

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

**Навчально-науковий інститут прикладного системного аналізу
Кафедра математичних методів системного аналізу**

До захисту допущено:
Завідувач кафедри
_____ Оксана ТИМОЩУК
«___» _____ 2026 р.

Дипломна робота

на здобуття ступеня бакалавра

**за освітньо-професійною програмою «Системний аналіз і управління»
спеціальності 124 «Системний аналіз»**

**на тему: «Інтелектуальна система прогнозування погодинних цін на
основі кластеризації режимів ринку електроенергії України»**

Виконала:

студентка IV курсу, групи КА-23

Кіндзерська Анастасія Олександрівна

Керівник:

професор кафедри ММСА, д.т.н., проф.

Недашківська Надія Іванівна

Консультант з економічного розділу:

доцент кафедри ЕК, к.е.н.

Рощина Надія Василівна

Консультант з нормоконтролю:

доцент кафедри ММСА, к.ф.-м.н.

Статкевич Віталій Михайлович

Рецензент:

к.т.н., доцент кафедри ІІІ ІПСА

Шаповал Наталія Віталіївна

Засвідчую, що у цій дипломній роботі
немає запозичень з праць інших авторів
без відповідних посилань.

Студентка _____

Київ – 2026 року

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
Навчально-науковий інститут прикладного системного аналізу
Кафедра математичних методів системного аналізу

Рівень вищої освіти – перший (бакалаврський)
 Спеціальність – 124 «Системний аналіз»
 Освітня програма «Системний аналіз і управління»

ЗАТВЕРДЖУЮ
 Завідувач кафедри
 _____ Оксана ТИМОЩУК
 «__» _____ 2026 р.

ЗАВДАННЯ
на дипломну роботу студентці
Кіндзерській Анастасії Олександрівні

1. Тема роботи «Інтелектуальна система прогнозування погодинних цін на основі кластеризації режимів ринку електроенергії України», керівник роботи Недашківська Надія Іванівна, доктор технічних наук, професор, затверджені наказом по університету від 26.02.2026 р. № 1944–с.
2. Термін подання студентом роботи «__» червня 2026 р.
3. Вихідні дані до роботи: науково-методична література за темою роботи; погодинні дані цін, імпорту, експорту та балансуєчих показників електроенергії України з 2022 по 2025 роки; програмні засоби Python.
4. Зміст роботи: теоретико-методичні основи аналізу, кластеризації та прогнозування функціональних даних; побудова ознак і вибір конфігурації кластеризації; прогнозування на основі результатів кластеризації; аналіз результатів кластеризації, поділу на режими та результатів прогнозування на годину та на добу вперед; функціонально-вартісний аналіз програмного продукту.
5. Перелік ілюстративного матеріалу (із зазначенням плакатів, презентацій тощо): аналіз сирих даних, порівняння методів кластеризації, інтерпретація режимів ринку електроенергії, порівняння результатів прогнозування, презентація.
6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Функціонально-вартісний аналіз	Рощина Н.В., доц., к.е.н.		

7. Дата видачі завдання _____

Календарний план

№ з/п	Назва етапів виконання дипломної роботи	Термін виконання етапів роботи	Примітка
1	Затвердження теми бакалаврської дипломної роботи (БДР), опанування ДСТУ 3008:2015	26.02.2026- 28.02.2026	виконано
2	Огляд літератури з аналізу функціональних даних, кластеризації, прогнозування	01.03.2026- 24.03.2026	виконано
3	Формування датасету: пошук даних, первинний аналіз та очищення	20.03.2026- 31.03.2026	виконано
4	Побудова ознак і перетворення даних у функціональний формат	01.04.2026- 15.05.2026	виконано
5	Реалізація та порівняння методів кластеризації та інтерпретація режимів ринку	03.05.2026- 14.05.2026	виконано
6	Реалізація та порівняння прогнозування різними способами	14.05.2026- 18.05.2026	виконано
7	Оформлення пояснювальної записки, перевірка структури, висновків, джерел і додатків, підготовка презентації	19.05.2026- 30.05.2026	виконано
8	Передзахист БДР		виконано

Студентка _____ Анастасія КІНДЗЕРСЬКА

Керівник _____ Надія НЕДАШКІВСЬКА

РЕФЕРАТ

Дипломна робота: 118 с., 14 табл., 19 рис., 2 дод., 28 посилань.

РИНОК ЕЛЕКТРОЕНЕРГІЇ УКРАЇНИ, ПОГОДИННІ ЦІНИ, ПРОГНОЗУВАННЯ, КЛАСТЕРИЗАЦІЯ, ФУНКЦІОНАЛЬНІ ДАНІ, FUNWEIGHTCLUST, XGBOOST, ЧАСОВІ РЯДИ.

Мета дослідження – підвищення точності та інтерпретованості прогнозування погодинних цін електроенергії в Україні шляхом побудови інтелектуальної системи, що поєднує кластеризацію типових режимів функціонування ринку електроенергії та моделі машинного навчання для прогнозування цін.

Об'єкт дослідження – процес формування погодинних цін на ринку електроенергії України та добові режими, що описуються часовими рядами цін, імпорту, експорту та балансуючих показників.

Предмет дослідження – методи кластеризації добових профілів ринку електроенергії та методи машинного навчання для прогнозування погодинних цін з урахуванням виявлених режимів ринку.

Методи дослідження – методи функціонального аналізу даних для подання кожної доби як функціонального спостереження; алгоритм funWeightClust для кластеризації режимів ринку електроенергії; алгоритми K-means і DBSCAN для порівняльного аналізу результатів кластеризації; алгоритм XGBoost для прогнозування погодинних цін електроенергії; методи оцінювання якості кластеризації на основі BIC, ICL та Silhouette; методи оцінювання якості прогнозування на основі MAE, RMSE, MAPE, графічного порівняння фактичних і прогнозних кривих, а також аналізу обчислювальної ефективності за часом виконання.

У роботі досліджено проблему підвищення точності прогнозування погодинних цін електроенергії на основі виявлення типових режимів функціонування ринку.

ABSTRACT

Thesis: 118 pages, 14 tables, 19 figures, 2 appendices, 28 references.

UKRAINIAN ELECTRICITY MARKET, HOURLY PRICES, FORECASTING, CLUSTERING, FUNCTIONAL DATA, FUNWEIGHTCLUST, XGBOOST, TIME SERIES.

The purpose of the research – to increase the accuracy and interpretability of hourly electricity price forecasting in Ukraine by building an intelligent system that combines clustering of typical modes of operation of the electricity market and machine learning models for price forecasting.

The object of the research – the process of hourly price formation in the electricity market of Ukraine and the daily modes of its operation, described by time series of prices, imports, exports and balancing indicators.

The subject of the research – methods of clustering daily profiles of the electricity market and machine learning methods for forecasting hourly prices considering the identified market modes.

Research methods – functional data analysis methods for presenting each day as a functional observation; funWeightClust algorithm for clustering electricity market regimes; K-means and DBSCAN algorithms for comparative analysis of clustering results; XGBoost algorithm for forecasting hourly electricity prices; clustering quality assessment methods based on BIC and ICL; forecasting quality assessment methods based on MAE, RMSE, MAPE, graphical comparison of actual and forecast curves, as well as analysis of computational efficiency by execution time.

The study investigates the problem of increasing the accuracy of hourly electricity price forecasting based on identifying typical market operating modes.

ЗМІСТ

ВСТУП.....	10
РОЗДІЛ 1 АНАЛІЗ РИНКУ ЕЛЕКТРОЕНЕРГІЇ УКРАЇНИ ТА ПІДХОДІВ ДО КЛАСТЕРИЗАЦІЇ Й ПРОГНОЗУВАННЯ ПОГОДИННИХ ЦІН.....	12
1.1 Особливості функціонування ринку електроенергії України.....	12
1.1.1 Структура ринку електроенергії України.....	13
1.1.2 Ринок «на добу наперед», внутрішньодобовий та балансуєчий ринки.....	13
1.1.3 Роль торгових зон та вибір зони «ОЕС України» для дослідження..	14
1.2 Аналіз факторів, що впливають на погодинні ціни електроенергії.....	15
1.3 Аналіз існуючих підходів до кластеризації часових і функціональних даних.....	16
1.3.1 Класичні методи кластеризації.....	17
1.3.2 Функціональні дані та функціональна кластеризація.....	17
1.4 Аналіз методів прогнозування часових рядів електроенергії.....	17
1.4.1 Статистичні методи прогнозування.....	17
1.4.2 Методи машинного навчання для прогнозування цін.....	18
1.4.3 Градієнтний бустинг і XGBoost.....	18
1.5 Формулювання задачі дослідження.....	19
1.6 Висновки до розділу 1.....	21
РОЗДІЛ 2 ТЕОРЕТИЧНІ ОСНОВИ КЛАСТЕРИЗАЦІЇ РЕЖИМІВ РИНКУ ТА ПРОГНОЗУВАННЯ ПОГОДИННИХ ЦІН ЕЛЕКТРОЕНЕРГІЇ УКРАЇНИ.....	22
2.1 Основи функціонального аналізу даних.....	22
2.1.1 Подання доби як функціонального спостереження.....	22
2.1.2 Базисні функції та апроксимація погодинних кривих.....	23
2.1.3 Формування функціональних предикторів і цільової змінної.....	24
2.2 Теоретичні основи алгоритму funWeightClust.....	26
2.2.1 Ідея алгоритму та область застосування.....	26
2.2.2 Математична модель кластеризації функціональних даних.....	28
2.3 Альтернативні методи кластеризації для порівняння.....	30
2.3.1 Метод K-means.....	30
2.3.2 Метод DBSCAN.....	31
2.3.3 Переваги та обмеження різних алгоритмів кластеризації.....	31
2.4 Теоретичні основи прогнозування погодинних цін.....	32
2.4.1 Постановка задачі регресії для прогнозування на 1 та 24 години	

вперед.....	32
2.4.2 Формування календарних, лагових, ковзних та режимних ознак..	33
2.4.3 Особливості використання кластерних міток у прогнозуванні.....	36
2.5 ТЕОРЕТИЧНІ ОСНОВИ АЛГОРИТМУ XGBoost.....	37
2.5.1 Принцип роботи градієнтного бустингу.....	37
2.5.2 Архітектура та параметри XGBoost.....	38
2.5.3 Переваги XGBoost для табличних часових даних.....	38
2.6 КРИТЕРІЇ ОЦІНЮВАННЯ ЯКОСТІ КЛАСТЕРИЗАЦІЇ ТА ПРОГНОЗУВАННЯ.....	39
2.6.1 Критерії BIC, ICL та Silhouette.....	39
2.6.2 Метрики MAE, RMSE, MAPE.....	41
2.6.3 Графічний аналіз прогнозних і фактичних кривих.....	42
2.7 Висновки до розділу 2.....	42
РОЗДІЛ 3 ПРОГРАМНА РЕАЛІЗАЦІЯ ІНТЕЛЕКТУАЛЬНОЇ СИСТЕМИ ПРОГНОЗУВАННЯ ТА АНАЛІЗ РЕЗУЛЬТАТІВ КЛАСТЕРИЗАЦІЇ І ПРОГНОЗУВАННЯ.....	44
3.1 Підготовка та аналіз вхідних даних.....	44
3.1.1 Джерела даних та структура вхідних таблиць.....	44
3.1.2 Первинний огляд та аудит даних.....	45
3.1.3 Очищення даних і обробка пропусків.....	50
3.1.4 Формування єдиного погодинного датасету.....	51
3.2 Побудова функціонального представлення добових профілів.....	52
3.2.1 Вибір ознак для кластеризації режимів ринку.....	52
3.2.2 Перетворення погодинних рядів у функціональні спостереження... 53	
3.2.3 Вибір параметрів базису та масштабування даних.....	54
3.3 Реалізація кластеризації режимів ринку.....	55
3.3.1 Налаштування та запуск алгоритму funWeightClust.....	55
3.3.2 Порівняння результатів з K-means і DBSCAN.....	56
3.3.3 Вибір оптимальної кількості кластерів.....	60
3.3.4 Аналіз і інтерпретація отриманих режимів ринку.....	61
3.4 Підготовка датасету для прогнозування погодинних цін.....	63
3.4.1 Формування цільової змінної та лагових ознак.....	63
3.4.2 Додавання міток кластерів до погодинного датасету.....	64
3.4.3 Розподіл даних на тренувальну, валідаційну та тестову вибірки..	64
3.5 РЕАЛІЗАЦІЯ МОДЕЛЕЙ ПРОГНОЗУВАННЯ НА ОСНОВІ XGBOOST.....	65
3.5.1 Базова модель без урахування кластеризації.....	65
3.5.2 Модель з використанням мітки кластера як ознаки.....	65
3.5.3 Окремі моделі для кожного кластера.....	66

3.6 Аналіз результатів прогнозування.....	66
3.6.1 Порівняння моделей за метриками MAE, RMSE, MAPE.....	66
3.6.2 Порівняння прогнозних і фактичних добових профілів.....	68
3.6.3 Аналіз впливу кластеризації на точність прогнозу.....	70
3.6.4 Експеримент з прогнозуванням кластера завтрашньої доби.....	71
3.7 Опис програмної реалізації системи.....	73
3.7.1 Структура програмного проєкту.....	74
3.7.2 Використані бібліотеки та інструменти.....	76
3.7.3 Інтерфейс користувача та приклади роботи системи.....	76
3.8 Висновки до розділу 3.....	78
РОЗДІЛ 4 ФУНКЦІОНАЛЬНО-ВАРТІСНИЙ АНАЛІЗ ПРОГРАМНОГО	
ПРОДУКТУ.....	80
4.1 Постановка задачі.....	80
4.2 Обґрунтування функцій програмного продукту.....	81
4.3 Вибір параметрів для програмного продукту.....	83
4.4 Експертний аналіз параметрів.....	84
4.5 Аналіз якості реалізації варіантів функцій.....	87
4.6 Економічний аналіз розробки.....	89
4.7 Вибір оптимального варіанту реалізації програмного продукту.....	92
4.8 Висновки до розділу 4.....	93
ВИСНОВКИ.....	94
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ.....	96
ДОДАТОК А. ЛІСТИНГ ПРОГРАМНОГО КОДУ.....	101
ДОДАТОК Б. ГРАФІЧНІ МАТЕРІАЛИ.....	108

ВСТУП

Ринок електроенергії займає важливе місце в сучасній економіці, оскільки впливає на функціонування промисловості, транспорту, комунальної інфраструктури та побутового сектору. Через високу мінливість енергетичних систем, збільшення перетоків енергії між державами та вплив зовнішніх факторів задача аналізу та прогнозування погодинних цін на електроенергію набуває ще більшої актуальності.

Сьогодні ринок електроенергії України функціонує в умовах значної невизначеності через часті обстріли по енергетичній інфраструктурі, які порушують потік електроенергії та спричиняють серйозний дефіцит в системі. В таких умовах складніше робити прогнози погодинної ціни на електроенергію, однак такі прогнози все рівно важливі, щоб мати змогу підготуватись до різних сценаріїв стану системи та продовжувати функціонувати на міжнародному ринку.

На формування погодинних цін впливають обсяги імпорту та експорту електроенергії, балансування системи, співвідношення попиту і пропозиції, а також особливості роботи різних сегментів ринку. Також часовим рядам електроенергетичних показників властива добова сезонність, різкі коливання та наявність різних режимів функціонування ринку. У зв'язку з цим класичні підходи до прогнозування не завжди дозволяють достатньо точно відобразити реальну динаміку цін.

Одним з варіантів підвищення точності прогнозування є попереднє виявлення типових режимів роботи ринку електроенергії. Якщо розглядати кожен день не просто як набір окремих значень, а як цілісний добовий профіль, то з'являється можливість дослідити приховану структуру ринку, виділити характерні типи днів та використати цю інформацію у подальшому прогнозуванні. Такий підхід поєднує методи функціонального аналізу даних, кластеризації та машинного навчання і дозволяє не лише побудувати точнішу

прогнозну модель, а й отримати більш змістовну інтерпретацію поведінки ринку.

Результати такого дослідження можуть бути корисними для енергетичних компаній, трейдерів, аналітичних центрів та операторів ринку для підвищення якості короткострокового прогнозування, аналізу ризиків та підтримки прийняття рішень. Крім того, запропонований підхід може бути використаний у подальших дослідженнях часових рядів та функціональних даних в енергетиці, а також у створенні інтелектуальних систем для моніторингу та прогнозування ринкових процесів.

РОЗДІЛ 1 АНАЛІЗ РИНКУ ЕЛЕКТРОЕНЕРГІЇ УКРАЇНИ ТА ПІДХОДІВ ДО КЛАСТЕРИЗАЦІЇ Й ПРОГНОЗУВАННЯ ПОГОДИННИХ ЦІН

1.1 Особливості функціонування ринку електроенергії України

Модель ринку електроенергії України функціонує на основі Закону України «Про ринок електричної енергії» [1], який визначає загальні засади організації ринку, права й обов'язки учасників та основні сегменти, на яких здійснюється купівля-продаж електроенергії. Ці сегменти поділяються на: ринок двосторонніх договорів, ринок «на добу вперед», внутрішньодобовий ринок, балансуєчий ринок та ринок допоміжних послуг. Така структура забезпечує можливість як довгострокового, так і короткострокового планування купівлі-продажу електроенергії, а також врегулювання відхилень між прогнозованими та фактичними обсягами виробництва і споживання.

Особливе значення для формування погодинних цін мають ринок «на добу вперед» та внутрішньодобовий ринок. Їхнє функціонування забезпечує ДП «Оператор ринку», яке організовує купівлю-продаж електричної енергії на цих сегментах та визначає за результатами торгів ціни та обсяги купівлі-продажу [2].

Водночас важливу роль у функціонуванні ринку відіграє балансуєчий ринок, який використовується для врегулювання небалансів, що виникають через відмінність між запланованими та фактичними обсягами виробництва або споживання електроенергії. На формування погодинних цін можуть суттєво впливати обсяги імпорту та експорту електроенергії, а також показники балансування вгору і вниз.

Додатково стан і результати функціонування ринку систематично аналізуються НКРЕКП у межах офіційного моніторингу ринку електричної енергії [1].

1.1.1 Структура ринку електроенергії України

До ключових учасників ринку належать виробники електроенергії, оператор системи передачі, оператори систем розподілу, електропостачальники, трейдери та споживачі. Торги на РДН і ВДР організовує АТ «Оператор ринку», тоді як НЕК «Укренерго» виконує функції оператора системи передачі та управляє балансуючим ринком. Ринок допоміжних послуг забезпечує надійність і стійкість роботи енергосистеми.

1.1.2 Ринок «на добу наперед», внутрішньодобовий та балансуючий ринки

Відповідно до Закону України «Про ринок електричної енергії», ринок «на добу наперед» – сегмент ринку, на якому здійснюється купівля-продаж електроенергії на наступну за днем проведення торгів добу. У межах цього сегмента учасники подають заявки наперед, а за результатами торгів формується погодинна ринкова ціна, яка виступає одним із основних індикаторів стану ринку.

Внутрішньодобовий ринок (ВДР) [3] використовується для уточнення комерційних позицій учасників ближче до моменту фактичної поставки електроенергії. Таким чином, ВДР виконує функцію оперативного коригування ринкових позицій і дозволяє зменшувати потенційні небаланси.

Організацію функціонування ринку «на добу наперед» та внутрішньодобового ринку забезпечує АТ «Оператор ринку».

Балансуючий ринок пов'язаний із забезпеченням фізичного балансу виробництва та споживання електроенергії в енергосистемі. Управління балансуючим ринком здійснює оператор системи передачі – НЕК

«Укренерго». Згідно з Правилами ринку [4], балансуючий ринок управляється окремо для кожної зони.

Для задачі кластеризації та прогнозування погодинних цін ці три сегменти мають важливе значення. Ринок «на добу наперед» формує базовий погодинний ціновий сигнал, внутрішньодобовий ринок відображає короткострокові уточнення ринкової ситуації, а балансуючий ринок характеризує реакцію енергосистеми на фактичні відхилення та необхідність балансування.

1.1.3 Роль торгових зон та вибір зони «ОЕС України» для дослідження

У межах торгових зон здійснюються торги та відображаються баланси попиту і пропозиції. Поділ ринку на торгові зони пов'язаний із технічними особливостями функціонування енергосистеми, обмеженнями мережі, режимами міжсистемних перетоків та правилами організації торгів [4].

До 24 лютого 2022 на ринку електроенергії України діяли дві зони: «ОЕС України» та «Острів Бурштинської ТЕС» (ОЕС – об'єднана енергосистема). Після в Україні діє одна торгова зона «ОЕС України», яка синхронізована з європейською мережею ENTSO-E.

ОЕС України складається з атомних, теплових, гідравлічних і гідроакумуючих електростанцій, теплоелектроцентралей та електростанцій з відновлювальних джерел енергії, магістральних електричних мереж Укренерго та розподільчих електромереж (обленерго), які об'єднані спільним режимом виробництва, передачі та розподілу електричної та теплової енергії.

Оскільки зараз діє лише одна торгова зона, для аналізу в цьому дослідженні був обраний період після 24 лютого 2022 та відповідно синхронізована з ENTSO-E «ОЕС України».

1.2 Аналіз факторів, що впливають на погодинні ціни електроенергії

Ринок електроенергії характеризується необхідністю безперервного балансування виробництва і споживання в реальному часі, що робить ціну чутливою до змін у стані енергосистеми. Тому аналіз факторів, які впливають на погодинні ціни, є необхідною передумовою для побудови коректної моделі кластеризації режимів ринку та подальшого прогнозування.

Одним із важливих факторів є співвідношення попиту і пропозиції в межах конкретної години. У години підвищеного навантаження система потребує залучення більшої кількості генеруючих потужностей, що може призводити до зростання цін. У години зниженого навантаження, навпаки, ціни можуть зменшуватися. Хоча в цьому дослідженні прямий показник попиту не використовується, добовий профіль ціни частково відображає баланс між споживанням і доступною генерацією.

Суттєвий вплив на погодинні ціни мають обсяги імпорту та експорту електроенергії. Імпорт може впливати на зниження внутрішнього дефіциту електроенергії та, відповідно, пом'якшувати зростання цін у години високого навантаження. Натомість експорт може зменшувати обсяг електроенергії, доступної на внутрішньому ринку, що за певних умов здатне підвищувати ціновий рівень. На українському ринку, особливо після синхронізації ОЕС України з ENTSO-E, міждержавні перетоки є важливим елементом ринкової рівноваги, тому обсяги імпорту та експорту доцільно враховувати як предиктори в задачі аналізу режимів ринку [5].

Також важливими чинниками є показники балансуючого ринку, зокрема обсяги балансування вгору та вниз, а також відповідні ціни балансуючої енергії. Балансування вгору виникає тоді, коли системі потрібно збільшити відпуск електроенергії для покриття фактичного дефіциту, тоді як

балансування вниз використовується для зменшення генерації у випадках надлишку електроенергії. Зростання обсягів балансування вгору зазвичай сигналізує про напружений стан системи та може супроводжуватися вищими цінами, тоді як підвищення обсягів балансування вниз може бути пов'язане з профіцитом і м'якшою ціновою динамікою. Таким чином, балансуючі показники відображають фактичну реакцію енергосистеми на відхилення від планових режимів і є важливими індикаторами ринкового стану.

Крім того, на погодинні ціни також впливають часові закономірності, насамперед добова сезонність. Для ринку електроенергії характерні повторювані добові профілі: у нічні години ціни, як правило, нижчі, тоді як у ранкові та вечірні піки вони можуть зростати. Форма добового профілю змінюється залежно від дня тижня, сезону, стану генерації та мережевих обмежень. Саме тому кожен добу доцільно розглядати як окреме функціональне спостереження, а не лише як набір незалежних погодинних точок.

Отже, погодинні ціни електроенергії в Україні формуються під впливом багатьох взаємопов'язаних факторів, серед яких найбільше значення для цього дослідження мають імпорт, експорт, обсяги балансування вгору та вниз, а також ціни балансуючого ринку. У поєднанні з внутрішньою добовою структурою цін ці показники формують характерні режими функціонування ринку.

1.3 Аналіз існуючих підходів до кластеризації часових і функціональних даних

Для виявлення типових режимів ринку можуть застосовуватися як класичні методи кластеризації, так і функціональна кластеризація [19, 22]. У цьому розділі наведено лише короткий огляд підходів, тоді як їх детальний механізм розглянуто в розділі 2.

1.3.1 Класичні методи кластеризації

Як базові алгоритми для порівняння у роботі використано K-means і DBSCAN. Перший потребує заздалегідь заданої кількості кластерів і добре працює для компактних груп, другий ґрунтується на щільності, не потребує задання K і виявляє викиди. Детальний опис їх принципів роботи, переваг і обмежень наведено в підрозділах 2.3.1–2.3.2.

1.3.2 Функціональні дані та функціональна кластеризація

Функціональний підхід розглядає кожну добу не як набір незалежних погодинних значень, а як цілісний профіль. Це дає змогу враховувати форму кривої, її гладкість і часову залежність між точками, що є важливим для ринку електроенергії. Теоретичні основи функціонального аналізу даних і модельної функціональної кластеризації детально подано в підрозділі 2.1.

1.4 Аналіз методів прогнозування часових рядів електроенергії

Для прогнозування погодинних цін можуть застосовуватися як статистичні методи, так і методи машинного навчання. У цьому розділі наведено короткий огляд підходів, а детальні теоретичні засади XGBoost і побудови ознак описано в підрозділах 2.4–2.5.

1.4.1 Статистичні методи прогнозування

До статистичних методів прогнозування належать моделі ковзного середнього, експоненційного згладжування, AR, MA, ARMA та ARIMA [6].

Вони добре працюють для трендових і сезонних рядів, але мають обмеження при складних нелінійних залежностях і багатфакторному впливі, характерних для ринку електроенергії [24].

1.4.2 Методи машинного навчання для прогнозування цін

Методи машинного навчання дозволяють враховувати лагові, календарні та ринкові ознаки без жорстких припущень про лінійність. У задачі прогнозування цін електроенергії вони є більш гнучкими, ніж класичні статистичні моделі, однак потребують якісної підготовки даних і коректного часового розбиття вибірки.

Окрім ансамблевих методів, для прогнозування цінових часових рядів застосовують і нейромережеві моделі, зокрема рекурентні та гібридні архітектури, здатні вловлювати складні часові залежності [7]. Перспективним напрямом є також генеративні моделі часових рядів на основі архітектури «кодувальник–декодувальник», які дають змогу моделювати динаміку ряду та формувати прогнозні траєкторії [8].

1.4.3 Градієнтний бустинг і XGBoost

Градієнтний бустинг – це ансамблевий метод машинного навчання, який формує прогнозну модель за рахунок послідовного поєднання слабких моделей. Часто для цього використовують дерева рішень. У цій послідовності кожна наступна модель навчається так, щоб зменшити помилки попередніх, внаслідок чого поступово покращується загальна точність прогнозування [9].

У роботі як основний алгоритм прогнозування обрано XGBoost. Алгоритм добре працює з табличними даними та різнорідними ознаками і дає змогу моделювати складні нелінійні залежності. Детальний опис принципу роботи й параметрів XGBoost наведено в підрозділі 2.5.

1.5 Формулювання задачі дослідження

У результаті аналізу предметної області, особливостей функціонування ринку електроенергії України, структури погодинних часових рядів, а також існуючих підходів до кластеризації та прогнозування, зроблено висновок, що задача дослідження полягає у побудові інтелектуальної системи, здатної виявляти типові режими функціонування ринку електроенергії та використовувати їх для прогнозування погодинних цін.

Кожна доба розглядається як цілісне функціональне спостереження, представлене погодинними профілями ціни електроенергії та функціональних предикторів. До таких предикторів у межах даного дослідження належать обсяги імпорту, обсяги експорту, обсяги балансування вгору, обсяги балансування вниз, а також ціни балансуючого ринку вгору та вниз.

Формально задачу дослідження можна подати таким чином. Введемо такі позначення.

Дано:

$D = \{d_1, d_2, \dots, d_N\}$ – множина днів спостереження ринку електроенергії України;

$t \in \{1, 2, \dots, 24\}$ – година доби;

$Y_i(t)$ – погодинна ціна електроенергії у добу d_i ;

$X_{i1}(t), X_{i2}(t), \dots, X_{im}(t)$ – погодинні функціональні предиктори для доби d_i : обсяг імпорту, обсяг експорту, обсяг балансування вгору, обсяг балансування вниз, ціна балансування вгору та ціна балансування вниз.

Результат:

R_i – режим ринку, до якого належить доба d_i ;

$\hat{Y}_i(t)$ – прогнозоване значення погодинної ціни.

Нехай $f(X, Y)$ – функція кластеризації, що на основі функціональних даних визначає режими ринку, а $g(X, R)$ – функція прогнозування, що на основі функціональних предикторів та виділеного режиму ринку формує прогноз погодинних цін. Тоді задача дослідження полягає у знаходженні таких функцій:

$$R = f(X, Y) \quad (1.1)$$

$$\hat{Y} = g(X, R) \quad (1.2)$$

де формула (1.1) визначає задачу кластеризації, що відповідає виділенню типових режимів функціонування ринку електроенергії, а формула (1.2) – задачу регресії (прогнозування погодинних цін) з урахуванням виявленого режиму ринку.

Таким чином, у роботі ставляться такі основні задачі:

- 1) проаналізувати особливості функціонування ринку електроенергії України та фактори, що впливають на погодинні ціни;
- 2) сформувати єдиний погодинний набір даних для торгової зони ОЕС України;
- 3) виконати підготовку, очищення та перетворення даних у функціональне представлення;
- 4) дослідити можливість виділення типових режимів ринку за допомогою алгоритму `funWeightClust` [10];
- 5) виконати порівняння результатів кластеризації з базовими алгоритмами K-means і DBSCAN;
- 6) побудувати модель прогнозування погодинних цін на основі XGBoost;
- 7) порівняти результати прогнозування без кластеризації, з урахуванням кластерної мітки та в межах окремих кластерів;
- 8) оцінити, чи покращує попередня кластеризація режимів ринку точність прогнозування погодинних цін електроенергії.

1.6 Висновки до розділу 1

У розділі охарактеризовано структуру ринку електроенергії України, обґрунтовано вибір торгової зони «ОЕС України» та визначено основні фактори, що впливають на погодинні ціни: імпорт, експорт, показники балансування і добову сезонність.

Показано, що для аналізу добових профілів доцільно поєднувати кластеризацію режимів ринку та методи машинного навчання для прогнозування, а як основні інструменти роботи обрано функціональну кластеризацію і XGBoost.

На цій основі сформульовано задачу дослідження – побудову інтелектуальної системи прогнозування погодинних цін із урахуванням режимів ринку.

РОЗДІЛ 2 ТЕОРЕТИЧНІ ОСНОВИ КЛАСТЕРИЗАЦІЇ РЕЖИМІВ РИНКУ ТА ПРОГНОЗУВАННЯ ПОГОДИННИХ ЦІН ЕЛЕКТРОЕНЕРГІЇ УКРАЇНИ

2.1 Основи функціонального аналізу даних

Функціональний аналіз даних (Functional Data Analysis, FDA) – це галузь статистики, в якій спостереження розглядаються як функції, визначені на певному неперервному інтервалі. Класичні методи багатовимірного статистичного аналізу інтерпретують вимірювання в кожен момент часу як незалежні координати багатовимірного простору. На відміну від цього, функціональний підхід використовує гладкість як ключове джерело інформації про досліджуваний об'єкт. Фундаментальний виклад методів функціонального аналізу даних наведено в монографії J. O. Ramsay і B. W. Silverman «Functional Data Analysis» [11], де показано, що такий підхід дає змогу побачити в даних закономірності, які залишаються прихованими при застосуванні стандартних багатовимірних методів.

У задачі аналізу ринку електроенергії функціональний підхід доцільно використати, оскільки погодинні значення ціни, імпорту, експорту чи балансування в межах однієї доби утворюють цілісний добовий профіль, який відображає режим функціонування ринку протягом 24 годин і містить інформацію не лише про окремі години, а й про загальну структуру добової динаміки.

2.1.1 Подання доби як функціонального спостереження

У межах функціонального підходу кожна доба розглядається як одне функціональне спостереження. Нехай $Y_i(t)$ – значення певної змінної

(наприклад, ціни електроенергії) у годину $t \in \{1, 2, \dots, 24\}$ для доби d_i . Тоді добова крива може бути формально записана як функція $Y_i: [1, 24] \rightarrow R$, навіть якщо в реальних даних вона представлена дискретними вимірюваннями.

Такий перехід від дискретного до неперервного представлення дає кілька важливих переваг. По-перше, погодинні значення розглядаються як вибіркові точки гладкого процесу, а не як набір незалежних чисел. По-друге, з'являється можливість оперувати похідними, інтегралами та іншими аналітичними характеристиками функції. По-третє, спрощується порівняння кривих за формою, оскільки гладкі функції природно виражають типові ринкові патерни – нічний провал, ранкове зростання споживання, вечірній піковий період.

Вхідними даними для аналізу є множина діб $D = \{d_1, d_2, \dots, d_N\}$, де для кожної доби d_i зафіксовано:

- 1) цільову функціональну змінну $Y_i(t)$ – погодинну ціну внутрішньодобового ринку;
- 2) набір функціональних предикторів $X_{i1}(t), X_{i2}(t), \dots, X_{im}(t)$ – погодинні обсяги імпорту та експорту, обсяги і ціни балансування у напрямку вгору і вниз.

Метою функціонального аналізу є виявлення закономірностей у формах цих кривих та використання отриманої інформації для подальшої кластеризації режимів ринку.

2.1.2 Базисні функції та апроксимація погодинних кривих

Для практичної роботи з функціональними даними необхідно знайти спосіб переходу від дискретних спостережень $\{Y_i(t_1), Y_i(t_2), \dots, Y_i(t_{24})\}$ до

неперервної функції $Y_i(t)$. Стандартний підхід полягає у поданні функції як лінійної комбінації базисних функцій:

$$Y_i(t) = \sum_{k=1}^K c_{ik} \varphi_k(t)$$

де $\{\varphi_k(t)\}_{k=1}^K$ – система базисних функцій, а c_{ik} – коефіцієнти, які знаходяться

шляхом апроксимації спостережуваних значень методом найменших квадратів. Розмірність вихідного представлення таким чином зменшується з 24 (кількості годин) до K (кількості базисних функцій), що одночасно згладжує шум і зберігає основні особливості кривої [11, 12].

Серед найпоширеніших базисних систем виділяють Фур'є-базис, який підходить для періодичних даних, та В-сплайн базиси для неперіодичних функцій з локальними особливостями [11, 18]. У контексті добових профілів ціни електроенергії ні періодична, ні неперіодична поведінка не є визначальними: добова крива має повторюваний характер, але водночас містить індивідуальні відхилення, пов'язані зі змінами попиту, аварійними ситуаціями та сезонними чинниками. Тому як основний базис у роботі обрано В-сплайн, адже він гнучкий, локально визначений та обчислювально ефективний.

В-сплайн порядку d є кусково-поліноміальною функцією степеня $d - 1$ із заданими вузлами. Кожна базисна функція $\varphi_{jl}(k)$ є ненульовою лише на обмеженій ділянці інтервалу спостереження, що забезпечує локальність впливу окремих коефіцієнтів і чисельну стабільність обчислень.

2.1.3 Формування функціональних предикторів і цільової змінної

Для постановки задачі прогнозування у функціональній формі необхідно розрізняти цільову функцію (відповідь, response) та функціональні предиктори (коваріати). У межах цієї роботи відповіддю $Y(t)$ є погодинна

ціна електроенергії на ВДР, а предикторами – функціональні криві обсягів та цін балансуючого ринку, а також обсягів імпорту й експорту.

Кожна змінна окремо проходить етап перетворення в В-сплайн представлення:

1. Будується сітка $\{t_1, t_2, \dots, t_{24}\}$ годинних точок доби.
2. Для кожної доби d_i та кожної змінної формується об'єкт `FDataGrid`, який відповідає дискретній фіксації функції на сітці.
3. Дискретне представлення апроксимується В-сплайн базисом (об'єкт `FDataBasis`), що дає набір коефіцієнтів $\{c_{ik}\}$.
4. Отримані функціональні представлення передаються в алгоритм кластеризації як вхідні дані.

В алгоритмі `funWeightClust` предиктори і відповідь подаються як окремі об'єкти $X(t)$ та $Y(t)$, що відповідає природі моделі функціональної лінійної регресії, на якій ґрунтується метод [10].

Програмною реалізацією функціонального подання у мові програмування Python виступають об'єкти класу `FDataBasis` з бібліотеки `scikit-fda`. Цей клас зберігає кожну функцію не як набір дискретних точок, а як вектор коефіцієнтів $(c_{i1}, c_{i2}, \dots, c_{iK})$, через які функція виражається у

вигляді $Y_i(t) = \sum_{k=1}^K c_{ik} \varphi_k(t)$. Перехід від дискретного подання (об'єкт

`FDataGrid`) до функціонального (`FDataBasis`) виконується методом `.to_basis()`, який внутрішньо реалізує апроксимацію методом найменших квадратів. Перевагами такого подання є:

- 1) зменшення розмірності – замість n дискретних точок зберігається K коефіцієнтів, де зазвичай $K \leq n$;
- 2) природне забезпечення гладкості – будь-яка лінійна комбінація гладких базисних функцій є гладкою;

3) аналітичні операції – інтеграли, похідні та скалярні добутки функцій зводяться до операцій над коефіцієнтами;

4) функціональна відстань – близькість двох функцій можна обчислити як L^2 -норму різниці їхніх коефіцієнтних векторів з урахуванням метричної матриці базису.

Алгоритми кластеризації функціональних даних, зокрема `funWeightClust`, очікують вхідні дані саме у форматі `FDataBasis`, оскільки внутрішні операції (функціональний аналіз головних компонент, оцінка коефіцієнтів функціональної регресії всередині кластерів) виражаються через коефіцієнти базисного розкладу [10].

2.2 Теоретичні основи алгоритму `funWeightClust`

2.2.1 Ідея алгоритму та область застосування

Алгоритм `funWeightClust` був запропонований у 2025 році С. Anton, С. Popescu та І. Smith з Університету MacEwan (Канада) і опублікований у відкритому доступі на arXiv [10]. Це метод кластеризації функціональних даних, що ґрунтується на сім'ї економних (parsimonious) моделей кластерно-зважених регресій (cluster weighted models, CWM).

Наведемо невелику довідку щодо вище згаданих понять.

Економні (parsimonious) моделі – це моделі з обмеженнями на структуру коваріаційних матриць і регресійних коефіцієнтів, які зменшують загальну кількість вільних параметрів. Замість того, щоб для кожного кластера оцінювати незалежну повну коваріаційну матрицю розміру $p \times p$, частина її властивостей (власні значення, орієнтація осей, внутрішня розмірність) може вважатися спільною для всіх кластерів. Алгоритм `funWeightClust` підтримує сім'ю з шести таких моделей з кодованими назвами

AKJBKQKDK, AKJBQKDK, AKBKQKDK, AKBQKDK, ABKQKDK, ABQKDK, де літери позначають властивості коваріаційної структури, а наявність літери K у відповідній позиції означає, що цей параметр може варіюватись між кластерами. Найбільш гнучкою (з усіма параметрами незалежними між кластерами) є модель AKJBKQKDK, яка обрана для основних експериментів роботи. У випадку малих обсягів даних доцільніше використовувати простіші моделі з меншою кількістю параметрів для уникнення перенавчання.

Кластерно-зважені моделі (Cluster Weighted Models, CWM) – це сім'я імовірнісних моделей, у яких сумісний розподіл функціональних предикторів $X(t)$ і функціональної відповіді $Y(t)$ моделюється як зважена суміш по G кластерах. Принципова відмінність CWM від класичних методів кластеризації полягає в тому, що кластеризація виконується одночасно у двох вимірах: за схожістю самих предикторів і за характером функціональної регресії $Y(t)$ на $X(t)$ всередині кластера. Завдяки цьому CWM розрізняє доби, які мають подібні значення предикторів, але демонструють принципово різний характер залежності між предиктором і ціною [20, 23].

На відміну від класичних методів кластеризації, які працюють лише зі змінними одного типу (тільки предиктори або тільки відповідь), `funWeightClust` одночасно використовує і функціональні предиктори, і функціональну відповідь, моделюючи їхній сумісний розподіл усередині кожного кластера.

Алгоритм продовжує методологію `funHDDC` (Functional High Dimensional Data Clustering) [13], яка вже довела свою ефективність у роботі з функціональними даними високої розмірності. Основне нововведення `funWeightClust` полягає в тому, що в межах кожного кластера будується власна функціональна лінійна регресія, яка пов'язує предиктор і відповідь. Як наслідок, кластери визначаються не лише за схожістю форм окремих кривих, а й за характером взаємозв'язку між предиктором і цільовою змінною [10].

Ця властивість робить funWeightClust особливо доречним для аналізу режимів ринку електроенергії, де економічна логіка визначає не лише типову форму ціни, але й те, як на ціну впливає, наприклад, ціна балансуючого ринку. У різних режимах функціонування ринку (звичайний, кризовий, перехідний) одне й те саме значення предиктора може давати суттєво відмінні рівні ціни ВДР, що відображається у різних коефіцієнтах функціональної регресії всередині кластерів.

2.2.2 Математична модель кластеризації функціональних даних

Нехай $X_i(t)$ – функціональний предиктор для i -го спостереження (добы), а $Y_i(t)$ – відповідна функціональна відповідь. Цільова сумісна щільність моделюється як суміш умовно функціональних розподілів:

$$f(X, Y) = \sum_{g=1}^G \pi_g \cdot f_g(X) \cdot f_g(X),$$

де G – кількість кластерів, π_g – апіорні ймовірності належності до кластера g , $f_g(X)$ – щільність розподілу предиктора в межах кластера g , а $f_g(X)$ – умовна щільність відповіді при заданому значенні предиктора всередині того самого кластера [10].

Для практичної реалізації алгоритм використовує функціональний аналіз головних компонент (Functional Principal Component Analysis, FPCA), який дає змогу зменшити розмірність функціонального простору і ефективно оцінити параметри моделі.

Функціональний аналіз головних компонент (Functional Principal Component Analysis, FPCA) є природним розширенням класичного аналізу головних компонент (PCA) на простір функцій. Якщо PCA шукає лінійні комбінації змінних, які пояснюють найбільшу частку дисперсії в скінченновимірних даних, то FPCA шукає базисні функції $\xi_1(t)$, $\xi_2(t)$, ..., які

розкладають функціональну варіацію вибірки. Кожна функціональна крива $X_i(t)$ при цьому подається у вигляді:

$$X_i(t) = \mu(t) + \sum_{j=1}^q \alpha_{ij} \xi_j(t),$$

де $\mu(t)$ – середня крива вибірки, $\xi_j(t)$ – функціональні головні компоненти, упорядковані за спаданням поясненої дисперсії, а α_{ij} – скалярні коефіцієнти (FPC scores), які характеризують відхилення кривої $X_i(t)$ від середньої вздовж j -ї головної компоненти.

Ключова перевага FPCA полягає у можливості суттєвого зменшення розмірності функціонального простору без значної втрати інформації. Замість роботи з функцією як з нескінченновимірним об'єктом достатньо зберігати кілька десятків коефіцієнтів α_{ij} , які пояснюють переважну частку варіації. Це робить FPCA ключовим інструментом для статистичного моделювання функціональних даних, оскільки оцінка коваріаційних матриць у вихідному функціональному просторі є практично неможливою.

В алгоритмі funWeightClust FPCA застосовується окремо у кожному кластері: алгоритм автоматично визначає функціональний підпростір кожної групи, у якому моделюються розподіл предикторів та функціональна регресія відповіді на предиктор. Для визначення розмірності кластерного підпростору d_k використовується критерій Cattell, який знаходить «лікоть» у графіку власних значень.

Далі параметри оцінюються за допомогою алгоритму очікування-максимізації (Expectation-Maximization, EM) у двох ітеративно повторюваних кроках:

1. Е-крок (Expectation) – обчислення апостеріорних ймовірностей належності кожного спостереження до кожного кластера на основі поточних оцінок параметрів.

2. М-крок (Maximization) – оновлення параметрів моделі (середніх, коваріаційних структур, регресійних коефіцієнтів) шляхом максимізації функції правдоподібності з урахуванням знайдених апостеріорних ймовірностей.

Алгоритм передбачає сімейство економних моделей, які відрізняються припущеннями про структуру коваріаційних матриць у кластерах [10]. У дослідженні використано модель ABQKDK для предиктора з моделлю ЕП для відповіді.

2.3 Альтернативні методи кластеризації для порівняння

Для оцінки якості та обґрунтування вибору алгоритму funWeightClust у роботі також застосовано класичні методи кластеризації – K-means і DBSCAN.

2.3.1 Метод K-means

K-means є одним із найвідоміших алгоритмів кластеризації. Він шукає таке розбиття множини об'єктів на K кластерів, при якому мінімізується сума квадратів відстаней від кожного об'єкта до центра його кластера:

$$J = \sum_{k=1}^K \sum_{x \in C_k} \|x - \mu_k\|^2$$

де μ_k – центр кластера C_k .

Алгоритм працює ітеративно, тобто на кожному кроці кожний об'єкт відноситься до найближчого центра, після чого центри перераховуються як середні значення об'єктів у кожному кластері. Процедура повторюється до збіжності.

K-means є простим і обчислювально ефективним методом, але вимагає попередньо задавати кількість кластерів K та припускає, що кластери мають приблизно однаковий розмір і сферичну форму. У випадку функціональних даних K-means працює з 24-вимірними векторами погодинних значень, не використовуючи інформацію про гладкість і функціональну природу кривих.

2.3.2 Метод DBSCAN

DBSCAN (Density-Based Spatial Clustering of Applications with Noise) – це алгоритм кластеризації на основі щільності, запропонований М. Ester, Н.-Р. Kriegel, J. Sander та Х. Xu у 1996 році [14]. На відміну від K-means, DBSCAN не вимагає заздалегідь задавати кількість кластерів і здатен виявляти кластери довільної форми.

Алгоритм працює з двома параметрами: радіусом сусідства ϵ та мінімальною кількістю точок у ϵ -сусідстві (`min_samples`). Точка вважається основною (`core point`), якщо в її ϵ -сусідстві міститься не менше `min_samples` точок. Кластери формуються як зв'язні компоненти основних точок та точок, досяжних від них за щільністю. Точки, які не належать жодному кластеру, маркуються як шум.

Перевагами DBSCAN є здатність обробляти кластери довільної форми та автоматично виділяти аномалії. Однак для функціональних даних DBSCAN, як і K-means, не використовує функціональну структуру і працює лише з евклідовою відстанню між векторами.

2.3.3 Переваги та обмеження різних алгоритмів кластеризації

Порівняння трьох розглянутих методів за теоретичними характеристиками наведено в таблиці 2.1.

Таблиця 2.1 – Порівняння funWeightClust, K-means, DBSCAN

Характеристика	funWeightClust	K-means	DBSCAN
Тип методу	Модельний (mixture-based)	геометричний	щільнісний
Робота з функціональними даними	Так, через FPCA	Ні (вектори)	Ні (вектори)
Враховує зв'язок $X \rightarrow Y$	Так	Ні	Ні
Потребує задавати K (кількість кластерів)	Так (з вибором за BIC)	Так	Ні
Виявляє аномалії	Ні	Ні	Так
Обчислювальна складність	Висока	Низька	Середня

2.4 Теоретичні основи прогнозування погодинних цін

2.4.1 Постановка задачі регресії для прогнозування на 1 та 24 години вперед

Прогнозування погодинної ціни електроенергії формулюється як задача регресії: знайти відображення $g: R^p \rightarrow R$, яке за вектором ознак X_t передбачає значення цільової змінної $\hat{y}_{t+h}$ для горизонту прогнозування h . У роботі розглядаються два горизонти:

- 1) one-step-ahead ($h = 1$) – прогноз ціни наступної години;
- 2) day-ahead ($h = 24$) – прогноз ціни через 24 години, що відповідає реальній практиці торгів на ВДР, де учасники подають заявки на наступну добу заздалегідь.

Для уникнення витоку даних (data leakage) у моделі використовуються лише ті ознаки, які реально доступні в момент прогнозу. Зокрема, для day-ahead прогнозу не використовуються лагові ознаки з лагом менше 24

годин, оскільки вони відповідають значенням ціни, які ще невідомі на момент побудови прогнозу.

Цільова змінна логарифмується: $\tilde{y} = \log(1 + y)$. Логарифмічне перетворення зменшує вплив екстремальних значень ціни на навчання моделі та покращує стабільність прогнозу. Після отримання прогнозу в логарифмічній шкалі виконується зворотне перетворення:

$$\hat{y} = \exp(\tilde{y}) - 1,$$

після чого розраховуються метрики якості в оригінальних одиницях (грн/МВт·год).

2.4.2 Формування календарних, лагових, ковзних та режимних ознак

Якість прогнозування моделі суттєво залежить від простору ознак, у якому вона навчається. Хоча алгоритм здатен виявляти складні нелінійні залежності, він не може створити інформацію з нічого: якщо ознаки не несуть прогностичної інформації про цільову змінну, точність прогнозу буде обмеженою. У задачі прогнозування погодинних цін електроенергії можна виділити три групи ознак, які відповідають різним джерелам інформації про майбутню ціну: календарні ознаки, лагові ознаки та режимні ознаки. Розглянемо кожну групу окремо з обґрунтуванням її ролі.

Календарні ознаки характеризують позицію поточної години в часі:

1) година доби (1–24) – відображає внутрішньодобову динаміку споживання, для якої характерна наявність нічного провалу, ранкового й вечірнього піків;

2) день тижня (0–6) – показує тижневу циклічність попиту, при якій робочі дні характеризуються вищим рівнем споживання порівняно з вихідними;

3) бінарний індикатор вихідного дня (is_weekend) – спрощене подання тижневого циклу, корисне для виявлення нелінійних взаємодій з іншими ознаками;

4) місяць (1–12) і квартал (1–4) – відображають сезонну динаміку, зокрема підвищене споживання в опалювальний сезон;

5) день року (1–365) – забезпечує безперервну часову координату, корисну для моделювання довгострокових трендів;

6) кругове кодування години через $\sin(\frac{2\pi h}{24})$ та $\cos(\frac{2\pi h}{24})$.

Кругове кодування години є важливим, оскільки година має циклічну природу: година 24 з точки зору моделі повинна бути близькою до години 1, а не далекою. Звичайне числове представлення цього не забезпечує, тоді як $\sin/\cos$ подання на одиничному колі вирішує задачу безпосередньо.

Календарні ознаки самі по собі мають обмежену прогностичну силу, оскільки рік від року спостерігаються відхилення від середнього сезонного профілю. Однак у поєднанні з іншими ознаками вони дозволяють виявляти умовні залежності.

Лагові ознаки ціни відображають значення цільової змінної у попередні моменти часу. У роботі використано такі лаги:

1) y_{t-1} , y_{t-2} – ціна попередньої та позапопередньої години. Ці ознаки відображають короткострокову інерцію ринку: ціна в момент $t + 1$ зазвичай близька до ціни в момент t , оскільки попит, генерація та структура балансування за одну годину змінюються поступово. Для прогнозу на 1 годину вперед це важлива група ознак;

2) y_{t-24} , y_{t-48} – ціна тієї самої години вчора та позавчора. Ці ознаки показують добовий цикл ринку: типовий рівень ціни о 10-й ранку сьогодні скоріше за все буде близьким до рівня о 10-й вранці вчора. Лаг $t - 24$ є особливо важливим для day-ahead прогнозу, де лаги менше доби недоступні;

3) y_{t-168} – ціна тієї самої години тиждень тому. Цей лаг відображає тижневу циклічність ринку: понеділок цього тижня схожий на понеділок попереднього тижня, неділя – на неділю тощо.

Сукупно ці ознаки дозволяють моделі розрізняти короткочасні коливання, добові цикли та тижневі тренди.

Ковзні характеристики ціни – агрегуючі ознаки за певний попередній період:

1) середнє за останні 24 години – характеризує загальний рівень добового профілю в період, що передує прогнозу;

2) середнє за останні 168 годин – згладжує добові коливання та показує тижневий тренд;

3) стандартне відхилення за 24 години – характеризує волатильність. Високе стандартне відхилення свідчить про великі коливання ціни всередині доби, що є типовим для кризових режимів.

Важливо, що ковзні характеристики обчислюються з попереднім зсувом часового ряду на один крок ($\text{shift}(1)$). Це гарантує, що поточне значення ціни не входить у власні ковзні ознаки, що виключає витік цільової змінної у простір ознак.

Лагові ознаки предикторів (імпорт, експорт, обсяги і ціни балансування) обчислюються з лагом 24 години – значення тієї ж години попередньої доби. Для day-ahead моделі поточні значення цих предикторів не використовуються, оскільки вони невідомі на момент прогнозу.

Режимні ознаки – це нова категорія ознак, яку запропоновано в межах цієї роботи на основі результатів кластеризації. Вони полягають у додаванні до простору ознак мітки кластера попередньої доби (cluster_lag_1d), що несе агрегуючу інформацію про тип режиму ринку, в якому функціонує система.

На відміну від календарних і лагових ознак, які описують окремі вимірні характеристики ринку, режимні ознаки відображають його

інтегральний стан. Деталі формування і використання режимних ознак розглядаються в наступному підрозділі.

2.4.3 Особливості використання кластерних міток у прогнозуванні

Найпростіший підхід – додати номер кластера поточної доби як категоріальну ознаку – є некоректним, оскільки в момент прогнозу кластер поточної доби невідомий: алгоритм кластеризації потребує повних даних добового профілю, які стають доступними лише після завершення доби.

Для розв'язання цієї проблеми у дослідженні використано ознаку `cluster_lag_1d` – номер кластера попередньої доби. Використовується така ознака, адже вчорашня доба вже завершилася, її повний профіль відомий, і для неї можна однозначно визначити кластер. Припущення, що режим ринку змінюється повільно і вчорашній режим є інформативним для прогнозу на сьогодні (або завтра), є економічно обґрунтованим.

У роботі реалізовано три варіанти прогнозування з різним рівнем використання кластерної інформації:

1. Базова модель – XGBoost без будь-якої інформації про кластери.
2. Модель з кластером як ознакою – XGBoost, у простір ознак якого додано `cluster_lag_1d`.
3. Окремі моделі по кластерах – для кожного значення `cluster_lag_1d` навчається окрема модель XGBoost.

Порівняння цих варіантів дозволяє оцінити, чи дійсно інформація про режим ринку допомагає прогнозуванню, і у якій формі це використання найефективніше.

2.5 Теоретичні основи алгоритму XGBoost

2.5.1 Принцип роботи градієнтного бустингу

XGBoost (eXtreme Gradient Boosting) – це масштабована система градієнтного бустингу дерев рішень, представлена Т. Chen і С. Guestrin у 2016 році [15]. Алгоритм належить до сімейства методів градієнтного бустингу (Gradient Boosting), ідея якого полягає в послідовному додаванні слабких базових моделей таким чином, щоб кожна наступна модель компенсувала помилки попередніх.

Формально, якщо позначити прогноз ансамблю на t -му кроці як $y_i^{(t)}$, то нова модель додається за правилом:

$$y_i^{(t)} = y_i^{(t-1)} + \eta \cdot f_t(x_i)$$

де f_t – нове дерево, а η – швидкість навчання (learning rate). Дерево f_t обирається таким чином, щоб мінімізувати функціонал:

$$L^{(t)} = \sum_{i=1}^n l(y_i, y_i^{(t-1)} + f_t(x_i)) + \Omega(f_t)$$

де l – функція втрат (для регресії квадратична), а $\Omega(f)$ – регуляризаційний член, що штрафувє складність дерева (кількість листків і норму вагів у листках). Розкладання $L^{(t)}$ у ряд Тейлора другого порядку дає аналітичні формули для оптимальних вагів у листках та критерію поділу.

Ключова відмінність XGBoost від класичного градієнтного бустингу полягає у явному включенні регуляризації, обробці пропущених значень, паралелізації побудови дерев і розширенні градієнтної інформації до другого порядку (використання гессіана) [15].

2.5.2 Архітектура та параметри XGBoost

XGBoost використовує низку гіперпараметрів, які визначають структуру і поведінку моделі. Основні з них:

- 1) `n_estimators` – кількість дерев в ансамблі (у роботі = 400);
- 2) `max_depth` – максимальна глибина дерева (у роботі = 6);
- 3) `learning_rate` (η) – швидкість навчання (у роботі = 0.05);
- 4) `subsample` – частка даних, що використовується для побудови кожного дерева (у роботі = 0.8);
- 5) `colsample_bytree` – частка ознак, що використовується для побудови кожного дерева (у роботі = 0.8);
- 6) `objective` – функція втрат (для регресії `reg:squarederror`).

Параметри `subsample` і `colsample_bytree` додають елемент випадковості до тренування, що зменшує перенавчання та покращує узагальнюючу здатність моделі. Помірно глибокі дерева ($d = 6$) у поєднанні з низькою швидкістю навчання дозволяють моделі поступово виявляти складні нелінійні взаємодії між ознаками без перенавчання.

2.5.3 Переваги XGBoost для табличних часових даних

XGBoost добре підходить для задач прогнозування на табличних даних завдяки кільком перевагам [15]:

1. Висока якість прогнозу на структурованих даних з ознаками різної природи (числові, категоріальні).
2. Здатність моделювати нелінійні залежності та взаємодії між ознаками без явного задання їхньої форми.
3. Стійкість до пропусків.
4. Регуляризація запобігає перенавчанню при великій кількості ознак.

5. Інтерпретованість через важливість ознак (feature importance) та інструменти типу SHAP (SHapley Additive exPlanations).

6. Масштабованість і ефективність обчислень – XGBoost підтримує паралелізацію і працює швидко навіть на великих наборах даних.

Для прогнозування цін електроенергії XGBoost є доречним, оскільки дані мають табличну структуру, помітні нелінійні залежності між предикторами та цільовою змінною, а лагові ознаки і кругові кодування часу легко інтегруються у простір ознак.

2.6 Критерії оцінювання якості кластеризації та прогнозування

2.6.1 Критерії BIC, ICL та Silhouette

Для оцінки якості кластеризації у моделі funWeightClust використовуються два інформаційних критерії – BIC та ICL.

BIC (Bayesian Information Criterion) оцінює якість моделі з урахуванням балансу між точністю припасування і складністю [9]. Критерій обчислюється за такою формулою [10]:

$$BIC = L^{(m_f)} - \frac{\tau}{2} \log n$$

де $L^{(m_f)}$ – максимальне значення логарифмічної правдоподібності, m_f – остання ітерація алгоритму перед збіжністю, τ – число вільних параметрів, а n – обсяг вибірки. За цією формулою більше значення критерію відповідає кращому результату.

ICL (Integrated Completed Likelihood) є модифікацією BIC, у якій додатково враховується ентропія розподілу спостережень за кластерами [10]. ICL особливо корисний у задачах кластеризації, бо штрафувє моделі з кластерами, які перетинаються.

$$ICL = BIC(K) - \sum_{k=1}^K \sum_{i=1}^n \tau_{ik} \log \tau_{ik}$$

Поєднане використання BIC та ICL дозволяє вибрати кількість кластерів, при якій модель одночасно добре припасована до даних і дає змістовно розділені кластери.

Коефіцієнт силуету (Silhouette coefficient) – це метрика оцінки якості кластеризації, що характеризує компактність кластерів і ступінь їх розділення в просторі ознак [21]. Для кожного спостереження обчислюється індивідуальний коефіцієнт силуету:

$$s(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}}$$

де $a(i)$ – середня відстань від точки i до всіх інших точок свого кластера, а $b(i)$ – середня відстань від точки i до точок найближчого сусіднього кластера.

Загальний коефіцієнт силуету для всієї кластеризації обчислюється як середнє значення індивідуальних коефіцієнтів:

$$S = \frac{1}{n} \sum_{i=1}^n s(i)$$

Значення S завжди знаходиться в діапазоні $[1, -1]$: значення вище за 0.5 свідчить про обґрунтовано хорошу структуру кластерів, від 0.25 до 0.5 – про слабку структуру, нижче 0.25 – про відсутність чіткої кластерної структури. Перевагою коефіцієнта силуету є відсутність припущень про форму кластерів та можливість порівняння кластеризацій з різною кількістю кластерів.

Слід зазначити, що коефіцієнт силуету є геометричною метрикою, тобто він оцінює компактність і розділеність кластерів виключно в евклідовому просторі ознак, не враховуючи функціональну природу даних чи зв'язок між предикторами і відповіддю. Тому для функціональних методів кластеризації (зокрема funWeightClust) значення силуету може систематично поступатися геометричним методам (K-means), що оптимізують саме компактність кластерів за своєю цільовою функцією.

2.6.2 Метрики MAE, RMSE, MAPE

Для оцінки якості прогнозу у роботі використано три метрики [6].

MAE (Mean Absolute Error) – середня абсолютна похибка. Формула має такий вигляд:

$$MAE = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i|$$

MAE має ту саму одиницю вимірювання, що й цільова змінна (грн/МВт·год), і легко інтерпретується (можемо інтерпретувати як те, на скільки гривень у середньому модель помиляється).

RMSE (Root Mean Squared Error) – корінь середньоквадратичної похибки:

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2}$$

RMSE сильніше штрафує великі помилки, ніж MAE, і тому є чутливою до екстремальних відхилень. Якщо модель добре працює на середніх цінах, але значно помиляється на пікових значеннях, значення RMSE буде помітно більшим, ніж MAE.

MAPE (Mean Absolute Percentage Error) – середня абсолютна відсоткова похибка:

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

MAPE дозволяє інтерпретувати помилку у відносних одиницях. Однак вона має істотний недолік: при дуже малих значеннях y_i MAPE різко зростає, що для ринку електроенергії з періодами майже нульових нічних цін призводить до завищених значень. У роботі MAPE наводиться як додаткова метрика, але основними вважаються MAE та RMSE.

Додатково розраховується коефіцієнт детермінації R^2 , який показує частку дисперсії цільової змінної, яка пояснюється моделлю [9].

$$R^2 = 1 - \frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{\sum_{i=1}^n (y_i - \bar{y})^2}$$

де y_i – фактичне значення, $\hat{y}_i$ – прогнозне значення, $\bar{y}$ – середнє значення цільової змінної.

2.6.3 Графічний аналіз прогнозних і фактичних кривих

Числові метрики дають загальну оцінку якості моделі, але не показують, які саме особливості ціни модель відтворює добре, а які – погано. Тому в дослідженні застосовується графічний аналіз. Поєднання числових метрик і графічного аналізу дозволяє зробити обґрунтовані висновки щодо реальної якості та практичної придатності моделі прогнозування.

2.7 Висновки до розділу 2

У цьому розділі було викладено теоретичну основу для побудови інтелектуальної системи прогнозування погодинних цін електроенергії на основі кластеризації режимів ринку. Розглянуто базові поняття функціонального аналізу даних, які дозволяють представити кожну добу як єдине функціональне спостереження і працювати з гладкими кривими через апроксимацію В-сплайн базисом.

Описано теоретичні основи алгоритму funWeightClust. Для порівняння наведено два класичні методи кластеризації – K-means та DBSCAN – і вказано їхні принципові обмеження у застосуванні до функціональних даних.

Сформульовано задачу прогнозування погодинних цін як задачу регресії з горизонтами 1 година (one-step-ahead) і 24 години (day-ahead). Описано принципи формування простору ознак (календарні, лагові, ковзні характеристики) та наведено стратегію використання кластерної інформації.

Описано теоретичні основи алгоритму XGBoost. Розглянуто його ключові переваги, що роблять його придатним для задачі прогнозування цін електроенергії.

Сформовано систему критеріїв оцінювання якості кластеризації (BIC, ICL, Silhouette) і прогнозування (MAE, RMSE, MAPE, R^2 , графічний аналіз).

Усі розглянуті теоретичні елементи будуть використані у практичній частині роботи у наступному розділі для побудови, навчання, оцінки та порівняння моделей кластеризації та прогнозування.

РОЗДІЛ 3 ПРОГРАМНА РЕАЛІЗАЦІЯ ІНТЕЛЕКТУАЛЬНОЇ СИСТЕМИ ПРОГНОЗУВАННЯ ТА АНАЛІЗ РЕЗУЛЬТАТІВ КЛАСТЕРИЗАЦІЇ І ПРОГНОЗУВАННЯ

3.1 Підготовка та аналіз вхідних даних

3.1.1 Джерела даних та структура вхідних таблиць

Для побудови інтелектуальної системи прогнозування погодинних цін електроенергії на основі кластеризації режимів ринку використано відкриті дані ринку електроенергії України з порталу Energy Map [5, 16, 17]. Усі вхідні таблиці охоплюють період з 24 лютого 2022 року по 30 вересня 2025 року включно (обрання цього періоду було обґрунтовано в першому розділі) та містять погодинні значення для торгової зони «ОЕС України (синхронізована з ENTSO-E)».

Датасет складається з трьох таблиць у форматі Microsoft Excel:

1. Результати торгів ВДР – погодинні ціни внутрішньодобового ринку (у програмній реалізації таблиця матиме назву *vdr*). Містить колонки «Доба постачання», «Година» (1-24), «Ціна, грн/МВт·год», «Торгова зона», а також додаткові метрики (мінімальна, максимальна та остання ціна за годину, обсяги продажу та купівлі у МВт·год, заявлені обсяги продажу та купівлі, попит, гранична ціна). «Ціна» є цільовою змінною роботи.

2. Результати балансуючого ринку – погодинні обсяги та середньозважені ціни врегулювання небалансу у двох напрямках (*balancing*). Містить колонки «Дата», «Операційний період», що відповідає годині роботи, «Торгова зона», а також «Обсяг у напрямку вгору», «Обсяг у напрямку вниз», «Середньозважена ціна у напрямку вгору», «Середньозважена ціна у

напрямку вниз», які виступають функціональними предикторами для кластеризації та регресійними ознаками для XGBoost.

3. Імпорт-експорт електроенергії – погодинні обсяги перетоків між Україною та сусідніми країнами (import_export). Містить колонки «Дата», «Час», «Торгова зона», «Митний режим», «Країна», «Обсяг». Дані подано у розрізі п'яти країн (Польща, Молдова, Румунія, Словаччина, Угорщина) та митного режиму (імпорт/експорт). Для кожної години доби в таблиці може бути до 10 рядків (5 країн $\times$ 2 напрямки).

Усі три таблиці мають спільну часову структуру, де кожна доба представлена 24 рядками. Це дає змогу об'єднати їх у єдиний погодинний датасет з ключем «дата + година». Загальний обсяг сирих даних становить близько 32 тисяч рядків для кожної з таблиць ВДР і балансування та близько 318 тисяч рядків для імпорту/експорту (через розгортання за країнами).

3.1.2 Первинний огляд та аудит даних

Перед очищенням і агрегацією було виконано аудит вхідних даних, метою якого було виявити аномалії, оцінити повноту та зрозуміти базові статистичні характеристики ринку. Були проведені такі етапи дослідження:

1) перевірка діапазонів дат та покриття для кожної з трьох таблиць. Таблиця import_export містить загалом діапазон від 2022-02-24 до 2025-10-12 (1327 діб), таблиця balancing – від 2022-02-24 до 2025-09-30 (1315 діб), а таблиця vdr – від 2022-02-24 до 2025-10-13 (1327 діб). Отже, був взятий спільний діапазон від 2022-02-24 до 2025-09-30 (1315 діб);

2) підрахунок пропущених значень у кожній колонці. Таблиця import_export не містила пропуски. Таблиця balancing містила по 57 пропусків в колонках «Обсяг у напрямку вгору» та «Середньозважена ціна у напрямку вгору», а також по 450 у відповідних колонках балансування вниз.

Таблиця *vdr* містила по 74 пропуски в колонках «Мінімальна ціна», «Максимальна ціна».

- 3) візуалізація середньої добової ціни ВДР і обсягів імпорту/експорту впродовж узятго періоду часу (рис. 3.1 та рис. 3.2);
- 4) побудова прикладів добових профілів цін для деяких діб (рис. 3.3);
- 5) аналіз розподілів цін балансування вгору та вниз (рис. 3.4);
- 6) розрахунок середніх погодинних профілів для всіх предикторів та цільової змінної (рис. 3.5).

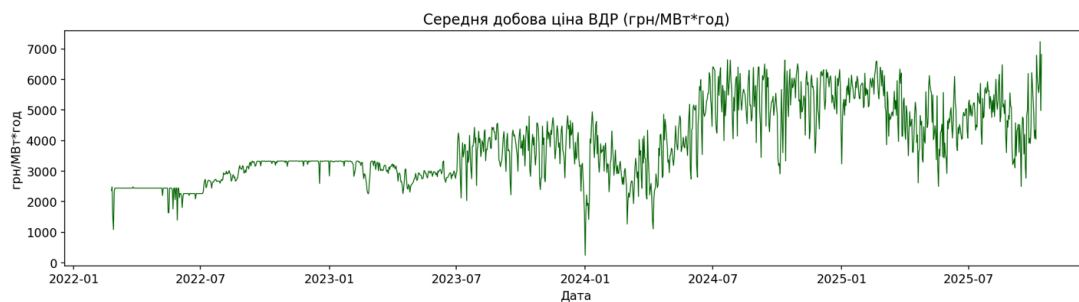


Рисунок 3.1 – Середня добова ціна ВДР

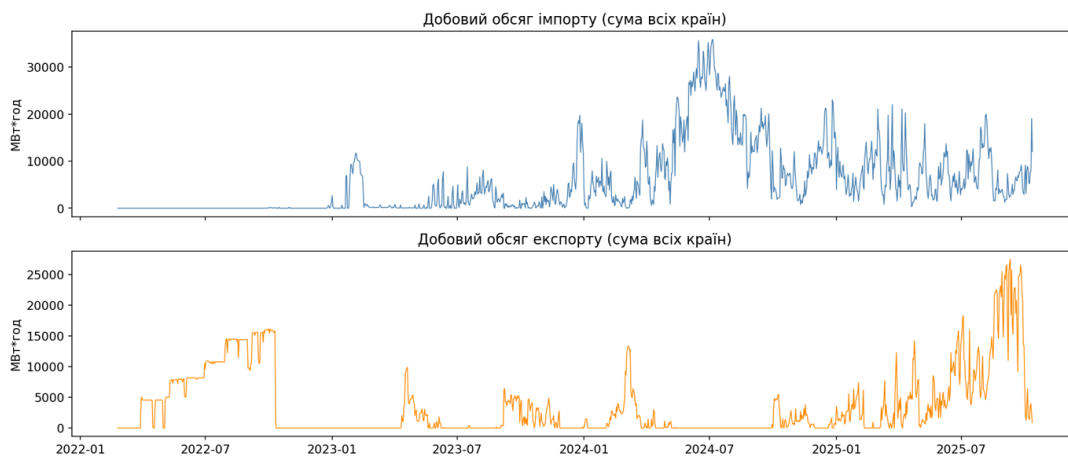


Рисунок 3.2 – Обсяги імпорту/експорту по всіх країнах

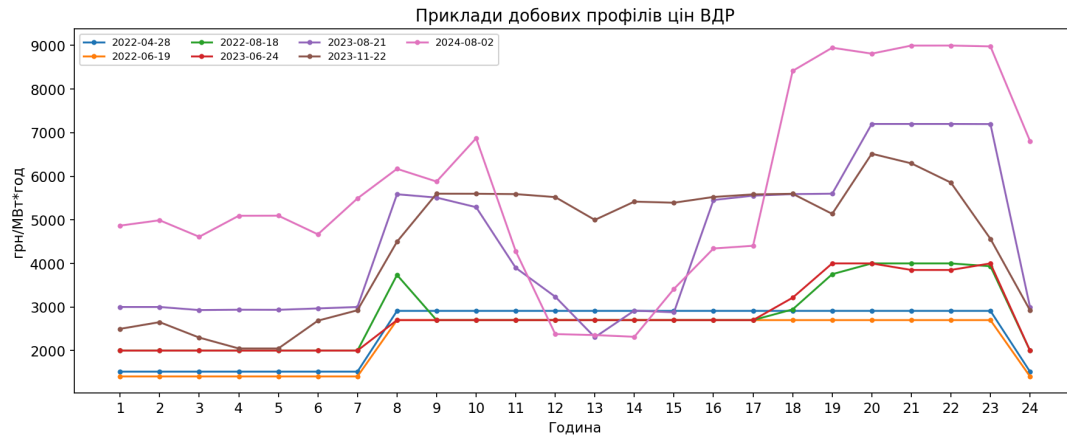


Рисунок 3.3 – Приклади добових профілів цін ВДР

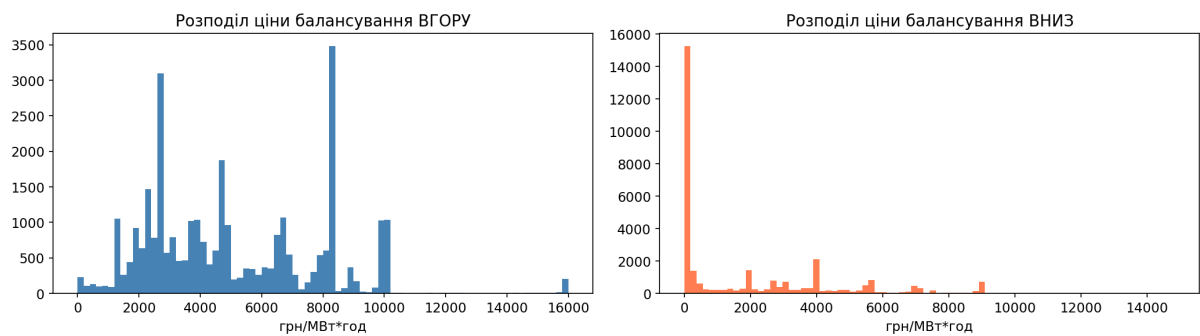


Рисунок 3.4 – Розподіли цін балансування вгору та вниз

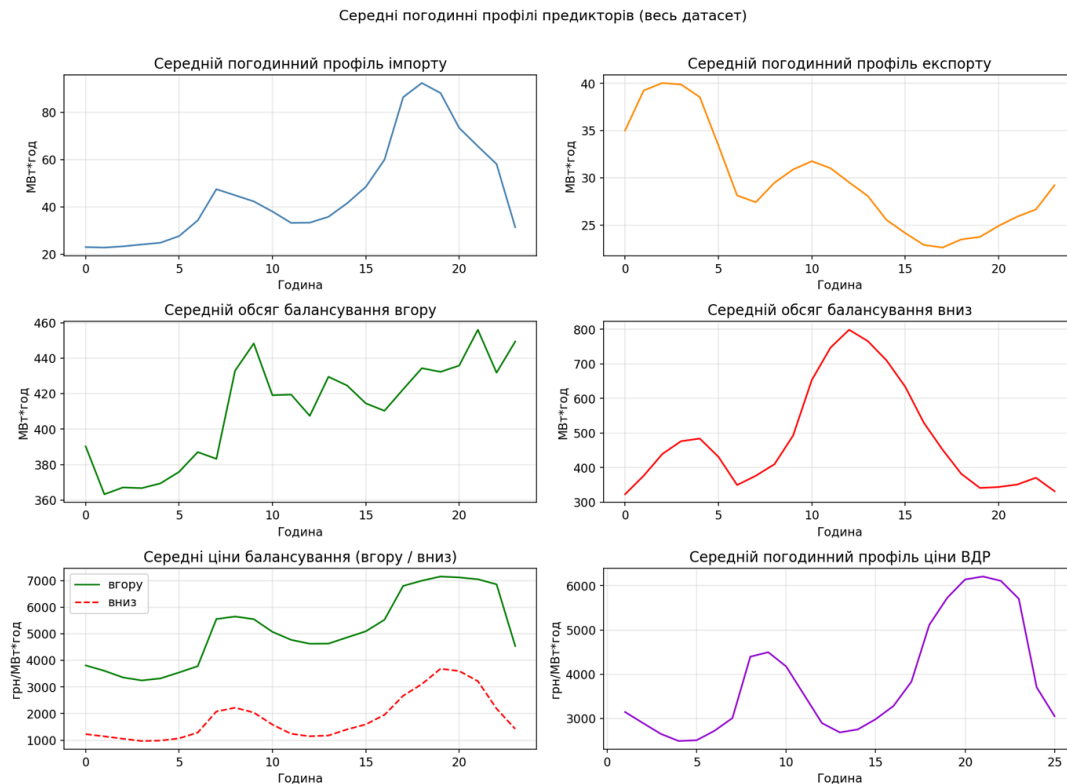


Рисунок 3.5 – Середні погодинні профілі предикторів та цільової змінної

Динаміка середньої добової ціни ВДР (рис. 3.1) демонструє висхідний тренд протягом всього періоду спостереження. На початку 2022 року середня добова ціна стабільно тримається близько 2400 грн/МВт·год, що відповідає періоду стабільності відразу після початку повномасштабного вторгнення. До кінця 2022 і протягом 2023 року спостерігається поступове зростання до рівня 3000-3500 грн/МВт·год. Потім з середини 2023 йде стрибок до 4000-4500 грн/МВт·год. Найпомітніший стрибок припадає на середину 2024 року, коли середня ціна сягає 5000-6500 грн/МВт·год, що збігається в часі з масовими атаками на енергетичну інфраструктуру України. У 2025 році ціна досі на високому рівні з періодичними піками понад 7000 грн/МВт·год.

Обсяги імпорту та експорту (рис. 3.2) показують протилежні в часі режими функціонування. Імпорт практично відсутній у 2022 році (близько 0 МВт·год для більшості днів), починає поступово зростати з кінця 2022 року і досягає піку в середині 2024 року з обсягами 30 000-35 000 МВт·год на добу. Експорт, навпаки, був активним у 2022 році (4000-15 000 МВт·год), потім практично зникає у 2023 році з кількома піками протягом 2023-2024 років і поновлюється у 2025 році з високими обсягами понад 25 000 МВт·год. Така динаміка чітко відображає фази балансу енергосистеми: спочатку Україна експортувала надлишок, потім переходила на імпорт через дефіцит і повернулася до експорту після відновлення генеруючих потужностей у 2025 році.

Розподіли цін балансуючого ринку (рис. 3.4) виявляють різні характеристики двох напрямків. Розподіл цін балансування у напрямку вгору є мультимодальним з кількома вираженими піками (2500, 5500, 8000 грн/МВт·год), що відображає типові цінові режими дефіциту електроенергії. Розподіл цін у напрямку вниз асиметричний з домінуючим піком близько нуля і коротким хвостом до 8000 грн/МВт·год – це свідчить про те, що в більшості випадків балансування у напрямку вниз відбувалось за дуже низькими цінами.

Середні погодинні профілі предикторів (рис. 3.5) показують типову внутрішньодобову динаміку ринку. Середній профіль ціни ВДР має двомодальну форму з ранковим піком близько 9-ї години (~4500 грн/МВт·год), денним провалом близько 13-14 годин (~2700 грн/МВт·год) і вираженим вечірнім піком близько 20-22 годин (~6000 грн/МВт·год). Найнижчі ціни спостерігаються в нічні години (3-5 ранку, ~2500 грн/МВт·год). Аналогічну форму має профіль цін балансування – обидва піки тут синхронізовані з піками ціни ВДР, що підтверджує сильний зв'язок між цими двома показниками. Профіль імпорту досягає максимуму у вечірні години (17–18 годин, до 90 МВт·год), коли система найбільше потребує додаткової електроенергії, тоді як експорт навпаки максимальний у ранкові години (1–4 години), коли є надлишок генерації при низькому споживанні.

Приклади добових профілів цін ВДР (рис. 3.3) демонструють неоднорідність функціонування ринку у різні періоди. Були взяті випадкові 7 діб протягом усього періоду. У 2022 році добові профілі мали досить стабільний вигляд через регулювання цін з фіксованими рівнями. У 2023 році з'являється виражена внутрішньодобова варіативність: ранковий і вечірній піки чітко виокремлюються. У 2024 році (профіль 02.08.2024) ціна тримається на високому рівні протягом всього дня з ранковими значеннями вже близько 5000 грн і вечірніми піками до 9000 грн, що відображає період найбільшого дефіциту електроенергії.

Аудит підтвердив доцільність кластеризації режимів ринку: різні періоди функціонування ринку демонструють різні форми добових профілів.

Також аудит виявив такі проблемні місця в даних:

- 1) у колонці з цінами ВДР є 74 пустих значень у 2022 році;
- 2) пропуски в обсягах балансування – 57 рядків у напрямку вгору і 450 рядків у напрямку вниз;
- 3) нульові добові профілі у 2022 році з адміністративним регулюванням створюють штучну сегрегацію кластерів за часом;
- 4) послідовні нулі експорту в 2023 році.

Усі виявлені проблеми обробляються на етапі очищення даних.

3.1.3 Очищення даних і обробка пропусків

Після аудиту було проведено очищення даних та був отриманий чистий регулярний погодинний датасет. Далі наведені усі етапи обробки.

Уніфікація часових індексів. Усі три таблиці мають різні позначення часу: ВДР використовує номер години 1–24, балансування і імпорт/експорт – інтервали типу «00:00–01:00». Реалізовано функцію `parse_hour_from_range`, яка переводить рядкове представлення в цілочисельне. Дата зведена до нормалізованого формату `datetime` без часової складової. Уніфікація дозволила виконувати об'єднання таблиць за ключем (дата, година).

Виправлення нерегулярностей через переведення годинника. В Україні щороку переводять годинник (останній тиждень березня – на літній час, останній тиждень жовтня – на зимовий). Це породжує дві аномалії:

1) доби з 25 годинами (зимовий час) – у даних з'являється дублікат години, який видаляється правилом `drop_duplicates`;

2) доби з 23 годинами (літній час) – одна година відсутня, у роботі вона відновлюється лінійною інтерполяцією (обчислюється середнє значення) між сусідніми годинами через спеціальну функцію `interpolate_missing_hours`.

Для ВДР у деяких добах присутня штучна година 25, яка не має фізичного сенсу і видаляється окремо.

Заповнення пропусків у колонках балансування. Для обсягів балансування пропуски інтерпретуються як відсутність балансуючих заявок і заповнюються нулями. Для цін балансування пропуски заповнюються медіаною по відповідній годині доби – це зберігає типовий добовий профіль ціни і не вносить штучних викидів. Заповнення нулями штучно знизило б середній рівень цін балансування і ввело б у модель хибну інформацію про доступність дешевої електроенергії. Натомість медіана за годиною дає

нейтральне значення, яке відповідає типовій ціні балансування у відповідну годину доби, не порушуючи структури даних. Вибір медіани (а не середнього) обумовлений сильною асиметрією розподілу цін балансування з довгим правим хвостом.

Обробка нульових цін. Виявлено 74 рядки з нульовою ціною у 2022 році. Замість видалення, що порушило б регулярність часового ряду, нулі замінюються пропусками і заповнюються лінійною інтерполяцією.

Логарифмічне перетворення цільової змінної. Через високу волатильність цін ВДР (від десятків до 14 997 грн/МВт·год) до колонки ціни застосовано перетворення $\log(1 + y)$. Логарифмування зменшує асиметрію розподілу та робить навчання XGBoost стійкішим до екстремальних значень. У вихідному датасеті збережено оригінальну ціну та її логарифм.

Заповнення пропущених діб. Для трьох таблиць виявлено різницю в часовому покритті: одна доба (2025-06-20) була повністю відсутня у VDR, але присутня у балансуванні та імпорту/експорту. Замість виключення доби побудовано повну сітку дат у спільному діапазоні і застосовано лінійну інтерполяцію для відсутніх рядків. Це гарантує безперервність часового ряду, що є критичною умовою для коректного формування лагових ознак.

3.1.4 Формування єдиного погодинного датасету

На останньому етапі очищення даних виконується агрегація імпорту/експорту та об'єднання трьох таблиць в один погодинний датасет.

Агрегація імпорту/експорту. Початкова таблиця містить рядки в розрізі країн і напрямків. За допомогою операцій `groupby` і `unstack` дані згортаються до двох колонок: «імпорт_МВт» (сума по всіх країнах) та «експорт_МВт» (сума по всіх країнах) для кожної комбінації (дата, година).

Далі виконується послідовне об'єднання трьох очищених таблиць методом `merge` за ключем (дата, година). Результуюча master-таблиця має такі

колонки: дата, година, ціна_МВт, log_ціна_МВт, імпорт_МВт, експорт_МВт, бал_вгору_МВт, бал_вниз_МВт, бал_вгору_грн, бал_вниз_грн.

Фінальний датасет налічує 31 560 рядків (1315 діб $\times$ 24 години). Ця таблиця є вхідною для подальших етапів побудови функціональних спостережень для кластеризації та формування ознак для XGBoost.

3.2 Побудова функціонального представлення добових профілів

3.2.1 Вибір ознак для кластеризації режимів ринку

Перед безпосереднім запуском алгоритму кластеризації funWeightClust необхідно перетворити погодинні дані у функціональний формат і визначити, які саме функціональні змінні будуть використані як предиктори для алгоритму.

Master-таблиця в погодинному форматі шляхом pivot-агрегації перетворюється на функціональну таблицю, де кожна доба представлена одним рядком з 24 погодинними значеннями для кожної з 7 змінних. Деталі цього перетворення розглядаються в підрозділі 3.2.2.

Експериментальний вибір предикторів. Алгоритм funWeightClust може приймати як один функціональний предиктор, так і кілька одночасно. Однак збільшення кількості предикторів збільшує розмірність функціонального простору і може погіршити збіжність ЕМ-алгоритму при обмеженому обсязі даних. Тому оптимальну конфігурацію визначено експериментально.

На першому етапі алгоритм funWeightClust запущено для 2 кластерів окремо для кожного з шести предикторів: імпорту, експорту, обсягів і цін балансування у двох напрямках. Цільовою змінною в усіх запусках виступала ціна ВДР. Результати показали:

1) найвище значення BIC (−30 752) досягається для предиктора бал_вгору_грн (ціна балансування у напрямку вгору);

- 2) наступний за якістю предиктор – експорт_МВт ($BIC = -32\,856$);
- 3) предиктори, пов'язані з обсягами балансування, дають значно гірші результати.

Детальніші результати показані в таблиці 3.1.

Таблиця 3.1 – Результати кластеризації з одним предиктором

predictor	converged	bic	icl	cluster_sizes
імпорт_МВт	True	-34111.610319	-34070.205125	[814, 501]
експорт_МВт	True	-32856.693268	-32820.974528	[942, 373]
бал_вгору_МВт	True	-42797.722461	-42715.323838	[542, 773]
бал_вниз_МВт	False	-38396.571146	-38356.202607	[319, 996]
бал_вгору_грн	True	-30752.974323	-30720.964762	[1010, 305]
бал_вниз_грн	False	-36510.115578	-36463.220483	[952, 363]

На другому етапі перевірено комбінації пар предикторів. Однак для більшості пар алгоритм не зміг збігтися ($status=failed$) через високу розмірність функціонального простору і обмежений обсяг тренувальної вибірки. Рідкісні випадки збіжності давали результати, які поступалися найкращому одиничному предиктору.

На основі цих експериментів за основний предиктор для funWeightClust обрано бал_вгору_грн. Ціна балансування у напрямку вгору відображає рівень дефіциту електроенергії в системі, який є ключовою характеристикою режиму ринку. Цільовою функцією залишається ціна ВДР ціна_МВт.

3.2.2 Перетворення погодинних рядів у функціональні спостереження

Для побудови функціональних спостережень виконано pivot-агрегацію master-таблиці. Тоді кожна доба, яка в погодинному форматі представлена 24 рядками, перетворюється на одне функціональне спостереження – рядок з 24

значеннями кожної змінної. Колонки відповідної таблиці отримують назви виду $\langle \text{змінна} \rangle_1, \langle \text{змінна} \rangle_2, \dots, \langle \text{змінна} \rangle_24$.

Вихідна таблиця має 1315 рядків (по одному на добу) і 169 колонок (1 колонка дати + 7 змінних $\times$ 24 години). У результаті отримали функціональне подання всього періоду спостережень.

Для подальшого використання таблиця перетворюється в об'єкти `FDataBasis` бібліотеки `scikit-fda`. Послідовність перетворення наступна:

1. Дані змінної у вигляді матриці 1315×24 передаються у конструктор `FDataGrid` разом із сіткою точок (спільна сітка для предиктора та відповіді $\{1, 2, \dots, 24\}$). Така постановка означає, що алгоритм `funWeightClust` моделює сумісну структуру цін балансування і цін ВДР протягом усього добового циклу, а не лише в окремі його частини. Це дозволяє виявляти цілісні режими функціонування ринку, у яких певна форма предиктора (наприклад, високі ціни балансування протягом ранкових годин) супроводжується певною формою відповіді (наприклад, високими цінами ВДР у вечірні години).
2. Викликається метод `to_basis(BSplineBasis(n_basis=4))`, який апроксимує дискретні точки В-сплайн-базисом.
3. Об'єкт `FDataBasis` має матрицю коефіцієнтів розмірністю 1315×4 , яка є компактним функціональним поданням добових кривих.

3.2.3 Вибір параметрів базису та масштабування даних

Вибір кількості базисних функцій є важливим етапом, адже вони контролюють гладкість представлення. Менші значення дають більш згладжені криві, але можуть втратити деталі, у той час як більші значення зберігають деталі, але можливе перенавчання.

Як кількість базисних функцій В-сплайн базису обрано значення `n_basis=4`. Експериментально перевірено також варіанти `n_basis=5` і `n_basis=6` у багатопредикторній постановці, однак для цих конфігурацій алгоритм не

зміг досягти збіжності через занадто високу розмірність функціонального простору. Кубічний В-сплайн з чотирма базисними функціями дає 4 коефіцієнти на криву, що достатньо для гладкого подання загальної форми добового профілю. Значення нижче 4 (наприклад, `n_basis=3`) втрачає виразність, а вище (`n_basis=6`) ускладнює оцінку параметрів алгоритму.

Перед кластеризацією виконується стандартизація (z-score) кожної змінної окремо. Це необхідно тому, що різні предиктори мають принципово різні масштаби: ціна вимірюється тисячами гривень, обсяги – десятками й сотнями МВт·год. Стандартизація реалізована через `StandardScaler` з бібліотеки `scikit-learn` окремо для кожної з 7 змінних. Параметри нормалізації зберігаються для подальшого застосування зворотного перетворення при візуалізації результатів.

3.3 Реалізація кластеризації режимів ринку

3.3.1 Налаштування та запуск алгоритму `funWeightClust`

Алгоритм `funWeightClust` реалізовано як Python-пакет, який отримано з GitHub-репозиторію авторів і інтегровано до проекту. Алгоритм викликається через головну функцію `funweightclust(datax, datay, K, model, modely, ...)`.

У роботі використано таку конфігурацію алгоритму:

- 1) `datax = [fd_бал_вгору_грн]` – список з одного функціонального предиктора;
- 2) `datay = fd_ціна_МВт` – функціональна цільова змінна;
- 3) `K` – кількість кластерів (тестовано значення 2, 3, 4, 5);
- 4) `model = "ABQKDK"` – економна модель для предиктора (має середній рівень гнучкості);
- 5) `modely = "ЕП"` – модель для відповіді. У ній закладений однаковий об'єм між кластерами, сферична форма та орієнтація без поворотів;

- 6) `init = "kmeans"` – ініціалізація K-means;
- 7) `criterion = "bic"` – критерій вибору моделі;
- 8) `nb_rep = 10` – кількість повторних запусків з різними початковими наближеннями (для стабільності збіжності);
- 9) `seed = 42` – фіксація випадковості для відтворюваності результатів.

Як основну функціональну модель алгоритму обрано конфігурацію ABQKDK-EП. Вибір моделі обумовлений потребою забезпечити стабільну збіжність ЕМ-алгоритму на досліджуваному наборі даних. Попередні експерименти показали, що використання найгнучкіших моделей сімейства `funWeightClust` (наприклад, AKJBKQKDK-VVV) призводить до нестійкої збіжності при заданому обсязі вибірки і однопредикторній постановці задачі – алгоритм або не виконував критерій збіжності, або повертав вироджені кластери з кількістю спостережень меншою за параметр `min_individuals`. Натомість більш економна модель ABQKDK-EП забезпечила стабільну збіжність, що дозволило отримати відтворювані результати кластеризації.

Для кожного значення кількості кластерів K алгоритм запускався окремо, і зберігався отриманий об'єкт `FunWeightClust` з полями `cl` (мітки кластерів), `bic`, `icl`, `converged`.

3.3.2 Порівняння результатів з K-means і DBSCAN

Для оцінки переваг і недоліків `funWeightClust` проведено порівняння з результатами алгоритмів K-means та DBSCAN. Для реалізації була використана бібліотека `scikit-learn`. K-means та DBSCAN запускались як для просто стандартизованих значень (24 колонки `бал_вгору_грн_{h}`), так і для BSpline коефіцієнтів цього предиктора, на яких і запускався `funWeightClust`. Обидва підходи показали схожі результати, тому нижче наведені значення коефіцієнту силуету для першого підходу.

K-means запущено з параметром `n_clusters=4` (співпадає з оптимальним K для `funWeightClust`), `random_state=42`, `n_init=10`.

DBSCAN запущено з параметрами `min_samples=10` та `eps=2`, значення якого було обрано на основі графіку k -distance для визначення оптимального `eps` (рис. 3.6).

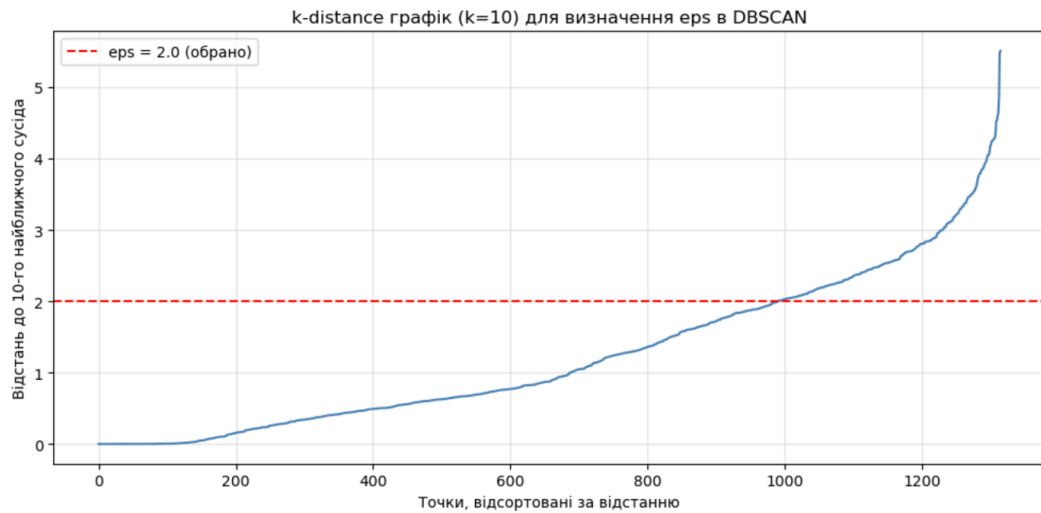


Рисунок 3.6 – k -distance графік для визначення `eps` в DBSCAN

Алгоритм виявив 3 кластери та 213 шумових точок (~16% даних). Далі шумові точки було оброблено методом найближчого сусіда, де для кожної шумової точки знайдено найближчу не-шумову точку в евклідовому просторі ознак і присвоєно їй мітку кластера. Реалізація використовує клас `NearestNeighbors` з бібліотеки `scikit-learn` з параметром `n_neighbors=1`. Така обробка дозволяє використовувати DBSCAN на однакових умовах з K-means і `funWeightClust`, які за визначенням не маркують точки як шум.

Далі порівнюємо результати кластеризації цих трьох алгоритмів. Слід зазначити, що автори алгоритму `funWeightClust` у первинній публікації [10] оцінюють якість кластеризації переважно за метрикою Adjusted Rand Index (ARI), яка порівнює знайдену кластеризацію з істинною структурою даних. Така оцінка можлива лише для синтетичних або бенчмаркових наборів даних, де істинна кластерна структура відома.

У реальних даних ринку електроенергії України істинна кластеризація режимів відсутня – саме виявлення цих режимів і є метою дослідження. Тому пряме використання ARI для оцінки якості кластеризації неможливе.

Для порівняння результатів використана метрика Silhouette, значення якої наведено в таблиці 3.2. Інформаційні критерії BIC та ICL природно визначені лише для імовірнісних (model-based) методів кластеризації, які мають явну функцію правдоподібності. До таких належать funWeightClust, а алгоритми K-means і DBSCAN не є імовірнісними методами і не мають функції правдоподібності, тому формально для них BIC не визначений.

Таблиця 3.2 – Порівняння алгоритмів кластеризації за Silhouette

Метод	К кластерів	Silhouette
funWeightClust	4	0.183
K-means	4	0.451
DBSCAN	3	0.504

Отже, як і було зазначено раніше, за коефіцієнтом силуету funWeightClust поступається K-means та DBSCAN. Однак ця метрика оцінює лише геометричну компактність кластерів і не враховує функціональну природу даних і зв'язок між предиктором та цільовою змінною. Тому для змістовного порівняння методів використано також практичну метрику – якість прогнозування ціни, побудованого на основі кластерів.

Було виконано додатковий візуальний аналіз за трьома напрямками: порівняння середніх добових профілів ціни ВДР, 2D-проекція об'єктів у просторі BSpline-коефіцієнтів та аналіз розподілу міток кластерів у часі.

Порівняння середніх профілів (рис. 3.7) показує, що funWeightClust виділяє чотири змістовно різні режими з чіткою внутрішньодобовою формою: базовий режим зі слабкою варіацією, режим із помірним денним провалом, режим із підвищеними денними значеннями та режим із найбільш глибоким опівденним провалом і високими ранковими та вечірніми цінами. K-means також формує чотири кластери, але вони більше орієнтовані на

геометричне розділення в просторі ознак і дають менш інтерпретовану часову динаміку. DBSCAN фактично стискає структуру до трьох режимів, один з яких є дуже малим, тому його розбиття є грубішим.

На 2D-проекції PCA (рис. 3.8) видно, що K-means і DBSCAN дають візуально компактніші кластери, оскільки вони безпосередньо оптимізують геометричне розділення у просторі коефіцієнтів. Натомість кластери funWeightClust частково перекриваються в перших двох головних компонентах. Це узгоджується з природою алгоритму: він враховує не лише геометрію предиктора, а й спільну структуру предиктора та цільової змінної, тому кластери можуть бути менш компактними, але змістовно ближчими до реальних режимів ринку.

Розподіл міток у часі (рис. 3.9) показує, що всі три методи виявляють виражену часову неоднорідність ринку. Водночас для K-means і особливо для DBSCAN кластери формують майже хронологічні блоки. funWeightClust також виділяє домінуючий базовий кластер для більшої частини 2022–початку 2024 років, але додатково відносить частину діб 2024–2025 років до окремих підкластерів, що відображають відмінні форми добових профілів, а не лише календарний період. Це робить його зручнішим для подальшого використання в прогнозуванні.

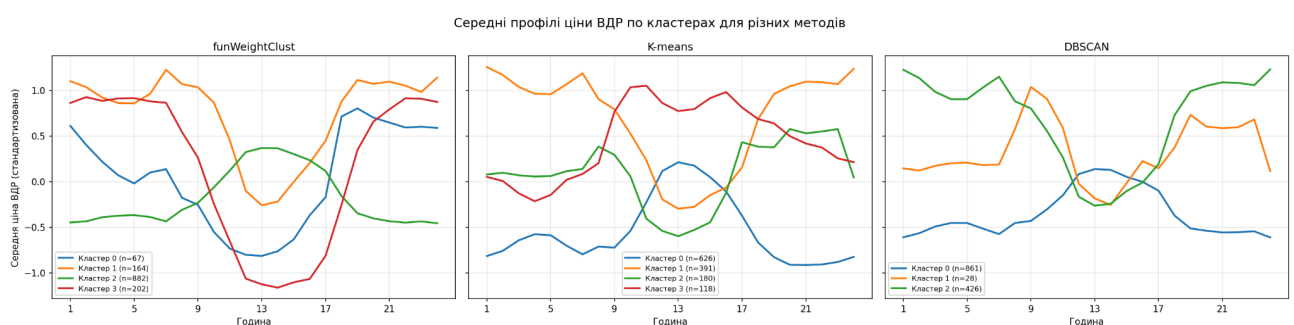


Рисунок 3.7 – Порівняння середніх профілів ціни ВДР по кластерах для різних методів

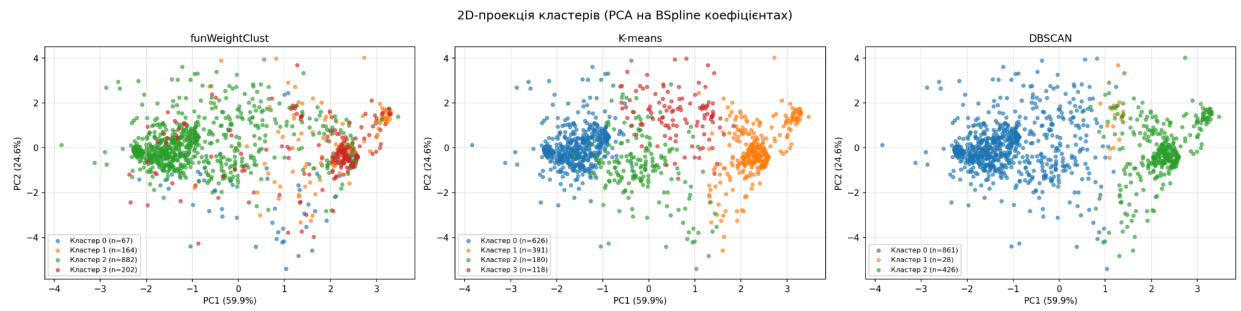


Рисунок 3.8 – 2D-проекція кластерів у просторі BSpline-коефіцієнтів

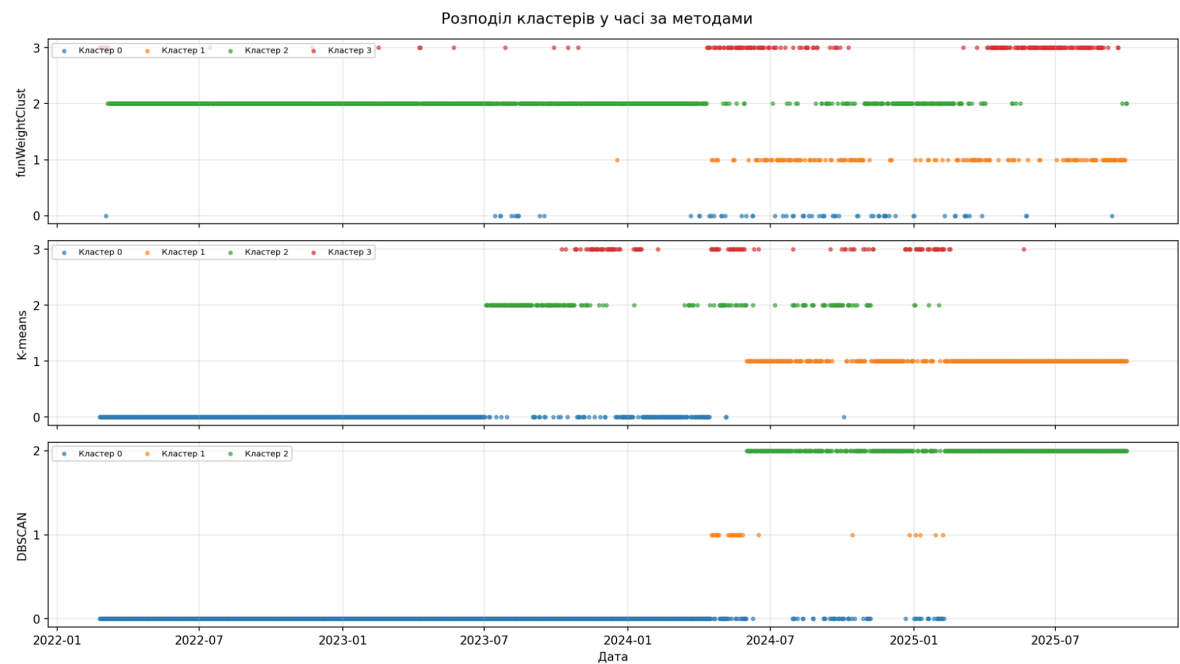


Рисунок 3.9 – Розподіл кластерів у часі для різних методів

3.3.3 Вибір оптимальної кількості кластерів

Для funWeightClust послідовно перевірено $K = 2, 3, 4, 5$. Результати представлено в таблиці 3.3.

Таблиця 3.3 – Вибір кількості кластерів funWeightClust

K	converged	bic	icl	cluster_sizes
2	True	-30752.974323	-30720.964762	[1010, 305]
3	True	-29812.982994	-29759.149621	[89, 273, 953]
4	True	-28635.359300	-28563.589044	[67, 164, 882, 202]
5	False	-27959.759933	-27864.801756	[109, 152, 843, 156, 55]

Значення BIC послідовно зростає при збільшенні K , але вже при алгоритм не збігається. Як оптимальна кількість кластерів обрано $K=4$, оскільки алгоритм збігається (`converged=True`), різниця в BIC порівняно з досить значна (1177 одиниць), відсутні аномально малі кластери.

3.3.4 Аналіз і інтерпретація отриманих режимів ринку

Розподіл діб за чотирма кластерами нерівномірний – від 67 до 882 діб. Проаналізуємо середні профілі цін в кожному кластері (рис. 3.10).

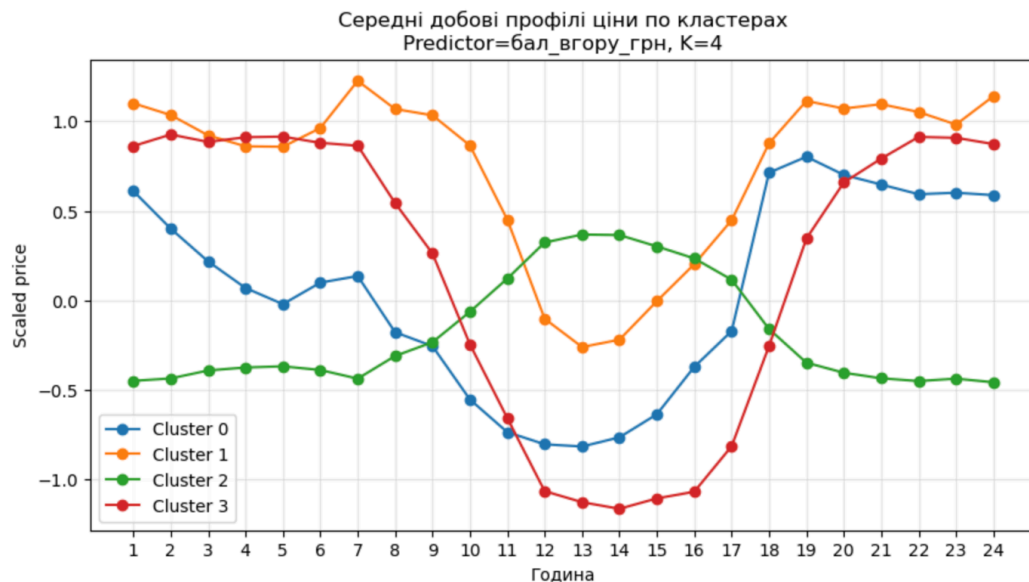


Рисунок 3.10 – Середні добові профілі ціни по кластерах

Кластер 0 (67 діб) є найменшим і відображає рідкісні, але виразні дні зі стресовою конфігурацією цін. Його середній профіль характеризується підвищеним рівнем цін у нічні та вечірні години, помірним ранковим спадом і відносно затяжним денним провалом. На часовій шкалі цей кластер з'являється переважно у 2024–2025 роках і не формує суцільного блоку, тобто відповідає не окремому календарному відрізку, а епізодичним дням із нетиповою внутрішньодобовою динамікою. Такі дні можна інтерпретувати як перехідні або атипові режими, коли ринок одночасно поєднує ознаки дефіциту у пікові години та часткового послаблення напруги вдень.

Кластер 1 (164 доби) описує режим із загалом високим ціновим рівнем протягом більшої частини доби та відносно вузьким, але помітним провалом у середині дня. У цьому режимі нічні, ранкові та вечірні значення залишаються вищими за середні, а опівдні відбувається тимчасове здешевлення. Така форма відповідає дням підвищеної напруги на ринку, коли дефіцит не зникає навіть у непікові години, але денне вікно все ж дає короткий період часткового вирівнювання. За графіком розподілу в часі цей кластер особливо часто трапляється в пізнішій частині вибірки, тобто у фазі тривало високих цін 2024–2025 років.

Кластер 2 (882 доби) є базовим і найбільш масовим режимом ринку. Його профіль має порівняно невелику амплітуду коливань: уночі й у вечірні години ціни нижчі за середні, а в середині дня спостерігається підвищення. Саме цей кластер домінує у 2022 році та значній частині 2023 – початку 2024 років, тобто охоплює періоди відносно стандартного функціонування ринку.

Кластер 3 (202 доби) має найбільш контрастну форму: високі значення в нічні та ранкові години, найглибший серед усіх кластерів провал у середині дня і повторне різке підвищення у вечірній період. Такий профіль відповідає режиму максимальної внутрішньодобової поляризації цін. Цей кластер припадає переважно на 2024–2025 роки, що узгоджується з періодом найбільшої нестабільності енергосистеми.

Отже, коротко ці кластери можна назвати так:

1. Кластер 0 – дефіцитний.
2. Кластер 1 – високий.
3. Кластер 2 – стабільний.
4. Кластер 3 – піковий.

3.4 Підготовка датасету для прогнозування погодинних цін

3.4.1 Формування цільової змінної та лагових ознак

На цьому етапі master-таблиця доповнюється часовими, лаговими та режимними ознаками.

Цільові змінні формуються через зсув:

1) $\text{target_1h} = \log_ціна_MB.t.shift(-1)$ – логарифм ціни наступної години (для one-step-ahead);

2) $\text{target_24h} = \log_ціна_MB.t.shift(-24)$ – логарифм ціни через 24 години (для day-ahead).

Лагові ознаки ціни обчислюються над логарифмованою колонкою:

1) price_lag_1 , price_lag_2 – ціна попередньої та позапопередньої години;

2) price_lag_24 , price_lag_48 , price_lag_168 – ціна тієї самої години 1, 2 і 7 діб тому.

Ковзні характеристики (price_roll_mean_3 , $\text{price_roll_mean_24}$, $\text{price_roll_mean_168}$, price_roll_std_24) обчислюються над попередньо зсунутим рядом $\log_ціна_MB.t.shift(1).rolling(...)$, що гарантує відсутність витоку поточного значення цільової змінної в простір ознак.

Календарні ознаки утворюються з часової мітки: день тижня, місяць, квартал, бінарний індикатор вихідного, день року, кругове кодування години через $\text{hour_sin} = \sin(2\pi \cdot \text{година}/24)$ і $\text{hour_cos} = \cos(2\pi \cdot \text{година}/24)$.

Лагові ознаки предикторів формуються зсувом на 24 години, де для кожного з 6 предикторів додається колонка $\text{<predictor>_lag_24}$, що відповідає значенню тієї самої години попередньої доби.

3.4.2 Додавання міток кластерів до погодинного датасету

Мітки кластерів `funWeightClust` зберігаються на рівні діб і приєднуються до погодинного датасету за датою. Для прогнозування використовується не кластер поточної доби, а ознака `cluster_lag_1d`; повне обґрунтування наведено в п. 2.4.3.

Після цього аналогічно додаються мітки K-means і DBSCAN для порівняльного аналізу.

3.4.3 Розподіл даних на тренувальну, валідаційну та тестову вибірки

Для часових рядів не можна використовувати випадкову розбивку, адже це призводить до витоку інформації з майбутнього в навчальну вибірку. Тому реалізовано хронологічний поділ даних у пропорції 70/15/15.

Спочатку датасет фільтрується від рядків з пустими значеннями, які виникають у перших 168 годинах через лагові ознаки тижневого горизонту. Потім дані розподіляються наступним чином:

- 1) тренувальна вибірка (70%) – використовується для навчання моделі;
- 2) валідаційна вибірка (15%) – використовується для відстеження якості під час навчання та раннього зупинення (early stopping);
- 3) тестова вибірка (15%) – оцінюється фінальна якість моделі.

Для one-step-ahead датасет містить 31 392 рядки, які розподілено як 21 974 / 4 709 / 4 709. Для day-ahead датасет коротший через додаткові пропуски від 24-годинного зсуву цільової змінної, тому там приблизно 31 369 рядків.

3.5 Реалізація моделей прогнозування на основі XGBoost

Для кожного з двох горизонтів прогнозування (1 година і 24 години) реалізовано три варіанти моделей. Усі вони використовують спільну конфігурацію XGBoost: `n_estimators=400`, `max_depth=6`, `learning_rate=0.05`, `subsample=0.8`, `colsample_bytree=0.8`, `objective='reg:squarederror'`, `random_state=42`.

3.5.1 Базова модель без урахування кластеризації

Базова модель приймає на вхід всі ознаки, за винятком колонки `cluster_lag_1d`. Це еталонний варіант, який дозволяє оцінити якість прогнозування без використання інформації про режим ринку.

Передача даних у XGBoost виконується через стандартний інтерфейс `XGBRegressor.fit(X_train, y_train, eval_set=[(X_valid, y_valid)])`. Прогноз отримується методом `predict(X_test)` і повертається в логарифмічній шкалі. Для подальшого розрахунку метрик у гривнях застосовується зворотне перетворення `np.expml(pred)`.

3.5.2 Модель з використанням мітки кластера як ознаки

У другому варіанті додається ще ознака `cluster_lag_1d`. Інші ознаки залишаються тими самими. Гіпотеза цього варіанта: інформація про режим ринку попередньої доби може допомогти моделі краще розрізняти типові патерни цін у різних ринкових умовах.

3.5.3 Окремі моделі для кожного кластера

У третьому варіанті для кожного значення `cluster_lag_1d` навчається окрема модель XGBoost. Гіпотеза цього варіанта: оскільки кожен кластер відповідає принципово іншому режиму ринку, модель для кожного режиму може давати кращі результати, ніж єдина універсальна модель.

Реалізація проходить ітерацією по унікальних значеннях `cluster_lag_1d` у тренувальній вибірці. Для кожного значення:

1. Тренувальна, валідаційна і тестова вибірки фільтруються лише за відповідними рядками.
2. Якщо у тренувальній вибірці менше ніж 50 рядків, у валідаційній менше ніж 10 або у тестовій менше ніж 10 – модель пропускається через недостатній обсяг даних.
3. Інакше навчається окрема модель XGBoost з тими самими гіперпараметрами.

Прогноз для тестової вибірки формується поетапно: для кожної доби тесту визначається значення `cluster_lag_1d`, після чого прогноз будується саме тією моделлю, яка була навчена на цьому кластері. Прогнози від різних моделей об'єднуються в єдиний ряд, на якому розраховуються загальні метрики якості.

3.6 Аналіз результатів прогнозування

3.6.1 Порівняння моделей за метриками MAE, RMSE, MAPE

Результати прогнозування для трьох варіантів моделі при горизонті `one-step-ahead` наведено в таблиці 3.4.

Таблиця 3.4 – Метрики one-step-ahead прогнозування

Модель	MAE	RMSE	MAPE, %	R ²
One-step Базова	1127.37	1498.77	134.43	0.7071
One-step З кластером	1130.47	1514.54	125.58	0.7009
One-step Окремі по кластерах	1138.23	1523.50	162.36	0.6974

Для горизонту one-step-ahead базова модель показує найкращу точність: MAE = 1127 грн/МВт·год, R² = 0.707. Додавання кластерної ознаки або поділ на окремі моделі не дають покращення, навпаки, спостерігається незначне погіршення MAE та R² (різниця менше 1%). Це зумовлено тим, що для прогнозу на 1 годину вперед визначальну роль грають лагові ознаки price_lag_1 і price_lag_24, які вже несуть основну інформацію про поточний стан ринку. Інформація про кластер вчорашньої доби в цьому випадку є надлишковою.

Результати для горизонту day-ahead вже інші (таблиця 3.5).

Таблиця 3.5 – Метрики day-ahead прогнозування

Модель	MAE	RMSE	MAPE, %	R ²
Day-ahead Базова	1427.30	1826.09	249.65	0.5649
Day-ahead З кластером	1337.40	1722.99	237.00	0.6127
Day-ahead Окремі по кластерах	1694.17	2161.68	466.73	0.3903

Для day-ahead модель з кластером як ознакою забезпечує помітне покращення відносно базової моделі: MAE зменшується з 1427 до 1337 грн/МВт·год (–6.3%), R² зростає з 0.565 до 0.613 (+8.5%). Це підтверджує гіпотезу, що при більшому горизонті прогнозування інформація про режим ринку стає важливою – вона компенсує відсутність короткострокових лагів, які при day-ahead недоступні.

Окремі моделі по кластерах для day-ahead показують гірший результат, що пояснюється обмеженим обсягом даних у малих кластерах (особливо в кластері 0 з 67 діб). Окремі моделі недостатньо узагальнюються.

Високі значення MAPE (понад 100%) пояснюються наявністю годин з низькими цінами (від десятків грн/МВт·год у нічний час), при яких ділення на y_i у формулі MAPE дає аномально великі відсоткові помилки. Метрика MAPE у такому випадку не є надійним показником, тому основну увагу слід приділяти MAE та R^2 .

Окремий експеримент, у якому замість міток funWeightClust використовуються мітки K-means і DBSCAN, дозволив оцінити вплив самого алгоритму кластеризації на якість прогнозу (таблиця 3.6).

Таблиця 3.6 – Порівняння методів кластеризації через прогнозування (day-ahead, модель з кластером як ознакою)

Модель	MAE	RMSE	MAPE, %	R^2
Day-ahead funWeightClust	1337.40	1722.99	237.00	0.6127
Day-ahead K-means	1370.32	1755.56	254.98	0.5979
Day-ahead DBSCAN	1337.63	1722.88	238.06	0.6127
Day-ahead Базова	1427.30	1826.09	249.65	0.5649

funWeightClust дає найкращий або еквівалентний DBSCAN результат, перевершуючи K-means на 33 грн/МВт·год за MAE. Хоча різниця з DBSCAN мінімальна, перевага funWeightClust полягає у відсутності необхідності окремої обробки шумових точок, що методично робить підхід чистішим.

3.6.2 Порівняння прогнозних і фактичних добових профілів

Графічний аналіз прогнозних і фактичних кривих на тестовій вибірці підтверджує висновки з числових метрик. Для one-step-ahead прогнозу (рис. 3.11) всі три варіанти моделі добре відтворюють добову форму кривої з невеликим відставанням на пікових значеннях. Базова модель і модель з кластером дають візуально майже ідентичні результати, окремі моделі по кластерах помітно занижують пікові ціни.

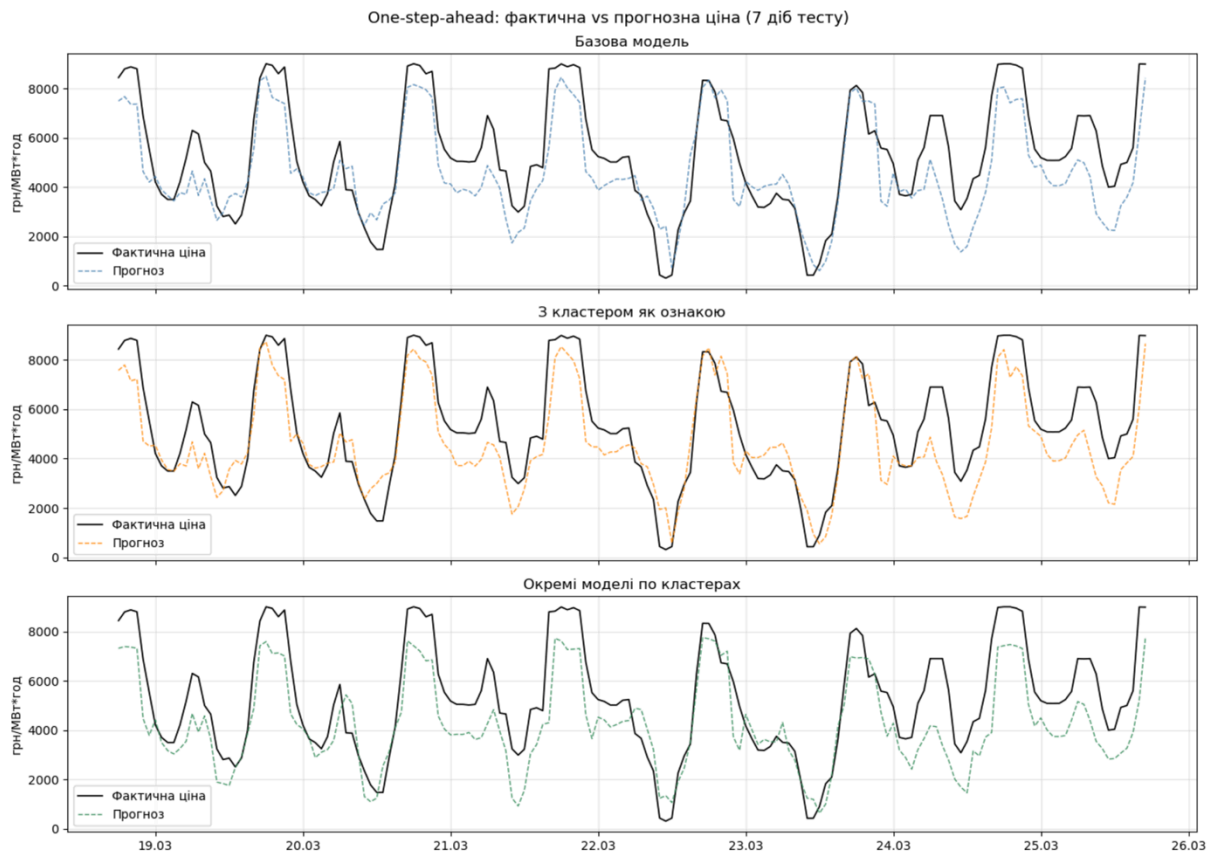


Рисунок 3.11 – One-step-ahead: фактична vs прогнозна ціна

Для day-ahead (рис. 3.12) базова модель систематично згладжує добові коливання: вона добре передбачає середній рівень ціни, але слабо відтворює різкі піки і провали. Модель з кластером як ознакою помітно краще схоплює загальний рівень добового профілю в межах кожного режиму, але також не повністю відтворює екстремальні значення. Найбільші похибки спостерігаються в днях з аномально високими цінами, тому що це події, які моделям важко передбачити заздалегідь без додаткових зовнішніх ознак.

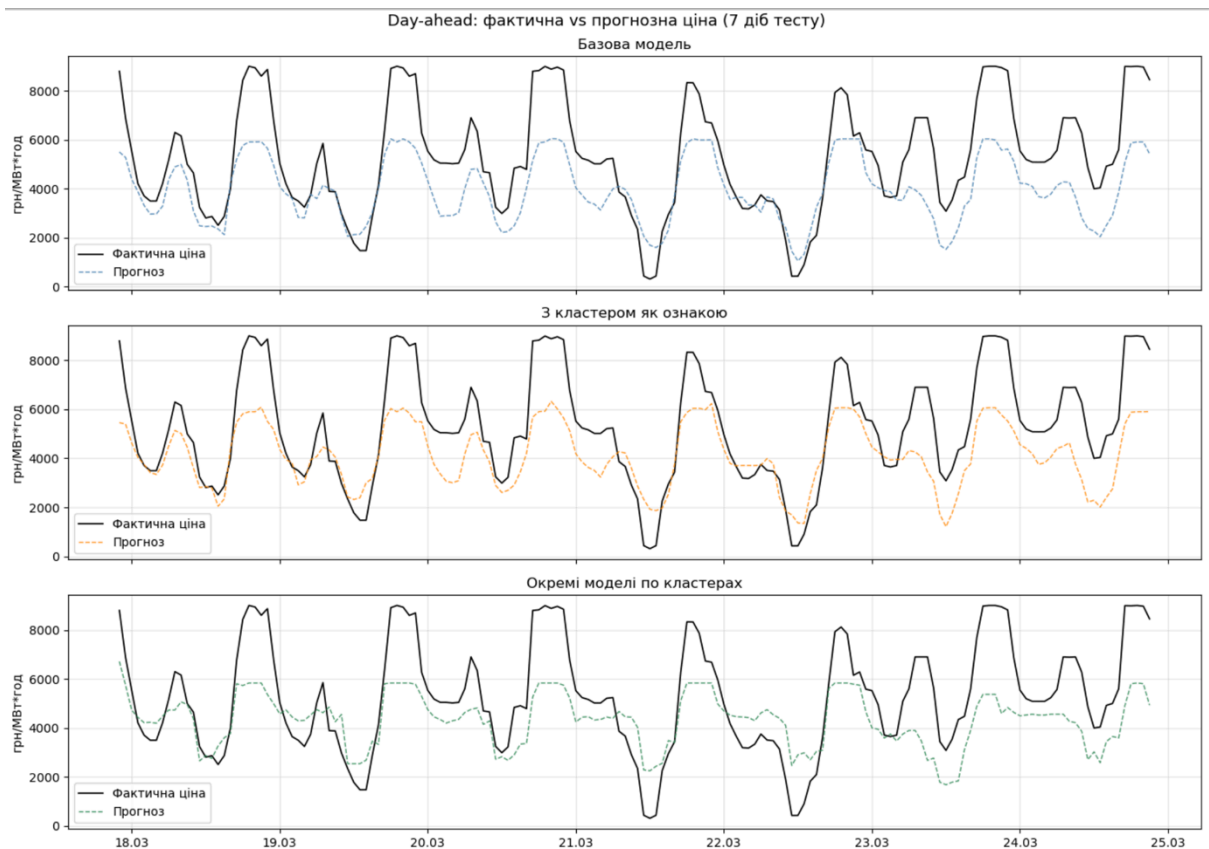


Рисунок 3.12 Day-ahead: фактична vs прогнозна ціна

3.6.3 Аналіз впливу кластеризації на точність прогнозу

Узагальнені висновки щодо впливу кластеризації на якість прогнозування:

1. Для one-step-ahead кластеризація не покращує якість. На короткому горизонті (1 година) лагові ознаки самі по собі несуть достатню інформацію про поточний стан ринку. Інформація про кластер вчорашньої доби виявляється надлишковою.

2. Для day-ahead кластеризація покращує якість на 6–8%. На довшому горизонті, коли короткострокові лаги недоступні, інформація про режим ринку компенсує цю прогалину. Покращення MAE з 1427 до 1337 грн/МВт·год є практично значущим.

3. Додавання колонки з міткою кластера ефективніше за окремі моделі. У day-ahead прогнозуванні XGBoost з `cluster_lag_1d` як однією з ознак працює краще, ніж окремі моделі XGBoost для кожного кластера. Це зумовлено тим, що XGBoost здатен сам вивчити взаємодії між кластером та іншими ознаками, не втрачаючи переваг великого обсягу тренувальних даних.

4. Алгоритм `funWeightClust` демонструє перевагу над класичними методами кластеризації при використанні в комбінації з XGBoost для day-ahead прогнозування, хоча різниця з DBSCAN є невеликою. Більш суттєвою є перевага над K-means.

Таким чином, основна гіпотеза дипломної роботи – про доцільність кластеризації режимів ринку для покращення прогнозування – підтверджується для day-ahead горизонту, який є основним для практичного застосування на ВДР.

3.6.4 Експеримент з прогнозуванням кластера завтрашньої доби

Як варіант можливого покращення day-ahead прогнозування було проведено додатковий експеримент, у якому спочатку прогнозується кластер завтрашньої доби, а потім цей прогнозований режим використовується в ціновій моделі XGBoost.

У межах експерименту було порівняно три варіанти day-ahead прогнозування: базову модель без кластерів; модель з ознакою `cluster_lag_1d`; модель, у якій використовується прогнозований кластер завтрашньої доби. Найкращий результат показав третій варіант (рис. 3.13): $MAE = 1249.19$ грн/МВт·год, $RMSE = 1629.86$ грн/МВт·год, $R^2 = 0.6528$. Для порівняння, базова модель дала $MAE = 1441.78$ грн/МВт·год і $R^2 = 0.5551$, а модель з `cluster_lag_1d` – $MAE = 1363.63$ грн/МВт·год і $R^2 = 0.6012$.

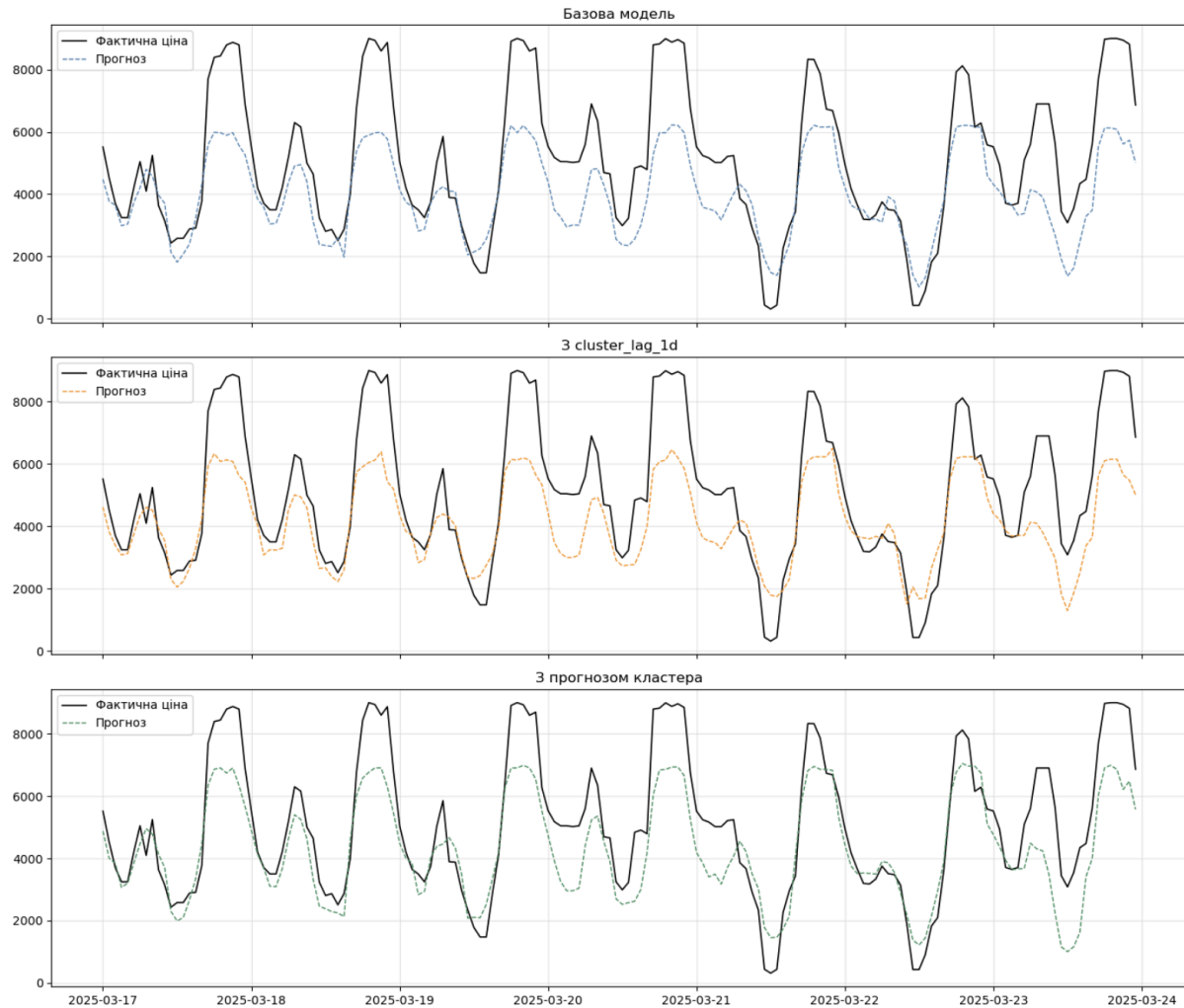


Рисунок 3.13 – Порівняння прогнозування базової моделі, моделі з кластерною ознакою та моделі з прогнозованим кластером

Отримані результати свідчать, що попереднє прогнозування режиму ринку може покращувати прогнозування погодинних цін. Водночас цей підхід у межах дипломного проєкту та реалізованого інтерфейсу не використовується через його експериментальний характер, появу додаткового етапу моделювання, що ускладнює архітектуру системи, а також ризик накопичення помилок: помилка на етапі прогнозу кластера може передаватися далі в модель прогнозування ціни. Тому в інтерфейсі залишено базові та більш стабільні варіанти моделей, а експеримент з прогнозуванням кластера розглядається як окремий напрям потенційного покращення.

3.7 Опис програмної реалізації системи

Програмна реалізація системи спирається на дві комплементарні моделі: архітектурну (рис. 3.14), яка показує компоненти та зв'язки між ними, та функціональну (рис. 3.15), яка фіксує послідовність функцій і потоки даних між ними.



Рисунок 3.14 – Архітектура системи

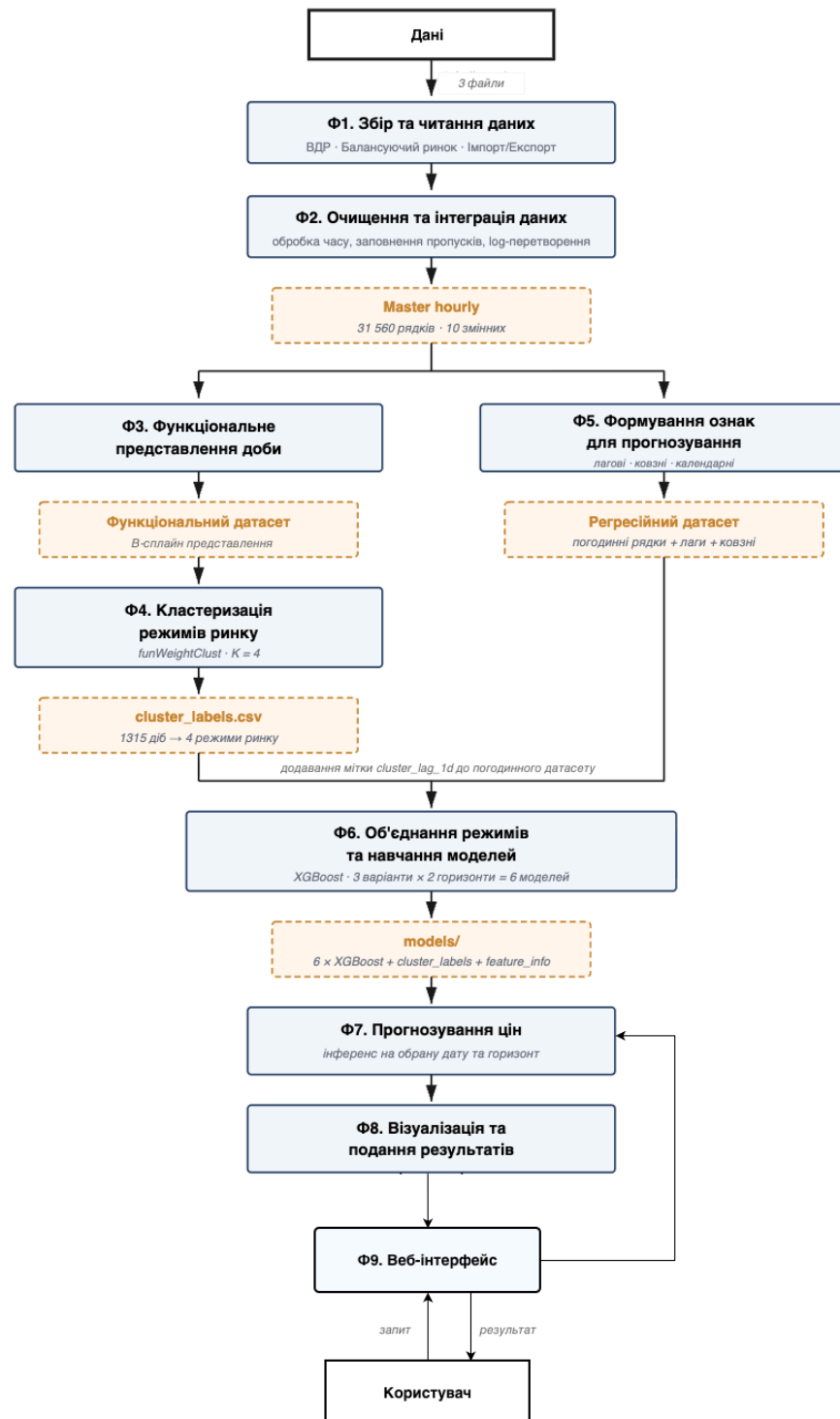


Рисунок 3.15 – Функціональна схема системи

3.7.1 Структура програмного проєкту

Програмний проєкт має модульну структуру, яка відокремлює різні етапи обробки даних, моделювання та інтерфейсу:

diploma_project/

├── data/

| ├── raw/ – оригінальні файли xlsx

| ├── interim/ – проміжні файли

| └── processed/ – фінальні датасети для моделювання

├── notebooks/

| ├── 01_data_review.ipynb – аудит сирих даних

| ├── 02_data_cleaning.ipynb – очищення і агрегація

| └── 03_feature_building.ipynb – побудова функціональних і регресійних

ознак

| ├── 04_funweightclust_experiments.ipynb – кластеризація

| └── 05_price_forecasting.ipynb – навчання та оцінка моделей

├── external/

| └── funWeightClust-Python-main/ – пакет funWeightClust

├── models/

| ├── xgb_1h_base.json – навчені моделі

| ├── xgb_1h_cluster.json

| ├── xgb_24h_base.json

| ├── xgb_24h_cluster.json

| ├── xgb_*_per_cluster.pkl – окремі моделі по кластерах

| ├── feature_info.json – список ознак

| └── cluster_labels.csv – мітки кластерів

├── app/

| └── app.py – Streamlit інтерфейс системи

└── outputs/

 ├── figures/ – графіки

 ├── metrics/ – таблиці метрик

 └── tables/ – таблиці результатів

3.7.2 Використані бібліотеки та інструменти

Реалізація виконана мовою Python 3.11. Ключові бібліотеки: pandas і NumPy для обробки даних; scikit-fda для функціонального подання; scikit-learn для K-means, DBSCAN і допоміжних процедур; xgboost для прогнозування; matplotlib для візуалізації; Streamlit для веб-інтерфейсу.

Розробка велась у Visual Studio Code в окремому Conda-середовищі.

Інтерфейс дозволяє обрати дату, горизонт прогнозування і варіант моделі та побудувати прогноз з візуалізацією фактичних і прогнозних значень.

3.7.3 Інтерфейс користувача та приклади роботи системи

Для демонстрації роботи розробленої системи реалізовано користувацький інтерфейс, який забезпечує зручний запуск прогнозування погодинних цін електроенергії та наочне відображення результатів. Інтерфейс дозволяє вибрати дату прогнозу, горизонт прогнозування — one-step-ahead або day-ahead, а також варіант моделі: базову модель без кластерів, модель з кластером як ознакою або окремі моделі по кластерах. Для one-step-ahead прогнозування додатково задається поточна година, на основі якої формується прогноз наступної години.

У верхній частині інтерфейсу відображається інформація про обрану дату, день тижня та режим ринку, якому відповідає ця доба (рис. 3.16). Режим ринку подається у вигляді номера кластера та короткої змістовної інтерпретації, наприклад, стабільний режим, високий режим або піковий.

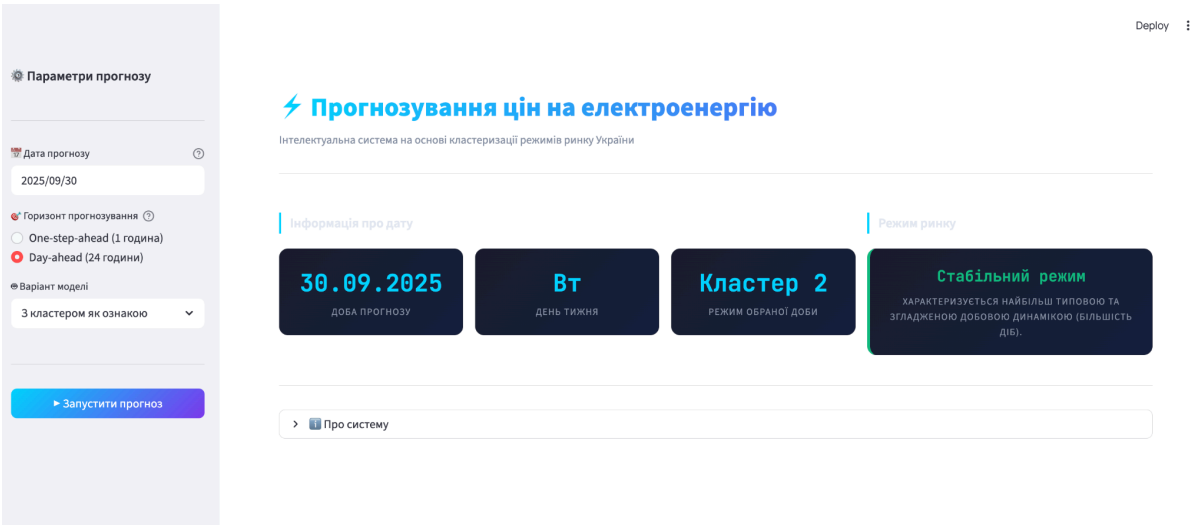


Рисунок 3.16 – Верхня частина інтерфейсу

У режимі one-step-ahead система показує прогнозовану ціну наступної години, фактичне значення ціни та абсолютну похибку (рис. 3.17). Додатково будується графік добового профілю ціни, на якому позначається фактична крива за добу та точка прогнозу для відповідної години.



Рисунок 3.17 – Прогнозування в режимі one-step-ahead

У режимі day-ahead інтерфейс формує погодинний прогноз на всю добу (рис. 3.18). Результати подаються у вигляді таблиці з прогнозними та фактичними значеннями для кожної години, графіка порівняння добового прогнозу з фактичним профілем, а також підсумкових метрик якості, зокрема MAE та RMSE за добу.

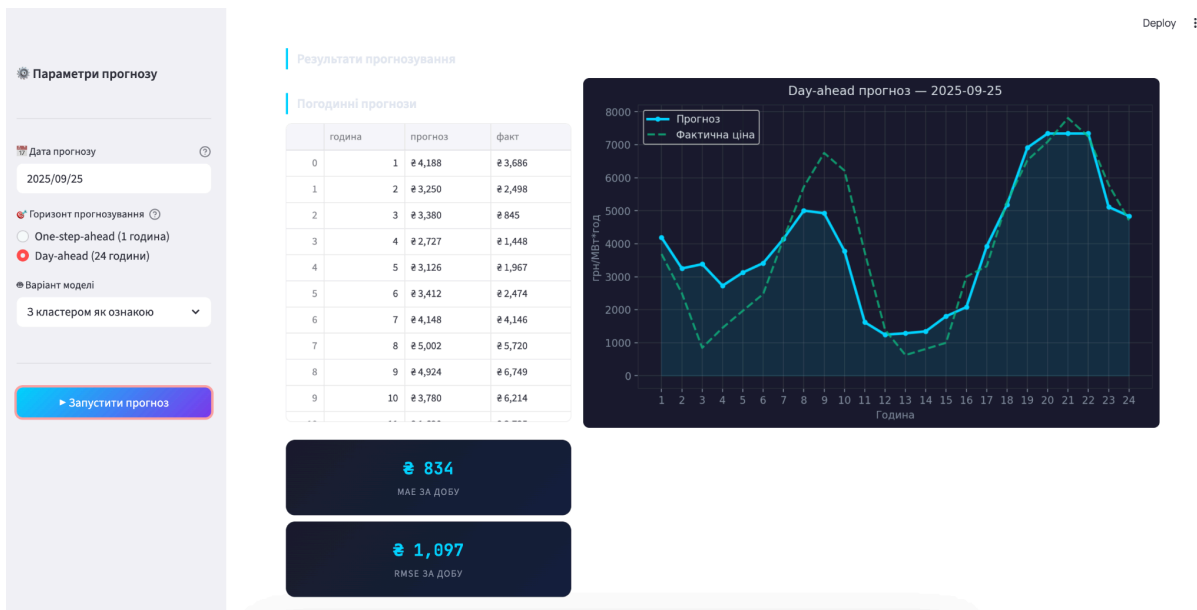


Рисунок 3.18 – Прогнозування в режимі day-ahead

Приклади роботи системи показали, що інтерфейс дає змогу оперативно порівнювати різні варіанти моделей і спостерігати, як змінюється точність прогнозу залежно від вибраного підходу.

3.8 Висновки до розділу 3

У розділі описано програмну реалізацію інтелектуальної системи прогнозування погодинних цін електроенергії.

На етапі підготовки даних виконано об'єднання трьох незалежних джерел (ВДР, балансуючий ринок, імпорту/експорту) в єдиний погодинний датасет. Проведено очищення даних з обробкою переведень годинника, заповнення пропусків нулями і медіаною, інтерполяцію нульових і відсутніх значень. До цільової змінної застосовано логарифмічне перетворення для зменшення впливу екстремальних значень.

На етапі побудови функціонального представлення кожна доба перетворена на функціональне спостереження з 24 значеннями кожної з 7 змінних. Експериментально визначено оптимальні параметри В-сплайн

базису та найкращий одиничний предиктор для кластеризації, яким виявилась ціна балансування у напрямку вгору.

Кластеризація режимів ринку виконана алгоритмом `funWeightClust` з оптимальною кількістю кластерів $K = 4$. Виявлено чотири економічно змістовні режими ринку: дефіцитний, високий, стабільний, піковий.

Прогнозування виконано за допомогою алгоритму `XGBoost` для двох горизонтів (1 година і 24 години) у трьох варіантах конфігурації. Найкращі результати на тестовій вибірці:

1. One-step-ahead: $MAE = 1127$ грн/МВт·год, $R^2 = 0.707$ (базова модель без кластерів).

2. Day-ahead: $MAE = 1337$ грн/МВт·год, $R^2 = 0.613$ (модель з кластером як ознакою).

Для day-ahead прогнозування додавання кластерної ознаки покращує MAE на 6.3% порівняно з базовою моделлю. Для one-step-ahead кластеризація не дає переваги.

Алгоритм `funWeightClust` перевершує `K-means` за якістю прогнозування на основі кластерів і еквівалентний `DBSCAN`, при цьому не потребує додаткової обробки шумових точок.

Проведено експеримент прогнозування цін на основі попереднього прогнозування кластера цільової доби. Як результат якість прогнозування цін покращилась.

Розроблено інтерактивний веб-інтерфейс системи на основі `Streamlit`, який надає прогноз ціни електроенергії для обраної дати з візуалізацією добового профілю і автоматичним визначенням режиму ринку.

РОЗДІЛ 4 ФУНКЦІОНАЛЬНО-ВАРТІСНИЙ АНАЛІЗ ПРОГРАМНОГО ПРОДУКТУ

Функціонально-вартісний аналіз (ФВА) застосовують тоді, коли потрібно зіставити технічну реалізацію продукту з витратами на його створення і знайти рішення, яке дає найкращий баланс цих двох сторін. Для інтелектуальної системи прогнозування цін на електроенергію такий аналіз має практичну підставу: ключове проєктне рішення – вибір методу кластеризації режимів ринку – водночас визначає і якість прогнозу, і обсяг роботи. Отже, дешевше рішення не обов'язково є вигіднішим, і навпаки.

Аналіз будується у два логічні етапи. Спочатку проводиться функціональна частина – визначаються функції продукту, формулюються критерії оцінки та обчислюється технічна якість кожного з можливих рішень. Потім виконується вартісна частина – складається кошторис витрат на розробку за кожним варіантом. Об'єднання результатів обох етапів через коефіцієнт техніко-економічного рівня дає підставу для остаточного вибору.

4.1 Постановка задачі

Завданням цього розділу є оцінити витрати на створення системи прогнозування погодинних цін та визначити, який із технічно можливих варіантів її реалізації є найдоцільнішим.

Система призначена для побудови прогнозу цін на два горизонти (на одну годину вперед та на добу вперед) з урахуванням режиму, у якому функціонує ринок. До неї висуваються такі вимоги:

- 1) автоматизоване опрацювання погодинних ринкових даних (ВДР, балансуючий ринок, потоки імпорту та експорту);
- 2) виділення режимів ринку через кластеризацію добових профілів;

- 3) формування прогнозу цін із урахуванням виявленого режиму;
- 4) наочне подання прогнозу та порівняння з фактичними значеннями;
- 5) зручність користування для побудови прогнозу на задану дату;
- 6) здатність до оновлення моделей у міру надходження нових даних.

Досягнення цих вимог передбачає аналіз методів функціональної кластеризації, реалізацію обраного методу, побудову прогнозних моделей і створення інтерфейсу та перевірку працездатності системи.

4.2 Обґрунтування функцій програмного продукту

Головна функція продукту полягає в автоматизованому прогнозуванні погодинних цін на електроенергію на основі кластеризації режимів ринку. Цю головну функцію забезпечують три основні функції, що відповідають архітектурним блокам системи:

- 1) F1 – кластеризація режимів ринку;
- 2) F2 – прогнозування цін;
- 3) F3 – програмна та технологічна реалізація.

Для кожної основної функції існує кілька способів реалізації. Варіанти реалізації цих функцій наведені у морфологічній карті (рис. 4.1).

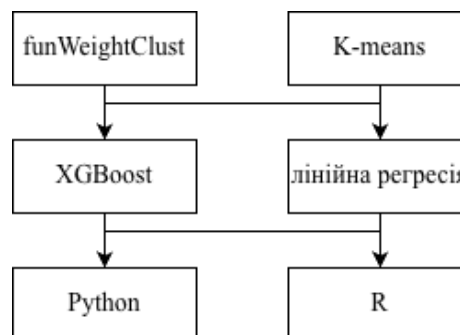


Рисунок 4.1 – Морфологічна карта

Функцію F1 можна реалізувати або через алгоритм funWeightClust, або через класичний K-means.

Функцію F2 можна побудувати на основі градієнтного бустингу XGBoost або на основі лінійної регресії.

Функцію F3 можна реалізувати мовою Python або мовою R.

Порівняння варіантів за їхніми перевагами та недоліками наведено в таблиці 4.1.

Таблиця 4.1 – Аналіз варіантів реалізації основних функцій

Функція	Варіант	Переваги	Недоліки
F1	funWeight Clust	Працює з функціональною природою даних, враховує залежність відповіді від предиктора, виділяє інтерпретовні режими ринку.	Складніший у реалізації, чутливий до обсягу даних.
F1	K-means	Простий, швидкий, стабільний, є в стандартній бібліотеці.	Ігнорує функціональну структуру, розділяє доби лише за рівнем цін.
F2	XGBoost	Точний на табличних даних, стійкий до викидів, моделює нелінійності.	Потребує підбору гіперпараметрів.
F2	Лінійна регресія	Простота, швидке навчання.	Не враховує нелінійність цін.
F3	Python	Розвинена екосистема ML, наявність scikit-fda.	Нижча швидкодія через інтерпретованість, більша витрата оперативної пам'яті.
F3	R	Сильні засоби функціонального аналізу.	Складніша інтеграція з вебінтерфейсом.

Для F2 обрано XGBoost, бо лінійна модель не здатна відтворити нелінійну поведінку цін електроенергії. Для F3 обрано Python переважно через наявність бібліотеки scikit-fda, без якої реалізація функціональної кластеризації була б складнішою.

Натомість для функції F1 обидва варіанти зберігаються для подальшого порівняння, оскільки саме вони були досліджені в роботі та дають різні результати. Таким чином, для економічного аналізу беруться два варіанти:

1. Варіант 1: funWeightClust + XGBoost + Python.
2. Варіант 2: K-means + XGBoost + Python.

4.3 Вибір параметрів для програмного продукту

Щоб оцінити технічну якість кожного варіанта, потрібен набір вимірюваних характеристик. Обрано чотири параметри, що охоплюють як наукову, так і інженерну сторону рішення:

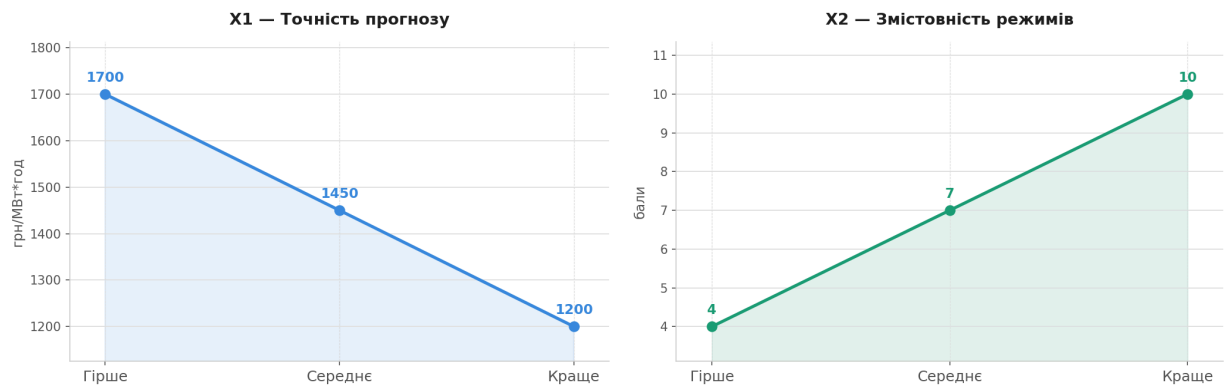
- 1) X1 – точність прогнозування (метрика MAE на горизонті day-ahead);
- 2) X2 – змістовність виявлених режимів (відповідність кластерів реальним станам ринку);
- 3) X3 – швидкодія алгоритму кластеризації;
- 4) X4 – надійність збіжності алгоритму на наявному обсязі даних.

Для кожного параметра встановлено діапазон можливих значень від найгіршого до найкращого, виходячи з вимог до системи та умов її роботи (табл. 4.2).

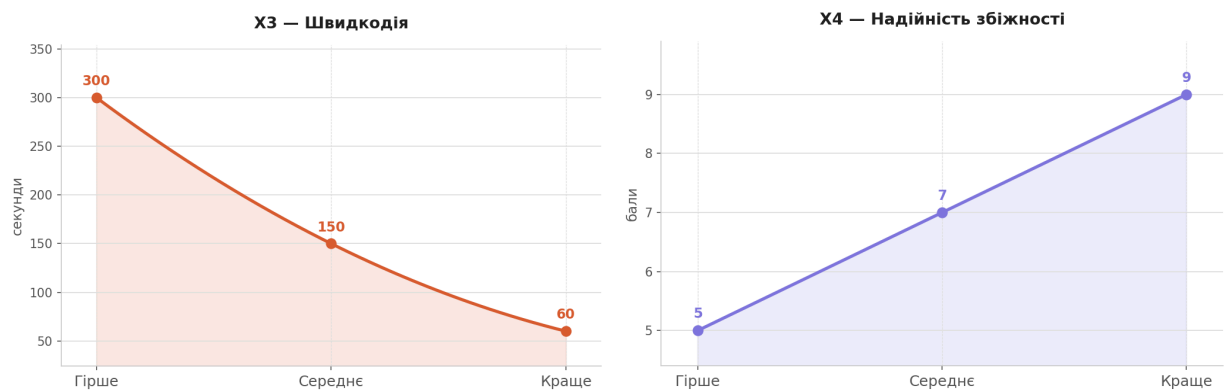
Таблиця 4.2 – Діапазони значень параметрів

Параметр	Позначення	Одиниці	Гірше	Середнє	Краще
Точність прогнозу	X1	грн/МВт*год	1700	1450	1200
Змістовність режимів	X2	бали	4	7	10
Швидкодія	X3	секунди	300	150	60
Надійність збіжності	X4	бали	5	7	9

Нижче наведені графіки для кожного параметру, побудовані на основі таблиці 4.2 (рис. 4.2 – рис. 4.5).



Рисунки 4.2 – X1 (точність прогнозу) та 4.3 – X2 (змістовність режимів)



Рисунки 4.4 – X3 (швидкодія) та 4.5 – X4 (надійність збіжності)

4.4 Експертний аналіз параметрів

Параметри мають неоднакову важливість, тому кожному з них потрібно надати вагу. Для визначення вагомості параметрів залучено комісію з семи незалежних експертів: три спеціалісти з машинного навчання, два аналітики ринку електроенергії та два фахівці з розробки програмних систем. Кожен експерт упорядкував параметри за важливістю, присвоюючи ранг 1 найважливішому та ранг 4 найменш важливому. Результати ранжування наведено в таблиці 4.3.

Таблиця 4.3 – Результати оцінки параметрів від експертів

Позначення параметра	Назва параметра	Одиниці виміру	Ранг параметра за оцінкою експерта							R_i	Δ_i	Δ_i^2
			1	2	3	4	5	6	7			
X1	Точність прогнозу	грн/МВт*год	1	2	1	2	1	2	1	10	-7,5	56,25
X2	Змістовність режимів	бали	2	1	2	1	2	1	2	11	-6,5	42,25
X3	Швидкість	секунди	4	4	4	4	4	4	4	28	10,5	110,25
X4	Надійність збіжності	бали	3	3	3	3	3	3	3	21	3,5	12,25
	Сума		10	10	10	10	10	10	10	70	0	221

Перевірка: загальна сума рангів має дорівнювати $N \cdot n(n + 1)/2$.

$$\sum R_i = 10 + 11 + 28 + 21 = 70 = \frac{7 \cdot 4 \cdot 5}{2} = 70.$$

Середня сума рангів:

$$T = \frac{\sum R_i}{n} = \frac{70}{4} = 17,5.$$

Відхилення суми рангів кожного параметра від середнього значення обраховано за формулою:

$$\Delta_i = R_i - T.$$

Загальна сума квадратів відхилень:

$$S = \sum_{i=1}^N \Delta_i^2 = 221.$$

Достовірність експертних оцінок підтверджується коефіцієнтом узгодженості, який обчислюється за формулою:

$$W = \frac{12S}{N^2(n^3-n)} = \frac{12 \cdot 221}{49(64-4)} = 0,90 > W_k = 0,67$$

де S – сума квадратів відхилень сумарних рангів параметрів від їхнього середнього значення, N – кількість експертів, n – кількість параметрів. Отримане значення перевищує нормативний поріг, тож узгодженість думок експертів вважається прийнятною, а ранжування – достовірним.

Далі проведемо попарне порівняння всіх параметрів на основі результатів ранжування (табл. 4.4).

Таблиця 4.4 – Попарне порівняння параметрів

Параметри	Експерти							Оцінка	Значення
	1	2	3	4	5	6	7		
X1 і X2	>	<	>	<	>	<	>	>	1,5
X1 і X3	>	>	>	>	>	>	>	>	1,5
X1 і X4	>	>	>	>	>	>	>	>	1,5
X2 і X3	>	>	>	>	>	>	>	>	1,5
X2 і X4	>	>	>	>	>	>	>	>	1,5
X3 і X4	<	<	<	<	<	<	<	<	0,5

Ступінь переваги одного параметра над іншим визначається за формулою:

$$a_{ij} = \begin{cases} 1.5, & X_i > X_j, \\ 1, & X_i = X_j, \\ 0.5, & X_i < X_j \end{cases}$$

З отриманих числових оцінок переваги складемо матрицю $A = \|a_{ij}\|$.

Для кожного параметра розрахуємо вагомість K згідно з формулою:

$$K = \frac{b_i}{\sum_{i=1}^n b_i},$$

$$\text{де } b_i = \sum_{j=1}^N a_{ij}.$$

Відносні оцінки розраховуємо декілька разів, поки наступні значення не відрізняються від попередніх (менше ніж 2%). Починаючи з другого кроку, відносні оцінки розраховуються за формулами:

$$K = \frac{b'_i}{\sum_{i=1}^n b'_i}$$

$$b'_i = \sum_{j=1}^N a_{ij} b_j.$$

Результати розрахунків наводяться в таблиці 4.5.

Таблиця 4.5 – Розрахунок вагомості коефіцієнтів

Параметри x_i	Параметри x_j				Перша ітер.		Друга ітер.	
	X1	X2	X3	X4	b_i	K_{bi}	b_i^1	K_{bi}^1
X1	1	1,5	1,5	1,5	5,5	0,3438	21,25	0,3603
X2	0,5	1	1,5	1,5	4,5	0,2812	16,25	0,2706
X3	0,5	0,5	1	0,5	2,5	0,1562	9,25	0,1568
X4	0,5	0,5	1,5	1	3,5	0,2188	12,25	0,2076
Всього					16	1	59	1

З таблиці 4.5 бачимо, що різниця між значеннями коефіцієнтів вагомості не перевищує 2%, тому результати другої ітерації приймаються як остаточні.

4.5 Аналіз якості реалізації варіантів функцій

Технічну якість варіанта визначає інтегральний показник – зважена сума бальних оцінок усіх параметрів:

$$K_K(j) = \sum_{i=1}^n K_{b_i,j} B_{i,j},$$

де n – кількість параметрів, $K_{b_i,j}$ – коефіцієнт вагомості i -го параметра, $B_{i,j}$ – оцінка i -го параметра в балах.

Бальні оцінки визначаються за графіками на основі абсолютних значень параметрів для кожного варіанту.

Значення в балах для кожного параметру наведені в таблиці 4.6.

Таблиця 4.6 – Бальні оцінки параметрів

Параметр	B1	B2	Обґрунтування
X1	9	7	У funWeightClust краще значення, ніж K-means
X2	9	3	funWeightClust знайшов режими, K-means поділив за роками
X3	3	9	funWeightClust працює довше за K-means
X4	6	9	При K=5 funWeightClust не збігся, K-means завжди стабільний

Результати зведено в таблицю 4.7.

Таблиця 4.7 – Розрахунок показників рівня якості варіантів реалізації основних функцій програмного продукту

Основні функції	Варіант реалізації	Параметри	Абсолютне значення	Бальна оцінка	Коефіцієнт вагомості	Коефіцієнт рівня якості
F1	А – funWeightClust	X1	1337	9	0,3603	3,24
		X2	9	9	0,2706	2,43
		X3	600	3	0,1568	0,47
		X4	6	6	0,2076	1,24
F2	Б – K-means	X1	1370	7	0,3603	2,52
		X2	3	3	0,2706	0,81
		X3	2	9	0,1568	1,41
		X4	9	9	0,2076	1,86

Варіант 1 (funWeightClust):

$$K_{K_1} = 7,38$$

Варіант 2 (K-means):

$$K_{K_2} = 6,61$$

Варіант 1 (funWeightClust) перевищує варіант 2 (K-means) за показником якості: 7,38 проти 6,61.

4.6 Економічний аналіз розробки

Щоб визначити вартість розробки системи, виконаємо розрахунок трудомісткості. Розробку поділено на три завдання. Загальна трудомісткість завдання визначається як

$$T_o = T_p \cdot K_{\Pi} \cdot K_{СК} \cdot K_M \cdot K_{СТ} \cdot K_{СТ.М}$$

де T_p – базова трудомісткість, K_{Π} – поправочний коефіцієнт, $K_{СК}$ – коефіцієнт на складність вхідної інформації ($K_{СК}=1$), K_M – коефіцієнт рівня мови програмування ($K_M=1$), $K_{СТ}$ – коефіцієнт використання стандартних модулів і прикладних програм, $K_{СТ.М}$ – коефіцієнт стандартного математичного забезпечення ($K_{СТ.М}=1$).

Перше завдання (кластеризація) має найвищий ступінь новизни та першу групу складності: $T_p = 90$ людино-днів, $K_{\Pi} = 1,7$. Обидва варіанти потребують побудови В-сплайн представлення через scikit-fda – це спільна частина. Проте funWeightClust додатково потребує інтеграції зовнішнього пакета та значної частки власного коду ($K_{СТ} = 0,50$), тоді як для K-means ключовий алгоритм береться зі scikit-learn і обсяг власного коду менший ($K_{СТ} = 0,45$):

$$T_{1,1} = 90 \cdot 1,7 \cdot 0,5 = 76,5 \text{ людино-днів};$$

$$T_{1,2} = 90 \cdot 1,7 \cdot 0,45 = 68,85 \text{ людино-днів.}$$

Друге завдання (прогнозування XGBoost): $T_p = 50$, $K_{\Pi} = 0,9$, $K_{СТ} = 0,6$:

$$T_2 = 50 \cdot 0,9 \cdot 0,6 = 27,0 \text{ людино-днів.}$$

Третє завдання (інтерфейс): $T_p = 34$, $K_{\Pi} = 0,8$, $K_{СТ} = 0,7$:

$$T_3 = 34 \cdot 0,8 \cdot 0,7 = 19,04 \text{ людино-дні.}$$

У людино-годинах:

$$T_1 = (76,5 + 27,0 + 19,04) \cdot 8 = 980,32 \text{ людино-годин};$$

$$T_2 = (68,85 + 27,0 + 19,04) \cdot 8 = 919,12 \text{ людино-годин.}$$

До роботи залучено три фахівці з різними окладами, що перевищують мінімальну заробітну плату 2026 року (8647 грн): керівник проєкту – 55 000 грн, інженер з ML – 45 000 грн, розробник інтерфейсу – 38 000 грн. Усереднена погодинна ставка:

$$C_{\Gamma} = \frac{M}{T_m \cdot t} = \frac{55000+45000+38000}{3 \cdot 21 \cdot 8} = 273,81 \frac{\text{грн}}{\text{год}}.$$

Заробітна плата розраховується за формулою:

$$C = C_{\Gamma} \cdot T_i \cdot K,$$

де C_{Γ} – величина погодинної оплати праці, T_i – трудомісткість відповідного завдання, K – норматив, що враховує додаткову заробітну плату.

З урахуванням додаткової заробітної плати (20 %):

$$C_{\text{зп1}} = 273,81 \cdot 980,32 \cdot 1,2 = 322\,105,70 \text{ грн,}$$

$$C_{\text{зп2}} = 273,81 \cdot 919,12 \cdot 1,2 = 301\,997,10 \text{ грн.}$$

З урахуванням єдиного соціального внеску (22 % від фонду оплати праці):

$$C_{\text{від1}} = 322\,105,70 \cdot 0,22 = 70\,863,25 \text{ грн,}$$

$$C_{\text{від2}} = 301\,997,10 \cdot 0,22 = 66\,439,36 \text{ грн.}$$

Розрахуємо вартість машинного часу. Розробка ведеться на двох робочих станціях (по 35 000 грн, потужність 0,4 кВт). Ефективний річний фонд часу роботи однієї станції:

$$T_{\text{еф}} = (D_K - D_B - D_C - D_P) \cdot t_3 \cdot K_B,$$

$$T_{\text{еф}} = (365 - 104 - 12 - 16) \cdot 8 \cdot 0,4 = 745,6 \text{ год,}$$

де D_K – календарна кількість днів у році, D_B , D_C – кількість вихідних та святкових днів відповідно, D_P – кількість днів планових ремонтів устаткування, t_3 – кількість робочих годин в день, K_B – коефіцієнт використання приладу у часі протягом зміни.

Річні витрати на утримання та експлуатацію обладнання складаються з таких пунктів.

Загальна заробітна плата спеціаліста з обслуговування (ML-інженер, оклад 45 000 грн, коефіцієнт зайнятості 0,2), включає основну та додаткову:

$$C_{\text{зп}} = (12 \cdot 45000 \cdot 0,2) + (12 \cdot 45000 \cdot 0,2 \cdot 0,2) = 129\,600 \text{ грн.}$$

Відрахування на ЄСВ:

$$C_{\text{від}} = 129600 \cdot 0,22 = 28\,512 \text{ грн.}$$

Амортизаційні відрахування (норма амортизації 25%, коефіцієнт транспортно-монтажних витрат $K_{\text{тм}} = 1,4$):

$$C_{\text{а}} = K_{\text{тм}} \cdot K_{\text{а}} \cdot C_{\text{пр}} = 1,4 \cdot 0,25 \cdot 35000 = 12\,250 \text{ грн,}$$

де $K_{\text{а}}$ – річна норма амортизації, $C_{\text{пр}}$ – ціна приладу.

Витрати на ремонт та профілактику (8% від вартості з $K_{\text{тм}}$):

$$C_{\text{р}} = K_{\text{тм}} \cdot C_{\text{пр}} \cdot K_{\text{р}} = 1,4 \cdot 35000 \cdot 0,08 = 3920 \text{ грн,}$$

де $K_{\text{р}}$ – відсоток витрат на поточні ремонти.

Витрати на електроенергію визначено за тарифом для непобутових споживачів другого класу напруги, що на 2026 рік становить 13,64 грн/кВт·год:

$$C_{\text{ел}} = T_{\text{еф}} \cdot N_{\text{с}} \cdot K_{\text{з}} \cdot C_{\text{ен}} = 745,6 \cdot 0,4 \cdot 0,4 \cdot 13,64 = 1\,627,2 \text{ грн,}$$

де $N_{\text{с}}$ – середньо-споживча потужність приладу, $K_{\text{з}}$ – коефіцієнт зайнятості приладу, $C_{\text{ен}}$ – тариф за 1 кВт-годин електроенергії.

Накладні витрати (67% від окладу спеціаліста) $C_{\text{н}} = 30\,150$ грн. Разом річні експлуатаційні витрати:

$$C_{\text{екс}} = C_{\text{зп}} + C_{\text{від}} + C_{\text{а}} + C_{\text{р}} + C_{\text{ел}} + C_{\text{н}}$$

$$C_{\text{екс}} = 129600 + 28512 + 12250 + 3920 + 1627,2 + 30150 = 206\,059,2 \text{ грн.}$$

Собівартість однієї машино-години:

$$C_{\text{м-г}} = \frac{C_{\text{екс}}}{T_{\text{эф}}} = \frac{206\,059,2}{745,6} = 276,37 \text{ грн/год.}$$

Тоді витрати на машинний час становлять:

$$C_{\text{м1}} = C_{\text{м-г}} \cdot T_1 = 276,37 \cdot 980,32 = 270\,931,04 \text{ грн,}$$

$$C_{\text{м2}} = C_{\text{м-г}} \cdot T_1 = 276,37 \cdot 919,12 = 254\,017,19 \text{ грн.}$$

Накладні витрати на розробку (67% від заробітної плати розробників):

$$C_{\text{н1}} = 322\,105,70 \cdot 0,67 = 215\,810,81 \text{ грн,}$$

$$C_{\text{н2}} = 301\,997,10 \cdot 0,67 = 202\,338,06 \text{ грн.}$$

Тоді повна вартість розробки:

$$C_{\text{пп}} = C_{\text{зп}} + C_{\text{від}} + C_{\text{м}} + C_{\text{н}}$$

$$C_{\text{пп1}} = 322\,105,70 + 70\,863,25 + 270\,931,04 + 215\,810,81 = \\ = 879\,719,8 \text{ грн,}$$

$$C_{\text{пп2}} = 301\,997,10 + 66\,439,36 + 254\,017,19 + 202\,338,06 = 824\,791,71 \text{ грн.}$$

Варіант 1 (funWeightClust) дорожчий за варіант 2 (K-means) на 54 928,09 грн. Ця різниця цілком зумовлена вищою трудомісткістю реалізації алгоритму funWeightClust, яка відображається в усіх витратах.

4.7 Вибір оптимального варіанту реалізації програмного продукту

Щоб врахувати одночасно і якість, і вартість, обчислюється коефіцієнт техніко-економічного рівня:

$$K_{\text{тер}j} = \frac{K_{Kj}}{C_{\Phi j}},$$

де K_{Kj} – коефіцієнт рівня якості j-го варіанту, $C_{\Phi j}$ – вартість розробки j-го варіанту у грн.

$$K_{\text{тер1}} = \frac{7,38}{879\,719,8} = 8,39 \cdot 10^{-6}, K_{\text{тер2}} = \frac{6,61}{824\,791,71} = 8,01 \cdot 10^{-6}$$

Варіант 1 (funWeightClust) має вищий коефіцієнт техніко-економічного рівня і є оптимальним. Хоча він дорожчий, якість перевищує варіант 2 на 7,8 %, тому перевага у якості перекриває різницю у вартості. Вирішальним є параметр змістовності режимів: funWeightClust виявляє функціонально осмислені режими ринку, а не лише хронологічні групи, що визначає практичну цінність системи.

Для реалізації обрано варіант 1: funWeightClust + XGBoost + Python (scikit-fda, scikit-learn, XGBoost, Streamlit).

4.8 Висновки до розділу 4

Проведено функціонально-вартісний аналіз інтелектуальної системи прогнозування погодинних цін на електроенергію.

Сформовано два варіанти реалізації, що відрізняються методом кластеризації (funWeightClust та K-means).

Обрано чотири параметри технічної оцінки з їх перетворенням в бальну шкалу та проведено експертний аналіз вагомості з перевіркою узгодженості. За результатами розрахунку funWeightClust набрав 7,38 бала, K-means – 6,61.

За коефіцієнтом техніко-економічного рівня обрано варіант 1: вища вартість компенсується кращою якістю, зосередженою в точності прогнозування та змістовності кластеризації режимів ринку.

ВИСНОВКИ

У дипломній роботі була досліджена задача побудови інтелектуальної системи прогнозування погодинних цін електроенергії в Україні на основі кластеризації режимів функціонування ринку. Актуальність роботи зумовлена високою мінливістю ринку електроенергії, значним впливом балансування, імпорту, експорту та інших короткострокових факторів, що ускладнюють прогнозування цін і потребують використання більш гнучких підходів до аналізу даних.

У роботі сформовано єдиний погодинний датасет для торгової зони ОЕС України, виконано очищення даних, обробку пропусків, побудову функціонального представлення добових профілів та формування ознак для задач кластеризації і прогнозування. Для подання добових кривих використано базисне наближення, що дало змогу зменшити розмірність даних і зберегти основні характеристики форми профілів.

Для виділення типових режимів ринку реалізовано алгоритм `funWeightClust` та виконано його порівняння з алгоритмами K-means і DBSCAN. За результатами експериментів встановлено, що функціональна кластеризація забезпечує більш змістовний і краще інтерпретований поділ добових профілів на режими ринку, ніж класичні методи. Виділені кластери було інтерпретовано як характерні режими функціонування ринку, зокрема стабільний, високий, дефіцитний і піковий.

Для прогнозування погодинних цін побудовано моделі на основі XGBoost у кількох постановках: базову модель без кластеризації, модель із використанням кластерної мітки як додаткової ознаки та окремі моделі в межах кластерів. Проведене порівняння показало, що врахування інформації про режими ринку може покращувати якість прогнозування, однак ефект залежить від способу включення кластерної інформації до моделі та від горизонту прогнозу.

Окремо було досліджено експериментальний варіант покращення day-ahead прогнозування через попередній прогноз кластера завтрашньої доби та подальше використання цього прогнозу в моделі XGBoost. Отримані результати показали, що такий підхід має потенціал для покращення точності.

Практичне значення роботи полягає в тому, що розроблений підхід може бути використаний для короткострокового аналізу ринку електроенергії, виявлення типових режимів його функціонування та підтримки прийняття рішень у задачах моніторингу і прогнозування. Реалізована система об'єднує підготовку даних, кластеризацію, прогнозування та користувацький інтерфейс для запуску прогнозів.

Дослідження має певні обмеження. Модель не враховує частину зовнішніх факторів, які також можуть впливати на ціни, зокрема погодні умови, навантаження, генерацію з ВДЕ чи аварійні події. Крім того, результат залежить від обраної конфігурації предикторів, параметрів кластеризації та способу використання кластерних міток у прогнозуванні.

Подальший розвиток проєкту доцільно спрямувати на розширення набору ознак, побудову окремої моделі для прогнозування режиму наступної доби, дослідження альтернативних алгоритмів прогнозування та вдосконалення інтерфейсу системи. Перспективним є також перехід до ймовірнісного прогнозування, яке дозволило б оцінювати не лише точкове значення ціни, а й невизначеність прогнозу.

Отже, у роботі розроблено та досліджено підхід до прогнозування погодинних цін електроенергії на основі кластеризації режимів ринку України. Отримані результати підтверджують доцільність використання функціонального подання добових профілів і демонструють, що кластеризація режимів ринку є корисним інструментом як для інтерпретації ринкових станів, так і для подальшого вдосконалення моделей прогнозування.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Про ринок електричної енергії : Закон України від 13.04.2017 № 2019-VIII. URL: <https://zakon.rada.gov.ua/laws/show/2019-19#Text>.
2. Ціновий центр : результати торгів на ринку «на добу наперед» та внутрішньодобовому ринку / Оператор ринку. URL: <https://www.oree.com.ua/index.php/pricectr>.
3. Словник термінів / Оператор ринку. URL: <https://www.oree.com.ua/index.php/vocabulary>.
4. Про затвердження Правил ринку : Постанова НКРЕКП від 14.03.2018 № 307. URL: <https://zakon.rada.gov.ua/laws/show/v0307874-18#Text>.
5. Імпорт та експорт електроенергії погодинно [Електронний ресурс] // Energy Map. Режим доступу: <https://energy-map.info/uk/datasets/56df70b0-6bc1-4c7d-a82f-284cf723438d>.
6. Hyndman R.J., Athanasopoulos G. Forecasting: Principles and Practice. 3rd ed. Melbourne, Australia : OTexts, 2021. URL: <https://otexts.com/fpp3/>.
7. Качан Д. С., Недашківська Н. І. Дослідження модифікації гібридної рекурентної нейронної мережі для прогнозування цін акцій. Information Technologies and Computer Modelling : матеріали міжнародної науково-практичної конференції. Івано-Франківськ, 21-24 травня 2024. С. 118–119. URL: https://www.researchgate.net/profile/Iryna-Melnyk-2/publication/389086186_INFORMACIJNI_TEHNOLOGII_TA_KOMP'UTERNE_MODELUVANNA_materiali_miznarodnoi_naukovo-prakticnoi_konferencii_21-24_travana_2024_roku_Ivano-Frankivsk_INFORMATION_TECHNOLOGIES_AND_COMPUTER_MODELLING_proceedi/links/67b46d738311ce680c678ab9/INFORMACIJNI-TEHNOLOGII-TA-KOMPUTERNE-MODELUVANNA-materiali-miznarodnoi-naukovo-prakticnoi-konferencii-21-24-travana-2024-roku-Ivano-Frankivsk-Information-Technologies-and-Computer-Modelling-proceedi.pdf

[24-roku-Ivano-Frankivsk-INFORMATION-TECHNOLOGIES-AND-COMPUTER-MODELLING-proceedin.pdf#page=118](#).

8. Nedashkovskaya N. I., Androsov D. V. Generative time series model based on encoder-decoder architecture. System Research and Information Technologies. 2022. №1. C. 97–109. URL: <https://journal.iasa.kpi.ua/article/view/259236/255872>.

9. Hastie T., Tibshirani R., Friedman J. The Elements of Statistical Learning. 2nd ed. New York : Springer, 2009. 745 p. URL: <https://www.sas.upenn.edu/~fdiebold/NoHesitations/BookAdvanced.pdf>.

10. Anton C., Smith I. Cluster weighted models for functional data. arXiv preprint. 2025. 36 p. URL: <https://arxiv.org/abs/2503.05159>.

11. Ramsay J.O., Silverman B.W. Functional Data Analysis. 2nd ed. New York : Springer, 2005. 426 p. DOI: 10.1007/b98888. URL: <https://link.springer.com/book/10.1007/b98888>.

12. Gertheiss J., Rügamer D., Liew B., Greven S. Functional Data Analysis: An Introduction and Recent Developments. arXiv preprint. 2023. 28 p. URL: <https://arxiv.org/abs/2312.05523>.

13. Schmutz A., Jacques J., Bouveyron C., Cheze L., Martin P. Clustering multivariate functional data in group-specific functional subspaces // Computational Statistics. 2020. Vol. 35. P. 1101–1131. DOI: 10.1007/s00180-020-00958-4. URL: <https://link.springer.com/article/10.1007/s00180-020-00958-4>.

14. Ester M., Kriegel H.-P., Sander J., Xu X. A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise // Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining (KDD'96), Portland, OR. AAAI Press, 1996. P. 226–231. URL: <https://file.biolab.si/papers/1996-DBSCAN-KDD.pdf>.

15. Chen T., Guestrin C. XGBoost: A Scalable Tree Boosting System // Proceedings of the 22nd ACM SIGKDD International Conference on

Knowledge Discovery and Data Mining, San Francisco, CA. 2016. P. 785–794. DOI: 10.1145/2939672.2939785. URL: <https://arxiv.org/abs/1603.02754>.

16. Результати торгів на ВДР [Електронний ресурс] // Energy Map. Режим доступу: <https://energy-map.info/uk/datasets/f0afe3f3-9d72-4e3e-a863-8812b8912bce>.

17. Результати балансуючого ринку [Електронний ресурс] // Energy Map. Режим доступу: <https://energy-map.info/uk/datasets/f89172b2-b9c6-4f84-b9da-18a26a437669>.

18. De Boor C. A Practical Guide to Splines. Revised ed. New York : Springer, 2001. 348 p. URL: https://www.researchgate.net/publication/200744645_A_Practical_Guide_to_Spline.

19. MacQueen J. Some methods for classification and analysis of multivariate observations // Proceedings of the 5th Berkeley Symposium on Mathematical Statistics and Probability. Vol. 1. University of California Press, 1967. P. 281–297. URL: <https://scispace.com/pdf/some-methods-for-classification-and-analysis-of-multivariate-4pswti19oz.pdf>.

20. Biernacki C., Celeux G., Govaert G. Assessing a Mixture Model for Clustering with the Integrated Completed Likelihood // IEEE Transactions on Pattern Analysis and Machine Intelligence. 2000. Vol. 22, no. 7. P. 719–725. DOI: 10.1109/34.865189. URL: https://www.researchgate.net/publication/3193130_Assessing_a_Mixture_Model_for_Clustering_with_the_Integrated_Completed_Likelihood.

21. Rousseeuw P.J. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis // Journal of Computational and Applied Mathematics. 1987. Vol. 20. P. 53–65. DOI: 10.1016/0377-0427(87)90125-7. URL: https://www.researchgate.net/publication/222451107_Rousseeuw_PJ_Silhouettes_A_Graphical_Aid_to_the_Interpretation_and_Validation_of_Cluster_Analysis_Comput_Appl_Math_20_53-65.

22. Aghabozorgi S., Shirkhorshidi A. S., Wah T. Y. Time-series clustering – A decade review. Information Systems. 2015. Vol. 53. P. 16–38.

URL: https://www.researchgate.net/publication/276075711_Time-series_clustering_-_A_decade_review.

23. Jacques J., Preda C. Model-based clustering for multivariate functional data. Computational Statistics & Data Analysis. 2014. Vol. 71. P. 92–106. URL: <https://www.sciencedirect.com/science/article/abs/pii/S0167947312004380>.

24. Аналіз нестационарних і нелінійних процесів в економіці та фінансах [Електронний ресурс] : навч. посіб. для студентів спеціальності F4 «Системний аналіз та наука про дані» / уклад.: П. І. Бідюк, О. Л. Тимошук, В. Г. Гуськова, Т. І. Просянкін-Жарова, Л. Б. Левенчук ; КПІ ім. Ігоря Сікорського. – Електрон. текст. дані (1 файл: 14,57 Мбайт). – Київ : КПІ ім. Ігоря Сікорського, 2025. – 378 с. – URL: <https://ela.kpi.ua/handle/123456789/77009>.

25. Недашківська Н. І. Інтелектуальний аналіз даних : практикум [Електронний ресурс] : навч. посіб. для студ. спеціальності 124 «Системний аналіз», освітніх програм «Системний аналіз і управління», «Системний аналіз фінансового ринку» / Н. І. Недашківська ; КПІ ім. Ігоря Сікорського. – Електрон. текст. дані (1 файл: 6 Мбайт). – Київ : КПІ ім. Ігоря Сікорського, 2021. – 105 с. – URL: <https://ela.kpi.ua/handle/123456789/53763>.

26. Недашківська Н. І. Методи і моделі інтелектуального аналізу даних : практикум [Електронний ресурс] : навч. посіб. для студ. спеціальності 122 «Комп'ютерні науки», освітньої програми «Системи і методи штучного інтелекту» / Н. І. Недашківська ; КПІ ім. Ігоря Сікорського. – Електрон. текст. дані (1 файл: 1,8 Мбайт). – Київ : КПІ ім. Ігоря Сікорського, 2019. – 70 с. – URL: <https://ela.kpi.ua/handle/123456789/53764>.

27. Недашківська Н.І. Сертифікований дистанційний курс «Інтелектуальний аналіз даних». Платформа дистанційного навчання «Сікорський», 2024 (Сертифікат серія ДК №0256 затверджено

Методичною радою КПІ ім. Ігоря Сікорського, протокол № 4 від 01.02.2024 р.)

28. Недашківська Н.І. Сертифікований дистанційний курс «Інтелектуальні системи підтримки прийняття рішень». Платформа дистанційного навчання «Сікорський», 2026 (Сертифікат серія ДК №0733 затверджено Методичною радою КПІ ім. Ігоря Сікорського, протокол № 8 від 04.06.2026 р.)

ДОДАТОК А. ЛІСТИНГ ПРОГРАМНОГО КОДУ

```
# Уніфікація часових індексів трьох джерел
def parse_hour_from_range(s):
    """'00:00-01:00' -> 1, '01:00-02:00' -> 2, ..., '23:00-00:00' -> 24"""
    return int(str(s).split(':')[0]) + 1

df_ie['дата'] = df_ie['Дата'].dt.normalize()
df_ie['година'] = df_ie['Час'].apply(parse_hour_from_range)

df_bal['дата'] = df_bal['Дата'].dt.normalize()
df_bal['година'] = df_bal['Операційний період'].apply(parse_hour_from_range)

df_vdr['дата'] = df_vdr['Доба постачання'].dt.normalize()
df_vdr['година'] = df_vdr['Година'].astype(int)

# Діагностика регулярності часового ряду
def diagnose_regularity(df, name, date_col='дата'):
    print(f'=== {name} ===')

    # Розподіл кількості годин по добах
    counts = df.groupby(date_col)['година'].count()
    dist = counts.value_counts().sort_index()
    for n_hours, n_days in dist.items():
        marker = 'OK' if n_hours == 24 else 'УВАГА'
        print(f' [{marker}] {n_hours} годин: {n_days} діб')

    # Пропущені доби
    all_dates = pd.date_range(df[date_col].min(), df[date_col].max(), freq='D')
    existing = pd.DatetimeIndex(df[date_col].unique())
    missing = all_dates.difference(existing)
    print(f' Очікується діб: {len(all_dates)}, наявних: {len(existing)}')
    if len(missing):
        print(f' Пропущених діб: {len(missing)} -> {list(missing.date)}')

# Обробка переведення годинника (23/25 годин у добі)
# 25 годин -> 24: видаляємо дублікат години
df.drop_duplicates(subset=['дата', 'година'], keep='first', inplace=True)

def interpolate_missing_hours(df):
    """Для кожної доби з < 24 годинами додає відсутні години
    лінійною інтерполяцією між сусідніми годинами."""
    num_cols = [c for c in df.select_dtypes(include='number').columns if c !=
'година']
    result = []

    for date, group in df.groupby('дата'):
        group = group.sort_values('година').reset_index(drop=True)
        missing_hours = set(range(1, 25)) - set(group['година'])
        if not missing_hours:
            result.append(group)
            continue

        new_rows = []
        for h in missing_hours:
            prev = group[group['година'] == h - 1]
            next_ = group[group['година'] == h + 1]
            if len(prev) > 0 and len(next_) > 0:
                new_row = prev.copy()
                for col in num_cols:
                    new_row[col] = (prev[col].values + next_[col].values) / 2
```

```

        elif len(prev) > 0:
            new_row = prev.copy()
        else:
            new_row = next_.copy()
            new_row['година'] = h
            new_rows.append(new_row)

    group = pd.concat([group] +
new_rows).sort_values('година').reset_index(drop=True)
    result.append(group)

    return pd.concat(result).reset_index(drop=True)

# Заповнення повністю пропущених днів
def fill_missing_days(df, date_min, date_max, name,
                      cols_to_interpolate=None, cols_fill_zero=None):
    cols_fill_zero = cols_fill_zero or []
    all_dates = pd.date_range(date_min, date_max, freq='D')
    all_hours = list(range(1, 25))

    full_grid = pd.DataFrame(
        [(d, h) for d in all_dates for h in all_hours],
        columns=['дата', 'година'])
    merged = full_grid.merge(df, on=['дата', 'година'], how='left')
    merged = merged.sort_values(['дата', 'година']).reset_index(drop=True)

    if not merged.isnull().any(axis=1).any():
        return merged

    if cols_to_interpolate is None:
        cols_to_interpolate = [c for c in
merged.select_dtypes(include='number').columns
                           if c != 'година']

    parts = []
    for h in all_hours:
        # інтерполяція в межах однієї
години
        chunk = merged[merged['година'] == h].copy().sort_values('дата')
        existing = [c for c in cols_to_interpolate if c in chunk.columns]
        chunk[existing] = chunk[existing].interpolate(method='linear',
                                                    limit_direction='both')

        for col in cols_fill_zero:
            if col in chunk.columns:
                chunk[col] = chunk[col].fillna(0)
        parts.append(chunk)

    return pd.concat(parts, ignore_index=True).sort_values(
        ['дата', 'година']).reset_index(drop=True)

# Заповнення пропусків балансуєного ринку
# Обсяги -> 0 (балансування не відбувалось)
for col in ['бал_вгору_МВт', 'бал_вниз_МВт']:
    bal_clean[col] = bal_clean[col].fillna(0)

# Ціни -> медіана по годині
for col in ['бал_вгору_грн', 'бал_вниз_грн']:
    hourly_median = bal_clean.groupby('година')[col].transform('median')
    bal_clean[col] = bal_clean[col].fillna(hourly_median)

# Об'єднання у master-таблицю та логарифмування ціни
master = (vdr_clean
          .merge(ie_agg, on=['дата', 'година'], how='left')
          .merge(bal_clean, on=['дата', 'година'], how='left'))

```

```

).sort_values(['дата', 'година']).reset_index(drop=True)

# Нульові ціни -> NaN -> лінійна інтерполяція
master['ціна_МВт'] = master['ціна_МВт'].replace(0,
np.nan).interpolate(method='linear')

# Логарифмування цільової змінної: loglp(x) = log(x + 1)
master['log_ціна_МВт'] = np.loglp(master['ціна_МВт'])

# Матриця кореляцій предикторів
corr = master[['ціна_МВт', 'імпорт_МВт', 'експорт_МВт',
               'бал_вгору_МВт', 'бал_вгору_грн',
               'бал_вниз_МВт', 'бал_вниз_грн']].corr()

fig, ax = plt.subplots(figsize=(8, 6))
im = ax.imshow(corr, cmap='RdBu_r', vmin=-1, vmax=1)
plt.colorbar(im, ax=ax)
ax.set_xticks(range(len(corr.columns)));
ax.set_yticks(range(len(corr.columns)));
ax.set_xticklabels(corr.columns, rotation=45, ha='right')
ax.set_yticklabels(corr.columns)
for i in range(len(corr)):
    for j in range(len(corr)):
        ax.text(j, i, f'{corr.iloc[i, j]:.2f}', ha='center', va='center')
plt.tight_layout()

# Побудова добових функціональних спостережень
VARIABLES = ['ціна_МВт', 'імпорт_МВт', 'експорт_МВт',
             'бал_вгору_МВт', 'бал_вгору_грн',
             'бал_вниз_МВт', 'бал_вниз_грн']

daily_parts = []
for var in VARIABLES:
    pivot = master.pivot(index='дата', columns='година', values=var)
    pivot.columns = [f'{var}_{h}' for h in pivot.columns]
    daily_parts.append(pivot)

daily = pd.concat(daily_parts, axis=1).reset_index()
# Результат: 1 рядок = 1 доба, 7 змінних × 24 години = 168 колонок ознак

# Стандартизація функціональних спостережень
from sklearn.preprocessing import StandardScaler

scalers = {}
daily_scaled = daily.copy()
for var in VARIABLES:
    cols = [f'{var}_{h}' for h in range(1, 25)]
    scaler = StandardScaler()
    daily_scaled[cols] = scaler.fit_transform(daily[cols])
    scalers[var] = scaler

# Перетворення у функціональні об'єкти на B-сплайн базисі
import numpy as np
from skfda.representation.grid import FDataGrid
from skfda.representation.basis import BSplineBasis

grid = np.arange(1, 25)

def get_hourly_cols(df, prefix):
    cols = [c for c in df.columns if c.startswith(prefix + "_")]
    return sorted(cols, key=lambda x: int(x.split("_")[-1]))

```

```

def build_fd_for_predictors(daily_df, response_prefix, predictor_prefixes,
n_basis=4):
    basis = BSplineBasis(domain_range=(1, 24), n_basis=n_basis)

    y_cols = get_hourly_cols(daily_df, response_prefix)
    Y = daily_df[y_cols].to_numpy()
    fd_y = FDataGrid(data_matrix=Y, grid_points=grid).to_basis(basis)

    fd_x_list = []
    for p in predictor_prefixes:
        X = daily_df[get_hourly_cols(daily_df, p)].to_numpy()
        fd_x_list.append(FDataGrid(data_matrix=X,
grid_points=grid).to_basis(basis))

    fd_x = fd_x_list[0] if len(fd_x_list) == 1 else fd_x_list
    return fd_y, fd_x

# Виклик funWeightClust
import funweightclust as fwc

res = fwc.funweightclust(
    datax=fd_x,          # функціональні предиктори (В-сплайн)
    datay=fd_y,          # функціональний відгук (ціна ВДР)
    K=4,                 # кількість кластерів (обрано за BIC, див. Лістинг
4.4)
    model="ABQKDK",      # модель коваріації для предикторів
    modely="EII",        # модель коваріації для відгуку
    init="kmeans",
    nb_rep=10,           # кількість рестартів
    threshold=0.1,
    min_individuals=2,
    show=False
)

labels = np.array(res.cl)      # мітки кластерів
print("Converged:", res.converged, "| BIC:", res.bic, "| ICL:", res.icl)

# Відбір найінформативнішого предиктора
candidate_predictors = ["імпорт_МВт", "експорт_МВт", "бал_вгору_МВт",
                        "бал_вниз_МВт", "бал_вгору_грн", "бал_вниз_грн"]

one_pred_results = []
for pred in candidate_predictors:
    fd_y_tmp, fd_x_tmp = build_fd_for_predictors(
        daily_model, response_prefix="ціна_МВт",
        predictor_prefixes=[pred], n_basis=4)

    res = fwc.funweightclust(datax=fd_x_tmp, datay=fd_y_tmp, K=2,
                             model="ABQKDK", modely="EII", init="kmeans",
                             nb_rep=10, min_individuals=2, threshold=0.1,
show=False)

    if not isinstance(res, dict):
        one_pred_results.append({
            "predictor": pred, "converged": bool(res.converged),
            "bic": float(res.bic), "icl": float(res.icl),
            "cluster_sizes": np.bincount(res.cl).tolist()})

one_pred_df = pd.DataFrame(one_pred_results)

# Вибір кількості кластерів K за BIC
k_results, k_objects = [], {}
for K in [2, 3, 4, 5]:

```

```

fd_y_tmp, fd_x_tmp = build_fd_for_predictors(
    daily_model, response_prefix="ціна_МВт",
    predictor_prefixes=["бал_впрог_рпн"], n_basis=4)

res = fwc.funweightclust(datax=fd_x_tmp, datay=fd_y_tmp, K=K,
                        model="ABQKDK", modely="EII", init="kmeans",
                        nb_rep=10, min_individuals=2, threshold=0.1,
show=False)
    if not isinstance(res, dict):
        k_results.append({"K": K, "converged": bool(res.converged),
                        "bic": float(res.bic), "icl": float(res.icl)})

    k_objects[K] = res

k_df = pd.DataFrame(k_results)
best_K = int(k_df[k_df.converged].sort_values("bic",
ascending=False).iloc[0]["K"])

# Порівняння з K-means та DBSCAN
from sklearn.cluster import KMeans, DBSCAN
from sklearn.metrics import silhouette_score

X_basis = fd_x_best.coefficients          # В-сплайн-коефіцієнти предиктора

labels_km = KMeans(n_clusters=4, random_state=42,
n_init=10).fit_predict(X_basis)
labels_db = DBSCAN(eps=0.5, min_samples=10).fit_predict(X_basis)
labels_fwc = clustered_dataset['cluster'].values

mask = labels_db != -1                    # без шумових точок DBSCAN
comparison = pd.DataFrame({
    'Метод': ['funWeightClust', 'K-means', 'DBSCAN'],
    'Silhouette ↑': [
        round(silhouette_score(X_basis, labels_fwc), 3),
        round(silhouette_score(X_basis, labels_km), 3),
        round(silhouette_score(X_basis[mask], labels_db[mask]), 3)]
})

# Кластер вчорашньої доби
master = master.merge(daily_clusters, on='дата', how='left')

cluster_map = daily_clusters.set_index('дата')['cluster_fwc']
master['cluster_lag_1d'] = (master['дата'] -
pd.Timedelta(days=1)).map(cluster_map)

# Часові, лагові та rolling-ознаки без витoku даних
df['timestamp'] = df['дата'] + pd.to_timedelta(df['година'] - 1, unit='h')
df = df.sort_values('timestamp').reset_index(drop=True)

# Часові ознаки (циклічне кодування години)
df['dayofweek'] = df['timestamp'].dt.dayofweek
df['month'] = df['timestamp'].dt.month
df['is_weekend'] = (df['dayofweek'] >= 5).astype(int)
df['hour_sin'] = np.sin(2 * np.pi * df['година'] / 24)
df['hour_cos'] = np.cos(2 * np.pi * df['година'] / 24)

# Лаги ціни (від логарифмованої ціни)
for lag in [1, 2, 24, 48, 168]:
    df[f'price_lag_{lag}'] = df['log_ціна_МВт'].shift(lag)

# Rolling-ознаки: shift(1) ПЕРЕД rolling() -> без витoku
df['price_roll_mean_3'] = df['log_ціна_МВт'].shift(1).rolling(3).mean()
df['price_roll_mean_24'] = df['log_ціна_МВт'].shift(1).rolling(24).mean()
df['price_roll_mean_168'] = df['log_ціна_МВт'].shift(1).rolling(168).mean()

```

```

df['price_roll_std_24'] = df['log_ціна_МВт'].shift(1).rolling(24).std()

# Лаги предикторів (значення тієї ж години вчора)
for col in ['імпорт_МВт', 'експорт_МВт', 'бал_вгору_МВт',
            'бал_вниз_МВт', 'бал_вгору_грн', 'бал_вниз_грн']:
    df[f'{col}_lag_24'] = df[col].shift(24)

# Цільові змінні
df['target_1h'] = df['log_ціна_МВт'].shift(-1) # на 1 годину вперед
df['target_24h'] = df['log_ціна_МВт'].shift(-24) # на добу вперед

# Метрики якості та параметри моделі
from sklearn.metrics import mean_absolute_error, mean_squared_error, r2_score

def mape(y_true, y_pred):
    y_true, y_pred = np.array(y_true), np.array(y_pred)
    mask = y_true != 0
    return np.mean(np.abs((y_true[mask] - y_pred[mask]) / y_true[mask])) * 100

def evaluate(y_true, y_pred, name='model'):
    return {'Модель': name,
            'MAE': round(mean_absolute_error(y_true, y_pred), 2),
            'RMSE': round(np.sqrt(mean_squared_error(y_true, y_pred)), 2),
            'MAPE, %': round(mape(y_true, y_pred), 2),
            'R²': round(r2_score(y_true, y_pred), 4)}

XGB_PARAMS = dict(n_estimators=400, max_depth=6, learning_rate=0.05,
                  subsample=0.8, colsample_bytree=0.8,
                  objective='reg:squarederror',
                  random_state=42, early_stopping_rounds=20)

# Хронологічний поділ на train/valid/test
df_1h = df.dropna(subset=features_1h_clust +
                  ['target_1h']).reset_index(drop=True)

n = len(df_1h)
train_1h = df_1h.iloc[:int(n * 0.70)].copy()
valid_1h = df_1h.iloc[int(n * 0.70):int(n * 0.85)].copy()
test_1h = df_1h.iloc[int(n * 0.85):].copy()

# Три варіанти моделі
target = 'target_1h'

# Варіант 1 – базова модель (без кластерів)
m1 = XGBRegressor(**XGB_PARAMS)
m1.fit(train_1h[features_1h_base], train_1h[target],
       eval_set=[(valid_1h[features_1h_base], valid_1h[target])],
       verbose=False)
pred_base = np.expm1(m1.predict(test_1h[features_1h_base]))
actual = np.expm1(test_1h[target].values)
met_base = evaluate(actual, pred_base, 'One-step | Базова')

# Варіант 2 – кластер як ознака
m2 = XGBRegressor(**XGB_PARAMS)
m2.fit(train_1h[features_1h_clust], train_1h[target],
       eval_set=[(valid_1h[features_1h_clust], valid_1h[target])],
       verbose=False)
pred_clust = np.expm1(m2.predict(test_1h[features_1h_clust]))
met_clust = evaluate(actual, pred_clust, 'One-step | 3 кластером')

# Варіант 3 – окрема модель для кожного кластера
models_sep, preds_sep = {}, []
for cl in sorted(train_1h['cluster_lag_1d'].dropna().unique()):

```

```

tr = train_1h[train_1h['cluster_lag_1d'] == cl]
va = valid_1h[valid_1h['cluster_lag_1d'] == cl]
te = test_1h[test_1h['cluster_lag_1d'] == cl]
if len(tr) < 50 or len(va) < 10 or len(te) < 10:
    continue
m = XGBRegressor(**XGB_PARAMS)
m.fit(tr[features_1h_base], tr[target],
      eval_set=[(va[features_1h_base], va[target])], verbose=False)
models_sep[cl] = m
part = te[['timestamp', target]].copy()
part['pred'] = np.expml(m.predict(te[features_1h_base]))
part['actual'] = np.expml(part[target])
preds_sep.append(part)

sep_df = pd.concat(preds_sep).sort_values('timestamp').reset_index(drop=True)
met_sep = evaluate(sep_df['actual'], sep_df['pred'], 'One-step | Окремі по
кластерах')

# Набір ознак для day-ahead
df['target_24h'] = df['log_ціна_МВт'].shift(-24)

features_24h_base = [
    'година', 'dayofweek', 'month', 'quarter', 'is_weekend',
    'hour_sin', 'hour_cos', 'day_of_year',
    'price_lag_24', 'price_lag_48', 'price_lag_168',          # тільки >= 24
    год
    'price_roll_mean_24', 'price_roll_mean_168', 'price_roll_std_24',
    'імпорт_МВт_lag_24', 'експорт_МВт_lag_24',
    'бал_вгору_МВт_lag_24', 'бал_вниз_МВт_lag_24',
    'бал_вгору_грн_lag_24', 'бал_вниз_грн_lag_24',
]
features_24h_clust = features_24h_base + ['cluster_lag_1d']

# Порівняння методів кластеризації у прогнозуванні
cluster_methods = {'funWeightClust': 'cluster_fwc',
                   'K-means': 'cluster_kmeans',
                   'DBSCAN': 'cluster_dbscan'}

comparison = []

for method_name, cluster_col in cluster_methods.items():
    cl_map = daily_clusters.set_index('дата')[cluster_col]
    df_tmp = df.copy()
    df_tmp['cluster_lag_1d'] = (df_tmp['дата'] -
pd.Timedelta(days=1)).map(cl_map)

    features_tmp = features_24h_base + ['cluster_lag_1d']
    d = df_tmp.dropna(subset=features_tmp +
['target_24h']).reset_index(drop=True)
    n = len(d)
    tr, va, te = d.iloc[:int(n*.70)], d.iloc[int(n*.70):int(n*.85)],
d.iloc[int(n*.85):]

    m = XGBRegressor(**XGB_PARAMS)
    m.fit(tr[features_tmp], tr['target_24h'],
          eval_set=[(va[features_tmp], va['target_24h'])], verbose=False)

    pred = np.expml(m.predict(te[features_tmp]))
    actual = np.expml(te['target_24h'].values)
    comparison.append(evaluate(actual, pred, f'Day-ahead | {method_name}'))
comparison.append(met_24h_base)          # + базова модель без кластерів
comparison_df = pd.DataFrame(comparison)

```

ДОДАТОК Б. ГРАФІЧНІ МАТЕРІАЛИ

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
ННІ прикладного системного аналізу · Кафедра ММСА

Інтелектуальна система прогнозування погодинних цін на основі кластеризації режимів ринку електроенергії України

Дипломна робота на здобуття ступеня бакалавра
Спеціальність 124 «Системний аналіз»

Виконала
Кіндзерська Анастасія Олександрівна
група КА-23

Науковий керівник
Недашківська Н. І.
професор кафедри ММСА, д.т.н., професор

Київ · 2026

Актуальність

- 1 Висока волатильність**
Ринок електроенергії України функціонує в умовах значної невизначеності — від атак на інфраструктуру до зміни режимів генерації.
- 2 Різні режими функціонування**
Добові профілі цін відрізняються в різні періоди: стабільний, високий, піковий, дефіцитний.
- 3 Обмеження класичних моделей**
Лінійні та статистичні підходи погано підходять для нелінійних залежностей, які присутні у вхідних даних.

Період
дослідження

2022–2025

1315

діб погодинних спостережень

31 560

погодинних рядків у датасеті

Мета, об'єкт і предмет дослідження

МЕТА

Підвищення точності та інтерпретованості прогнозування погодинних цін електроенергії в Україні шляхом побудови інтелектуальної системи, що поєднує кластеризацію режимів ринку та машинне навчання.

ОБ'ЄКТ

Процес формування погодинних цін на ринку електроенергії України та добові режими, описувані часовими рядами цін, імпорту, експорту та балансуючих показників.

ПРЕДМЕТ

Методи кластеризації добових профілів ринку та методи машинного навчання для прогнозування погодинних цін з урахуванням виявлених режимів.

Кіндзерська А. О., КА-23

Формальна постановка задачі

Позначення

$D = \{d_1, \dots, d_n\}$ — множина днів спостереження
 $t \in \{1, \dots, 24\}$ — година доби
 $Y_i(t)$ — погодинна ціна електроенергії у добу d_i
 $X_i(t)$ — функціональні предиктори:
імпорт, експорт, обсяги і ціни балансування
 R_i — режим ринку доби d_i
 $\hat{Y}_i(t)$ — прогнозоване значення ціни

Задача дослідження

Кластеризація режимів

$$R = f(X, Y)$$

Прогнозування з урахуванням режиму

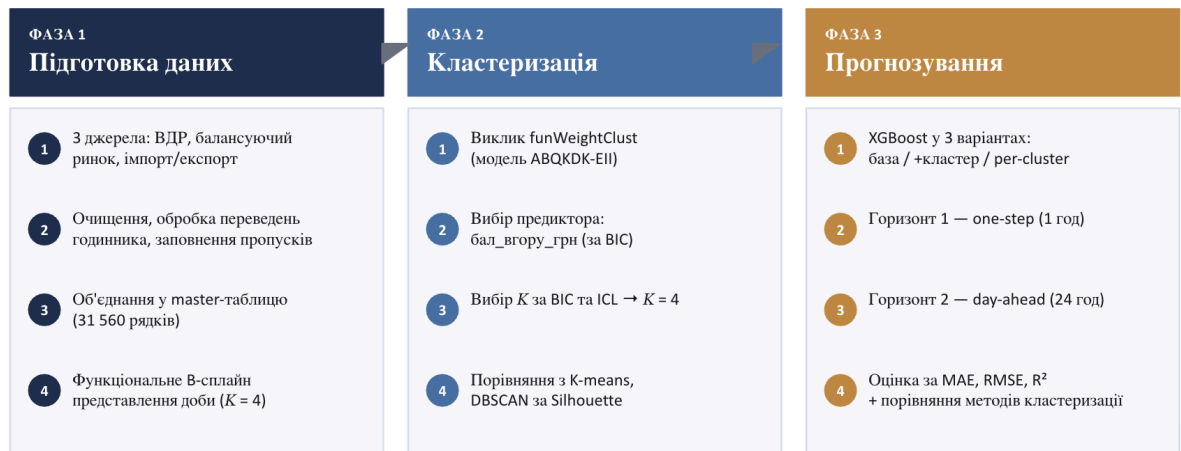
$$\hat{Y} = g(X, R)$$

Знайти функції f і g такі, що режим ринку R та прогноз $\hat{Y}$ максимально точно описують реальну динаміку.

Кіндзерська А. О., КА-23

Схема дослідження

Три послідовні фази: підготовка даних → кластеризація режимів → прогнозування цін



Кіндзерська А. О., КА-23

Схема архітектури системи



Функціональна схема

Повний потік даних: від трьох джерел до прогнозу у веб-інтерфейсі

9
фаз обробки
Ф1–Ф9

1315
діб у
датасеті

4
режими
ринку

6
моделей
XGBoost

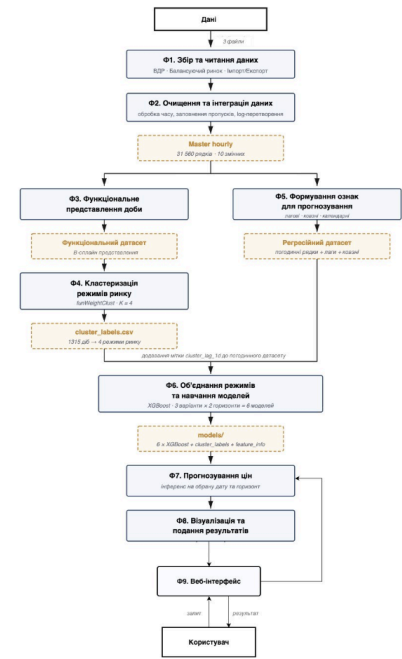
1 Підготовка даних (Ф1–Ф2)
3 файли об'єднуються в одну master-таблицю: 31 560 погодинних рядків, 10 змінних.

2 Дві паралельні гілки (Ф3 / Ф5)
Зліва — В-сплайн представлення доби для кластеризації; справа — лаги, ковзні й календарні ознаки для регресії.

3 Кластеризація режимів (Ф4)
funWeightClust виділяє K = 4 режими ринку

4 Навчання та прогноз (Ф6–Ф9)
6 моделей XGBoost (3 варіанти × 2 горизонти) → інференс на обрану дату → веб-інтерфейс.

Мітка cluster_lag_1d (режим учорашньої доби) додається в регресійний датасет. Режим поточної доби невідомий — це запобігає витoku даних.



Вхідні дані

Відкриті погодинні дані ринку електроенергії з порталу Energy Map, торгова зона «ОЕС України» (синхронізована з ENTSO-E)

ВДР
Внутрішньодобовий ринок

- Погодинна ціна (грн/МВт·год)
- Цільова змінна моделі

Балансуючий ринок
Врегулювання небалансу

- Обсяги балансування ↑ / ↓
- Ціни балансування ↑ / ↓

Імпорт / Експорт
Міждержавні перетоки

- Погодинні обсяги перетоків
- Агрегація по 5 країнах

1315
діб

7
функціональних
змінних

70/15/15
train / valid /
test, %

Функціональне представлення доби

Кожна доба — одне функціональне спостереження

24 погодинних значення кожної змінної згладжуються кубічним В-сплайн базисом:

Зменшення розмірності: 24 точки $\rightarrow K = 4$ коефіцієнти

Збереження форми добового профілю — нічний провал, ранковий і вечірній піки

В-сплайн розклад

$$Y_i(t) = \sum c_{ik} \phi_k(t)$$

$k = 1 \dots K$

Параметри:

- базис: кубічний В-сплайн, $K = 4$
- стандартизація (z-score) по кожній змінній

Кіндзерська А. О., КА-23

Алгоритм funWeightClust

Кластерно-зважена модель функціональних даних (Cluster Weighted Model)

$$f(X, Y) = \sum \pi_m \cdot f_m(X) \cdot f_m(Y | X) \quad m = 1 \dots G$$

1 Сумісна модель X і Y

Одночасно враховує і схожість предикторів, і характер регресії $Y(t)$ на $X(t)$ усередині кластера.

2 FPCA + сім'я моделей

Функціональний аналіз головних компонент усередині кластера; обрана модель ABQKDK-EII забезпечує стабільну збіжність.

3 ЕМ-алгоритм

Е-крок — апостеріорні ймовірності; М-крок — оновлення параметрів. Вибір K за BIC та ICL.

Кіндзерська А. О., КА-23

funWeightClust — математична модель

FPCA усередині кластера

Functional Principal Component Analysis

$$X_i(t) = \mu(t) + \sum_j \alpha_{ij} \xi_j(t)$$

Кожна крива розкладається на середню $\mu(t)$ та функціональні головні компоненти $\xi_j(t)$

Розмірність d_k обирається за критерієм Cattell («лікоть» власних значень)

ЕМ-ітерація

Expectation-Maximization

Е-крок: оновлення апостеріорних ймовірностей належності спостереження до кластера

М-крок: оновлення параметрів моделі (π_m , середні μ_m , коваріаційні структури) максимізацією правдоподібності

→ Ітерації до збіжності; повторні запуски ($nb_rep = 10$) для стійкості

Сім'я моделей: шість економних моделей коваріаційної структури (AKJBKQKDK-VVV найгнучкіша)
Вхід: $Y(t)$ = ціна ВДР, $X(t)$ = бал_вгору_грн (1 з 6 предикторів, обрано за BIC; пари давали нестійку збіжність)

Кіндзерська А. О., КА-23

Критерії якості кластеризації

BIC

Bayesian Information Criterion

$$BIC = L^{(m)} - (\tau / 2) \cdot \log n$$

баланс точності та складності моделі; більше значення — краще

ICL

Integrated Completed Likelihood

$$ICL = BIC(K) - \sum_i \sum_k \tau_{ik} \cdot \log \tau_{ik}$$

BIC з додатковим штрафом за перетинання кластерів

Silhouette

Коефіцієнт силуету

$$s(i) = [b(i) - a(i)] / \max\{a(i), b(i)\}$$

геометрична компактність кластерів; $S \in [-1, 1]$ > 0.5 — гарна структура

Кіндзерська А. О., КА-23

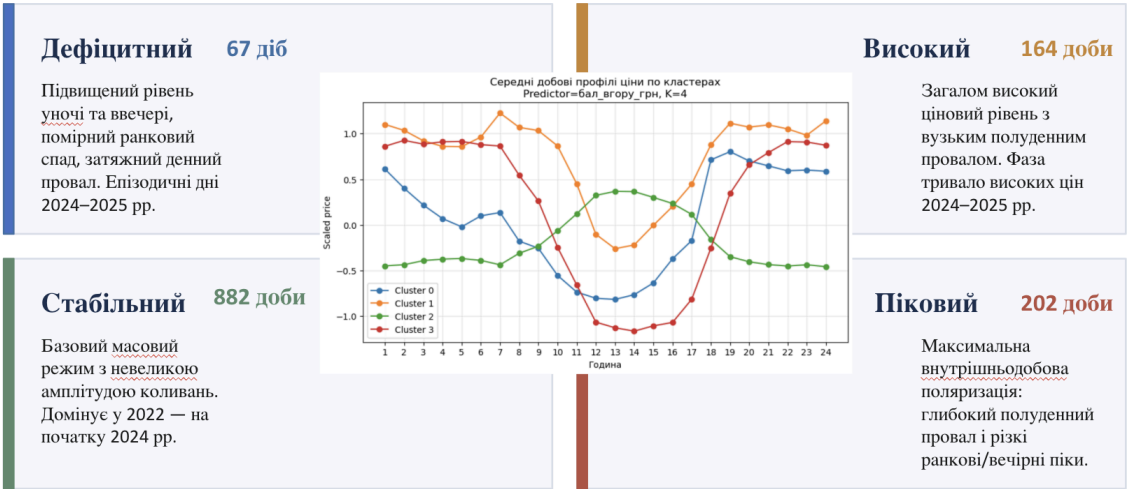
Критерії якості прогнозування

MAE <i>Mean Absolute Error</i>	$MAE = (1 / n) \cdot \sum y_i - \hat{y}_i $ <i>одиниці = грн/МВт·год; стійка до окремих викидів</i>
RMSE <i>Root Mean Square Error</i>	$RMSE = \sqrt{ (1 / n) \cdot \sum (y_i - \hat{y}_i)^2 }$ <i>сильніше штрафує великі помилки; чутлива до пікових значень</i>
MAPE <i>Mean Absolute Percentage Error</i>	$MAPE = (100\% / n) \cdot \sum (y_i - \hat{y}_i) / y_i $ <i>у відносних одиницях; ненадійна за малих y_i — використовується як допоміжна</i>

Кіндзерська А. О., КА-23

Виявлені режими ринку

$K = 4$ (обрано за BIC) · 1 функціональний предиктор: бал_вгору_грн — найкращий за BIC серед 6 кандидатів



Середні добові профілі ціни по кластерах
Predictor=бал_вгору_грн, K=4

Cluster 0
Cluster 1
Cluster 2
Cluster 3

Година

Кіндзерська А. О., КА-23

Порівняння методів кластеризації

Метод	K	Silhouette $\uparrow$	Враховує форму та залежність $X \rightarrow Y$
funWeightClust	4	0.183	так
K-means	4	0.451	ні
DBSCAN	3	0.504	ні

Висновок

Silhouette оцінює лише геометрію — funWeightClust поступається за нею. Але саме він враховує функціональну природу даних та зв'язок предиктора з цільовою змінною, тому подальше порівняння йде через якість прогнозування.

Кіндзерська А. О., КА-23

Прогнозування: 2 горизонти × 3 варіанти моделі

Усі моделі — XGBoost з однаковими гіперпараметрами; різниця лише у наборі ознак і обсязі тренувальної вибірки

ГОРИЗОНТИ ПРОГНОЗУ

One-step-ahead

1 год

Прогноз ціни наступної години — задача оперативного управління.

Доступні: усі лаги (1, 2, 24, 48, 168) + поточні значення предикторів

Day-ahead

24 год

Прогноз тієї ж години завтра — відповідає реальним торгам на ВДР.

Доступні: лише лаги ≥ 24 год; короткі лаги і поточні предиктори виключені

ВАРІАНТИ МОДЕЛІ

1

Базова

XGBoost без кластерної інформації — еталон для порівняння.

2

3 кластером як ознакою

До набору ознак додається cluster_lag_1d — мітка режиму вчорашньої доби.

3

Окремі моделі по кластерах

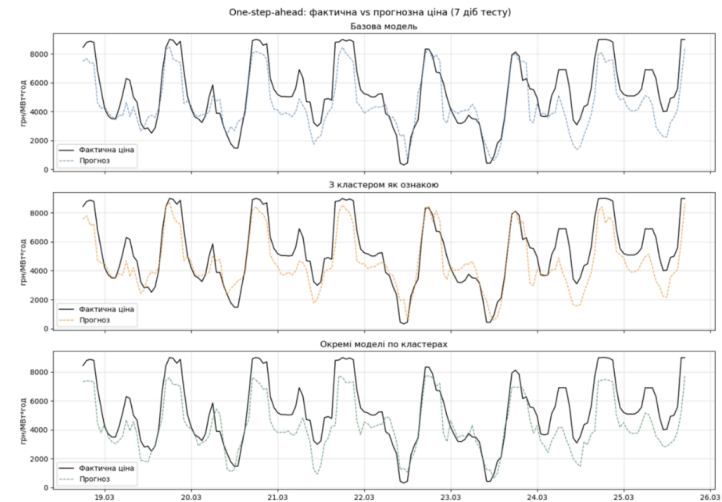
Для кожного режиму — окрема модель XGBoost (4 моделі за 4 кластерами).

⚠ Кластер поточної доби невідомий у момент прогнозу — використовується мітка вчорашньої доби (запобігання витоку даних)

Кіндзерська А. О., КА-23

Результати: One-step-ahead (1 год)

3 варіанти моделі XGBoost



Кіндзерська А. О., КА-23

Ключовий висновок

Усі 3 варіанти дають практично однакову якість прогнозу. Базова модель та з кластером справляються трохи краще.

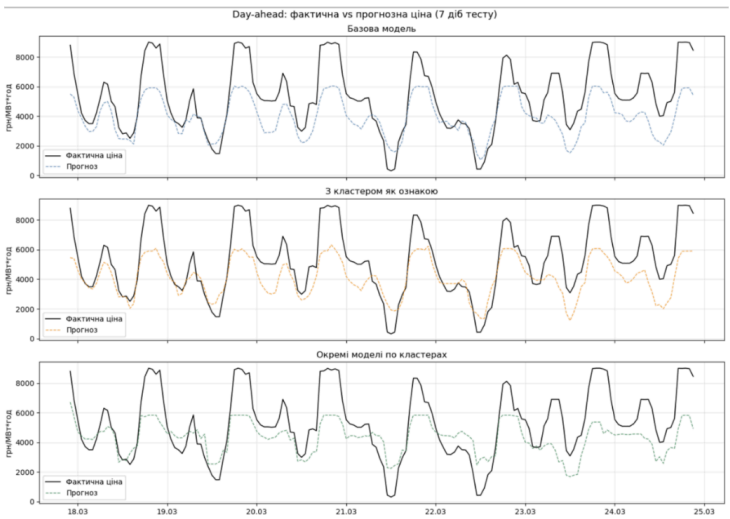
Чому?

Лагові ознаки (`price_lag_1`, `price_lag_24`) вже несуть основну інформацію — кластерна мітка надлишкова

R^2 базової моделі = 0.707
MAE базової моделі = 1127.37

Результати: Day-ahead (24 год)

Day-ahead (24 год)



Кіндзерська А. О., КА-23

Ключовий результат

-6.3 %

MAE для day-ahead з `funWeightClust` порівняно з базовою моделлю

Чому?

Без коротких лагів (< 24 год) інформація про режим ринку компенсує цю прогалину

R^2 зростає: 0.565 → 0.613

Висновки

1

Побудовано пайплайн обробки даних

Сформовано єдиний погодинний датасет для ОЕС України (1315 діб, 2022–2025) з очищенням, обробкою переведень годинника та логарифмуванням ціни.

2

Виділено 4 змістовні режими ринку

Алгоритм funWeightClust дав інтерпретовані кластери — дефіцитний, високий, стабільний, піковий — що відповідають реальній динаміці ринку.

3

Кластеризація покращує day-ahead прогноз

Включення мітки кластера як ознаки знижує MAE з 1427 до 1337 грн/МВт-год (–6.3 %, R^2 зростає з 0.565 до 0.613).

4

Перевага funWeightClust над K-means

За точністю прогнозу funWeightClust перевершує K-means; з DBSCAN — паритет, але без потреби окремої обробки шумових точок.

Кіндзерська А. О., КА-23

Дякую за увагу!

Кіндзерська А. О., КА-23