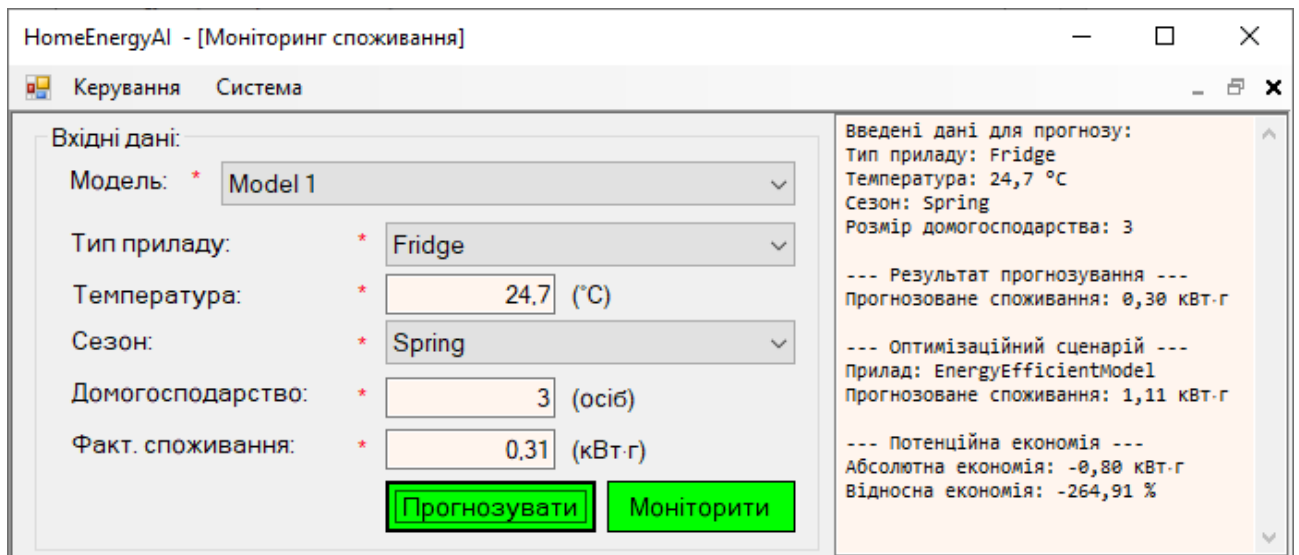


Рисунок 4.24 – Імітація прогнозування споживання електроенергії

На рис. 4.25 наведено ще один приклад експлуатації модуля моніторингу, де користувач вніс параметри для оцінки споживання комп'ютера восени при температурі 5,8 °C та наявності п'яти мешканців у домогосподарстві. У результаті модель сформувала прогнозоване значення енергоспоживання на рівні 1,05 кВт·г, що узгоджується з типовими значеннями для пристроїв цього класу при таких умовах експлуатації.

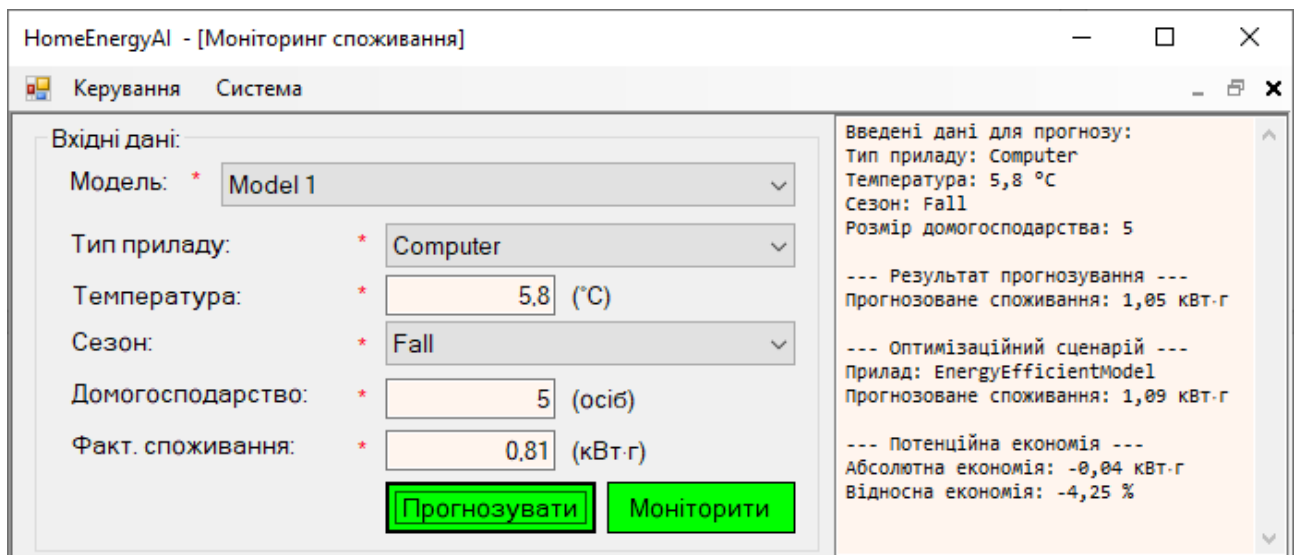


Рисунок 4.25 – Імітація прогнозування споживання електроенергії

У рамках моделювання альтернативного сценарію за допомогою енергоефективного приладу система оцінила споживання на рівні 1,09 кВт·г, що

| | | | | |
|-------|------|----------|--------|------|
| Змін. | Арк. | № докум. | Підпис | Дата |
| | | | | |

виявилось вищим за базове. На основі цього було обчислено негативну абсолютну економію $-0,04$ кВт·г і відносне зниження ефективності на рівні $-4,25\%$. Такий результат засвідчує, що поточні умови використання вже є достатньо оптимізованими, і запропонована заміна моделі не призводить до енергетичного зиску.

Для оцінки ефективності побудованої моделі доцільним є систематизований аналіз її поведінки за різних умов використання побутових приладів. З цією метою сформовано таблицю, в якій зіставляються ключові характеристики: базове прогнозоване споживання, показник при зміненому (умовно оптимізованому) сценарії, абсолютна та відносна економія. Табл. 4.2 узагальнює ці аспекти на прикладі кількох контрольних прогнозів.

Таблиця 4.2 – Опис атрибутів тренувального набору даних

| Тип приладу | Прогноз (базовий), кВт·г | Прогноз (оптимізований), кВт·г | Відносна економія, % |
|------------------|--------------------------|--------------------------------|----------------------|
| Fridge | 0,30 | 1,11 | -264,91 |
| Computer | 1,05 | 1,09 | -4,25 |
| Microwave | 1,25 | 0,95 | 24,00 |
| Air Conditioning | 3,45 | 2,91 | 15,65 |

Проаналізовані результати свідчать про різноспрямовану динаміку залежно від типу приладу та контексту експлуатації. Найбільш показовим є випадок мікрохвильової печі, де оптимізаційний сценарій призвів до зменшення прогнозованого навантаження майже на чверть, що підтверджує потенціал для енергоощадного використання при дотриманні належних умов. Схожий ефект виявлено для кондиціонера, де зниження споживання на понад 15% може мати вагоме значення з огляду на високі абсолютні показники. Водночас для холодильника та комп'ютера спостерігається протилежна ситуація: модель фіксує зростання споживання у «оптимізованому» варіанті, що свідчить про

недостатність користувацького контексту або вже досягнутий енергетичний баланс.

Висновки до розділу

У рамках даного розділу було реалізовано повноцінний цикл побудови та експериментального дослідження системи прогнозування енергоспоживання побутових приладів на основі методу машинного навчання. Здійснено поетапну реалізацію моделі у програмному середовищі C# з використанням ML.NET, описано процес завантаження даних, трансформацій, тренування, оцінювання точності моделі та її інтеграції у WinForms-застосунок. Для навчання було сформовано набір даних із відкритого джерела Smart Home Energy Consumption, який проаналізовано за низкою ключових напрямів – тип приладу, розмір домогосподарства, сезон, температура. Проведено кореляційний аналіз і досліджено варіації енергоспоживання в контексті зовнішніх факторів, що дало змогу виявити відсутність лінійних залежностей, однак підтвердило необхідність використання багатфакторного моделювання.

Модель машинного навчання, заснована на методі швидкого дерева, показала високу ефективність: MAE = 0,4764, MSE = 0,3437, RMSE = 0,5863, а коефіцієнт детермінації $R^2 = 0,7553$, що свідчить про добру здатність до узагальнення. У межах експериментів протестовано декілька контрольних сценаріїв, у яких зіставлялись базові та умовно оптимізовані прогнози. Для частини приладів зафіксовано позитивну динаміку зниження енергоспоживання до 24 % та 15,65 % відповідно. Водночас виявлено приклади зворотної тенденції, що засвідчує необхідність контекстного підходу до оптимізації. Отримані результати лягли в основу побудови системи моніторингу, яка не лише генерує прогнози, але й дозволяє оцінити доцільність впровадження енергоефективних сценаріїв.

| | | | | |
|-------|------|----------|--------|------|
| | | | | |
| Змін. | Арк. | № докум. | Підпис | Дата |

ВИСНОВКИ

Дипломний проєкт присвячений розробці інформаційної системи для аналізу ефективності використання енергії в побутових приладах на основі методів машинного навчання. Головною метою роботи було створення інтелектуального модуля, що здатен виконувати прогнозування енергоспоживання з урахуванням зовнішніх та поведінкових факторів, а також формувати обґрунтовані оцінки потенційної економії.

У першому розділі була проаналізована актуальність моніторингу енергоспоживання як в економічному, так і в екологічному вимірі. Розглянуто чинники, що формують споживання в побуті, виокремлено основні підходи до оцінювання ефективності, включно з методами прогнозного моделювання, індексами енергоефективності та балансувальними схемами на основі Smart Grid. Окрема увага приділена можливостям застосування машинного навчання, визначено основні напрямки його використання, джерела даних та обмеження. Проведено огляд існуючих програмних рішень (The Sense Electricity, Home Assistant, Wattics), що дозволило аргументувати доцільність створення нового рішення з фокусом на інтеграцію штучного інтелекту, простоту у використанні та безкоштовну доступність. Сформульовано чітку постановку задачі із зазначенням вхідних, вихідних даних, функціональних і нефункціональних вимог до системи.

Другий розділ висвітлює порівняння методів машинного навчання для задач прогнозування, включно з аналізом швидкого дерева, методу опорних векторів і наївного байєсівського класифікатора. На підставі переваг у точності та швидкості обрано метод швидкого дерева. Побудовано математичну модель, яка описує процес регресійного прогнозування на множині ознак. Здійснено аналіз платформ і інструментів реалізації, в результаті чого для створення системи обрано мову програмування C# та середовище Visual Studio 2022.

Третій розділ був присвячений вибору бібліотеки машинного навчання. Розглянуто ML.NET, Accord.NET та Infer.NET, з яких найбільш придатною

| | | | | | | |
|-------|------|----------|--------|------|--------------------|------|
| | | | | | ІАЛЦ.045480.004 ПЗ | Арк. |
| Змін. | Арк. | № докум. | Підпис | Дата | | 86 |

виявилась ML.NET завдяки простоті інтеграції в середовище C# та підтримці необхідних функцій. Розроблено базові алгоритми для етапів тренування моделі й моніторингу споживання. Запроєктовано трьохрівневу архітектуру системи з чітким поділом на рівень взаємодії, бізнес-логіки та зберігання. Побудовано діаграми класів, що формалізують структуру програмного рішення.

У четвертому розділі реалізовано повноцінний прототип інформаційної системи. Створено модель машинного навчання у середовищі ML.NET, налаштовано завантаження та попередню обробку даних, побудовано трансформаційний конвеєр і виконано тренування на основі датасету Smart Home Energy Consumption. Здійснено розширений аналіз цього набору: вивчено вплив типу приладу, сезону, температури та розміру домогосподарства на споживання, побудовано графіки, теплові карти, діаграми розмаху. Результати експериментального тестування показали достатню точність моделі ($MAE = 0,4764$; $RMSE = 0,5863$; $R^2 = 0,7553$). Система також пройшла імітаційні випробування на різних прикладах прогнозування, в результаті яких було продемонстровано як випадки ефективного зниження навантаження (наприклад, для Microwave – економія 24 %), так і приклади зворотного ефекту, що дозволяє уникати невіправданих дій з боку користувача.

Проведена робота демонструє застосування сучасних методів машинного навчання для вирішення задач енергоменеджменту на побутовому рівні. Результати дослідження підтвердили здатність побудованої системи не лише виконувати точне прогнозування, а й слугувати інструментом підтримки прийняття рішень щодо енергоощадних стратегій. Сформоване програмне рішення є масштабованим, адаптивним і може бути використане як основа для впровадження в розумних будинках або платформах побутового моніторингу.

| | | | | |
|-------|------|----------|--------|------|
| | | | | |
| Змін. | Арк. | № докум. | Підпис | Дата |

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Alimohamadi R., Jahangir M. Multi-Objective optimization of energy consumption pattern in order to provide thermal comfort and reduce costs in a residential building. *Energy Conversion and Management*. 2024. Vol. 1, 305 p.
2. Аверкуня М., Дубровук О. Прогнозування впливу потенціалу працівника іт-сектору на його заробітну плату. *Scientific Notes of Ostroh Academy National University, "Economics" Series*. 2021. Vol. 22, No 50, pp. 82-87.
3. Bafjaish S. Comparative analysis of Naive Bayesian techniques in health-related for classification task. *Journal of Soft Computing and Data Mining*. 2020. Vol. 1, No 2, pp. 1-10.
4. Chen H., Hu S., Hua R., Zhao X. Improved naive Bayes classification algorithm for traffic risk management. *EURASIP Journal on Advances in Signal Processing*. 2021. Vol. 1, No 30, 150 p.
5. Come funziona Home Assistant | SmartWorld. URL: <https://www.smartworld.it/guide/come-funziona-home-assistant.html> (дата звернення 03.05.2025).
6. Dutta P., El-Kenawy, S., Guma A., Dhoska K. An Energy Consumption Monitoring and Control System in Buildings using Internet of Things // *International Journal of Advanced Computer Science and Applications*. 2023. Vol. 14, No 6, pp. 1–8.
7. Home Assistant Review - Dream Green House. URL: <https://www.dreamgreenhouse.com/reviews/homeassistant/index.php> (дата звернення 03.05.2025).
8. Hussien A., Maksoud A., Al-Dahhan A., Abdeen A., Baker T. Machine learning model for predicting long-term energy consumption in buildings // *Discover Internet of Things*. 2025. Vol. 5, pp. 18-25.
9. Integrating Machine Learning with Fullstack Development Using ML.NET. URL: <https://www.researchgate.net/profile/Thiyagarajan-Mani>

| | | | | |
|-------|------|----------|--------|------|
| | | | | |
| Змін. | Арк. | № докум. | Підпис | Дата |

ІАЛЦ.045480.004 ПЗ

Арк.

88

Chettier/publication/390300770_Integrating_Machine_Learning_with_Full_stack_Development_Using_MLNET/links/67e8136695231d5ba59f3ad1/Integrating-Machine-Learning-with-Fullstack-Development-Using-MLNET.pdf (дата звернення 03.05.2025).

10. Kong X., Zhu S., Huo X., Li S., Li Y., Zhang S. A Household Energy Efficiency Index Assessment Method Based on Non-Intrusive Load Monitoring Data // Applied Sciences. 2020. Vol. 10, No 11, 820 p.
11. Liu C., Song Y., Li R., Ma W., Hao J. L., Qiang G. Three-level modular grid system for sustainable construction of industrialized residential buildings: A case study in China. Journal of Cleaner Production, 2023. 395 p.
12. Mamadaliev D., Touko P., Kim J., Kim, S. Esfd-yolov8n: early smoke and fire detection method based on an improved yolov8n model. Fire. 2024. Vol. 7, No 9, 300 p.
13. Piñeiro C., Abuín J., Pichel J. (2020). Very Fast Tree: speeding up the estimation of phylogenies for large alignments through parallelization and vectorization strategies. Bioinformatics. 2020. Vol. 36, No 17, pp. 658-659.
14. Price M. J. C# 11 and . NET 7 - Modern Cross-Platform Development Fundamentals: Start Building Websites and Services with ASP. NET Core 7, Blazor, and EF Core 7, 7th Edition. Packt Publishing, Limited, 2022. 826 p.
15. Rider for C# - The Best Visual Studio Alternative IDE . URL: <https://developer.okta.com/blog/2020/11/30/rider-csharp-visual-studio-alternative> (дата звернення 03.05.2025).
16. Sense Energy Monitor Reviews & Ratings 2025. URL: <https://www.trustradius.com/products/sense/reviews> (дата звернення 03.05.2025).
17. Sharma A., Gupta R., Singh M. Design of Realtime Electricity Consumption Monitoring System Integrated with Campus Web // TEM Journal. 2023. Vol. 12, No 2, pp. 840–849.

| | | | | |
|-------|------|----------|--------|------|
| | | | | |
| Змін. | Арк. | № докум. | Підпис | Дата |

ІАЛЦ.045480.004 ПЗ

Арк.
89

18. Singh V., Poonia R., Kumar S., Dass P., Agarwal P., Bhatnagar V., Raja L. Prediction of COVID-19 corona virus pandemic based on time series data using Support Vector Machine. Journal of Discrete Mathematical Sciences and Cryptography. 2020. Vol. 23, No 8, pp. 583-597.
19. Smart Home Energy Consumption. URL: <https://www.kaggle.com/datasets/mexwell/smart-home-energy-consumption> (дата звернення 23.05.2025).
20. The Sense Electricity Dashboard App - Dave Margulius On Climate. URL: <https://davemargulius.com/my-home-electricity-dashboard-app-explained/> (дата звернення 03.05.2025).
21. Verma, R. Extending Visual Studio. In Visual Studio Extensibility Development: Extending Visual Studio IDE for Productivity, Quality, Tooling, Analysis, and Artificial Intelligence. 2023. pp. 73-113.
22. Visual Studio Code Reviews & Product Details. URL: <https://www.g2.com/products/visual-studio-code/reviews> (дата звернення 03.05.2025).
23. Wattics | Software Reviews & Alternatives. URL: <https://sourceforge.net/software/product/Wattics/> (дата звернення 03.05.2025).
24. Wattics Pricing, Features, and Reviews (May 2025). URL: <https://www.softwaresuggest.com/wattics> (дата звернення 03.05.2025).
25. Zhang F., O'Donnell L. Support vector regression. In Machine learning. Academic Press. 2020. pp. 123-140.
26. Zhang L., Shen F., Chen F., Lin Z. Origin and evolution of the 2019 novel coronavirus. Clinical Infectious Diseases. 2020. Vol. 71, No 15, pp. 882-883.
27. Zhou C., Xu C., Cui Z. Progressive Bayesian inference for scribble-supervised semantic segmentation. In Proceedings of the AAAI Conference on Artificial Intelligence. 2023. Vol. 37, No 3, pp. 751-759.

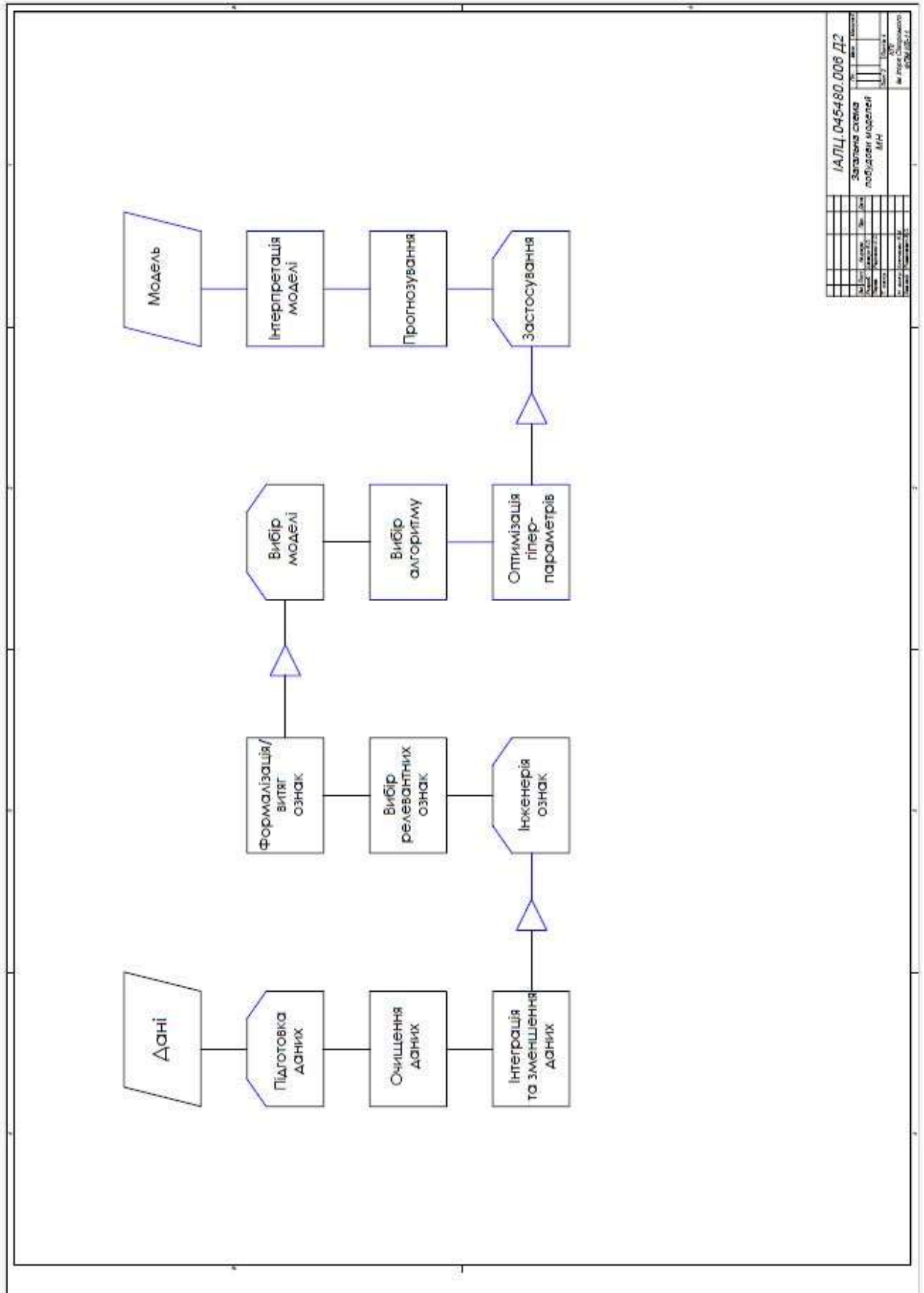
28. Гальперін А., Кочнев Д. Робота з Eclipse: Навчальний посібник. - Харків: Видавництво Харківського національного університету імені В.Н. Каразіна, 2017. - 230 с.
29. Гладун А.В. Трьохрівнева архітектура побудови веб-систем. Вісник Національного університету "Львівська політехніка". 2018. – 315 с.
30. Городенко В.П., Марценюк О.П. Програмування на мові Python. Теорія та практика: Навчальний посібник. - К.: Центр навчальної літератури, 2016. - 480 с.
31. Сліпченко В. Г., Мамалига В. М., Полягушко Л. Г. Комп'ютерний моніторинг еко-енерго-економічних процесів та систем: навч. посіб. – Київ: КПІ ім. Ігоря Сікорського, 2022. – 120 с.

| | | | | |
|-------|------|----------|--------|------|
| | | | | |
| Змін. | Арк. | № докум. | Підпис | Дата |

ІАЛЦ.045480.004 ПЗ

ДОДАТКИ

ДОДАТОК 2



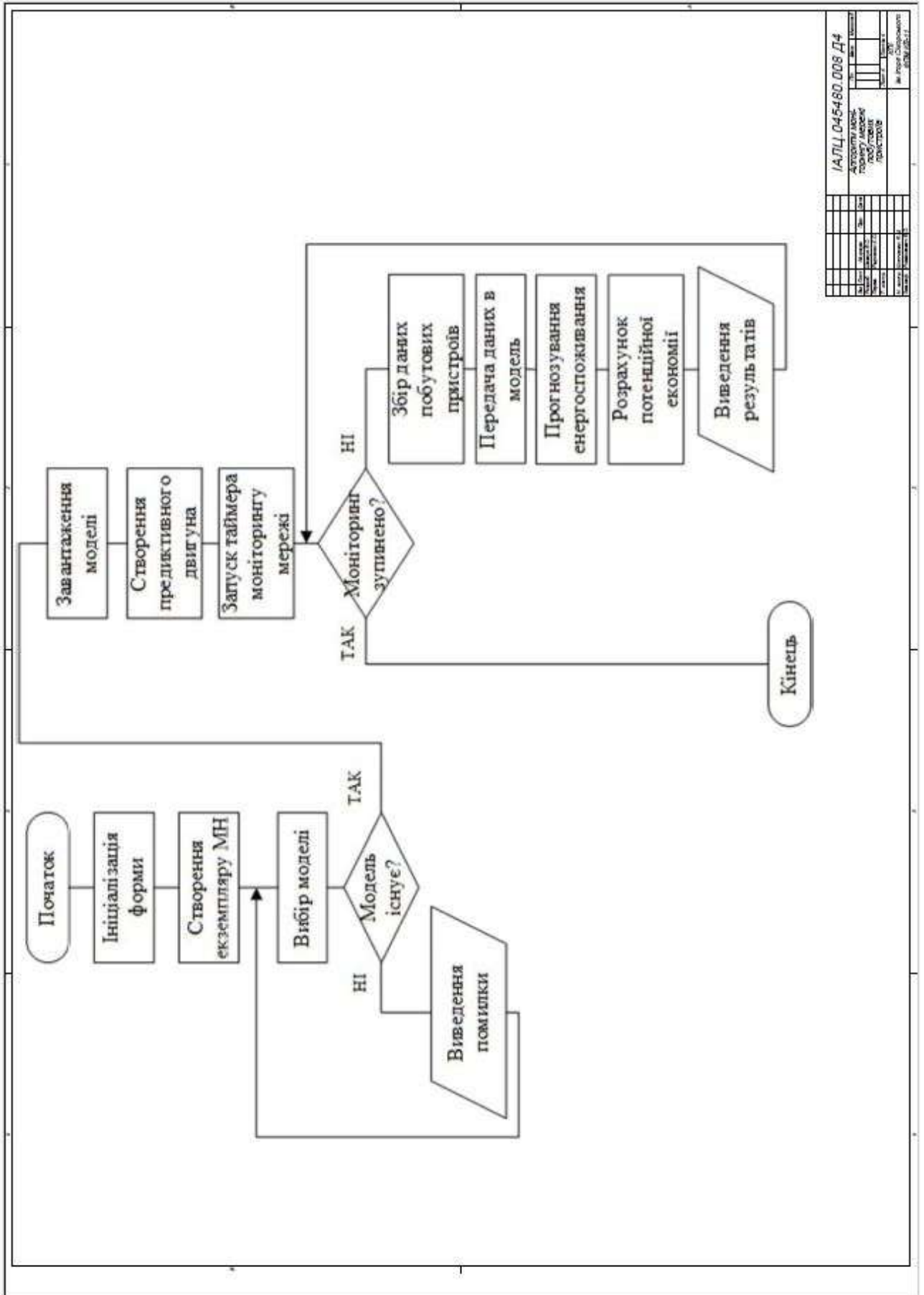
ІАЛЦ 045480.008 Д2

| | | | | | | | | | |
|----|----|----|----|----|----|----|----|----|-----|
| № | № | № | № | № | № | № | № | № | № |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 |
| 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 |
| 31 | 32 | 33 | 34 | 35 | 36 | 37 | 38 | 39 | 40 |
| 41 | 42 | 43 | 44 | 45 | 46 | 47 | 48 | 49 | 50 |
| 51 | 52 | 53 | 54 | 55 | 56 | 57 | 58 | 59 | 60 |
| 61 | 62 | 63 | 64 | 65 | 66 | 67 | 68 | 69 | 70 |
| 71 | 72 | 73 | 74 | 75 | 76 | 77 | 78 | 79 | 80 |
| 81 | 82 | 83 | 84 | 85 | 86 | 87 | 88 | 89 | 90 |
| 91 | 92 | 93 | 94 | 95 | 96 | 97 | 98 | 99 | 100 |

Загальна сума
лободових моделей
даних

№ стор. Облікового
зрешіт. 001.1

ДОДАТОК 4



ІАЛЦ 045480.008 Д4

| | | | |
|----|------------|--------|-----------|
| № | Дата | Зробив | Перевірив |
| 1 | 2023.05.15 | І.І.І. | І.І.І. |
| 2 | 2023.05.15 | І.І.І. | І.І.І. |
| 3 | 2023.05.15 | І.І.І. | І.І.І. |
| 4 | 2023.05.15 | І.І.І. | І.І.І. |
| 5 | 2023.05.15 | І.І.І. | І.І.І. |
| 6 | 2023.05.15 | І.І.І. | І.І.І. |
| 7 | 2023.05.15 | І.І.І. | І.І.І. |
| 8 | 2023.05.15 | І.І.І. | І.І.І. |
| 9 | 2023.05.15 | І.І.І. | І.І.І. |
| 10 | 2023.05.15 | І.І.І. | І.І.І. |

ж. Голова Департаменту: І.І.І.
м. Київ, 2023 р.

ДОДАТОК 5

ЛІСТИНГ 1. Код класу «MonitoringForm»

```
using EnergyPredictionApp.AppCode;
using EnergyPredictionApp.Forms.Systems;
using EnergyPredictionApp.Providers;
using Microsoft.ML;
using System;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace EnergyPredictionApp.Forms.Controls {
    public partial class MonitoringForm : Form {
        private ValidationMy _Validation = new ValidationMy();

        private Models _SelectedModels = new Models();
        private MLContext mlContext = new MLContext();
        private PredictionEngine<EnergyData, EnergyPredictionResult> predictionEngine;
        private ModelsProvider _ModelsProvider = new ModelsProvider();
        private List<Models> _ModelsList = new List<Models>();
        private bool _IsModelsLoad = false;
        private LogsProvider _LogsProvider = new LogsProvider();

        string filePath = "data.csv";
        private List<EnergyData> _ApplianceData = new List<EnergyData>();

        public MonitoringForm() {
            InitializeComponent();
            LoadAllData();
        }

        private void ModelsCBox_SelectedValueChanged(object sender, EventArgs e) {
            if (_IsModelsLoad && IsModelExist()) {
                _SelectedModels = _ModelsProvider.SelectedModelsByModelsId(
                    Convert.ToInt32(ModelsCBox.SelectedValue));
                LoadData(_SelectedModels.ModelsFileModel);
            }
        }

        private void PredictBtn_Click(object sender, EventArgs e) {
            // Перевірка коректності введення
            if (IsAllInputsCorrect() && IsModelExist()) {
                // Формування об'єкта з введених даних
                EnergyData testApplianceData = new EnergyData {
                    ApplianceType = ApplianceTypeCBox.Text.Trim(),
                    Temperature = float.Parse(TemperatureTBox.Text),
                }
            }
        }
    }
}
```

```

Season = SeasonCBox.Text.Trim(),
HouseholdSize = float.Parse(HouseholdSizeTBox.Text)
};

ResultTBox.Clear();

// Виведення введених даних
var inputInfo = new StringBuilder();
inputInfo.AppendLine("Введені дані для прогнозу:");
inputInfo.AppendLine($"Тип приладу: {testApplianceData.ApplianceType}");
inputInfo.AppendLine($"Температура: {testApplianceData.Temperature} °C");
inputInfo.AppendLine($"Сезон: {testApplianceData.Season}");
inputInfo.AppendLine($"Розмір домогосподарства: {testApplianceData.HouseholdSize}");
// Основний прогноз
var predictionBase = predictionEngine.Predict(testApplianceData);

// Створення оптимізованого сценарію (наприклад, заміна приладу на
енергоефективний)
EnergyData optimizedData = new EnergyData {
    ApplianceType = "EnergyEfficientModel", // умовна назва
    Temperature = testApplianceData.Temperature,
    Season = testApplianceData.Season,
    HouseholdSize = testApplianceData.HouseholdSize
};
var predictionOpt =
    predictionEngine.Predict(optimizedData);
// Обчислення економії
float deltaE = predictionBase.PredictedEnergy -
    predictionOpt.PredictedEnergy;
float eta = (predictionBase.PredictedEnergy > 0) ?
    (deltaE / predictionBase.PredictedEnergy) * 100.0f : 0.0f;

// Формування результату
var predictionInfo = new StringBuilder();
predictionInfo.AppendLine("\r\n--- Результат прогнозування ---");
predictionInfo.AppendLine($"Прогнозоване споживання: " +
    $"{predictionBase.PredictedEnergy:F2} кВт·г");
predictionInfo.AppendLine("\r\n--- Оптимізаційний сценарій ---");
predictionInfo.AppendLine($"Прилад: {optimizedData.ApplianceType}");
predictionInfo.AppendLine($"Прогнозоване споживання: " +
    $"{predictionOpt.PredictedEnergy:F2} кВт·г");
predictionInfo.AppendLine("\r\n--- Потенційна економія ---");
predictionInfo.AppendLine($"Абсолютна економія: {deltaE:F2} кВт·г");
predictionInfo.AppendLine($"Відносна економія: {eta:F2} %");

// Логування дії
_LogsProvider.InsertLogs(
    LoginForm.CurrentUser.UserId,
    $"Проведено прогнозування для приладу: {testApplianceData.ApplianceType}",
    DateTime.Now
);

```

```

// Виведення результату у поле
ResultTBox.Text = inputInfo.ToString() + predictionInfo.ToString();
}
}

private void MonitoringBtn_Click(object sender, EventArgs e) {
    if (IsModelExist()) {
        if (MoniroringTimer.Enabled) {
            MoniroringTimer.Enabled = false;
            MonitoringBtn.Text = "Моніторити";
            _LogsProvider.InsertLogs(LoginForm.CurrentUser.UsersId,
                "Було зупинено генерацію випадкових даних для моделі " +
                ModelsCBox.Text, DateTime.Now);
        } else {
            MoniroringTimer.Enabled = true;
            MonitoringBtn.Text = "Зупинити";
            _LogsProvider.InsertLogs(LoginForm.CurrentUser.UsersId,
                "Було запущено генерацію випадкових даних для імітації мережевого трафіку " +
                ModelsCBox.Text, DateTime.Now);
        }
    }
}

private void MonitoringTimer_Tick(object sender, EventArgs e) {
    if (_ApplianceData.Any()) {
        Random rand = new Random();
        int index = rand.Next(_ApplianceData.Count);
        EnergyData randomData = _ApplianceData[index];

        // Копіювання для об'єкта передбачення
        EnergyData mlData = new EnergyData {
            ApplianceType = randomData.ApplianceType,
            Temperature = randomData.Temperature,
            Season = randomData.Season,
            HouseholdSize = randomData.HouseholdSize,
            EnergyConsumption = randomData.EnergyConsumption // для порівняння
        };

        // Основний прогноз
        var predictionBase = predictionEngine.Predict(mlData);

        // Імітація енергоефективнішого сценарію (заміна на менш енергоспоживаючий прилад)
        EnergyData optimizedData = new EnergyData {
            ApplianceType = "EnergyEfficientModel", // умовна заміна
            Temperature = mlData.Temperature,
            Season = mlData.Season,
            HouseholdSize = mlData.HouseholdSize
        };
        var predictionOpt = predictionEngine.Predict(optimizedData);
    }
}

```

```

// Обчислення економії
float deltaE = predictionBase.PredictedEnergy - predictionOpt.PredictedEnergy;
float eta = (predictionBase.PredictedEnergy > 0) ? (deltaE / predictionBase.PredictedEnergy) *
100.0f : 0.0f;

// Виведення у UI
ResultTBox.Invoke((MethodInvoker)() => {
    var dataInfo = new StringBuilder();
    dataInfo.AppendLine("Моніторинг пристроїв");
    dataInfo.AppendLine($"Тип приладу: {mlData.ApplianceType}");
    dataInfo.AppendLine($"Температура: {mlData.Temperature} °C");
    dataInfo.AppendLine($"Сезон: {mlData.Season}");
    dataInfo.AppendLine($"Домогосподарство: {mlData.HouseholdSize} осіб");
    dataInfo.AppendLine($"Факт. споживання: {mlData.EnergyConsumption:F2} кВт·г");

    dataInfo.AppendLine("\r\n--- Прогнозування ---");
    dataInfo.AppendLine($"Прогнозоване споживання: {predictionBase.PredictedEnergy:F2}
кВт·г");

    dataInfo.AppendLine("\r\n--- Оптимізаційний сценарій ---");
    dataInfo.AppendLine($"Прилад: {optimizedData.ApplianceType}");
    dataInfo.AppendLine($"Прогнозоване споживання: {predictionOpt.PredictedEnergy:F2}
кВт·г");

    dataInfo.AppendLine("\r\n--- Потенційна економія ---");
    dataInfo.AppendLine($"Абсолютна економія: {deltaE:F2} кВт·г");
    dataInfo.AppendLine($"Відносна економія: {eta:F2} %");

    ResultTBox.Text = dataInfo.ToString();
});
}
}

private void LoadAllData() {
    ApplianceTypeCBox.SelectedIndex = 0;
    SeasonCBox.SelectedIndex = 0;
    _ModelsList = _ModelsProvider.GetAllModels();
    ModelsCBox.DataSource = _ModelsList;
    ModelsCBox.ValueMember = "ModelsId";
    ModelsCBox.DisplayMember = "ModelsName";
    _IsModelsLoad = true;
    _ApplianceData = LoadApplianceData(filePath);
    ModelsCBox_SelectedValueChanged(ModelsCBox, EventArgs.Empty);
}

private void LoadData(string filePath) {
    string localPath = Application.StartupPath + filePath;
    DataViewSchema modelSchema;
    try {
        // Завантаження моделі

```

```

ITransformer model =
    mlContext.Model.Load(localPath, out modelSchema);
predictionEngine =
    mlContext.Model.CreatePredictionEngine<EnergyData,
    EnergyPredictionResult>(model);
} catch (Exception ex) {
    MessageBox.Show("Помилка при завантаженні моделі: " + ex.Message,
    "Помилка", MessageBoxButtons.OK, MessageBoxIcon.Error);
}
}

// Метод зчитування CSV-файлу з EnergyData
public List<EnergyData> LoadApplianceData(string filePath) {
    List<EnergyData> applianceList = new List<EnergyData>();

    try {
        using (var reader = new StreamReader(filePath)) {
            if (!reader.EndOfStream) {
                reader.ReadLine(); // пропуск заголовка
            }

            while (!reader.EndOfStream) {
                var line = reader.ReadLine();
                if (!string.IsNullOrEmpty(line)) {
                    var appliance = ParseApplianceFromCsv(line);
                    applianceList.Add(appliance);
                }
            }
        }
    } catch (Exception ex) {
        MessageBox.Show(" ✘ Помилка при зчитуванні файлу: " + ex.Message, "Помилка",
        MessageBoxButtons.OK, MessageBoxIcon.Error);
    }

    return applianceList;
}

// Метод парсингу рядка CSV → EnergyData
public EnergyData ParseApplianceFromCsv(string line) {
    var values = line.Split(',');
    var culture = CultureInfo.InvariantCulture;

    return new EnergyData {
        ApplianceType = values[1],
        EnergyConsumption = float.Parse(values[2], culture),
        Temperature = float.Parse(values[5], culture),
        Season = values[6],
        HouseholdSize = float.Parse(values[7], culture)
    };
}

```

```
}
```

```
private bool IsAllInputsCorrect() {  
    bool isCorrect = true;  
  
    if (_Validation.IsDataEntering(ApplianceTypeCBox.Text)) {  
        ApplianceTypeValidationLbl.Text = NamesMy.ProgramButtons.RequiredValidation;  
    } else {  
        ApplianceTypeValidationLbl.Text = NamesMy.ProgramButtons.ErrorValidation;  
        isCorrect = false;  
    }  
  
    if (_Validation.IsDataConvertToDouble(TemperatureTBox.Text)) {  
        TemperatureValidationLbl.Text = NamesMy.ProgramButtons.RequiredValidation;  
    } else {  
        TemperatureValidationLbl.Text = NamesMy.ProgramButtons.ErrorValidation;  
        isCorrect = false;  
    }  
  
    if (_Validation.IsDataEntering(SeasonCBox.Text)) {  
        SeasonValidationLbl.Text = NamesMy.ProgramButtons.RequiredValidation;  
    } else {  
        SeasonValidationLbl.Text = NamesMy.ProgramButtons.ErrorValidation;  
        isCorrect = false;  
    }  
  
    if (_Validation.IsDataConvertToDouble(HouseholdSizeTBox.Text)) {  
        HouseholdSizeValidationLbl.Text = NamesMy.ProgramButtons.RequiredValidation;  
    } else {  
        HouseholdSizeValidationLbl.Text = NamesMy.ProgramButtons.ErrorValidation;  
        isCorrect = false;  
    }  
  
    if (_Validation.IsDataConvertToDouble(EnergyConsumptionTBox.Text)) {  
        EnergyConsumptionValidationLbl.Text = NamesMy.ProgramButtons.RequiredValidation;  
    } else {  
        EnergyConsumptionValidationLbl.Text = NamesMy.ProgramButtons.ErrorValidation;  
        isCorrect = false;  
    }  
  
    return isCorrect;  
}
```

```
private bool IsModelExist() {  
    bool isCorrect = true;  
    if (Convert.ToInt32(ModelsCBox.SelectedValue) > 0) {  
        ModelsValidationLbl.Text = NamesMy.ProgramButtons.RequiredValidation;  
    } else {
```

```

        ModelsValidationLbl.Text = NamesMy.ProgramButtons.ErrorValidation;
        isCorrect = false;
    }
    return isCorrect;
}

```

```

}

```

Лістинг 2. Код класу «EnergyDataProvider»

```

using Microsoft.ML.Data;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace EnergyPredictionApp.Providers {
    internal class EnergyDataProvider {
    }
}

// Клас вхідних даних
public class EnergyData {
    [LoadColumn(1)] public string ApplianceType { get; set; }
    [LoadColumn(2)] public float EnergyConsumption { get; set; }
    [LoadColumn(5)] public float Temperature { get; set; }
    [LoadColumn(6)] public string Season { get; set; }
    [LoadColumn(7)] public float HouseholdSize { get; set; }
}

// Клас результату прогнозу
public class EnergyPredictionResult {
    [ColumnName("Score")]
    public float PredictedEnergy { get; set; }
}

```

Лістинг 3. Код класу «ModelsForm»

```

using EnergyPredictionApp.AppCode;
using EnergyPredictionApp.Forms.Systems;
using EnergyPredictionApp.Providers;
using Microsoft.ML;
using Microsoft.ML.Data;
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

```

```

using System.Windows.Forms;
using System.Globalization;
using System.IO;
using System.Diagnostics;

namespace EnergyPredictionApp.Forms.SysMS {
public partial class ModelsForm : Form {
private MLContext mlContext;
ITransformer model;
private IDataView _fullData;
private string _Path = "";

private int _selectedRowIndex = 0;
private ValidationMy _Validation = new ValidationMy();
private ModelsProvider _ModelsProvider = new ModelsProvider();
private List<Models> _ModelsList = new List<Models>();
private LogsProvider _LogsProvider = new LogsProvider();
private bool _IsModelTrain = false;
private Random _rand = new Random();

public ModelsForm() {
InitializeComponent();
DataLoad();
}

private void OpenBtn_Click(object sender, EventArgs e) {
// Створення діалогового вікна для відкриття файлу
OpenFileDialog openFileDialog = new OpenFileDialog {
Filter = "Text files (*.csv)|*.csv|All files (*.*)|*.*",
FilterIndex = 2,
RestoreDirectory = true
};

if (openFileDialog.ShowDialog() == DialogResult.OK) {
_Path = openFileDialog.FileName;
FileNameTBox.Text = openFileDialog.FileName;

// Ініціалізація MLContext
mlContext = new MLContext(seed: 1);
// Засікання часу
Stopwatch stopwatch = new Stopwatch();
stopwatch.Start();

// 1. Завантаження даних
_fullData = mlContext.Data.LoadFromTextFile<EnergyData>(
path: _Path,
hasHeader: true,
separatorChar: ',');
ReportTBox.Text = "Завантаження даних\r\n";
Application.DoEvents();
// Загальна статистика

```

```

var dataEnumerable =
    mlContext.Data.CreateEnumerable<EnergyData>(_fullData, reuseRowObject:
false).ToList();
int totalCount = dataEnumerable.Count;
float avgEnergy = dataEnumerable.Average(d => d.EnergyConsumption);
float minEnergy = dataEnumerable.Min(d => d.EnergyConsumption);
float maxEnergy = dataEnumerable.Max(d => d.EnergyConsumption);
ReportTBox.Text += ("Статистика вхідних даних:\r\n");
ReportTBox.Text += ($" - Загальна кількість взірців: {totalCount}\r\n");
ReportTBox.Text += ($" - Середнє значення споживання енергії: {avgEnergy:F2}
кВт·г\r\n");
ReportTBox.Text += ($" - Мінімальне значення: {minEnergy:F2} кВт·г\r\n");
ReportTBox.Text += ($" - Максимальне значення: {maxEnergy:F2} кВт·г\r\n");

// 2. Розділення даних
var trainTestSplit =
    mlContext.Data.TrainTestSplit(_fullData, testFraction: 0.2);
var trainData = trainTestSplit.TrainSet;
var testData = trainTestSplit.TestSet;

// 3. Побудова трансформацій
var dataPipeline = mlContext.Transforms.CopyColumns("Label", "EnergyConsumption")
    .Append(mlContext.Transforms.Categorical.OneHotEncoding("ApplianceTypeEncoded",
"ApplianceType"))
    .Append(mlContext.Transforms.Categorical.OneHotEncoding("SeasonEncoded",
"Season"))
    .Append(mlContext.Transforms.Concatenate("Features", "ApplianceTypeEncoded",
"Temperature", "SeasonEncoded", "HouseholdSize"));

// 4. Створення і навчання моделі
var trainer = mlContext.Regression.Trainers.FastTree();
var pipeline = dataPipeline.Append(trainer);
Console.WriteLine("\nНавчання моделі швидкого дерева...\r\n");
model = pipeline.Fit(trainData);
stopwatch.Stop(); // Зупинка таймера

// Виведення часу навчання
ReportTBox.Text += ($"\nЧас навчання моделі:" +
    $" {stopwatch.Elapsed.TotalSeconds:F2} секунд\r\n");

// 5. Оцінка моделі на тестовій вибірці
var predictions = model.Transform(testData);
var metrics =
    mlContext.Regression.Evaluate(predictions, labelColumnName: "EnergyConsumption");

ReportTBox.Text += ("\nМетрики моделі (тестова вибірка):\r\n");
ReportTBox.Text += ($" - MAE: {metrics.MeanAbsoluteError:F4}\r\n");
ReportTBox.Text += ($" - MSE: {metrics.MeanSquaredError:F4}\r\n");
ReportTBox.Text += ($" - RMSE: {metrics.RootMeanSquaredError:F4}\r\n");
ReportTBox.Text += ($" - R²: {metrics.RSquared:F4}\r\n");

```

```

// 6. Аналіз похибок (умовна "матриця помилок")
var actualData =
    mlContext.Data.CreateEnumerable<EnergyData>(testData, reuseRowObject: false).ToList();
var predictedData =
    mlContext.Data.CreateEnumerable<EnergyPredictionResult>(predictions,
    reuseRowObject: false).ToList();
// 8. Прогнозування на тестових прикладах вручну
var predEngine = mlContext.Model.CreatePredictionEngine<EnergyData,
    EnergyPredictionResult>(model);
var samples = new[] {
    new EnergyData { ApplianceType = "Fridge", Temperature = 5,
        Season = "Winter", HouseholdSize = 2 },
    new EnergyData { ApplianceType = "AirConditioning", Temperature = 38,
        Season = "Summer", HouseholdSize = 3 },
    new EnergyData { ApplianceType = "Microwave", Temperature = 15,
        Season = "Spring", HouseholdSize = 4 }
};
RaportTBox.Text += ("\r\nПрогнози на контрольних прикладах:\r\n");
foreach (var sample in samples) {
    var prediction = predEngine.Predict(sample);
    RaportTBox.Text += (" {sample.ApplianceType}, {sample.Season}," +
        $" {sample.Temperature}°C, {sample.HouseholdSize} " +
        $"оціб => {prediction.PredictedEnergy:F2} кВт·г\r\n");
}

RaportTBox.SelectionStart = RaportTBox.Text.Length;
RaportTBox.ScrollToCaret();

_IsModelTrain = true;
}
}

```

```

private void ModelsGridView_CellClick(object sender, DataGridViewCellEventArgs e) {
    if (e.ColumnIndex == 5 && ModelsGridView[0, e.RowIndex].Value.ToString() !=
        _ModelsList[0].Message) {
        if (MessageBox.Show("Ви дійсно хочете видалити цю модель?", "Видалити",
            MessageBoxButtons.YesNo) == DialogResult.Yes) {
            _ModelsProvider.DeleteModelsByModelsId(Convert.ToInt32(ModelsGridView[0,
                e.RowIndex].Value.ToString()));
            DataLoad();
        }
    }
}

```

```

private void SaveBtn_Click(object sender, EventArgs e) {
    if (IsDataEnteringCorrect()) {
        //Зберігання моделі
        string pathName = @"teach\" + GenerateFileName() + ".zip";
        string localProj =

```

```

System.IO.Path.GetDirectoryName(System.Reflection.Assembly.GetExecutingAssembly().Location);
    _ModelsProvider.InsertModels(ModelsNamesTBox.Text, pathName);
    mlContext.Model.Save(model, _fullData.Schema, localProj + pathName);
    ClearAllData();
    _LogsProvider.InsertLogs(LoginForm.CurrentUser.UsersId,
        "Було навчено модель " +
        ModelsNamesTBox.Text, DateTime.Now);
    MessageBox.Show("Дані успішно збережено!");
}
}

```

```

private void ClearBtn_Click(object sender, EventArgs e) {
    ClearAllData();
}

```

```

private void ExitBtn_Click(object sender, EventArgs e) {
    this.Close();
}

```

```

public string GenerateFileName() {
    DateTime now = DateTime.Now;
    string fileName = string.Format("{0}_{1}_{2}_{3}_{4}_{5}",
        now.Year, now.Month, now.Day, now.Hour, now.Minute, now.Second);

    return fileName;
}

```

```

private void ClearAllData() {
    _IsModelTrain = false;
    ModelsNamesTBox.Text = String.Empty;
    RaportTBox.Text = String.Empty;
    DataLoad();
}
private bool IsDataEnteringCorrect() {
    bool isCorrect = true;
    if (!_IsModelTrain) {
        MessageBox.Show("Неможливо зберегти дані. \r\nЩе не навчено модель!", "Увага!");
        isCorrect = false;
    }
    if (_Validation.IsDataEntering(ModelsNamesTBox.Text)) {
        ModelsNamesValidationLbl.Text = NamesMy.ProgramButtons.RequiredValidation;
    } else {
        ModelsNamesValidationLbl.Text = NamesMy.ProgramButtons.ErrorValidation;
        isCorrect = false;
    }
    return isCorrect;
}
}

```

```
private void DataLoad() {
    int firstRowIndex = 0;
    if (ModelsGridView.FirstDisplayedScrollingRowIndex > 0) {
        firstRowIndex = ModelsGridView.FirstDisplayedScrollingRowIndex;
    }
    try {
        _ModelsList = _ModelsProvider.GetAllModels();
        LoadDataInModelsGridView(_ModelsList);
        if (_selectedRowIndex == ModelsGridView.Rows.Count) {
            _selectedRowIndex = ModelsGridView.Rows.Count - 1;
        }
        if (_selectedRowIndex >= 0) {
            ModelsGridView.FirstDisplayedScrollingRowIndex = firstRowIndex;
            ModelsGridView.Rows[_selectedRowIndex].Selected = true;
        }
    } catch (Exception ex) {
        MessageBox.Show(ex.ToString());
    }
}
}
```

ДОДАТОК 6
Презентація

ДОДАТОК 7

Додаткові діаграми аналізу тренувальних даних моделі

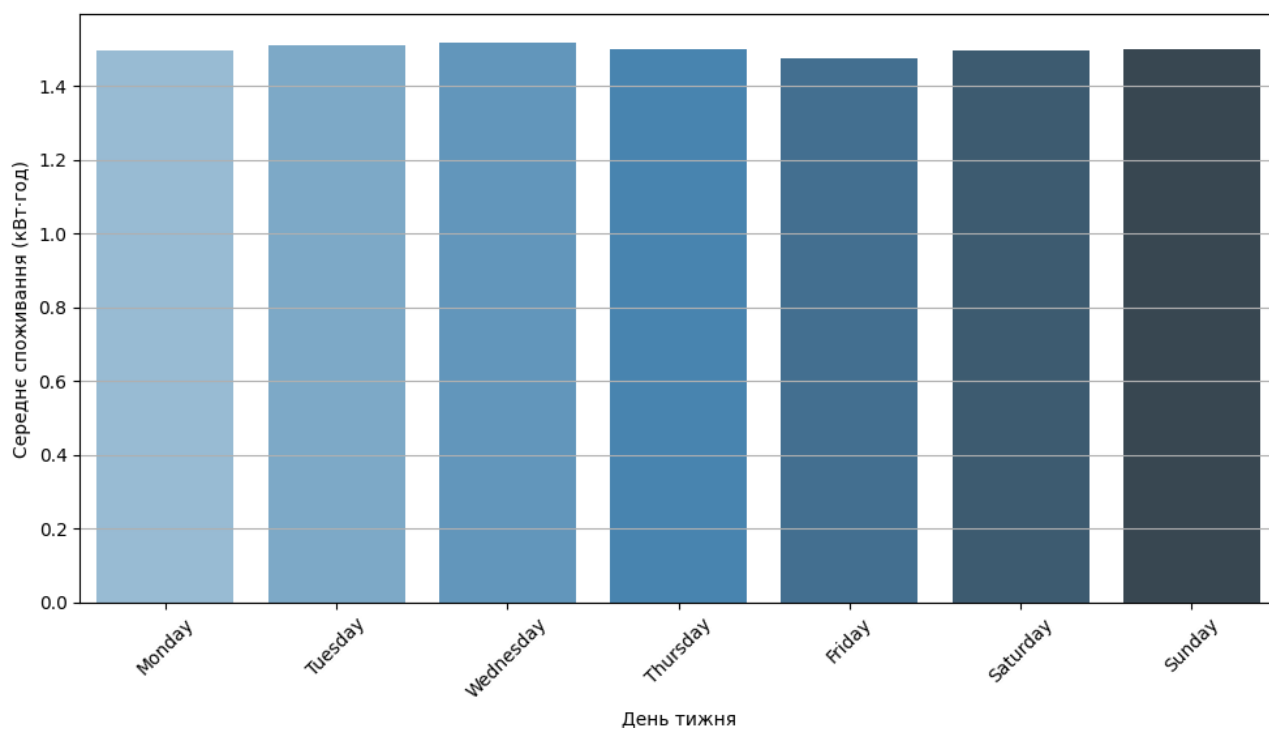


Рисунок 1.1 – Середнє енергоспоживання за днями тижня

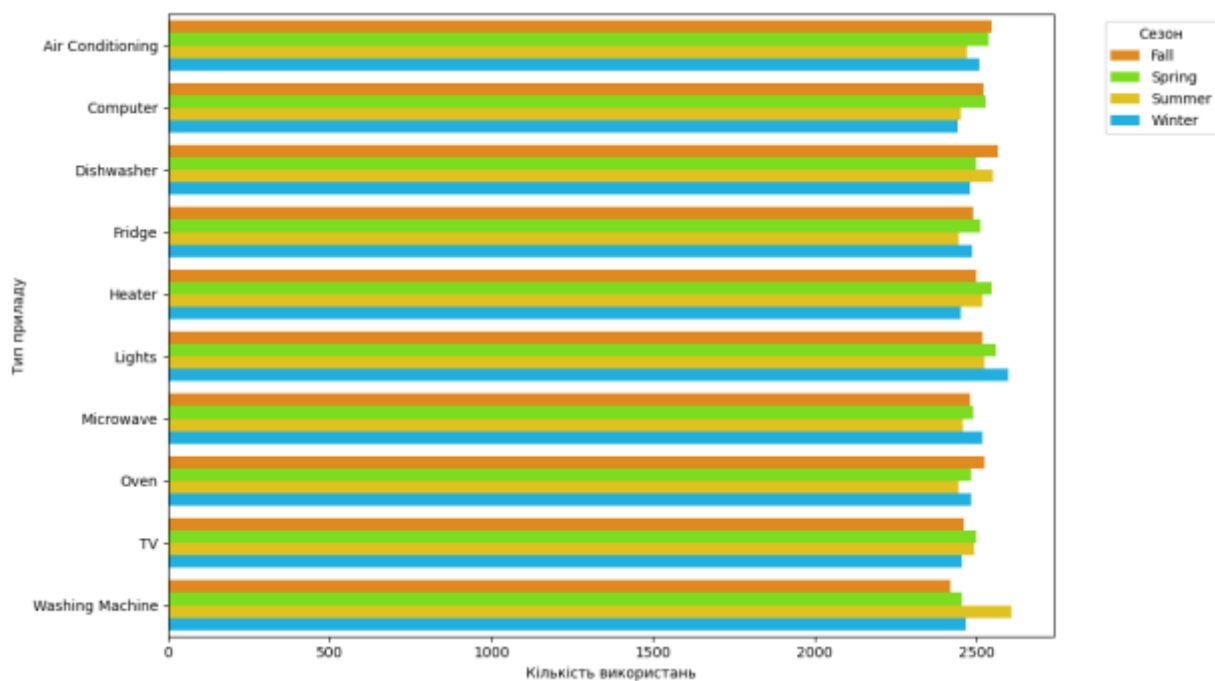


Рисунок 1.2 – Кількість використань приладів за сезонами

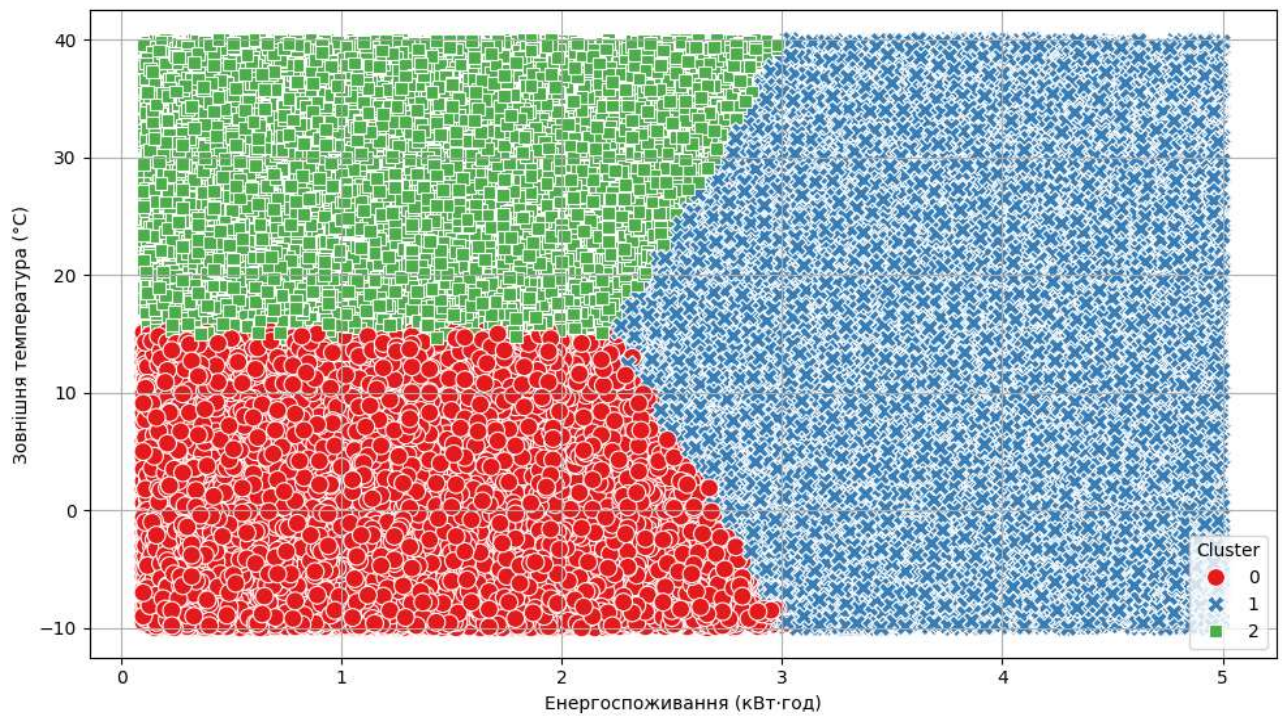


Рисунок 1.3 – Кластеризація домогосподарств

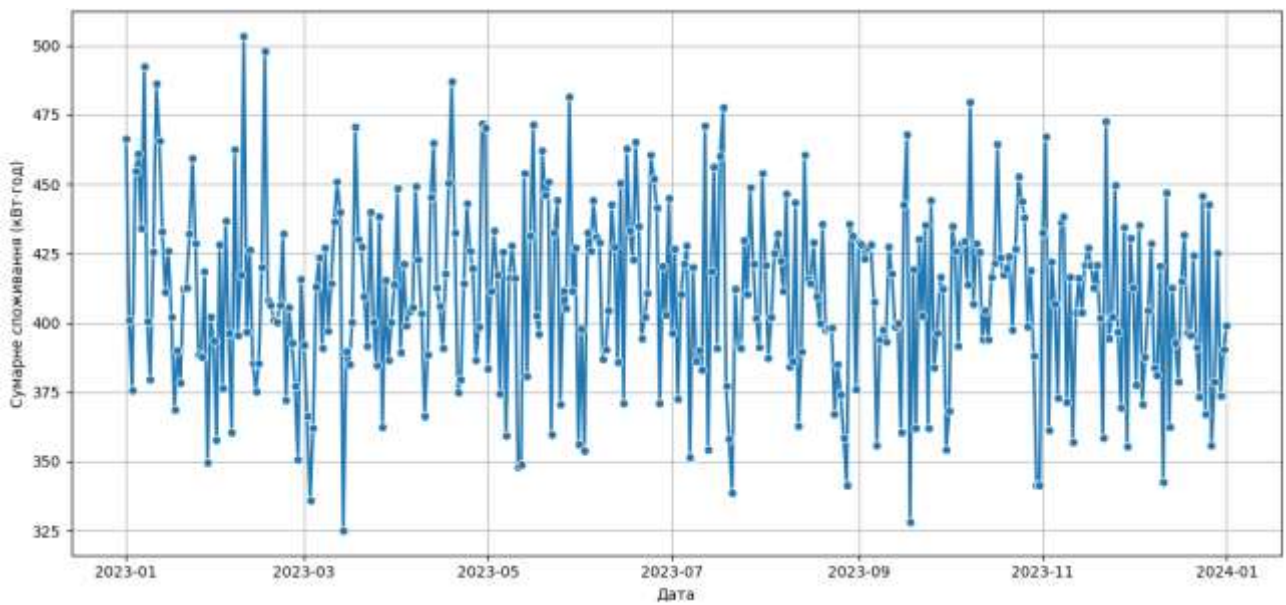


Рисунок 1.4 – Тренд енергоспоживання за датами

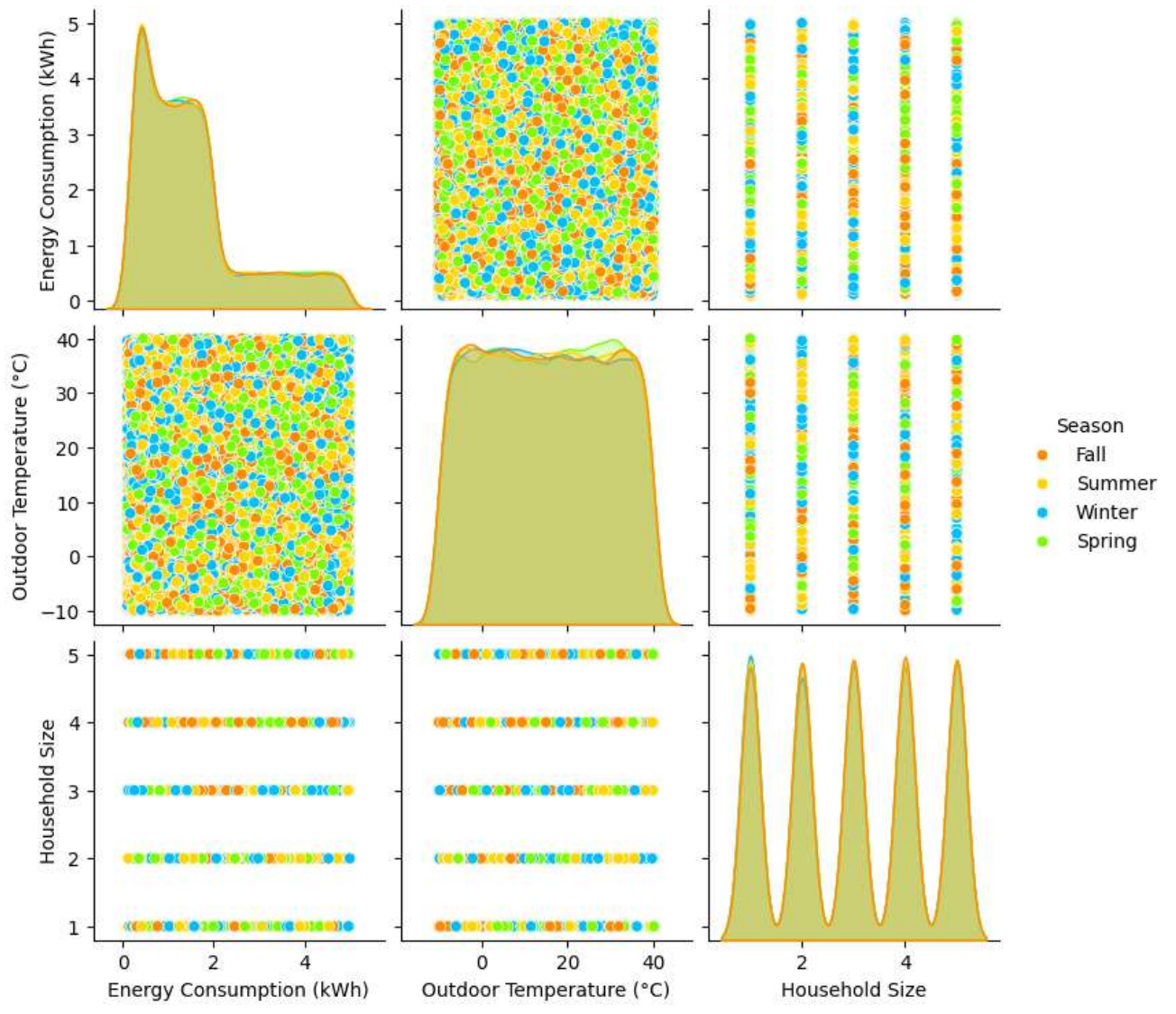


Рисунок 1.5 – Парні залежності між ознаками