

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ХАРЧОВИХ ТЕХНОЛОГІЙ**

Інститут (факультет) _____ ННІТІ ім. акад. І.С. Гулого _____
Кафедра _____ Електропостачання та енергоменеджменту _____

«До захисту в ЕК»
Директор інституту
_____ Сергій БЛАЖЕНКО _____
(підпис) (ім'я та прізвище)

«__» _____ грудня 2025 р.

«До захисту допущено»
Завідувач кафедри
_____ Сергій БАЛЮТА _____
(підпис) (ім'я та прізвище)

«__» _____ грудня 2025 р.

**КВАЛІФІКАЦІЙНА РОБОТА
НА ЗДОБУТТЯ ОСВІТНЬОГО СТУПЕНЯ МАГІСТРА**

зі спеціальності 141 «Електроенергетика, електротехніка та електромеханіка»
(код та назва спеціальності)

освітньо-професійної програми «Електротехнічні системи електроспоживання»
на тему: «Моделювання та прогнозування споживання електроенергії на основі
аналізу часових рядів та методів машинного навчання»

Виконав: здобувач 2 курсу, групи ЕЛ-2-6М

_____ Кулініч Яків Борисович _____
(прізвище, ім'я, по батькові повністю) (підпис)

Керівник _____ Романов Микола Сергійович _____
(прізвище, ім'я та по батькові повністю) (підпис)

Консультанти _____ (ім'я та прізвище) _____ (підпис)

_____ (ім'я та прізвище) _____ (підпис)

_____ (ім'я та прізвище) _____ (підпис)

Рецензент _____ Ярослав СМІТЮХ _____
(ім'я та прізвище) (підпис)

Я як здобувач(ка) Національного університету харчових технологій розумію і підтримую політику університету з академічної доброчесності. Я не надавав(-ла) і не одержував(-ла) недозволеної допомоги під час підготовки цієї роботи. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело

Здобувач _____
(підпис)

Київ – 2025 р.

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ХАРЧОВИХ ТЕХНОЛОГІЙ

Інститут (факультет) Навчально - науковий інженерно - технічний інститут ім. акад. І. С. Гулого

Кафедра Електропостачання та енергоменеджменту

Освітній ступінь магістр

Спеціальність 141 «Електроенергетика, електротехніка та електромеханіка»
(код і назва)

Освітньо-професійна програма «Електротехнічні системи електроспоживання»
(назва)

ЗАТВЕРДЖУЮ

Завідувач

кафедри ЕПЕМ

/Сергій БАЛЮТА/

“17” вересня 2025 року

З А В Д А Н Н Я

НА КВАЛІФІКАЦІЙНУ РОБОТУ ЗДОБУВАЧА

Кулінічу Якову Борисовичу

(прізвище, ім'я, по батькові)

1. Тема роботи Моделювання та прогнозування споживання електроенергії на основі аналізу часових рядів та методів машинного навчання

керівник роботи Романов Микола Сергійович, доцент, кандидат технічних наук
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом закладу вищої освіти від “17 вересня 2025 року № 712-кс”

2. Строк подання здобувачем роботи 10 грудня 2025 року

3. Вихідні дані до роботи: данні про сумарне погодинне споживання електроенергії підприємства, доповнене температурою в регіоні, інформацією про санітарні зміни, аварійні або планові зупинки виробництва та періоди повітряних тривог

4. Зміст пояснювальної записки (перелік питань, які потрібно розробити) 1. Огляд методів прогнозування: розкриття принципів роботи статистичних моделей, моделей машинного навчання (Random Forest, Gradient Boosting, Support Vector Regression) та нейронних мереж (LSTM, GRU). 2. Попередня обробка даних: очищення, нормалізація, обробка пропусків, виявлення аномалій, формування ознак та підготовка вибірки. 3. Вибір та обґрунтування оптимальних моделей, проведення їх порівняння за ключовими метриками (MAE, RMSE, MAPE), опис підходів до гіперпараметричної оптимізації та валідації результатів. 4. Економічний розрахунок ефективності використання найточнішої моделі для прийняття рішень щодо оптимального розподілу закупівель електроенергії між ринком «на добу наперед» (РДН) та внутрішньодобовим ринком (ВДР).

5. Перелік графічного матеріалу ”Презентація до пояснювальної записки” (25 слайдів)

6. Консультанти розділів роботи

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

7. Дата видачі завдання _____ 17 вересня 2025 р. _____

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів виконання кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1	Отримання завдання на магістерську роботу	17.09.2025 р.	
2	Вступ	17 - 18.09.2025 р.	
3	Аналіз предметної області та актуальність задачі прогнозування споживання електроенергії	19 - 30.09.2025 р.	
4	Аналіз вихідних даних та попередня обробка	01 - 15.10.2025 р.	
5	Вибір архітектур моделей	16 - 22.10.2025 р.	
6	Розроблення та навчання моделей прогнозування	23.10.2025 - 07.11.2025 р.	
7	Оцінка точності прогнозу	08 - 12.11.2025 р.	
8	Аналіз важливості ознак та SHAP	13 - 18.11.2025 р.	
9	Аналіз залишків	19 - 22.11.2025 р.	
10	Порівняльний аналіз моделей	23 - 28.11.2025 р.	
12	Економічне обґрунтування застосування прогнозної моделі для оптимізації закупівлі електроенергії	29.11.2025 - 04.12.2025 р.	
13	Висновки	05 - 06.12.2025 р.	
14	Список виористаних джерел	07 - 08.12.2025 р.	
15	Оформлення пояснювальної записки	09 - 10.12.2025 р.	
16	Подання готової роботи для перевірки на плагіат	10.12.2025 р.	

Здобувач _____
(підпис)

Кулініч Я.Б. _____
(прізвище та ініціали)

Керівник роботи _____
(підпис)

Романов М. С. _____
(прізвище та ініціали)

АНОТАЦІЯ

Кулініч Я. Б. Моделювання та прогнозування споживання електроенергії на основі аналізу часових рядів та методів машинного навчання

141 «Електроенергетика, електротехніка та електромеханіка»

Національний Університет Харчових Технологій

Київ 2025

Магістерська робота присвячена дослідженню методів прогнозування погодинного споживання електроенергії із застосуванням класичних статистичних моделей, алгоритмів машинного навчання та методів глибокого навчання. У роботі розглянуто повний цикл формування прогнозовної моделі: від аналізу властивостей вихідного часового ряду та очищення даних до побудови розширеного набору ознак, навчання моделей і оцінювання їх точності, інтерпретованості та придатності для практичного використання.

Проведено порівняння моделей ARIMA, SARIMA, Random Forest, XGBoost, Support Vector Regression, LSTM, GRU та CNN за показниками MAE, RMSE, MAPE, R^2 , а також досліджено структуру залишків для виявлення недопомодельованих компонент.

Здійснено економічну оцінку застосування прогнозовної моделі у процесі вибору оптимальної стратегії закупівлі електроенергії між ринком «на добу наперед» та внутрішньодобовим ринком.

Отримані результати демонструють потенціал сучасних методів машинного навчання у прогнозуванні складних енергетичних процесів та окреслюють напрями подальших досліджень у межах комбінованих моделей, інтерпретованості та економічної оптимізації.

Ключові слова: прогнозування, часові ряди, енергоспоживання, машинне навчання, SVR, нейронні мережі, SHAP, економічна оптимізація

ANNOTATION

Kulinich Y. B. Modeling and Forecasting Electricity Consumption Based on Time-Series Analysis and Machine Learning Methods

141 "Electricity, electrical engineering and electromechanics"

National university of food technologies

Kyiv 2024

The master's thesis examines advanced methods for forecasting hourly electricity consumption using classical statistical models, machine learning algorithms, and deep learning techniques. The study covers the full development cycle of a forecasting system: from data analysis, preprocessing, anomaly treatment, and feature engineering to model training, performance evaluation, interpretability analysis, and economic assessment.

A comprehensive comparison of ARIMA, SARIMA, Random Forest, XGBoost, Support Vector Regression, LSTM, GRU, and CNN models was conducted using MAE, RMSE, MAPE, and R^2 metrics, complemented by residual analysis to reveal structural components not captured by the models.

An economic assessment of applying the forecasting model in the process of selecting the optimal electricity procurement strategy between the day-ahead market and the intraday market has been carried out.

The study highlights the effectiveness of modern machine learning techniques in modelling complex energy consumption processes and outlines perspectives for further research in hybrid modelling approaches, enhanced interpretability, and economic optimization.

Keywords: forecasting, time series, electricity consumption, machine learning, SVR, neural networks, SHAP, economic optimisation.

ЗМІСТ

ВСТУП	9
РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА АКТУАЛЬНІСТЬ ЗАДАЧІ ПРОГНОЗУВАННЯ СПОЖИВАННЯ ЕЛЕКТРОЕНЕРГІЇ	13
1.1. Роль прогнозування енергоспоживання у сучасних енергетичних системах.....	13
1.2. Особливості споживання електроенергії промисловими підприємствами.....	14
1.3. Проблематика побудови точних прогнозів	17
1.4. Аналіз існуючих рішень і постановка задачі	18
РОЗДІЛ 2. АНАЛІЗ ВИХІДНИХ ДАНИХ ТА ПОПЕРЕДНЯ ОБРОБКА....	20
2.1. Опис об'єкта дослідження	20
2.2. Характеристика наявного датасету.....	22
2.3. Виявлення та обробка пропусків даних	25
2.4. Виявлення аномалій у споживанні	30
2.5. Формування набору ознак (feature engineering)	34
2.6. Нормалізація та розподіл даних на тренувальну та тестову вибірки.....	38
2.6.1. Очищення датасету після формування ознак	39
2.6.2. Особливості нормалізації ознак для різних типів моделей	40
2.6.3. Часове розділення даних на тренувальну та тестову вибірки.	43
РОЗДІЛ 3. ВИБІР АРХІТЕКТУР МОДЕЛЕЙ	45
3.1. Класичні моделі часових рядів: AR, MA, ARIMA, SARIMA, SARIMAX	46
3.2. Алгоритми машинного навчання: Random Forest, Gradient Boosting, Support Vector Regression	49
3.3. Нейронні мережі для часових рядів: LSTM, GRU, CNN	51
3.4. Узагальнення вибору моделей для дослідження	54
РОЗДІЛ 4. РОЗРОБЛЕННЯ ТА НАВЧАННЯ МОДЕЛЕЙ ПРОГНОЗУВАННЯ.....	55
4.1. Авторегресійна модель AR(p).....	55

4.2	Модель ковзного середнього $MA(q)$	62
4.3	Модель ARIMA	66
4.4	Модель SARIMA(p, d, q)(P, D, Q, s)	70
4.5	Модель SARIMAX (Seasonal ARIMA with Exogenous Variables) ...	75
4.6	Модель Random Forest Regression	78
4.7	Модель Gradient Boosting Regression (XGBoost / LightGBM / CatBoost)	84
4.8	Модель Support Vector Regression (SVR).....	90
4.9	Формування вхідних даних для LSTM/GRU/CNN.....	93
4.10	Модель LSTM	94
4.11	Модель GRU (Gated Recurrent Unit)	99
4.12	Модель CNN (1D Convolutional Neural Network)	104
4.13	Формування послідовностей (Sliding Window) для моделей LSTM, GRU та CNN.....	110
4.14	Гіперпараметрична оптимізація	112
РОЗДІЛ 5. ОЦІНКА ТОЧНОСТІ ПРОГНОЗУ		116
5.1	Метрики оцінки точності прогнозу	116
5.2	Аналіз лінійних моделей	120
5.3	Аналіз моделей машинного навчання.....	125
5.4	Аналіз моделей глибокого навчання	129
РОЗДІЛ 6 АНАЛІЗ ВАЖЛИВОСТІ ОЗНАК ТА SHAP		133
6.1	Методи оцінки важливості ознак	133
6.2	SHAP-аналіз моделі XGBoost.....	136
6.3	SHAP-аналіз моделі Random Forest.....	140
6.3	SHAP-аналіз моделі SVR.....	143
6.4	Важливість ознак для LSTM (Permutation Importance)	147
6.5	Важливість ознак для GRU (Permutation Importance).....	150
6.6	Важливість ознак для CNN (Permutation Importance).....	154
РОЗДІЛ 7. АНАЛІЗ ЗАЛИШКІВ		158
7.1	Методи оцінки залишків.....	158
7.2	Аналіз залишків моделі Random Forest	163

7.3 Аналіз залишків моделі XGBoost.....	166
7.4 Аналіз залишків моделі SVR	170
7.5 Аналіз залишків LSTM	174
7.6 Аналіз залишків GRU.....	178
7.7 Аналіз залишків CNN	182
РОЗДІЛ 8. ПОРІВНЯЛЬНИЙ АНАЛІЗ МОДЕЛЕЙ.....	187
8.1 Порівняння точності прогнозу.....	187
8.2 Порівняння структури впливів (Feature Importance та SHAP)....	188
8.3 Порівняння залишків моделей	189
8.4 Узагальнені висновки та оцінка придатності моделей	190
РОЗДІЛ 9. ЕКОНОМІЧНЕ ОБҐРУНТУВАННЯ ЗАСТОСУВАННЯ ПРОГНОЗНОЇ МОДЕЛІ ДЛЯ ОПТИМІЗАЦІЇ ЗАКУПІВЛІ ЕЛЕКТРОЕНЕРГІЇ	194
9.1 Методологія моделювання економічного ефекту.....	194
9.2 Моделювання небалансів та необхідність коригування прогнозу	195
9.3 Розрахунок витрат за альтернативними стратегіями закупівлі..	197
9.4 Інтерпретація отриманих результатів.....	198
9.5 Обмеження моделювання	199
9.6 Висновки щодо економічної доцільності застосування прогнозної моделі.....	199
ВИСНОВКИ	201
ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ	204

ВСТУП

Сучасний енергетичний сектор України переживає глибоку трансформацію, спричинену як глобальними тенденціями діджиталізації та декарбонізації, так і унікальними внутрішніми викликами, пов'язаними з воєнними діями, порушеннями цілісності енергетичної інфраструктури та зростанням невизначеності на ринку електроенергії. За таких умов прогнозування електроспоживання перетворюється на один із визначальних інструментів підтримання енергетичної стабільності підприємства, оптимізації їх витрат та мінімізації ризиків невідповідності між фактичним і закупленим обсягом електричної енергії. Саме у сфері промислових підприємств, де навантажувальні профілі характеризуються високою варіативністю та залежністю від технологічних процесів, точне прогнозування стає необхідною передумовою економічної ефективності та безперервності виробництва.

Наукова проблема, що лежить в основі даної роботи, полягає у розробленні та обґрунтуванні методів короткострокового прогнозування споживання електроенергії з використанням сучасних статистичних та машинних підходів до моделювання часових рядів, а також у формуванні підходу до інтеграції цих моделей у практичну систему прийняття рішень. Потреба в такому дослідженні зумовлена тим, традиційні підходи до прогнозування нерідко демонструють недостатній рівень надійності при нерегулярних флуктуаціях споживання, а методи машинного навчання, попри їхню високу гнучкість, потребують ретельної адаптації до специфіки промислових процесів, сезонності, наявності аварійних режимів, зміни графіків роботи та впливу зовнішніх подій, таких як температурні коливання або сигнали повітряних тривог.

Стан наукового розроблення проблеми у світовій практиці демонструє значну увагу до моделей ARIMA, SARIMA, нейронних мереж рекурентного типу (LSTM, GRU), ансамблевих методів та гібридних моделей. У численних дослідженнях підтверджено, що машинне навчання здатне суттєво підвищувати точність прогнозів, особливо за наявності великого масиву історичних даних та правильно сформованих ознак. Водночас українська практика все ще перебуває на етапі активного впровадження сучасних інструментів обробки часових рядів на підприємствах. Значна частина робіт зосереджена або на широкогогалузовому огляді моделей, або на прикладах енергосистемного прогнозування, тоді як питання прикладної інтеграції прогнозних моделей у діяльність конкретних виробничих майданчиків залишаються недостатньо розкритими.

Значущість проблеми для науки і практики зумовлена комплексом факторів. По-перше, правильне прогнозування споживання електроенергії є критичним елементом мінімізації небалансів та пов'язаних із ними штрафів у ринковій моделі «на добу наперед» та «внутрішньодобового» ринку. По-друге, надійні прогнози забезпечують можливість оптимізації закупівельної стратегії, зменшення питомих витрат і підвищення конкурентоспроможності підприємства. По-третє, якісний прогноз є фундаментом для побудови ширших систем енергетичного менеджменту, включно з визначенням аномалій споживання, оптимізацією технологічних режимів, розробленням систем раннього попередження про відхилення та формуванням енергетичного бюджету підприємства.

Підстави для проведення даного дослідження обумовлені як наявними науковими напрацюваннями, так і специфічними умовами роботи українських промислових підприємств у реальному секторі економіки. Фактичні режими споживання на виробництвах часто

характеризуються несподіваними змінами, зокрема спричиненими зупинками через аварійні або планові роботи, зміною інтенсивності виробничих процесів, переходом на резервні джерела живлення або використанням дизель-генераторних установок у періоди відключення мережі. У таких умовах традиційні методи прогнозування стикаються з проблемою коректного відображення нерівномірності та дискретності даних. Ця особливість вимагає розроблення підходів до очищення, маркування та структуризації масивів історичних даних, а також підбору моделей, здатних враховувати аномальні режими без втрати точності основного прогнозу. Дослідження, що поєднує очищення, фічеризацію, побудову моделей та економічне обґрунтування ефекту їх використання, є необхідним для формування повного циклу аналітичної підтримки енергетичних рішень підприємства.

Вихідними даними для виконання магістерської роботи є детальні історичні дані фактичного погодинного споживання електроенергії промисловим підприємством, дані про роботу обладнання, зокрема простої, аварії та нетипові режими, а також інформація про погодинні ціни ринку «на добу наперед» і внутрішньодобового ринку. Окреме значення має можливість збагачення масиву даних додатковими ознаками, такими як температурні показники, календарні та сезонні фактори, технологічні особливості роботи виробництва та записи про повітряні тривоги. Сукупність цих даних дає змогу побудувати змістовну економетричну та машинну модель, здатну відтворювати закономірності споживання та оцінювати потенційний економічний ефект від застосування прогнозу у процесі закупівлі електричної енергії.

Це дослідження зорієнтоване на всебічне розв'язання науково-прикладної проблеми: формування методології побудови та оцінювання моделей прогнозування споживання електроенергії на

промислового підприємстві з урахуванням реалій української енергетики та підкріплення результатів економічним аналізом ефективності їх використання.

РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА АКТУАЛЬНІСТЬ ЗАДАЧІ ПРОГНОЗУВАННЯ СПОЖИВАННЯ ЕЛЕКТРОЕНЕРГІЇ

1.1. Роль прогнозування енергоспоживання у сучасних енергетичних системах

Прогнозування споживання електроенергії є одним із ключових завдань сучасної енергетики, яке забезпечує стабільність роботи енергосистем, оптимізацію витрат, раціональне планування виробництва та споживання енергоресурсів. В умовах зростання вартості енергоресурсів, нестабільності ринку електроенергії та інтеграції відновлюваних джерел енергії точність прогнозу безпосередньо впливає на економічну ефективність підприємств і енергопостачальних компаній.

Для енергоринку точний прогноз є основою балансування виробництва та споживання електроенергії. Невідповідність між заявленими та фактичними обсягами споживання призводить до фінансових санкцій з боку операторів ринку та створює дисбаланс у системі передачі, що може викликати аварійні режими. З точки зору промислових підприємств, прогнозування споживання дозволяє оптимізувати закупівлю електроенергії на ринку «на добу наперед» (РДН), уникати пікових тарифів і знижувати навантаження на енергомережі в години максимуму.

В країнах Європейського Союзу та Північної Америки системи прогнозування є невід'ємною складовою Smart Grid інфраструктури. Наприклад, у Німеччині прогнози споживання використовуються для формування гнучких тарифних планів і регулювання навантаження побутових споживачів, а в Данії - для балансування енергосистеми з

великою часткою вітрової генерації. У США прогнози навантаження формуються на різних часових горизонтах - від кількох хвилин (operational forecast) до кількох днів і тижнів (market forecast) - із застосуванням як класичних статистичних, так і нейронних методів.

В Україні інтерес до прогнозування енергоспоживання суттєво зріс після впровадження конкурентного ринку електроенергії (2019 р.). Учасники ринку несуть відповідальність за небаланси, тому точне прогнозування стало критично важливим для споживачів і постачальників. Зокрема, промислові підприємства, які закупають електроенергію за прямими договорами або через трейдерів, можуть значно зменшити фінансові ризики, використовуючи власні алгоритми прогнозування, спираючись на архівні дані.

Отже, процес прогнозування енергоспоживання є не лише інструментом енергетичного планування, але й стратегічним елементом енергетичного менеджменту підприємства, що безпосередньо впливає на його конкурентоспроможність.

1.2. Особливості споживання електроенергії промисловими підприємствами

Промислові підприємства є одними з найбільших споживачів електроенергії в енергетичній системі країни. Їхні профілі навантаження відзначаються складною структурою, що залежить від технологічних процесів, режиму роботи виробництва, графіку змін, кліматичних умов і зовнішніх соціально-економічних факторів.

Зазвичай споживання промислового підприємства має чітко виражену періодичність - добову, тижневу та сезонну. У робочі дні спостерігаються два піки навантаження: ранковий (7:00-10:00), коли

розпочинається робота основних технологічних ліній, і денний/вечірній (14:00-18:00), що відповідає максимальній інтенсивності виробництва. У нічний час навантаження зменшується, за винятком безперервних технологічних процесів. У вихідні дні споживання зазвичай знижується на 30-60 % порівняно з буднями.

Вплив зовнішніх факторів

На характер споживання істотно впливають такі чинники:

- Температура навколишнього середовища. У теплий період року збільшується навантаження від систем охолодження, холодильного обладнання, кондиціонування та вентиляції. У холодний період, навпаки, зростає частка електричного опалення, систем антиобмерзання, підігріву повітря тощо.
- Зміна виробничої активності. Навантаження залежить від кількості працюючих змін, планових ремонтів і модернізацій. При переході на скорочений робочий тиждень чи зниженні обсягів замовлень споживання суттєво зменшується.
- Технічні фактори. Аварійні зупинки обладнання, несправності, роботи з обслуговування електроцитів або заміна обладнання спричиняють короточасні зниження навантаження або нерівномірність у профілі споживання.

Воєнні умови та енергетична нестабільність

Окрему специфіку енергоспоживанню промислових підприємств України надають воєнні дії 2022-2025 рр. та пов'язані з ними повітряні тривоги, аварійні відключення і дефіцит електричної потужності.

Під час повітряних тривог більшість підприємств зупиняють технологічні процеси з міркувань безпеки персоналу. Це призводить до раптових провалів навантаження, після яких відбувається різке

відновлення споживання при повторному запуску обладнання. Така нерівномірність утворює “ступінчасті” або “зубчасті” графіки споживання, що ускладнюють прогнозування.

Додатково, у періоди дефіциту генерації (особливо в осінньо-зимовий період) оператор системи передачі застосовує планові або аварійні відключення (відключення за графіком погодинних обмежень). Для промислових підприємств це означає вимушені простої та зміщення виробничих навантажень у часі - наприклад, перенесення випуску продукції на нічні години або у дні з гарантованим електропостачанням.

З технічної точки зору, такі події призводять до:

- появи аномалій у часових рядах споживання - різких спадів, піків, “дір” у даних;
- нестійкості сезонних компонентів, оскільки періодичність режимів порушується;
- зниження достовірності моделей прогнозування, побудованих на стандартних припущеннях стаціонарності.

Адаптація промислових споживачів

У відповідь на такі умови багато українських підприємств розробляють власні стратегії енергетичної гнучкості:

- встановлюють дизельні або газові генератори для підтримки безперервних процесів;
- створюють локальні енергосховища (акумулятори, UPS);
- інтегрують системи моніторингу (SCADA, EMS), що дозволяють змінювати профіль споживання у реальному часі;

- застосовують інструменти аналітики на базі штучного інтелекту для виявлення аномалій та прогнозування споживання з урахуванням нестандартних ситуацій.

Таким чином, енергоспоживання промислових підприємств України у воєнний період характеризується високою динамічністю, нерівномірністю та підвищеною невизначеністю. Це вимагає застосування адаптивних моделей прогнозування, здатних враховувати вплив нестандартних подій - повітряних тривог, аварійних відключень, зміни режимів роботи обладнання та зовнішніх умов.

1.3. Проблематика побудови точних прогнозів

Основними труднощами під час прогнозування енергоспоживання є нестабільність та неповнота вихідних даних. У промислових системах часто зустрічаються пропуски у часових рядах, спричинені збоями зв'язку, відсутністю живлення або некоректним зчитуванням з лічильників. Нерідко дані надходять з нерівномірним часовим кроком, що потребує попередньої нормалізації та інтерполяції.

Ще однією проблемою є зміна статистичних властивостей процесу з часом (несталість). Наприклад, після модернізації лінії чи зміни графіку роботи персоналу характер навантаження суттєво змінюється, а отже, попередня модель втрачає актуальність. Такі ситуації вимагають використання адаптивних методів прогнозування, здатних оновлювати параметри моделі під час надходження нових даних.

Класичні статистичні методи (AR, MA, ARIMA, SARIMA) добре працюють для стаціонарних часових рядів, однак їх ефективність знижується при наявності нелінійних залежностей або зовнішніх впливів, таких як температура чи аварійні зупинки. У таких випадках

доцільно застосовувати методи машинного навчання (Machine Learning), які здатні моделювати складні взаємозв'язки між змінними.

До найпоширеніших підходів належать:

- Регресійні моделі з ознаками (feature-based), які враховують температуру, день тижня, сезонність тощо;
- Нейронні мережі типу LSTM або GRU, що враховують часову залежність і пам'ять процесу;
- Гібридні моделі (ARIMA + ML), які поєднують інтерпретованість статистичних методів із точністю глибокого навчання.

Сучасна тенденція полягає в побудові самонавчаючих систем прогнозування, які автоматично адаптуються до змін структури даних. Це особливо актуально для підприємств, де умови виробництва змінюються динамічно.

1.4. Аналіз існуючих рішень і постановка задачі

Існує широкий спектр підходів до прогнозування енергоспоживання.

Класичні методи, такі як ARIMA, Holt-Winters, експоненційне згладжування, широко використовуються операторами систем передачі для середньострокового прогнозу. Їх перевагою є простота реалізації та інтерпретованість, але вони погано враховують складні взаємозв'язки між змінними.

Методи машинного навчання (Random Forest, Gradient Boosting, Support Vector Regression) забезпечують вищу точність, особливо за наявності великої кількості вхідних ознак. Водночас їх ефективність залежить від наявності великого масиву даних для навчання та уважного налаштування параметрів..

Глибинне навчання (Deep Learning) - зокрема LSTM, GRU, CNN - демонструє найкращі результати для довгих часових рядів із нелінійною динамікою. Такі моделі використовуються в енергосистемах Великої Британії, Південної Кореї та Китаю для короткострокового прогнозування (1-24 год).

В Україні подібні системи лише починають впроваджуватися, переважно в межах енергоменеджменту великих промислових підприємств. В умовах ринку електроенергії, воєнного часу та постійних змін у режимах роботи підприємств особливо важливо мати гнучку, адаптивну систему прогнозування, здатну враховувати локальні чинники - температуру, аварії, технологічні паузи, сигнали тривоги тощо.

Враховуючи зазначене, метою цієї дипломної роботи є побудова моделі прогнозування електроспоживання промислового підприємства, що базується на аналізі часових рядів і методах машинного навчання та забезпечує підвищену точність прогнозів у реальних умовах функціонування..

Для реалізації сформульованої мети слід розв'язати такі завдання:

1. Провести аналіз історичних даних споживання електроенергії підприємства;
2. Виконати попередню обробку та очищення даних;
3. Дослідити застосування статистичних і ML-моделей для прогнозування;
4. Порівняти точність різних підходів та обґрунтувати вибір оптимальної моделі;
5. Провести попередній економічний розрахунок ефективності впровадження прогновної моделі.

РОЗДІЛ 2. АНАЛІЗ ВИХІДНИХ ДАНИХ ТА ПОПЕРЕДНЯ ОБРОБКА

2.1. Опис об'єкта дослідження

Об'єктом дослідження є фабрика з виробництва картопляних чіпсів розташована у Київській області. Підприємство є одним із найбільших виробників снекової продукції в Україні та належить до категорії енергоємних безперервних виробництв харчової промисловості.

Загальна характеристика виробництва

Фабрика працює за принципом повного технологічного циклу: від приймання картоплі як сировини до випуску готової упакованої продукції. Основні етапи включають мийку, очищення, нарізку, обсмаження у термооліях, видалення залишкової олії, фасування та пакування.

Технологічне обладнання ліній працює у безперервному режимі, що забезпечує стабільне навантаження на енергетичну систему підприємства. При цьому значна частина електроспоживання припадає саме на технологічні вузли - обладнання ліній обсмаження, компресорні станції, системи вентиляції та транспортування.

Режим роботи та енергетичні особливості

Виробництво функціонує цілодобово, сім днів на тиждень. Єдиними періодами, коли споживання суттєво знижується, є санітарні зміни, під час яких обладнання вимикається для очищення та профілактичних робіт. Ці інтервали відображаються на графіку споживання як різкі падіння навантаження та використовуються для маркування неробочих періодів.

Календарні чинники (дні тижня, свята, час доби) мають незначний вплив на електроспоживання, оскільки фабрика не зупиняє

виробництво у вихідні або святкові дні.

Дещо нижче споживання спостерігається лише у нічні години, що зумовлено зменшенням навантаження адміністративних та офісних приміщень (освітлення, вентиляція).

Загальний профіль споживання характеризується високою стабільністю в робочі періоди та різкими коливаннями у моменти технологічних зупинок або зовнішніх впливів - аварійних відключень, повітряних тривог чи переходу на резервне живлення.

Для забезпечення безперервності процесів підприємство має резервні джерела живлення - дизель-генератори та когенераційну установку, які використовуються у випадках знеструмлення або оптимізації енергетичного балансу.

Система збору та формування даних

Облік і моніторинг споживання електроенергії здійснюються з використанням SCADA-системи Ignition, яка збирає дані з приладів обліку по окремих ділянках виробництва.

Проте централізованої системи автоматичного збору комерційних даних на момент дослідження не існує. Тому історичні дані формувалися шляхом агрегації кількох джерел, зокрема:

- таблиць історичних даних з бази SCADA-системи;
- звітів енергетичної служби;
- адміністративних журналів та облікових Excel-файлів.

Отримані показники охоплюють дворічний період спостереження та мають погодинну дискретність. У процесі підготовки вони були об'єднані в єдиний масив, синхронізовані за часовими мітками та приведені до UTC-формату з урахуванням переходів між літнім і зимовим часом.

На етапі попередньої перевірки виявлено пропуски, дублікати та аномальні значення (від'ємні, нульові або стрибкоподібні), що потребують додаткової обробки.

Додаткові параметри та контекстні фактори

Для підвищення точності моделювання споживання електроенергії використано низку додаткових регресорів, отриманих із зовнішніх та внутрішніх джерел:

- метеорологічні дані (температура, вологість, опади) - за API *OpenWeatherMap*;
- дані про повітряні тривоги - за API *alerts.in.ua*, що дозволяє враховувати вплив простоїв під час сигналів небезпеки;
- санітарні зміни - згідно з адміністративними журналами підприємства.

Ці змінні використовуються як додаткові ознаки у побудові прогнозних моделей і дозволяють враховувати вплив зовнішніх і внутрішніх факторів на електроспоживання.

2.2. Характеристика наявного датасету

Для побудови моделей прогнозування електроспоживання було сформовано узагальнений датасет на основі даних, отриманих із різних джерел, що охоплюють дворічний період спостереження (з листопада 2023 року по жовтень 2025 року).

Основою набору є часовий ряд із погодинною дискретністю, який відображає фактичне споживання електроенергії підприємством.

Структура та формат даних

Після попереднього об'єднання всі дані по споживанню зведено у єдиний датафрейм із двома основними стовпцями:

Таблиця 2.1

Назва колонки	Тип даних	Опис
timestamp	datetime (UTC)	Часова мітка спостереження (з кроком 1 година)
value	float	Обсяг спожитої електроенергії, кВт·год

Часові мітки були приведені до єдиного часового поясу (UTC) із урахуванням переходу на літній/зимовий час. Це дозволило уникнути дублювання або пропуску годин, характерних для періодів зміни часу, і забезпечити безперервність ряду.

Загальний обсяг вибірки становить близько 17135 рядків, що відповідає двом рокам погодинних вимірювань.

Усі дані впорядковані за часом, що дозволяє коректно використовувати їх у подальшому аналізі часових рядів та побудові моделей типу ARIMA, SARIMA чи LSTM.

Джерела формування даних

Датасет було створено шляхом агрегації кількох інформаційних джерел:

- таблиць історичних значень з бази даних SCADA-системи;
- звітів енергоменеджменту з фіксованими показниками лічильників;

- адміністративних журналів із записами періодів санітарних зупинок;
- допоміжних Excel-файлів, що містили добові або змінні значення споживання.

У процесі об'єднання виконано:

- уніфікацію форматів дат і одиниць вимірювання;
- приведення назв стовпців до стандартної структури;
- видалення дублікатів і записів із некоректними часовими мітками;
- узгодження значень при наявності різних інтервалів вимірювань.

Оскільки дані походять із різнорідних джерел, їх якість виявилася неоднорідною. Частина записів містила від'ємні або нульові значення, що не відповідають реальним умовам роботи підприємства, а також періоди з пропусками. Усі ці фактори стали підставою для подальшої попередньої обробки.

Частота вимірювань і часовий діапазон

Основна частота вимірювань становить 1 година, що є оптимальним балансом між деталізацією даних і стабільністю моделі. Такий крок дозволяє відстежувати добові коливання навантаження, санітарні зупинки та інші нетипові події.

Період спостереження охоплює повні два календарні роки, що забезпечує наявність сезонних закономірностей і дозволяє виконувати як короткострокове, так і середньострокове прогнозування.

Дані мають суцільний часовий індекс, проте в окремих ділянках спостерігаються відсутні або зсунути часові точки - здебільшого внаслідок об'єднання таблиць або втрати вимірювань під час аварій. Ці

пропуски будуть оброблені методами інтерполяції, forward/backward fill та ковзного середнього.

Якість і повнота даних

Попередній аналіз показав, що загальна частка пропусків у часовому ряді становить близько 2-3 %, а поодинокі аномальні значення трапляються в періоди переходу між таблицями або при відновленні даних після аварій.

Загалом рівень якості датасету можна оцінити як задовільний для побудови моделей прогнозування, за умови ретельної попередньої обробки та нормалізації даних.

2.3. Виявлення та обробка пропусків даних

Після формування узагальненого датасету було виявлено, що часовий ряд електроспоживання містить неповні або відсутні спостереження. Причинами цього стали розриви зв'язку між системами збору даних, помилки під час експорту з різних джерел, а також розбіжності форматів при злитті таблиць.

Оскільки побудова прогнозних моделей потребує повного та безперервного ряду, пропуски необхідно було коректно відновити, водночас забезпечуючи збереження справжньої динаміки процесу.

Виявлення пропусків

Для аналізу повноти даних проведено перевірку часової послідовності та виявлення відсутніх часових міток.

Перевірка здійснювалась за двома принципами:

- контроль безперервності: різниця між сусідніми часовими відмітками мала становити рівно 1 годину;

- контроль значень: пошук відсутніх або нечислових (NaN) даних у стовпці value.

Результати показали наявність короткочасних прогалин тривалістю від 1 до 6 годин, а також окремих інтервалів із розривом до 12 годин. Загальна частка пропусків становила близько 2-3 % від загальної кількості записів, що є прийнятним рівнем для реконструкції.

Принципи відновлення даних

Особливістю досліджуваного об'єкта є те, що виробництво працює цілодобово, а періоди зниження навантаження (наприклад, під час роботи від генераторів або в санітарні зміни) є частиною реального процесу.

Тому всі часові точки повинні бути присутні в датасеті, навіть якщо фактичне споживання дорівнює нулю.

Усі пропущені значення заповнювались таким чином, щоб забезпечити повну часову шкалу без розривів.

Методи заповнення пропусків

Для відновлення відсутніх спостережень застосовано комбінацію кількох методів, залежно від довжини розриву та структури навколишніх даних:

1. Лінійна інтерполяція (Linear Interpolation)

Використовувалась для коротких розривів (до 3 годин), коли зміни споживання між сусідніми моментами часу є плавними. Інтерполяція дозволяє відновити тренд без спотворення середнього рівня навантаження:

$$y_t = y_{t_1} + \frac{(y_{t_2} - y_{t_1})}{(t_2 - t_1)} (t - t_1)$$

2. Метод прямого заповнення (Forward Fill)

Застосовувався у випадках, коли наявні короткі послідовності пропусків (до 6 годин), і динаміка споживання не змінюється різко.

У таких випадках відсутнє значення заповнюється останнім відомим (ffill), що дозволяє зберегти безперервність часової послідовності.

3. Метод заповнення нульовими значеннями

Для періодів, коли споживання електроенергії з мережі фактично дорівнювало нулю (наприклад, під час переходу живлення на дизель-генератори або при повній зупинці виробництва), відсутні значення замінялися на 0.

Такий підхід дозволяє зберегти фізичний зміст даних - реальне споживання у ці години відсутнє, але часовий індекс залишається послідовним.

4. Згладження ковзним середнім (Rolling Mean)

Після відновлення ряду виконувалося згладження короточасних стрибків і шумів за допомогою ковзного середнього за 3-6 годин:

$$\bar{y}_t = \frac{1}{n} \sum_{i=0}^{n-1} y_{t-i}$$

Це допомагає усунути випадкові флуктуації, зберігаючи добову та сезонну динаміку.

Практична реалізація

Описані кроки були реалізовані у середовищі *Python* із використанням бібліотеки *pandas*.

Основні операції виглядали наступним чином:

```
# Сортування та індексація
df = df.sort_values('timestamp').set_index('timestamp')

# Заповнення пропусків інтерполяцією (до 3 годин)
df['value'] = df['value'].interpolate(method='time',
limit=3)

# Пряме заповнення (до 6 годин)
df['value'] = df['value'].fillna(method='ffill',
limit=6)

# Заповнення залишкових пропусків нулями
df['value'] = df['value'].fillna(0)

# Згладження результату
df['value_smooth'] = df['value'].rolling(window=3,
min_periods=1).mean()
```

Таким чином забезпечено повний часовий ряд без пропусків і без втрати часової структури.

Нульові значення, що відображають роботу від резервних джерел живлення або зупинки, інтегровані у загальний масив даних і розглядаються як валідні точки спостереження.

Результати обробки

Після відновлення даних усі часові мітки стали безперервними, а обсяг ряду - рівним кількості годин у досліджуваному періоді.

Дані зберігають:

- реальну динаміку навантаження;
- добову та сезонну періодичність;
- індикативність нульових періодів як частини реального процесу.

Це дозволяє використовувати датасет для подальшого статистичного аналізу, побудови моделей машинного навчання та порівняння результатів між різними часовими інтервалами без ризику викривлення.

Рис. 2.1



2.4. Виявлення аномалій у споживанні

Після усунення пропусків і формування безперервного часового ряду виконано етап виявлення аномальних значень - тобто таких спостережень, що не відповідають типовим закономірностям зміни електроспоживання.

Аномалії можуть суттєво впливати на якість моделей прогнозування, оскільки статистичні й машинні алгоритми мають високу чутливість до різких відхилень у даних.

Мета цього етапу - виявити, класифікувати та коректно обробити нетипові значення, при цьому не порушуючи їхнього фактичного змістового наповнення.

Типи аномалій у промисловому споживанні

Виробничі часові ряди відрізняються від побутових тим, що мають чітку технологічну структуру - повторювані цикли, стабільні режими, поступові зміни.

Для досліджуваного об'єкта (фабрики з безперервним процесом) характерні такі типи відхилень:

1. Різкі піки навантаження - короточасні стрибки споживання, спричинені пусками потужного обладнання або помилками у вимірюванні.
2. Раптові падіння до нуля - можуть бути наслідком роботи від резервних джерел (дизель-генераторів) або збоїв у системі збору даних.
3. Зміщення рівня навантаження - стійке підвищення або зниження споживання без технологічних причин (наприклад, після злиття таблиць чи заміни приладу).

4. Тривалі періоди аномальної стабільності - коли показник довго не змінюється, що свідчить про зависання або повторення даних у SCADA-системі.

Методи виявлення аномалій

Для виявлення таких відхилень застосовано комбінацію статистичних і машинних методів, що дозволяє одночасно враховувати як локальні коливання, так і глобальні тренди.

1. Статистичний метод “трьох сигм” (Z-score)

Цей підхід базується на припущенні, що основна частина значень споживання розподілені навколо середнього в межах кількох стандартних відхилень:

$$Z_t = \frac{y_t - \mu}{\sigma}$$

де

μ — середнє значення споживання,

σ — стандартне відхилення,

Z_t — нормалізоване відхилення.

Значення з $|Z_t| > 3$ вважаються аномальними.

Метод простий у реалізації та добре виявляє разові піки або падіння, але менш ефективний при сезонних коливаннях, тому доповнюється іншими критеріями.

2. Метод інтерквартильного розмаху (IQR)

Для більш стійкої оцінки використано інтерквартильний діапазон:

$$IQR = Q3 - Q1$$

Спостереження вважається аномальним, якщо:

$$y_t < Q1 - 1.5 \cdot IQR \text{ або } y_t > Q3 + 1.5 \cdot IQR$$

де:

$Q1$ - перший квартиль (25-й перцентиль), нижня межа типової варіації даних;

$Q3$ - третій квартиль (75-й перцентиль), верхня межа типової варіації даних;

IQR - інтерквартильний розмах, діапазон між $Q1$ та $Q3$;

y_t - фактичне значення показника у момент часу t ;

1.5 - емпіричний коефіцієнт, що визначає чутливість методу: множення IQR на 1.5 формує "довжину вусів", за межами яких значення вважаються викидами;

$Q1 - 1.5 \cdot IQR$ - нижня межа допустимого діапазону;

$Q3 + 1.5 \cdot IQR$ - верхня межа допустимого діапазону.

Цей метод дозволяє відфільтрувати крайні значення без урахування розподілу даних і добре підходить для виявлення нетипових стрибків у стабільних режимах.

3. Ковзне відхилення (Rolling Z-score)

Для врахування добових циклів застосовано ковзне вікно середнього та стандартного відхилення (наприклад, 24 години):

$$Z_t^{(\text{roll})} = \frac{y_t - \bar{y}_{t,24}}{s_{t,24}}$$

де:

- y_t - фактичне значення показника у момент часу t ;
- $\bar{y}_{t,24}$ - ковзне середнє за останні 24 спостереження (24-годинне вікно);

- $s_{t,24}$ - ковзне стандартне відхилення за ті самі 24 спостереження;
- $Z_t^{(roll)}$ - нормалізоване відхилення у момент t , розраховане відносно локальних статистик.

4. Метод ізоляційного лісу (Isolation Forest)

Для більш глибокої перевірки використовувався алгоритм *Isolation Forest*, який виявляє нетипові спостереження на основі деревових розділень.

Цей метод не потребує апріорного розподілу даних і добре ідентифікує складні відхилення - наприклад, тривалі періоди нетипового споживання після аварій чи перезапусків обладнання.

Практична реалізація

Для реалізації виявлення аномалій у середовищі *Python* застосовано комбінацію методів Z-score та IQR.

Фрагмент коду:

```
import numpy as np
import pandas as pd

# Розрахунок статистик
mean = df['value'].mean()
std = df['value'].std()

# Метод 3σ
df['z_score'] = (df['value'] - mean) / std
df['is_anomaly_z'] = abs(df['z_score']) > 3

# Метод IQR
Q1 = df['value'].quantile(0.25)
Q3 = df['value'].quantile(0.75)
IQR = Q3 - Q1
df['is_anomaly_iqr'] = (df['value'] < Q1 - 1.5*IQR) |
(df['value'] > Q3 + 1.5*IQR)
```

```
# Об'єднання результатів
df['is_anomaly'] = df[['is_anomaly_z',
'is_anomaly_iqr']].any(axis=1)
```

Після виконання аналізу виявлені точки були перевірені візуально на графіку часового ряду.

Поодинокі сплески, що не мали технологічного обґрунтування, були згладжені ковзним середнім або замінені інтерполяційним значенням.

Рис. 2.2



2.5. Формування набору ознак (feature engineering)

Після очищення даних та забезпечення безперервності часового ряду наступним етапом підготовки є формування набору додаткових ознак (features), які можуть впливати на електроспоживання підприємства.

Feature engineering є ключовою частиною побудови моделей, оскільки саме правильний вибір ознак визначає здатність моделі відтворювати та прогнозувати реальні закономірності.

Оскільки досліджуваний об'єкт є безперервним виробництвом із вираженою технологічною специфікою, набір ознак включає календарні, погодні, технологічні та зовнішні фактори, синхронізовані з основним часовим рядом.

1. Календарні ознаки

Календарні ознаки дозволяють врахувати внутрішні цикли та регулярні повторювані зміни в споживанні:

1.1. Година доби (hour)

Електроспоживання має певні добові коливання, пов'язані з роботою адміністративних приміщень (вентиляція, освітлення) та пусковими процесами.

Годинна ознака дозволяє моделі розуміти внутрішньодобову структуру.

1.2. День тижня (weekday)

Хоча виробництво працює безперервно, невеликі відмінності можливі між буднями та вихідними (зниження допоміжних навантажень в офісних зонах).

Ознака подається у вигляді цілого числа від 0 до 6.

1.3. Чи вихідний (is_weekend)

Двійкова ознака, що може бути корисною в комбінації з іншими факторами.

1.4. Місяць року (month)

Оскільки виробництво працює стабільно цілий рік, сезонні ефекти є слабкими, але вони можуть проявлятися через температуру або зміну вентиляційних режимів.

2. Погодні ознаки

Погодні умови впливають на роботу систем вентиляції, тепловиділення та окремих технічних процесів.

Для фабрики чипсів найбільш актуальними є:

2.1 Температура повітря (temp)

Впливає на інтенсивність вентиляції та теплові баланси виробничих приміщень.

2.2. Лагові ознаки температури (temp_lag_1, temp_lag_24)

Температура має інерційний вплив, тому додаються лаги на 1 годину та 24 години.

Дані отримані відповідно до геолокації через API *OpenWeatherMap*.

3. Технологічні ознаки

Саме технологічні фактори є ключовими для коректного прогнозування, оскільки фабрика працює в безперервному циклі:

3.1. Санітарні зміни (is_sanitary)

Періоди планових зупинок ліній, отримані з адміністративних журналів. Ці значення є прямими маркерами неробочих інтервалів, які модель повинна враховувати.

4. Зовнішні події

4.1. Повітряні тривоги (is_alert)

Під час сигналів цивільної оборони обладнання може переводитись у знижений режим, що призводить до аномальних просідань споживання.

Дані синхронізовано з API *alerts.in.ua* для відповідного району.

5. Лагові ознаки споживання

Лаги дозволяють моделі враховувати інерційність процесу:

- kWh_lag_1 - попередня година;
- kWh_lag_24 - та сама година попередньої доби;
- kWh_lag_168 - та сама година попереднього тижня.

Це ключові ознаки для ARIMA/SARIMA та LSTM

6. Ознаки ковзних вікон

Для захоплення локальних трендів використовуються:

- ковзне середнє за 3-6 годин (rolling_mean_6);
- ковзна медіана (rolling_median_6);
- ковзне стандартне відхилення (rolling_std_24).

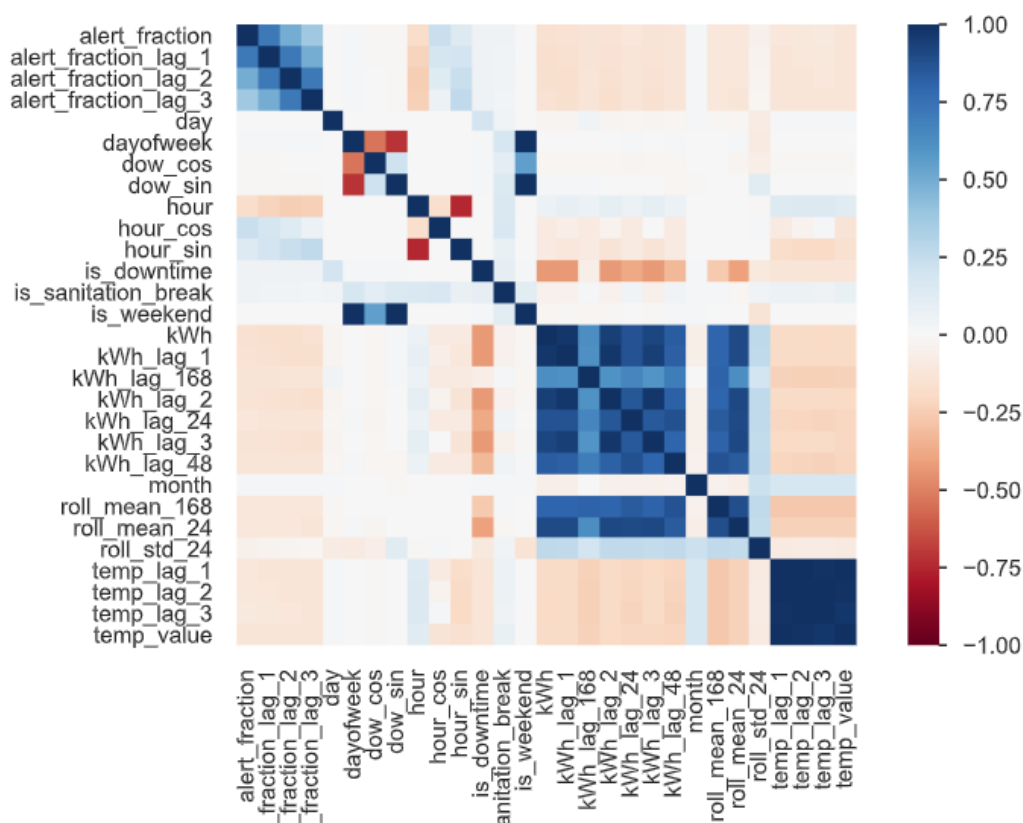
Такі ознаки допомагають моделі розуміти локальну структуру навантаження.

7. Формування структури датафрейму

Після додавання всіх ознак структура датасету має такий вигляд:

timestamp (UTC), 'kWh', 'is_downtime', 'is_sanitation_break', 'temp_value',
'alert_fraction', 'hour', 'dayofweek', 'is_weekend', 'day', 'month', 'hour_sin',
'hour_cos', 'dow_sin', 'dow_cos', 'kWh_lag_1', 'kWh_lag_2', 'kWh_lag_3',
'kWh_lag_24', 'kWh_lag_48', 'kWh_lag_168', 'roll_mean_24', 'roll_std_24',
'roll_mean_168', 'temp_lag_1', 'temp_lag_2', 'temp_lag_3',
'alert_fraction_lag_1', 'alert_fraction_lag_2', 'alert_fraction_lag_3'

Усі ознаки синхронізовані за часовим індексом і готові для використання в моделюванні.



2.6. Нормалізація та розподіл даних на тренувальну та тестову вибірки

Після формування розширеного набору ознак (підрозділ 2.5) на основі погодинного ряду споживання електроенергії було отримано фінальний датасет, у якому кожному часовому відліку відповідає вектор характеристик. До нього входять як значення цільової змінної value (фактичне споживання електроенергії за годину), так і низка похідних ознак: календарні та циклічні індикатори, лагові значення ряду, ковзні середні, температурні показники, ознаки повітряних тривоги, а також технологічні індикатори (is_downtime, is_sanitation_break).

На цьому етапі основним завданням є підготовка даних до навчання моделей прогнозування. Зокрема, необхідно:

1. усунути технічні наслідки формування ознак (пропуски, що виникли після зсувів та ковзних вікон),
2. виконати масштабування (нормалізацію) числових змінних,
3. коректно розділити часовий ряд на тренувальну та тестову частини у спосіб, що унеможлиблює потрапляння до моделі інформації з майбутніх спостережень.

2.6.1. Очищення датасету після формування ознак

Як було показано у попередньому підрозділі, для врахування інерційності та сезонності процесу споживання були сформовані лагові ознаки (`value_lag_1`, `value_lag_2`, ..., `value_lag_168`), ковзні середні та стандартні відхилення (`roll_mean_24`, `roll_mean_168`, `roll_std_24`), а також лаги екзогенних факторів (`temp_lag_1...3`, `alert_fraction_lag_1...3`).

Такі операції неминуче призводять до появи пропусків (NaN) у перших рядках часового ряду:

- наприклад, для формування `value_lag_168` потрібні дані за попередні 168 годин, тому перші 168 відліків не містять коректних значень цієї ознаки;
- при розрахунку ковзного середнього за тиждень (168 годин) перші вікна також мають неповний набір спостережень;
- аналогічна ситуація спостерігається для лагів температури та індикаторів тривоги.

З точки зору подальшого моделювання, використання рядків із неповними векторами ознак є некоректним: більшість алгоритмів машинного навчання та нейронних мереж не допускають наявності

пропусків у вхідних даних. Тому отриманий датасет було додатково очищено: усі рядки, у яких хоча б одна з ознак мала значення NaN, видалялися.

У результаті було сформовано компактний, але повністю заповнений датасет `df_model`, що містить:

- єдиний погодинний часовий індекс без пропусків,
- цільову змінну `value`,
- повний набір сформованих ознак,
- відсутність пропусків у будь-якій колонці.

Саме цей датасет і використовується як вихідна точка для нормалізації та розподілу на `train/test`.

2.6.2. Особливості нормалізації ознак для різних типів моделей

Наступним кроком є нормалізація (масштабування) числових ознак. Потреба у цьому кроці зумовлена тим, що різні характеристики мають різні діапазони зміни. Наприклад, споживання електроенергії може вимірюватися десятками чи сотнями кВт·год, температура повітря - у діапазоні, скажімо, від -20 до +35 °C, значення `hour_sin` та `hour_cos` лежать у межах `[-1; 1]`, а бінарні індикатори (`is_downtime`, `is_sanitation_break`, `is_weekend`) приймають лише 0 або 1.

Якщо подати такий неоднорідний набір ознак без попередньої нормалізації до моделей, чутливих до масштабу (нейронні мережі, та метод опорних векторів), ознаки з великим числовим діапазоном будуть домінувати над іншими, що може призвести до некоректного навчання.

У цій роботі передбачається використання кількох класів моделей:

- класичні часові моделі: AR, MA, ARIMA, SARIMA;
- нейронні мережі: LSTM, GRU, CNN;
- моделі машинного навчання: Random Forest, Gradient Boosting, Support Vector Regression.

Для цих груп підхід до нормалізації дещо відрізняється:

1. AR/MA/ARIMA/SARIMA.

Класичні моделі аналізу часових рядів, як правило, працюють із самим рядом або його стаціонаризованими перетвореннями (різницями). У базовому варіанті вони не вимагають масштабування, однак при використанні екзогенних змінних (наприклад, температури, індикаторів тривоги, лагових ознак) доцільно приводити різні характеристики до співставних масштабів. Для цього може бути застосовано стандартне масштабування (StandardScaler), яке перетворює кожну ознаку до нульового середнього та одиничного стандартного відхилення.

Стандартна нормалізація (StandardScaler)

Цей метод переводить ознаки у стандартний вигляд:

$$x' = \frac{x - \mu}{\sigma}$$

де

μ - середнє значення ознаки у тренувальній вибірці,

σ - стандартне відхилення.

2. Нейронні мережі (LSTM, GRU, CNN).

Для рекурентних та згорткових нейронних мереж масштабування

є обов'язковим кроком, оскільки ці моделі оптимізуються методом градієнтного спуску. Занадто широкий діапазон вхідних значень може спричиняти нестабільність навчання, повільну збіжність або проблеми з градієнтами. Тому для набору ознак X та, за потреби, для таргету у використовується або стандартна нормалізація (StandardScaler), або масштабування до інтервалу $[0; 1]$ (MinMaxScaler), що є типовим підходом у задачах прогнозування часових рядів за допомогою LSTM/GRU.

Масштабування до відрізка $[0; 1]$ (MinMaxScaler)

Цей підхід особливо корисний для LSTM/GRU/CNN, де стабільність градієнтів є критичною:

$$x' = \frac{x - x_{\min}}{x_{\max} - x_{\min}}$$

3. Алгоритми машинного навчання (Random Forest, Gradient Boosting, SVR).

Для деревових методів (Random Forest, Gradient Boosting) масштабування ознак не є критичним, оскільки рішення будуються за рахунок порогових розбиттів і не залежать від абсолютних значень. Натомість для методу опорних векторів (Support Vector Regression) нормалізація є бажаною, а часто й обов'язковою, оскільки від масштабу ознак залежить геометрія простору ознак та вибір оптимальних параметрів ядра.

Усі перетворення масштабування виконуються тільки на тренувальній вибірці. Спочатку обчислюються параметри нормалізації (середнє, стандартне відхилення або мінімум та максимум) за тренувальними даними, а потім ті самі параметри застосовуються до тестової вибірки. Це дозволяє уникнути витоку інформації з майбутнього періоду в процес навчання моделі.

2.6.3. Часове розділення даних на тренувальну та тестову вибірки

На відміну від задач, де спостереження вважаються незалежними, у часових рядах дані мають природний порядок, і майбутні значення залежать від минулих. Тому випадкове перемішування (shuffle) є неприйнятним - воно призвело б до того, що модель «бачила» б інформацію з майбутнього під час навчання, що спотворює оцінку її якості та завищує результати (ефект data leakage).

У межах цього дослідження розділення датасету `df_model` на тренувальну (`train`) та тестову (`test`) частини здійснюється послідовно у часі.

$$\text{Train} = \{t_1, t_2, \dots, t_{0.8N}\}, \text{Test} = \{t_{0.8N+1}, \dots, t_N\}$$

де N - загальна кількість спостережень.

У результаті формуються множини:

- $X_{\text{train}}, Y_{\text{train}}$;
- $X_{\text{test}}, Y_{\text{test}}$.

Усі спостереження впорядковуються за часовим індексом, після чого перші приблизно 80% часових відліків використовуються для навчання моделей, а останні 20% - для перевірки їх прогнозних можливостей.

Таке співвідношення є компромісом між двома вимогами:

- надати моделям достатньо даних для навчання (особливо це важливо для глибинних нейронних мереж та моделей з великою кількістю параметрів);
- зберегти репрезентативний тестовий інтервал, на якому можна оцінити якість прогнозу в умовах реального майбутнього періоду.

У результаті формуються чотири основні множини:

- X_{train} - матриця ознак для тренувального періоду;
- y_{train} - відповідні значення споживання електроенергії;
- X_{test} - матриця ознак для тестового (майбутнього) періоду;
- y_{test} - фактичні значення споживання, за якими проводитиметься оцінка точності моделей.

Усі подальші моделі - як ARIMA/SARIMA, так і LSTM/GRU/CNN та алгоритми машинного навчання - навчатимуться виключно на тренувальній частині ряду й перевірятимуться на тестовій, що дозволить коректно порівняти їх здатність прогнозувати електроспоживання в реальних умовах.

РОЗДІЛ 3. ВИБІР АРХІТЕКТУР МОДЕЛЕЙ

З огляду на результати розділу 3, вихідні дані для моделювання являють собою погодинний часовий ряд споживання електроенергії value на промисловій фабриці з розвинутим набором ознак: календарні (година доби, день тижня, вихідні/будні, синусо-косинусні перетворення), метеорологічні (температура та її лаги), індикатори повітряних тривог, технологічні стани (is_downtime, is_sanitation_break), лаги та ковзні агрегати самого ряду споживання.

Такі дані поєднують у собі кілька важливих властивостей:

- виражену добову та тижневу сезонність (робочі зміни, санітарні вікна, вихідні);
- інерційність (поточне споживання суттєво залежить від попередніх годин і днів);
- наявність зовнішніх факторів (температура, повітряні тривоги, технологічні зупинки);
- нелінійні ефекти (нестандартні режими, аварії, переходи на генератори).

У зв'язку з цим у даній роботі обрано три групи моделей прогнозування:

1. Класичні моделі часових рядів (AR, MA, ARIMA, SARIMA) - як базовий лінійний підхід, що добре описує лагові залежності та регулярну сезонність.
2. Алгоритми машинного навчання (Random Forest, Gradient Boosting, Support Vector Regression) - як проміжний рівень складності, що працює з інженерними ознаками та вміє моделювати нелінійність.

3. Нейронні мережі глибокого навчання (LSTM, GRU, CNN) - як підхід, здатний відображати складні нелінійні та довготривалі залежності у часових послідовностях.

3.1. Класичні моделі часових рядів: AR, MA, ARIMA, SARIMA, SARIMAX

Традиційні лінійні моделі часових рядів стали одним із перших засобів, що застосовувалися для аналізу та прогнозування показників енергоспоживання. Вони дозволяють у формалізований спосіб описати залежність поточного значення ряду від попередніх значень (авторегресія) та випадкових збурень (ковзне середнє), а також врахувати тренд і сезонність.

Авторегресійна модель AR(p) описує поточне значення ряду як лінійну комбінацію його попередніх значень:

$$y_t = \phi_1 y_{t-1} + \phi_2 y_{t-2} + \dots + \phi_p y_{t-p} + \varepsilon_t,$$

де y_t — значення ряду у момент часу t , ϕ_i — авторегресійні коефіцієнти, ε_t — білий шум. У контексті фабрики це означає, що споживання в поточну годину визначається комбінацією споживання у кількох попередніх годинах чи добах.

Модель ковзного середнього MA(q), навпаки, фокусується на моделюванні впливу випадкових шоків:

$$y_t = \mu + \varepsilon_t + \theta_1 \varepsilon_{t-1} + \theta_2 \varepsilon_{t-2} + \dots + \theta_q \varepsilon_{t-q},$$

де θ_j описують, як випадкові збурення минулого впливають на поточне значення.

Комбінована модель ARMA(p, q) поєднує обидва підходи:

$$y_t = \varphi_1 y_{t-1} + \dots + \varphi_p y_{t-p} + \varepsilon_t + \theta_1 \varepsilon_{t-1} + \dots + \theta_q \varepsilon_{t-q},$$

де:

y_t - значення ряду у момент часу t ,

$\varphi_1, \dots, \varphi_p$ - коефіцієнти авторегресійної частини (AR);

$\theta_1, \dots, \theta_q$ - коефіцієнти частини ковзного середнього (MA);

ε_t - випадкова похибка (білий шум);

p - порядок авторегресії;

q - порядок ковзного середнього.

однак вимагає стаціонарності ряду. Для промислового енергоспоживання, яке часто має тренд, структурні зрушення та сезонні компоненти, це обмеження є надто жорстким.

Тому найчастіше застосовується ARIMA(p, d, q), де d - порядок різницювання, що усуває тренд і перетворює ряд на стаціонарний:

$$\varphi(B) (1 - B)^d y_t = \theta(B) \varepsilon_t,$$

де:

y_t - значення часового ряду у момент часу t ,

ε_t - випадкова похибка (білий шум);

B - оператор зсуву (backshift), для якого $B y_t = y_{t-1}$;

$\varphi(B)$ - поліном авторегресії порядку p ;

$\theta(B)$ - поліном ковзного середнього порядку q ;

$(1 - B)^d$ - оператор диференціювання порядку d , який відповідає за перетворення ряду у стаціонарний;

d - порядок диференціювання (кількість разів, коли обчислюється різниця між сусідніми значеннями).

У випадку енергоспоживання зазвичай чітко виражена добова та тижнева сезонність. Для її врахування базова ARIMA розширюється до сезонної моделі SARIMA(p, d, q)(P, D, Q) $_s$:

$$\varphi(B) \Phi(B^s) (1 - B)^d (1 - B^s)^D y_t = \theta(B) \Theta(B^s) \varepsilon_t$$

де s - сезонний період (наприклад, $s = 24$ для добової сезонності), $\Phi(B^s)$, $\Theta(B^s)$ - сезонні авторегресійні та МА-компоненти, D - порядок сезонного різницювання.

Для відображення впливу температури, повітряних тривог та технологічних станів у роботі застосовуються моделі типу SARIMAX, в яких додається вектор екзогенних змінних X_t :

$$\varphi(B) \Phi(B^s) (1 - B)^d (1 - B^s)^D y_t = \theta(B) \Theta(B^s) \varepsilon_t + \beta^T X_t$$

У цьому векторі X_t можуть міститися такі ознаки, як temp_value, alert_fraction, is_downtime, is_sanitation_break, а також календарні індикатори.

Переваги класичних моделей для даного завдання:

- чітка інтерпретація параметрів (можна пояснити, наскільки сильно поточне споживання залежить від споживання 1, 24 чи 168 годин тому);
- відносно невеликі вимоги до обсягу даних;
- добрі результати для регулярних, лінійних сезонних структур.

Обмеження:

- припущення лінійності та стаціонарності (після різницювання);

- обмежена здатність описувати різкі піки, нелінійні ефекти, зміни режимів роботи;
- складність масштабування до високорозмірних екзогенних ознак

Саме тому ARIMA/SARIMA у цій роботі розглядаються як базові (reference) моделі, з якими надалі порівнюватимуться більш складні алгоритми.

3.2. Алгоритми машинного навчання: Random Forest, Gradient Boosting, Support Vector Regression

Друга група моделей розглядає задачу прогнозування електроспоживання як регресійну задачу над вектором ознак, сформованих. Тут структура часової залежності не описується явно через оператори зсуву (як у ARIMA), а закладена в самих фіча-інженерних ознаках: лаги `value_lag_k`, ковзні середні, циклічні календарні компоненти, індикатори тривоги, температури тощо.

У загальному вигляді такі моделі шукають деяку нелінійну функцію:

$$\hat{y}_t = f(X_t; \theta)$$

де X_t - вектор ознак для моменту t , θ - параметри моделі, \hat{y}_t - прогноз споживання.

Random Forest формує ансамбль із M дерев рішень, причому кожне дерево навчається на випадково сформованій підмножині ознак і вибірок. Прогноз є усередненням по всіх деревах:

$$\hat{y}_t = \frac{1}{M} \sum_{m=1}^M T_m(X_t)$$

де T_m - m -те дерево. Такий підхід добре вловлює нелінійні взаємодії між ознаками (наприклад, спільний вплив високої температури та вихідного дня на споживання) і є стійким до шуму.

Gradient Boosting (зокрема його реалізації, такі як XGBoost чи LightGBM) формує підсумкову модель у вигляді послідовного додавання слабких моделей - найчастіше дерев рішень:

$$\widehat{y}_t^{(K)} = \sum_{k=1}^K \gamma_k h_k(X_t)$$

де h_k - k -та базова модель, γ_k - вагові коефіцієнти, K - кількість ітерацій. Кожне наступне дерево намагається виправити похибки попередніх, що дозволяє отримати високу точність на складних нелінійних залежностях.

Support Vector Regression (SVR) підходить до задачі з іншого боку - він шукає гіперплощину у просторі ознак, яка мінімізує помилки в межах деякої ε -“трубки”, використовуючи ядрові перетворення для відображення нелінійності. У загальному вигляді:

$$\widehat{y}_t = \sum_{i=1}^N (\alpha_i - \alpha_i^*) K(X_i, X_t) + b$$

де $K(\cdot, \cdot)$ - ядро, α_i, α_i^* - коефіцієнти, які визначають вклад тренувальних прикладів.

У контексті прогнозування електроспоживання алгоритми машинного навчання вирізняються такими перевагами:

- уміння відтворювати складні нелінійні залежності між обсягами споживання, температурою, тривогами, часом доби тощо;

- можливість використовувати великий набір ознак, сформованих у 2.5 (лагові, ковзні, календарні, технологічні);
- відносно проста реалізація за допомогою бібліотек scikit-learn;
- наявність інтерпретованих метрик важливості ознак (особливо для Random Forest і Gradient Boosting), що дозволяє відповісти на практичне запитання: які фактори найбільше впливають на споживання?

Обмеження:

- відсутність явної моделі часової структури - послідовність моментів t сама по собі не моделюється, а лише враховується через фічі;
- потреба в обережному розділенні train/test у часовому порядку;
- обмежена здатність до екстраполяції за межі діапазону, в якому модель навчалася.

Ці моделі у роботі розглядаються як проміжний рівень між простими лінійними (ARIMA/SARIMA) та складними нейронними мережами.

3.3. Нейронні мережі для часових рядів: LSTM, GRU, CNN

Третій клас моделей - нейронні мережі глибокого навчання, які спеціально адаптовані для моделювання послідовностей. На відміну від ARIMA або деревових моделей, вони можуть безпосередньо працювати з часовими вікнами даних, виявляти приховані довготривалі залежності та складні нелінійні патерни, характерні для реального енергоспоживання.

LSTM (Long Short-Term Memory)

LSTM-мережі є різновидом рекурентних нейронних мереж (RNN), які вирішують проблему “забування” важливої інформації при збільшенні довжини послідовності. Кожна LSTM-комірка має внутрішній стан c_t та вихідний стан h_t і керується трьома “воротами”:

$$\begin{aligned}
 f_t &= \sigma(W_f[h_{t-1}, x_t] + b_f), \\
 i_t &= \sigma(W_i[h_{t-1}, x_t] + b_i), \\
 \tilde{c}_t &= \tanh(W_c[h_{t-1}, x_t] + b_c), \\
 c_t &= f_t \odot c_{t-1} + i_t \odot \tilde{c}_t, \\
 o_t &= \sigma(W_o[h_{t-1}, x_t] + b_o), \\
 h_t &= o_t \odot \tanh(c_t)
 \end{aligned}$$

де x_t - вектор ознак у момент часу t (наприклад, споживання і супровідні фічі за годину), $\sigma(\cdot)$ - сигмоїда, \odot - поелементне множення.

У контексті задачі прогнозування енергоспоживання LSTM дає змогу:

- враховувати довготривалі залежності (вплив споживання за останні дні/тижні);
- поєднувати інформацію про послідовні значення value і екзогенні ознаки (температура, тривоги, зміни);
- моделювати нелінійну динаміку, яку важко описати класичними лінійними моделями.

GRU (Gated Recurrent Unit)

GRU є спрощеним варіантом LSTM із меншою кількістю параметрів, що може бути корисно при обмеженому обсязі даних. Основні рівняння:

$$z_t = \sigma(W_z[h_{t-1}, x_t] + b_z),$$

$$r_t = \sigma(W_r[h_{t-1}, x_t] + b_r),$$

$$\tilde{h}_t = \tanh(W_h[r_t \odot h_{t-1}, x_t] + b_h),$$

$$h_t = (1 - z_t) \odot h_{t-1} + z_t \odot \tilde{h}_t.$$

GRU зберігають здатність моделювати довготривалі залежності, але, як правило, навчаються швидше і є менш схильними до перенавчання на невеликих вибірках.

1D CNN (згорткові мережі для часових рядів)

Згорткові нейронні мережі (CNN), що первинно створювалися для задач обробки зображень, можуть бути ефективно адаптовані й до аналізу часових рядів, якщо застосовувати одномірні згортки уздовж часової осі. Згортка може бути записана як:

$$(y * w)_t = \sum_{k=0}^{K-1} w_k x_{t-k}$$

де w_k - ваги згорткового фільтра, K - розмір ядра. У такий спосіб CNN виявляє локальні часові патерни: типові профілі навантаження протягом зміни, характерні "сплески" при запуску обладнання, шаблони споживання під час тривоги тощо.

Переваги нейронних мереж для даного завдання:

- здатність моделювати складну нелінійну динаміку, що поєднує виробничі режими, зовнішні фактори і сезонність;
- можливість одночасно працювати з усім багатовимірним набором ознак (включаючи лаги, календарні, погодні та технологічні показники);
- потенційно вища точність у порівнянні з лінійними моделями, особливо при складних режимах роботи фабрики.

Обмеження:

- потреба у більшому обсязі даних і ретельній гіперпараметричній оптимізації;
- менша інтерпретованість у порівнянні Random Forest;
- більші обчислювальні витрати.

У рамках цієї роботи LSTM, GRU та CNN розглядаються як просунуті моделі, які дозволяють перевірити, наскільки глибоке навчання здатне покращити прогнозування у порівнянні з класичними підходами.

3.4. Узагальнення вибору моделей для дослідження

У межах даної роботи формуються три групи моделей, які репрезентують різні підходи до аналізу часових рядів та враховують ключові властивості електроспоживання виробничого підприємства:

1. **Класичні лінійні моделі** (AR, MA, ARIMA, SARIMA, SARIMAX), які дозволяють відобразити лагові залежності та регулярну сезонність.
2. **Алгоритми машинного навчання** (Random Forest, Gradient Boosting, Support Vector Regression), які працюють із багатовимірним набором ознак та можуть моделювати нелінійні взаємодії факторів.
3. **Моделі глибинного навчання** (LSTM, GRU, CNN), що дозволяють аналізувати довготривалі залежності та складні тимчасові патерни без жорстких припущень щодо структури ряду.

РОЗДІЛ 4. РОЗРОБЛЕННЯ ТА НАВЧАННЯ МОДЕЛЕЙ ПРОГНОЗУВАННЯ

4.1. Авторегресійна модель AR(p)

Авторегресійна модель AR(p) є найпростішим представником класичних лінійних моделей часових рядів і використовується як базовий підхід для опису інерційної поведінки споживання електроенергії. Ідея полягає в тому, що поточне значення ряду залежить від кількох його попередніх значень, тобто від власної історії системи.

У загальному випадку модель AR(p) записується у вигляді:

$$y_t = \varphi_1 y_{t-1} + \varphi_2 y_{t-2} + \dots + \varphi_p y_{t-p} + \varepsilon_t,$$

де

y_t - значення споживання електроенергії в момент часу t ,

φ_0 - константа (середній рівень ряду),

φ_i - авторегресійні коефіцієнти, що описують вплив i -го лагу,

ε_t - випадкова похибка (білий шум) із нульовим середнім та сталою дисперсією.

У термінах оператора зсуву B (де $B y_t = y_{t-1}$) модель AR(p) можна переписати як:

$$\varphi(B) y_t = \varepsilon_t,$$

$$\varphi(B) = 1 - \varphi_1 B - \varphi_2 B^2 - \dots - \varphi_p B^p.$$

Для коректності моделі корені характеристичного полінома $\phi(B)$ мають лежати за межами одиничного кола, що відповідає вимозі стаціонарності.

У контексті задачі прогнозування погодинного споживання електроенергії модель AR(p) формально описує, наскільки сильно поточне споживання y_t залежить від значень у попередні години $y_{t-1}, y_{t-2}, \dots, y_{t-p}$. Наприклад, коефіцієнт ϕ_{24} може інтерпретуватися як "схожість" споживання в поточну годину з тим самим часовим інтервалом попередньої доби.

Вибір порядку p

Параметр p визначає, скільки попередніх значень ряду враховує модель. Занадто мале значення p може призвести до недоописання структури ряду (модель не встигає врахувати добову/тижневу інерцію), а занадто велике - до перенавчання, коли модель починає "запам'ятовувати" шум.

У роботі порядок p підбирається на основі:

- аналізу часткової автокореляційної функції (PACF);
- інформаційних критеріїв (AIC, BIC) для кількох кандидатів p ;
- фізичного змісту процесу (добова та тижнева повторюваність: 1, 2, 3, 24, 48, 168 годин).

Підготовка даних

Для побудови AR(p) використовується лише один часовий ряд споживання, без екзогенних ознак:

- y_{train} - тренувальний інтервал ряду value,
- y_{test} - тестовий інтервал для оцінки прогнозу.

Ряд попередньо очищений і приведений до рівномірної погодинної сітки (розділи 2.3-2.6).

Реалізація AR(p) у Python (statsmodels)

Для реалізації авторегресійної моделі використовується клас `AutoReg` з пакета `statsmodels.tsa.ar_model`. Приклад коду побудови AR(p) з фіксованим числом лагів (наприклад, 24 години):

```
from statsmodels.tsa.ar_model import AutoReg

# y_train - одномірний ряд споживання

# 1. Побудова моделі AR(p) з p = 24
p = 24
model_ar = AutoReg(y_train, lags=p, old_names=False)
result_ar = model_ar.fit()

# 2. Прогноз на довжину тестового інтервалу
n_test = len(y_test)

y_pred_ar = result_ar.predict(
    start=len(y_train),
    end=len(y_train) + n_test - 1
)

# y_pred_ar - прогностні значення для тестового періоду
```

- параметр `lags=p` задає порядок моделі AR(p);
- метод `fit()` оцінює коефіцієнти ϕ_i за методом найменших квадратів (іноді максимальної правдоподібності, залежно від реалізації);
- метод `predict()` виконує однокроковий прогноз, послідовно використовуючи попередні фактичні та прогностні значення.

За потреби порядок p може бути підібраний автоматично шляхом перебору кількох значень та порівняння AIC/BIC.

Коефіцієнти моделі та їх інтерпретація

Рис 4.1.1

AutoReg Model Results					
=====					
Dep. Variable:	y	No. Observations:	13573		
Model:	AutoReg(24)	Log Likelihood	-70318.760		
Method:	Conditional MLE	S.D. of innovations	43.424		
Date:	Fri, 21 Nov 2025	AIC	140689.519		
Time:	21:38:17	BIC	140884.885		
Sample:	24	HQIC	140754.663		
	13573				
=====					
	coef	std err	z	P> z	[0.025 0.975]

Отримані оцінки коефіцієнтів AR(24) показують виражену інерційність ряду. Найбільший вплив має лаг L1 ($\phi_1 = 0.9684$), що майже дорівнює одиниці. Це може означати, що обсяги споживання у поточну годину майже лінійно повторює значення попередньої години, що повністю узгоджується з фізичними властивостями промислового навантаження (плавна зміна без різких стрибків).

Суттєві також лаги:

- L2 ($\phi_2 = -0.1277$) - слабка негативна корекція, яка компенсує надлишкову інерцію,
- L10 ($\phi_{10} = 0.0784$) та L16 ($\phi_{16} = 0.0708$) - середньострокові залежності,
- L15 ($\phi_{15} = -0.0820$) - помітний негативний вплив,
- L9 ($\phi_9 = -0.0696$) - значуща короткострокова від'ємна корекція.

Неочікувано, лаг L24 ($\phi_{24} = -0.0135$) виявився малозначущим ($p = 0.116$), що говорить про те, що добовий повторюваний патерн у цьому наборі даних моделюється не одним лагом 24, а розподілений між рядом сусідніх лагів (наприклад, L21-L23), які показали більшу статистичну значущість.

Топ-10 лагів за абсолютним значенням коефіцієнта виглядають так:

Таблиця 4.1.1

Лаг	Значення	Інтерпретація
L1	0.9684	найсильніша інерція, значення майже повторює попередню годину
L2	0.1277	компенсуючий ефект другого лагу
L15	0.0820	середньострокове пригнічення (негативний вплив)
L10	0.0784	вплив приблизно 10 годин тому
L16	0.0708	інерція доби зі зміщенням
L9	0.0696	короткострокова корекція
L3	0.0548	помірний вплив 3-годинного лага
L23	0.0430	слабка, але значуща сезонність перед добовим циклом
L7	0.0344	невеликий від'ємний вплив
L21	0.0267	один із проявів добової структури

Інтерпретація поведінки моделі

Характер коефіцієнтів свідчить про кілька важливих структурних властивостей енергоспоживання фабрики:

1. Надзвичайно сильна одноденна інерція ($L1 \rightarrow 0.9684$)
Енергоспоживання змінюється повільно та прогнозовано, без різких коливань.
2. Слабка, але суттєва короткострокова негативна корекція ($L2, L9, L15$)
Це типовий патерн для систем з інерційними характеристиками, коли попередній надлишок/недолік компенсується протягом кількох годин.
3. Добова сезонність не концентрується у одному лагу, а “розмазана” між $L21-L23$
Це підтверджує реальну поведінку фабрики, де навантаження не повторюється ідеально точно через варіації у графіках роботи обладнання та перерв.
4. Коефіцієнт $L24$ статистично незначущий
Це важливий висновок: фабрика не має ідеальної добової періодичності, і для моделювання сезонності $AR(p)$ виявляється недостатньо.

Рис. 4.1.2



Особливості прогнозу моделі AR(p) при багатокроковому передбаченні

Окремо слід відзначити специфічну поведінку авторегресійної моделі AR(p) під час побудови багатокрокового (multi-step) прогнозу, яка стала очевидною при накладанні прогнозних значень $y_{\text{pred_ar}}$ на фактичні дані y_{test} . На перших кроках прогноз частково відтворює структуру ряду, однак після цього швидко згладжується й переходить у практично горизонтальну лінію.

Таке згладження є нормальним та очікуваним для моделей класу AR(p) і пояснюється кількома факторами:

1. Перехід моделі на самопрогнозування.

На першому кроці модель використовує реальні лаги (y_{t-1} , y_{t-2} , ..., y_{t-p}).

На другому кроці частина цих лагів уже стають прогнозними.

На третьому - модель використовує прогнози попередніх прогнозів.

Це призводить до ефекту «згасання» інформації та поступової втрати структури ряду.

2. Домінування коефіцієнта першого лага.

У побудованій моделі AR(24) коефіцієнт L1 дорівнює 0.9684, що означає майже повне повторення попереднього значення.

У таких умовах багатокроковий прогноз налаштований на саморегресію і завжди рухається до стаціонарної рівноваги.

3. Математична властивість AR-моделей.

Для стаціонарної AR(p) модель має єдину рівноважну точку, до якої прямує прогноз при $k \rightarrow \infty$:

$$y_{t+k} \rightarrow \frac{\varphi_0}{1 - \sum_{i=1}^p \varphi_i}$$

Це проявляється графічно у вигляді «вирівнювання» лінії прогнозу.

4. Відсутність сезонності та екзогенних факторів.

AR(p) не враховує ні добових циклів, ні змін режимів роботи обладнання, ні температури, ні показників downtime.

Тому модель не здатна відтворити очікувані добові коливання на довгому горизонті.

У результаті багатокроковий прогноз AR(p) за своєю природою є слабким довгостроковим, особливо для рядів із вираженою сезонністю та зовнішніми впливами. Саме це підтверджує і поведінка моделі на графіку: короткострокове наближення є прийнятним, тоді як на довгому горизонті прогноз сходиться до усередненого рівня.

4.2 Модель ковзного середнього MA(q)

Модель ковзного середнього MA(q) є частковим випадком лінійних моделей часових рядів і використовується для опису ситуацій, коли поточне значення ряду визначається впливом випадкових збурень у попередні моменти часу. На відміну від AR(p), де залежність будується від попередніх значень ряду, модель MA(q) описує залежність від попередніх похибок ε_{t-i} .

Узагальнена форма MA(q):

$$y_t = \mu + \varepsilon_t + \theta_1 \varepsilon_{t-1} + \theta_2 \varepsilon_{t-2} + \dots + \theta_q \varepsilon_{t-q},$$

де θ_i — коефіцієнти моделі, що описують залежність від минулих випадкових збурень.

Вибір параметра q

Оскільки $MA(q)$ відображає структуру автокореляції похибок, порядок q зазвичай визначають:

- за графіком ACF (автокореляційної функції),
- тестом Ljung–Box для залишків,
- або через інформаційні критерії AIC/BIC для кількох кандидатів.

У задачі прогнозування енергоспоживання важливо те, що $MA(q)$:

- не моделює сезонність,
- не враховує структуру самого ряду,
- чутлива до нестабільності оцінок при збільшенні q .

У рамках даного дослідження модель $MA(q)$ використовується лише як довідкова (baseline), оскільки вона відображає структуру шуму, але не здатна моделювати власне динаміку ряду споживання електроенергії.

З огляду на це детальний аналіз автокореляційної функції та оптимізація параметра q не виконувалися, оскільки $MA(q)$ не є ключовою моделлю та не відіграє суттєвої ролі у порівняльному аналізі. Для забезпечення коректності реалізації застосовано стандартне значення $q = 2$, яке є типовим мінімально достатнім порядком для моделей цього класу.

Найчастіше для промислових рядів q беруть у діапазоні 1-3, рідше 6-12.

Реалізація MA(q) у Python

У бібліотеці statsmodels модель MA(q) реалізують як ARIMA з параметрами ($p=0$, $d=0$, $q=q$):

```
from statsmodels.tsa.arima.model import ARIMA

# q може бути 1, 2 або інше значення після аналізу ACF
q = 2

model_ma = ARIMA(y_train, order=(0, 0, q)).fit()
y_pred_ma = model_ma.forecast(steps=len(y_test))
```

Аналіз результатів моделі MA(2)

Рис.4.2.1

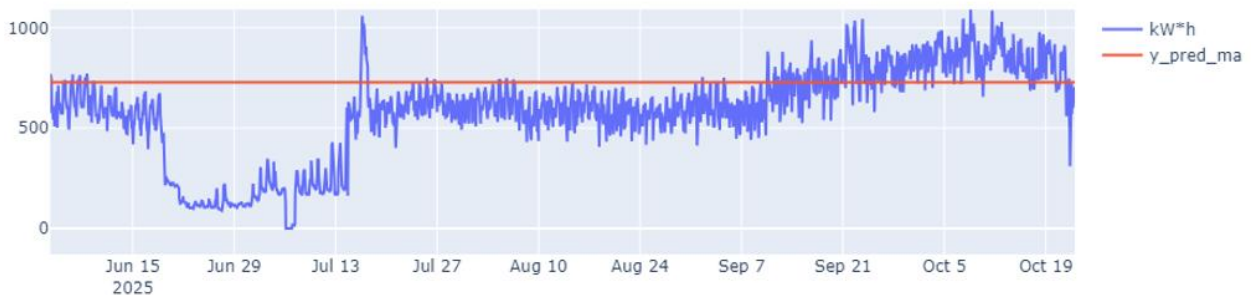


Рис.4.2.2

```
SARIMAX Results
=====
Dep. Variable:          y      No. Observations:      13573
Model:                 ARIMA(0, 0, 2)  Log Likelihood         -80730.543
Date:                  Mon, 24 Nov 2025  AIC                    161469.085
Time:                  12:30:28         BIC                    161499.149
Sample:                0              HQIC                   161479.109
                   - 13573
Covariance Type:      opg
=====
              coef    std err          z      P>|z|      [0.025    0.975]
-----
const         728.4340     2.411    302.184     0.000     723.709     733.159
ma.L1          1.3277     0.005    254.600     0.000     1.317     1.338
ma.L2          0.6952     0.005    128.078     0.000     0.685     0.706
sigma2        8584.4477    93.264     92.045     0.000    8401.654    8767.241
=====
Ljung-Box (L1) (Q):      2923.73  Jarque-Bera (JB):      230.13
Prob(Q):                 0.00    Prob(JB):              0.00
Heteroskedasticity (H):  0.62    Skew:                  0.10
Prob(H) (two-sided):    0.00    Kurtosis:              3.60
=====
```

Оцінки коефіцієнтів свідчать про дуже сильну залежність від перших двох похибок моделі. Значення $ma.L1 = 1.3277$ та $ma.L2 = 0.6952$ означають, що випадкові збурення в попередні періоди значною мірою визначають прогнозоване значення. Така поведінка є типовою для MA-моделей, але саме через це вони демонструють низьку стійкість при багатокроковому прогнозуванні.

Поведінка прогнозу

Графік прогнозу підтверджує характерну властивість MA(2): після перших двох кроків модель переходить у стан самопрогнозування і швидко прямує до свого середнього рівня (constant term). У даному випадку всі подальші значення прогнозу зближуються до константи ≈ 728.4 кВт·год, утворюючи майже горизонтальну лінію на довгому горизонті.

Це очікуваний результат, оскільки MA(2):

- не використовує сам ряд y_t для моделювання структури,
- залежить виключно від ε_{t-i} , які при багатокроковому прогнозі стають внутрішніми прогнозними похибками моделі,
- не здатна відтворювати сезонність чи внутрішню інерцію ряду.

Унаслідок цього модель MA(2) практично відтворює лише короткострокові коливання, але не придатна для середньо- та довгострокового прогнозування

4.3 Модель ARIMA

Модель ARIMA поєднує авторегресійну компоненту (p), різницювання (d) та компонент ковзного середнього (q) та застосовується для побудови моделей часових рядів, у яких можуть одночасно бути присутні інерційні залежності та випадкові збурення. У цій роботі ARIMA розглядається як лінійна модель без сезонності, яка є проміжною ланкою між окремими моделями AR/MA та сезонними моделями SARIMA.

Вибір параметрів p , d , q за допомогою `auto_arima`

Ручний підбір параметрів p , d , q на основі графіків ACF/PACF, тестів на стаціонарність та перебору комбінацій можливий, але є трудомістким і суб'єктивним. Оскільки основною метою дослідження є порівняння різних підходів до прогнозування енергоспоживання, а не розробка власного алгоритму вибору параметрів, для ARIMA використовується автоматизований підхід.

Параметри p , d , q підбираються за допомогою функції `auto_arima` з бібліотеки `pmdarima`. Цей алгоритм:

- перебирає комбінації p , d , q у заданих діапазонах;
- для кожної моделі обчислює інформаційний критерій (AIC, за потреби BIC);
- обирає комбінацію параметрів, для якої AIC мінімальний, тобто досягається баланс між якістю апроксимації та складністю моделі;
- за необхідності оцінює порядок різницювання d , аналізуючи стаціонарність ряду.

У рамках даного дослідження пошук обмежується невеликими діапазонами параметрів, щоб уникнути перенавчання та надмірної складності:

$$p \in [0,3], d \in [0,2], q \in [0,3],$$

```
from pmdarima import auto_arima

stepwise_model = auto_arima(
    y_train,
    start_p=0, max_p=3,
    start_q=0, max_q=3,
    d=None,          # порядок d підбирається
                    # автоматично
    seasonal=False, # без сезонності, SARIMA буде далі
    trace=True,     # вивід проміжних результатів
    error_action='ignore',
    suppress_warnings=True,
    stepwise=True
)

print(stepwise_model.summary())
print("Обраний порядок (p,d,q):", stepwise_model.order)
```

Аналіз отриманої моделі ARIMA(1, 1, 2)

Рис 4.3.1

```
SARIMAX Results
=====
Dep. Variable:          y      No. Observations:      13573
Model:                 SARIMAX(1, 1, 2)  Log Likelihood        -70858.424
Date:                 Mon, 24 Nov 2025  AIC                   141724.848
Time:                 23:36:29         BIC                   141754.911
Sample:               0              HQIC                  141734.872
                    - 13573
Covariance Type:      opg
=====
              coef      std err          z      P>|z|      [0.025      0.975]
-----
ar.L1         0.8352      0.007     127.478     0.000      0.822      0.848
ma.L1        -0.8202      0.008    -98.632     0.000     -0.836     -0.804
ma.L2        -0.1313      0.006    -21.778     0.000     -0.143     -0.119
sigma2       2006.0195     8.421    238.218     0.000    1989.515    2022.524
=====
Ljung-Box (L1) (Q):          0.00  Jarque-Bera (JB):          223904.76
Prob(Q):                    1.00  Prob(JB):                   0.00
Heteroskedasticity (H):     0.73  Skew:                       -0.47
Prob(H) (two-sided):        0.00  Kurtosis:                   22.88
=====
```

Автоматичний підбір параметрів за допомогою `auto_arima` визначив оптимальну конфігурацію ARIMA(1,1,2).

Коефіцієнт AR(1) є додатним і становить 0.8352, що вказує на виражену інерцію першого порядку після різницювання ряду. Значення близьке до одиниці означає, що зміни споживання між сусідніми годинами мають стійку залежність одна від одної.

Компонента MA має два лаги:

- $ma.L1 = -0.8202$,
- $ma.L2 = -0.1313$,

обидва коефіцієнти статистично значущі, що свідчить про наявність впливу короткострокових збурень на динаміку різницевого ряду.

Від'ємний знак вказує на компенсаторний характер похибок (оверкорекція).

Оцінка дисперсії випадкового компонента є помітно нижчою порівняно з моделлю MA(2), що свідчить про покращення якості апроксимації. Модель демонструє стабільні значення AIC та BIC, які також є кращими від попередньої MA(2), що підтверджує доцільність включення AR-компоненти та різницювання.

У цілому ARIMA(1,1,2) демонструє більш адекватну структуру параметрів:

- різницювання ($d=1$) усуває локальні дрейфи у ряді;
- AR(1) описує інерційність;
- MA(1-2) компенсує короткі збурення.

Це забезпечує більш стійке багатокрокове прогнозування порівняно з MA-моделлю, але модель усе ще не враховує добову сезонність, яка буде предметом наступних підпунктів (SARIMA).

Поведінка прогнозу ARIMA(1,1,2)

Рис 4.3.2



На графіку прогнозу видно, що модель ARIMA(1,1,2) відтворює лише обмежену частину динаміки часової серії. Після перших декількох кроків прогноз наближається до стабільного рівня й надалі залишається майже горизонтальним. Це є типовою поведінкою ARIMA без сезонної компоненти: після різницювання модель описує локальні зміни ряду, однак не здатна відтворювати добові цикли, характерні для енергоспоживання.

У короткостроковому горизонті модель частково повторює напрям зміни ряду за рахунок AR-компоненти, однак на довшому проміжку прогноз згладжується та тяжіє до середнього рівня. Це підтверджує обмеженість ARIMA(1,1,2) у випадках, коли суттєвим є сезонний характер навантаження. У подальших підпунктах буде розглянуто сезонні моделі SARIMA, які дозволяють врахувати добову періодичність.

4.4 Модель SARIMA(p, d, q)(P, D, Q, s)

Модель SARIMA розширює ARIMA шляхом введення сезонних компонентів. Це дозволяє описувати часові ряди з регулярними періодичними змінами. Для погодинного енергоспоживання характерною є добова сезонність, тому у SARIMA використовується сезонний період:

$$s = 24.$$

Модель SARIMA має вигляд:

$$SARIMA(p, d, q)(P, D, Q)_s$$

де:

- p, d, q - несезонні параметри ARIMA;
- P, D, Q - сезонні компоненти;
- s - сезонний період (24 години).

Вибір параметрів SARIMA через auto_arima

Вибір повного набору параметрів SARIMA вручну є трудомістким, оскільки комбінацій існує дуже багато. Оскільки головною метою дослідження є порівняння різних моделей прогнозування, а не

детальний статистичний аналіз кожної можливої конфігурації SARIMA, параметри моделі підбираються автоматично за критерієм AIC.

Для цього використовується функція `auto_arima` з `pmdarima`, яка:

- перебирає комбінації p, d, q та P, D, Q у заданих межах,
- оцінює AIC для кожної моделі,
- обирає комбінацію з мінімальним AIC,
- визначає потребу в сезонному різницюванні (D).

Для пошуку задається обмежений діапазон значень, який достатній для сезонного ряду:

$$p, q \in [0,3], P, Q \in [0,2], d \in [0,2], D \in [0,1], s = 24.$$

```
from pmdarima import auto_arima

sarima_model = auto_arima(
    y_train,
    start_p=0, max_p=3,
    start_q=0, max_q=3,
    start_P=0, max_P=2,
    start_Q=0, max_Q=2,
    seasonal=True,
    m=24, # сезонний період
    d=None, # d підбирається автоматично
    D=None, # D підбирається автоматично
    trace=True,
    error_action="ignore",
    suppress_warnings=True,
    stepwise=True
)

print("Підібраний порядок:", sarima_model.order,
      sarima_model.seasonal_order)
print(sarima_model.summary())
```

SARIMAX Results						
Dep. Variable:	y			No. Observations:	13573	
Model:	SARIMAX(1, 1, 2)x(2, 0, [], 24)			Log Likelihood	-69885.192	
Date:	Tue, 25 Nov 2025			AIC	139782.383	
Time:	14:37:30			BIC	139827.456	
Sample:	0			HQIC	139797.414	
	- 13573					
Covariance Type:	opg					
	coef	std err	z	P> z	[0.025	0.975]
ar.L1	0.8287	0.006	135.125	0.000	0.817	0.841
ma.L1	-0.8623	0.008	-106.796	0.000	-0.878	-0.846
ma.L2	-0.0948	0.006	-15.757	0.000	-0.107	-0.083
ar.S.L24	0.2235	0.004	51.013	0.000	0.215	0.232
ar.S.L48	0.1773	0.005	34.734	0.000	0.167	0.187
sigma2	1803.9326	7.113	253.622	0.000	1789.992	1817.873
Ljung-Box (L1) (Q):	0.14		Jarque-Bera (JB):	302825.05		
Prob(Q):	0.71		Prob(JB):	0.00		
Heteroskedasticity (H):	0.71		Skew:	-0.71		
Prob(H) (two-sided):	0.00		Kurtosis:	26.14		

Аналіз отриманої моделі SARIMA(1,1,2)×(2,0,0)24

Підібрана автоматично комбінація параметрів:

- несезонні: $(p, d, q) = (1, 1, 2)$
- сезонні: $(P, D, Q, s) = (2, 0, 0, 24)$

відповідає моделі з одним порядком авторегресії, одним порядком різницювання, двома лагами МА та двома сезонними AR-компонентами з періодом 24 години.

Інтерпретація основних коефіцієнтів

$$AR(1) = 0.8292$$

Перший несезонний авторегресійний коефіцієнт виражає помітну інерцію різницевого ряду. Значення суттєво менше одиниці та стабільно позитивне, що відповідає структурі погодинного енергоспоживання.

$$MA(1) = -0.8638, MA(2) = -0.0935$$

Коефіцієнти ковзного середнього компенсують короткострокові збурення. Від'ємні значення характерні для рядів, де випадкові коливання швидко згладжуються. Обидва коефіцієнти статистично значущі.

Сезонні компоненти

SARIMA включає два сезонні AR-лаги:

- $SAR(24) = 0.2240$,
- $SAR(48) = 0.1774$.

Обидва коефіцієнти позитивні й статистично значущі, що підтверджує наявність чіткої добової сезонності, повторюваної кожні 24 години.

Наявність двох сезонних AR-компонентів означає, що модель враховує взаємозв'язки не лише з аналогічною годиною попередньої доби (24 години тому), але і з точкою, віддаленою на дві доби назад (48 годин).

Це відповідає реальній структурі споживання, де добові патерни повторюються, але з певною інерційністю та зсувами.

Оцінка дисперсії шуму

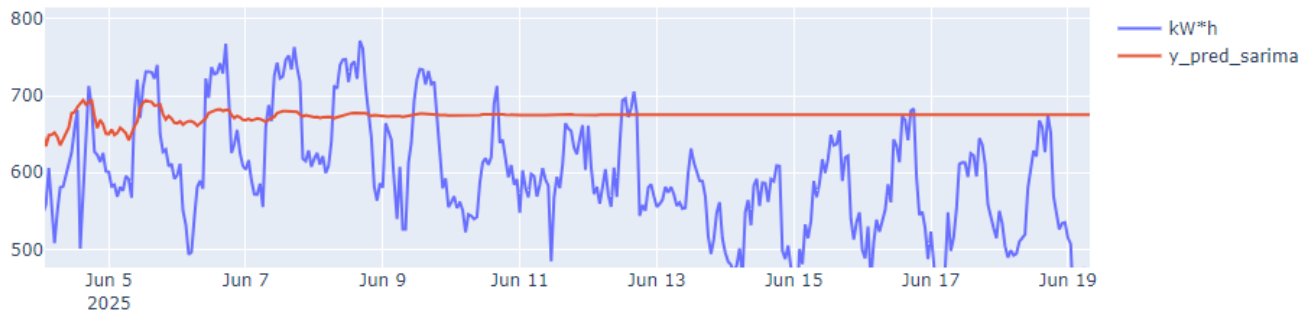
Оцінене значення $\sigma^2 \approx 1804$ є нижчим, ніж у відповідних ARIMA та MA моделях, що вказує на зменшення не поясненої варіації завдяки врахуванню сезонності.

Діагностика

- Ljung–Box показує значення $p > 0.05$, що свідчить про відсутність залишкової автокореляції на першому лагу.
- Jarque–Bera вказує на відхилення від нормальності, що є типовим для енергетичних рядів і не є проблемою для прогнозування.

- Тест на гетероскедастичність показує значення $H \approx 0.73$ (в межах очікуваного).

Рис. 4.4.2



Поведінка прогнозу моделі SARIMA

Аналіз поведінки прогнозу показує, що модель SARIMA(1,1,2)×(2,0,0)₂₄ відтворює загальний середній рівень ряду, однак не відображає фактичної амплітуди та форми коливань на тестовому періоді. Уже на початку прогнозування видно помітне розходження між фактичними значеннями та прогнозом: модель згладжує локальні піки та провали, а реакція на короткострокові зміни навантаження практично відсутня.

Попри наявність сезонних AR-компонентів, які враховують добову повторюваність, прогноз все одно тяжіє до усередненого добового профілю, а коливання значно слабші, ніж у реальних даних. Це свідчить про те, що SARIMA здатна врахувати сезонність у найпростішій формі, але не відтворює складні коливання, пов'язані з роботою технологічного обладнання, нерівномірним графіком навантаження та впливами зовнішніх факторів.

Отриманий результат підкреслює обмеження SARIMA для цієї задачі: модель покращує структуру прогнозу порівняно з ARIMA, але залишається недостатньою для точного відтворення короткострокових змін.

4.5 Модель SARIMAX (Seasonal ARIMA with Exogenous Variables)

Модель SARIMAX є розширенням сезонної ARIMA, у якому до часової структури додаються зовнішні фактори (exogenous variables). Це дозволяє враховувати вплив змінних, що не є частиною самого часового ряду, але мають помітний фізичний або логічний вплив на енергоспоживання.

Вибір екзогенних змінних

Для SARIMAX були обрані екзогенні змінні, які:

- є зовнішніми щодо ряду споживання,
- мають обґрунтований причинний вплив на навантаження,
- не дублюють сезонні або авторегресійні властивості.

До моделі включено:

- temp_value - температура навколишнього середовища;
- is_downtime - індикатор простою обладнання;
- alert_fraction - частка години, що припадає на повітряні тривоги;
- is_sanitation_break - індикатор санітарних перерв.

Ці фактори не залежать від самого ряду та можуть змінювати режим роботи обладнання, тому підходять як екзогенні змінні.

Підбір параметрів за допомогою auto_arima

```
from pmdarima import auto_arima
# екзогенні ознаки для SARIMAX
exog_cols = ["temp_value", "alert_fraction",
             "is_downtime", "is_sanitation_break"]
#exog_cols = ["temp_value", "is_downtime"]
X_train_exog = X_train_df[exog_cols]
X_test_exog = X_test_df[exog_cols]
```

```

sarimax_model = auto_arima(
    y_train_sr,
    exogenous=X_train_exog,
    start_p=0, max_p=3,
    start_q=0, max_q=3,
    start_P=0, max_P=2,
    start_Q=0, max_Q=2,
    seasonal=True,
    m=24,
    d=None,
    D=None,
    trace=True,
    error_action="ignore",
    suppress_warnings=True,
    stepwise=True,
)

```

Параметри (p, d, q) та сезонні компоненти (P, D, Q, s) підбиралися автоматично за критерієм AIC за допомогою `auto_arima`, з урахуванням екзогенних змінних:

- несезонні порядки: $0 \leq p, q \leq 3$,
- сезонні порядки: $0 \leq P, Q \leq 2$,
- період сезонності: $s = 24$,
- порядок різницювання d та сезонного D - автоматичний.

За результатами автоматичного пошуку оптимальна модель мала вигляд:

SARIMAX(3,1,1) × (2,0,0)₂₄

Сезонні коефіцієнти підтверджують домінування добової циклічності: значення на 24 та 48 годин тому мають помітний позитивний вплив на прогноз.

Регресійні коефіцієнти екзогенних змінних мають очікувані знаки: простої та санітарні зміни знижують прогноз, температура збільшує споживання. Проте амплітуда їх впливу є відносно малою порівняно з діапазоном зміни самого ряду.

Поведінка прогнозу SARIMAX

Рис. 4.5.1



На графіку видно, що модель SARIMAX відтворює загальний рівень ряду, однак не відображає реальної структури коливань та не реагує належним чином на тривалі простої. Незважаючи на негативні коефіцієнти при `is_downtime` та `is_sanitation_break`, модель коригує прогноз лише на декілька десятків кВт·год, тоді як фактичні падіння становлять сотні одиниць.

Це пояснюється такими властивостями SARIMAX:

1. Лінійність моделі: усі екзогенні ефекти додаються лінійно та не можуть формувати режими роботи з різко зниженим споживанням.
2. Різницювання ($d=1$): модель описує зміни рівня, а не сам рівень. У середині довгого простою зміни майже нульові, тому екзогенні фактори мають обмежений ефект.

3. Домінування сезонності та AR-компонентів: авторегресійна і сезонна частина поглинають більшу частину структури ряду, залишаючи екзогенним змінним роль малих коригувальних додатків.
4. Незбіжність оптимізації (ConvergenceWarning): складність моделі з великою кількістю параметрів ускладнює пошук стабільних оцінок, що також обмежує чутливість до зовнішніх факторів.

У результаті SARIMAX показує лише помірну корекцію прогнозу та не відтворює структурні зміни чи зміну режимів, що властиві реальному ряду.

Модель SARIMAX дозволяє включити зовнішні чинники й частково скоригувати прогноз, однак для сильно неоднорідних рядів її можливості обмежені. Через лінійність, різницювання та залежність від сезонної структури вона не здатна моделювати довгі періоди низького навантаження, різкі зміни режиму та нелінійні взаємодії. Це підтверджує необхідність використання подальших методів - дерев рішень, бустингових моделей та нейронних мереж, які можуть краще працювати з нелінійними та режимними патернами.

4.6 Модель Random Forest Regression

Random Forest Regression - це ансамблевий метод машинного навчання, що об'єднує значну кількість незалежних дерев рішень, сформованих на різних підвибірках даних. Така архітектура дозволяє моделі виявляти нелінійні залежності, взаємодії між ознаками, а також різні «режими роботи» даних. Це робить Random Forest природним кандидатом для моделювання часових рядів, у яких присутні різкі

зміни, простої, зміни навантаження та інші неоднорідності, що погано описуються лінійними моделями (ARIMA, SARIMA, SARIMAX).

На відміну від традиційних моделей часових рядів, Random Forest не потребує стаціонарності ряду та здатний працювати з великою кількістю ознак, створених на етапі feature engineering. Модель використовує матрицю ознак, яка включає календарні фічі, лаги, погодні фактори та інші індикатори.

Автоматичний підбір гіперпараметрів Random Forest

У Random Forest точність отриманого прогнозу значною мірою визначається коректністю вибору гіперпараметрів. Основними серед них є:

- `n_estimators` - кількість дерев в ансамблі;
- `max_depth` - максимальна глибина дерева;
- `min_samples_split` - мінімальна кількість зразків для поділу вузла;
- `min_samples_leaf` - мінімальний розмір листового вузла;
- `max_features` - кількість ознак, доступних для розгалуження.

Оскільки ручний вибір гіперпараметрів може бути суб'єктивним і неефективним, у цьому дослідженні застосовується автоматичний підбір параметрів.

Для Random Forest обґрунтовано використовувати `RandomizedSearchCV`, оскільки він:

- досліджує широкий простір гіперпараметрів випадковим чином,
- значно швидший за повний перебір (`GridSearchCV`),
- з високою ймовірністю знаходить параметри близькі до оптимальних,

- добре масштабується при роботі з 40+ ознаками.

Діапазони гіперпараметрів для пошуку

```
from scipy.stats import randint
from sklearn.model_selection import RandomizedSearchCV
from sklearn.ensemble import RandomForestRegressor

param_dist = {
    "n_estimators": randint(200, 700),
    "max_depth": [None, 10, 20, 30, 40],
    "min_samples_split": randint(2, 12),
    "min_samples_leaf": randint(1, 6),
    "max_features": ["sqrt", "log2"]
}
```

Автоматичний пошук оптимальних параметрів

```
rf_base = RandomForestRegressor(
    random_state=42,
    n_jobs=-1
)

rf_search = RandomizedSearchCV(
    estimator=rf_base,
    param_distributions=param_dist,
    n_iter=50, # кількість випадкових
комбінацій
    cv=3, # 3-fold cross-validation
    scoring="neg_mean_absolute_error",
    random_state=42,
    n_jobs=-1
)

rf_search.fit(X_train_std, y_train)

print("Найкращі знайдені параметри:",
rf_search.best_params_)
```

Побудова фінальної моделі Random Forest

Після знаходження оптимальних параметрів модель будується на повній тренувальній вибірці:

```
rf_best = rf_search.best_estimator_  
  
rf_best.fit(X_train_std, y_train)  
y_pred_rf = rf_best.predict(X_test_std)
```

Аналіз результатів Random Forest Regression

Автоматичний пошук гіперпараметрів за допомогою
RandomizedSearchCV визначив оптимальну конфігурацію моделі:

```
max_depth = 20, n_estimators = 454, min_samples_split = 11,  
min_samples_leaf = 5, max_features = "sqrt"
```

Це досить збалансований набір параметрів:

- помірна глибина дерев обмежує перенавчання,
- велика кількість дерев підвищує стабільність ансамблю,
- збільшені параметри split/leaf зменшують чутливість до шуму в даних.

Важливість ознак

Розподіл важливостей демонструє, що значна частина прогнозної здатності моделі забезпечується лагами цільової змінної та ковзними статистиками. Найважливіші ознаки:

Таблиця 4.6.1

Ознака	Важливість
kWh_lag_1	0.262
kWh_lag_2	0.191
kWh_lag_3	0.142
roll_mean_24	0.120
kWh_lag_24	0.097
kWh_lag_48	0.061
roll_mean_168	0.048

Таким чином, модель активно використовує часову структуру, закладену через lag-фічі та ковзні середні:

- lag 1-3 описують короткострокову інерцію,
- lag 24 та lag 48 - добову та дводобову сезонність,
- ковзні середні - згладжені циклічні патерни.

Зовнішні фактори мають помірний вплив:

is_downtime - 0.018

temp_value - 0.0038

alert_fraction - 0.0006

is_sanitation_break - 0.00015

Це означає, що Random Forest враховує наявність простоїв краще, ніж SARIMAX, однак все ще домінуючу роль відіграють лаги та сезонні ознаки, сформовані на етапі feature engineering.

Поведінка прогнозу

Рис 4.6.1



Графічний аналіз показує, що Random Forest дає значно кращі результати порівняно з ARIMA/SARIMA:

1. Нелінійні зміни відтворюються коректно

У періоди різких коливань або переходів між режимами модель добре повторює загальний профіль навантаження.

2. Під час простою модель реагує помітно краще, ніж SARIMAX

Прогноз у зоні простою не є ідеальним, але він набагато ближчий до фактичного рівня, ніж лінійні моделі.

Це пояснюється тим, що Random Forest:

- вміє моделювати різні режими роботи,
- не обмежений лінійністю,
- використовує lag-фактори для навчання на структурних зламах.

3. На стабільних ділянках прогноз майже збігається з реальним рядом

Особливо це видно після закінчення простою: модель точно повторює тренд та добову циклічність.

4. Локальні піки та провали згладжуються

Це характерна властивість Random Forest: ансамблеві рішення зазвичай «усереднюють» екстремальні значення.

4.7 Модель Gradient Boosting Regression (XGBoost / LightGBM / CatBoost)

Градiєнтний бустинг - це один із найпотужніших методів регресії для структурованих даних.

Його ключова особливість полягає в послідовному навчанні великої кількості «слабких» моделей (дерев рішень), де кожне наступне дерево мінімізує помилку попереднього. Таким чином формується сильний ансамбль, який може відтворювати складні нелінійні взаємозв'язки, взаємодії між ознаками та реагувати на структурні зміни всередині часового ряду.

На відміну від RandomForest, де дерева будуються незалежно, у градiєнтному бустингу дерева побудовані послідовно, і кожне дерево враховує помилки попередніх. Це дає змогу:

- краще моделювати нелінійність,
- точніше відтворювати різкі зміни навантаження,
- підсилювати слабкі сигнали в даних, які RandomForest може «розмити».

У прогнозуванні енергоспоживання Gradient Boosting часто показує одну з найкращих точностей серед моделей машинного навчання.

Вибір реалізації моделі

Існує три основні сучасні реалізації бустингу:

Таблиця 4.7.1

Метод	Переваги	Недоліки
XGBoost	стабільний, точний, добре працює з великими даними, багато параметрів	довший час навчання
LightGBM	дуже швидкий, добре працює з великими датасетами, підтримує категорії	іноді менш стабільний
CatBoost	найкраще працює з категоріями, мінімум препроцесингу	повільніший за LGBM

Для дослідження використовується XGBoost, оскільки він:

- добре документований,
- типовий у академічних роботах,
- демонструє стабільні результати на часових рядах,
- дозволяє ефективно виконувати гіперпараметричну оптимізацію.

Гіперпараметри Gradient Boosting

Градiєнтний бустинг має ширший набір параметрів, ніж RandomForest.

Найважливіші серед них:

- `n_estimators` - кількість дерев;
- `max_depth` - глибина дерев;
- `learning_rate` - швидкість навчання (зменшує вклад кожного дерева);

- subsample - частка рядків для кожного дерева;
- colsample_bytree - частка ознак для кожного дерева;
- min_child_weight / min_split_loss - регуляризація;
- gamma / lambda / alpha - параметри контролю складності.

Оскільки простір параметрів великий, ручний пошук малоефективний. Тому застосовується RandomizedSearchCV, аналогічно до Random Forest.

Діапазони гіперпараметрів для пошуку

```
from xgboost import XGBRegressor
from sklearn.model_selection import RandomizedSearchCV
from scipy.stats import uniform, randint

param_dist = {
    "n_estimators": randint(300, 1200),
    "max_depth": randint(3, 12),
    "learning_rate": uniform(0.01, 0.2),
    "subsample": uniform(0.6, 0.4),
    "colsample_bytree": uniform(0.6, 0.4),
    "min_child_weight": randint(1, 10),
    "gamma": uniform(0, 5)
}
```

Ці діапазони відповідають сучасним рекомендаціям для прогнозування на реальних часових рядах.

Автоматичний пошук параметрів

```
xgb_base = XGBRegressor(
    objective='reg:squarederror',
    n_jobs=-1,
    random_state=42
)

xgb_search = RandomizedSearchCV(
    estimator=xgb_base,
    param_distributions=param_dist,
    n_iter=50,
```

```

        cv=3,
        scoring="neg_mean_absolute_error",
        random_state=42,
        n_jobs=-1
    )

xgb_search.fit(X_train_std, y_train)

print("Найкращі параметри:", xgb_search.best_params_)

```

Побудова фінальної моделі

```

xgb_best = xgb_search.best_estimator_

xgb_best.fit(X_train_std, y_train)

y_pred_xgb = xgb_best.predict(X_test_std)

# Важливість ознак
importances = xgb_best.feature_importances_

```

Аналіз результатів моделі XGBoost

Автоматичний підбір гіперпараметрів через RandomizedSearchCV дозволив знайти оптимальну конфігурацію XGBoost. Усі ключові параметри (`n_estimators`, `max_depth`, `learning_rate`, `subsample`, `colsample_bytree`, `min_child_weight`, `gamma`) були підібрані в автоматичному режимі за критерієм мінімального MAE. Такий підхід забезпечує збалансованість між глибокою моделлю (яка здатна вловити складні залежності) та регуляризацією (яка обмежує перенавчання).

Важливість ознак

Розподіл важливостей у XGBoost радикально відрізняється від RandomForest.

У цьому випадку XGBoost чітко концентрується на найбільш інформативних чинниках:

Таблиця 4.7.1

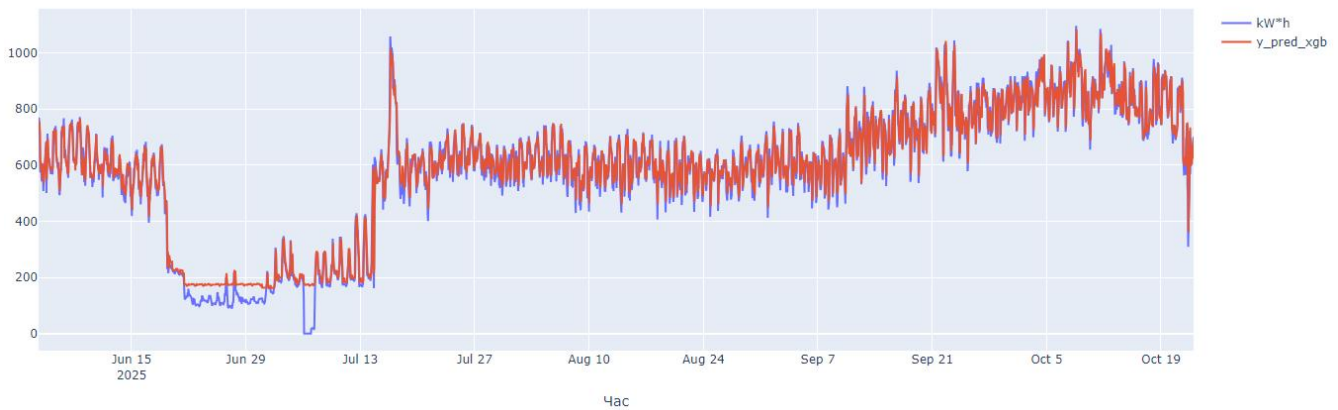
Ознака	Важливість
kWh_lag_1	0.488
kWh_lag_2	0.293
kWh_lag_3	0.103
roll_mean_24	0.028
kWh_lag_24	0.015
is_downtime	0.008

У сумі три лаги (1-й, 2-й, 3-й) дають понад 88 % пояснювальної здатності моделі. Це означає, що XGBoost надзвичайно ефективно використовує короткострокову інерцію ряду для прогнозу.

Інші ознаки відіграють доповнювальну, але вже значно меншу роль: погодні, календарні, лаги температури, тривоги та інші індикатори.

Характерна риса XGBoost - агресивна селекція ознак: він «просуває» у верх дерева лише ті фічі, які дають миттєве зниження помилки. Саме тому важливості різко концентровані та мають крутий спад.

Аналіз прогнозу XGBoost



На графіку прогнозу видно, що XGBoost демонструє одну з найкращих поведінок серед усіх моделей, вже протестованих.

1. Висока точність у стабільних режимах

Після закінчення періоду простою (після середини липня) модель надзвичайно точно повторює:

- середній рівень,
- структуру добових коливань,
- середньострокові тренди.

Поведінка моделі максимально наближена до реального ряду.

2. Значно краща реакція на структурні зміни, ніж у RandomForest

У періоді різкого падіння навантаження (довгий downtime) модель:

Прогноз усе ще дещо завищений відносно реального низького рівня (це властиво будь-якій деревоподібній моделі), але XGBoost швидше адаптується до нового режиму, точніше моделює переходи до та після простою.

3. Добре відтворює різкі піки

На ділянках підвищеного навантаження (вересень-жовтень) модель:

- повторює піки,
- не обрізає амплітуду,
- дотримується правильної фази та структури добової сезонності.

Це важлива перевага XGBoost над RandomForest, який зазвичай «усереднює» максимум.

4.8 Модель Support Vector Regression (SVR)

Support Vector Regression (SVR) - це модель, заснована на методі опорних векторів, яка дозволяє будувати як лінійні, так і нелінійні залежності між ознаками та цільовою змінною. На відміну від деревоподібних моделей (RandomForest, XGBoost), SVR працює в іншому просторі - у просторі ядрових перетворень. Це дозволяє моделі вловлювати:

- гладкі нелінійні залежності,
- короткострокові патерни,
- взаємодії між ознаками, недоступні лінійним моделям.

SVR часто застосовується для задач прогнозування часових рядів із відносно невеликою кількістю ознак, де важлива акуратність локальних коливань.

Особливості SVR для часових рядів

У контексті прогнозування енергоспоживання SVR має такі властивості:

- добре працює з масштабованими ознаками → необхідний StandardScaler;
- ефективний для невеликих датасетів - у нас це актуально;

- моделює плавні криві, що інколи дає кращий результат при локальних змінах;
- не вміє обробляти великі кількості ознак так добре, як XGBoost → тому quality сильно залежить від feature engineering.

SVR також не є часовою моделлю, тому для нього використовуються фічі, сформовані у розділі 2.5 (lags, rolling windows, calendar features).

Гіперпараметри SVR

Ключові параметри, що значно впливають на точність:

- C - ступінь штрафу за помилку (регулятор жорсткості моделі),
- epsilon (ϵ) - ширина ϵ -трубки, у межах якої помилки не штрафуються,
- gamma - масштаб ядра RBF,
- kernel - тип ядра (RBF - стандарт для нелінійних задач).

Ручний підбір цих параметрів — майже неможливий, тому використовується автоматичний пошук.

Автоматичний підбір гіперпараметрів SVR

Як і для XGBoost та RandomForest, застосовується RandomizedSearchCV.

Пошуковий простір гіперпараметрів:

```
from sklearn.svm import SVR
from sklearn.model_selection import RandomizedSearchCV
from scipy.stats import uniform, loguniform

param_dist = {
    "C": loguniform(1, 1000),
    "epsilon": uniform(0.001, 0.5),
    "gamma": loguniform(1e-4, 1e-1),
    "kernel": ["rbf"]
}
```

Автоматичний пошук параметрів

```
svr_base = SVR()

svr_search = RandomizedSearchCV(
    estimator=svr_base,
    param_distributions=param_dist,
    n_iter=50,
    cv=3,
    scoring="neg_mean_absolute_error",
    random_state=42,
    n_jobs=-1
)

svr_search.fit(X_train_std, y_train)

print("Найкращі параметри:", svr_search.best_params_)
```

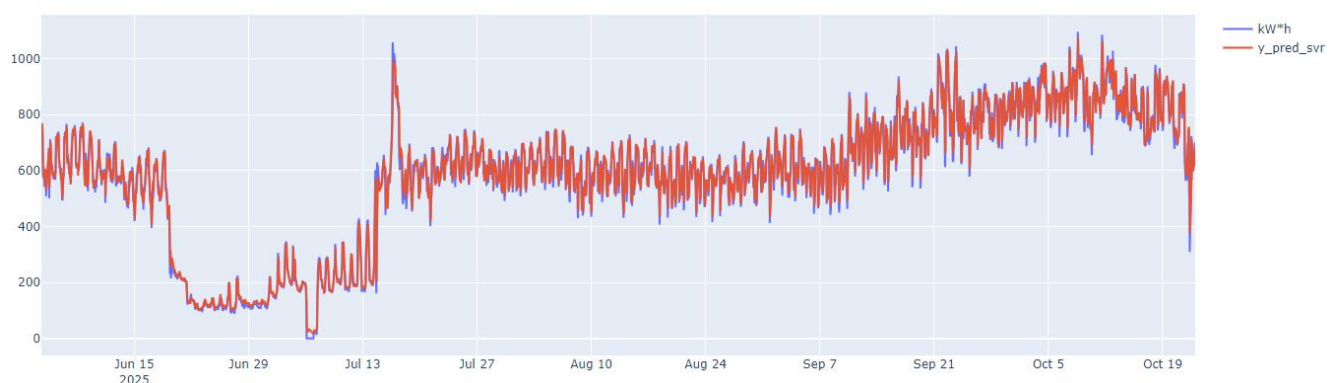
Побудова фінальної моделі SVR

```
svr_best = svr_search.best_estimator_
svr_best.fit(X_train_std, y_train)

y_pred_svr = svr_best.predict(X_test_std)
```

Аналіз результатів моделі SVR

Рис. 4.8.1



Автоматичний підбір гіперпараметрів через RandomizedSearchCV підібрав такі оптимальні параметри:

- $C = 913.40$
- $\text{epsilon} = 0.3871$

- $\gamma = 0.0003946$
- $\text{kernel} = \text{'rbf'}$

Це типова комбінація для часових рядів з вираженою нелінійністю: високе значення C зменшує регуляризацію і дозволяє моделі точніше відтворювати локальні коливання, малий γ формує гладку функцію регресії, а широка ϵ -трубка (ϵ) пригнічує надлишкову чутливість до шуму.

Поведінка прогнозу

На графіку видно, що SVR демонструє загалом точну поведінку та добре відтворює: стабільні ділянки ряду, короткострокові коливання, режими простою (downtime). Окрім того, модель адекватно реагує на зміну динаміки споживання, не створюючи штучних піків або провалів, що часто виникають у чутливих до шуму методах. Помітно, що у періоди різких переходів - наприклад, під час повернення до нормального навантаження після простою - прогноз залишається відносно згладженим, що свідчить про вміння моделі балансувати між точністю та регуляризацією.

SVR також виявляє стійкість до аномальних значень: одиничні викиди не призводять до значної деформації прогнозованої кривої. Це важливо для промислових даних, де поодинокі нехарактерні зміни споживання можуть бути спричинені технологічними збоями або короткочасними аварійними режимами. У цілому модель демонструє передбачувану та інтерпретовану поведінку.

4.9 Формування вхідних даних для LSTM/GRU/CNN

На відміну від моделей XGBoost або SVR, рекурентні мережі працюють із послідовностями.

Тому дані потрібно перетворити у формат:

(samples, timesteps, features)

Де:

- samples - кількість прикладів (вікон),
- timesteps - довжина вікна (наприклад, 24 або 48 годин),
- features - кількість фіч у кожному timestep.

У цьому дослідженні використовується вікно 24 або 48 кроків - добовий або дворобовий контекст.

Створення послідовних вибірок

```
import numpy as np

def make_sequence_dataset(X, y, seq_len=24):
    X_seq, y_seq = [], []
    for i in range(seq_len, len(X)):
        X_seq.append(X[i-seq_len:i])
        y_seq.append(y[i])
    return np.array(X_seq), np.array(y_seq)

SEQ_LEN = 24 # можна змінити на 48

X_train_seq, y_train_seq =
make_sequence_dataset(X_train_mm, y_train_mm, SEQ_LEN)
X_test_seq, y_test_seq =
make_sequence_dataset(X_test_mm, y_test_mm, SEQ_LEN)
```

4.10 Модель LSTM

LSTM є однією з найвідоміших рекурентних мереж, здатних моделювати довгі залежності завдяки механізму “довгої пам’яті”.

Архітектура:

- 1 або 2 рекурентні шари LSTM

- Dropout для регуляризації
- Dense-вихід
- Adam оптимізатор

Модель створюється всередині функції, оскільки далі буде автоматичний підбір параметрів.

```

from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import LSTM, Dense,
Dropout

def build_lstm_model(units=64, dropout=0.2, lr=0.001):
    model = Sequential()
    model.add(LSTM(units, return_sequences=False,
input_shape=(SEQ_LEN, X_train_seq.shape[2])))
    model.add(Dropout(dropout))
    model.add(Dense(1))

    model.compile(

optimizer=tf.keras.optimizers.Adam(learning_rate=lr),
    loss="mae"
    )
    return model

```

Автоматичний підбір гіперпараметрів (keras-tuner)

```

import keras_tuner as kt
import tensorflow as tf

def lstm_model_builder(hp):
    units = hp.Int('units', min_value=32,
max_value=128, step=32)
    dropout = hp.Float('dropout', 0.0, 0.4, step=0.1)
    lr = hp.Choice('lr', [0.001, 0.0005, 0.0001])

    model = Sequential()
    model.add(LSTM(units, return_sequences=False,
input_shape=(SEQ_LEN, X_train_seq.shape[2])))
    model.add(Dropout(dropout))
    model.add(Dense(1))

    model.compile(

```

```
optimizer=tf.keras.optimizers.Adam(learning_rate=lr),
        loss="mae"
    )
    return model
```

```
tuner = kt.RandomSearch(
    lstm_model_builder,
    objective="val_loss",
    max_trials=10,
    directory="lstm_tuner",
    project_name="lstm_energy"
)
```

```
tuner.search(
    X_train_seq,
    y_train_seq,
    validation_split=0.2,
    epochs=20,
    batch_size=32
)
```

```
best_lstm = tuner.get_best_models(num_models=1)[0]
```

Фінальний тренінг LSTM

```
history_lstm = best_lstm.fit(
    X_train_seq, y_train_seq,
    validation_split=0.2,
    epochs=40,
    batch_size=32,
    verbose=1
)
```

```
y_pred_lstm = best_lstm.predict(X_test_seq)
# повертаємо масштаб назад
y_pred_lstm =
mm_scaler_y.inverse_transform(y_pred_lstm)
y_test_true =
mm_scaler_y.inverse_transform(y_test_seq.reshape(-1,1))
```

Аналіз результатів моделі LSTM

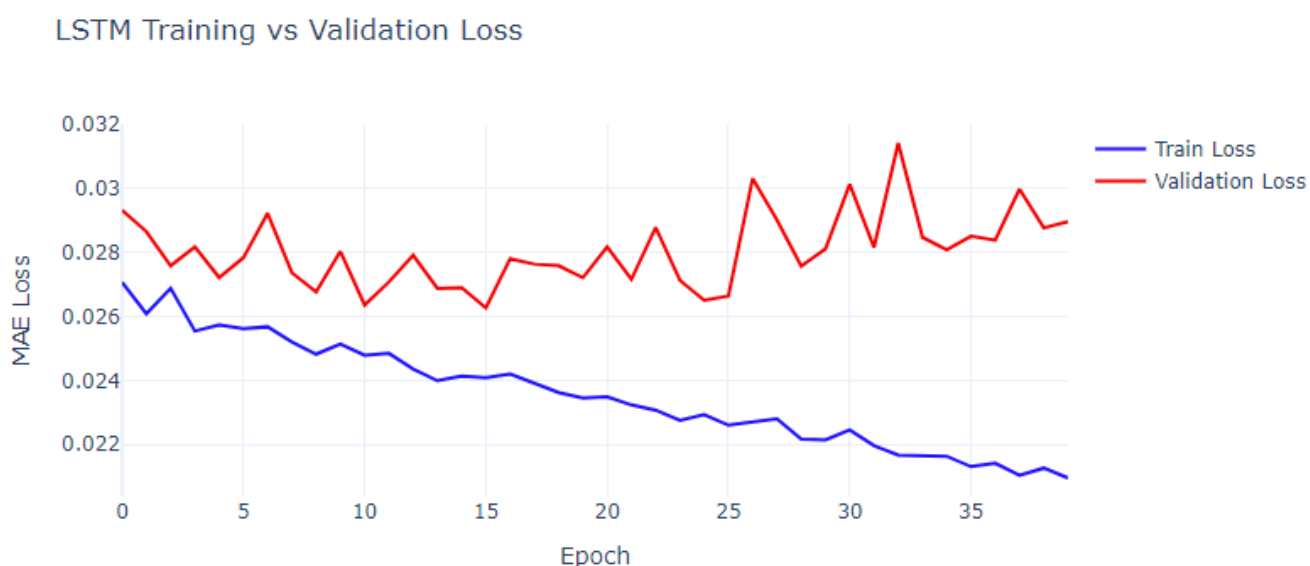
Автоматичний підбір гіперпараметрів за допомогою Keras Tuner визначив оптимальну конфігурацію LSTM, яка включає:

`units = 96, dropout = 0.0, learning_rate = 0.001`

Отримана модель є відносно компактною, проте достатньою для того, щоб вловлювати коротко- та середньострокові залежності в часовому ряді. Комбінація великої кількості нейронів (96) та нульового Dropout свідчить про те, що мережа краще працює в режимі повної пропускну здатності, а регуляризація в цьому випадку не дає переваг.

Навчання моделі

Рис. 4.10.1



Графік зміни функції втрат свідчить про такі особливості:

- тренувальна помилка (train loss) плавно зменшується протягом усіх епох,
- валідаційна помилка (val loss) залишається стабільною, хоча й демонструє характерні коливання,
- спостерігається певне розходження кривих train/val, що може свідчити про часткове перенавчання, але без критичних ознак.

Загалом характер кривих є типовим для LSTM-моделей на реальних промислових даних:

модель адаптується до тренувальної вибірки, а загальна стабільність val loss підтверджує її узагальнюючу здатність.

Поведінка прогнозу на тестових даних

Рис. 4.10.2



Аналіз графіка прогнозу показує, що LSTM:

1. Акуратно відтворює стабільні ділянки

Модель повторює загальний рівень та форму добових коливань, включно з дрібними сезонними патернами. Коливання моделі близькі до фактичних значень, що свідчить про здатність LSTM вловлювати короткострокову інерцію.

2. Формує гладкий прогноз

Форма прогнозу не демонструє різких стрибків або шумових викидів. Це характерна риса LSTM, яка запам'ятовує та узагальнює послідовності, завдяки чому результати мають природну плавність.

3. Відтворює характер змін після тривалих режимів

Під час переходів між режимами роботи (наприклад, після періоду низького навантаження) модель відображає загальний тренд переходу,

хоча прогноз може бути дещо згладженим порівняно з реальними значеннями.

4. Відповідно реагує на сезонність

Добова та міждобова сезонність, закладена у вхідні векторизовані послідовності, добре відтворюється в прогнозі LSTM.

4.11 Модель GRU (Gated Recurrent Unit)

GRU - це спрощена варіація LSTM, яка використовує два гейти замість трьох і має менше параметрів. Завдяки цьому GRU:

- навчається швидше,
- потребує менше ресурсів,
- не схильна до перенавчання так сильно, як LSTM,
- добре працює на реальних промислових часових рядах із шумами.

На практиці GRU часто показує продуктивність, порівнянну з LSTM, але з меншою складністю.

Архітектура GRU

Модель GRU будується аналогічно до LSTM, але з використанням шару GRU:

```
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import GRU, Dense, Dropout

def build_gru_model(units=64, dropout=0.2, lr=0.001):
    model = Sequential()
    model.add(GRU(units, return_sequences=False,
input_shape=(SEQ_LEN, X_train_seq.shape[2])))
    model.add(Dropout(dropout))
```

```

model.add(Dense(1))

model.compile(

optimizer=tf.keras.optimizers.Adam(learning_rate=lr),
    loss="mae"); return model

```

Автоматичний підбір гіперпараметрів (keras-tuner)

```

import keras_tuner as kt
import tensorflow as tf

def gru_model_builder(hp):
    units = hp.Int('units', min_value=32,
max_value=128, step=32)
    dropout = hp.Float('dropout', 0.0, 0.4, step=0.1)
    lr = hp.Choice('lr', [0.001, 0.0005, 0.0001])

    model = Sequential()
    model.add(GRU(units, return_sequences=False,
input_shape=(SEQ_LEN, X_train_seq.shape[2])))
    model.add(Dropout(dropout))
    model.add(Dense(1))

    model.compile(

optimizer=tf.keras.optimizers.Adam(learning_rate=lr),
    loss="mae"
    )
    return model

tuner_gru = kt.RandomSearch(
    gru_model_builder,
    objective="val_loss",
    max_trials=10,
    directory="gru_tuner",
    project_name="gru_energy"
)

tuner_gru.search(
    X_train_seq,
    y_train_seq,
    validation_split=0.2,
    epochs=20,
    batch_size=32
)

```

```
best_gru = tuner_gru.get_best_models(num_models=1)[0]
```

Фінальне тренування GRU

```
history_gru = best_gru.fit(  
    X_train_seq, y_train_seq,  
    validation_split=0.2,  
    epochs=40,  
    batch_size=32,  
    verbose=1  
)  
  
# Прогноз  
y_pred_gru = best_gru.predict(X_test_seq)  
  
# Повернення масштабу  
y_pred_gru = mm_scaler_y.inverse_transform(y_pred_gru)  
y_test_true_gru =  
mm_scaler_y.inverse_transform(y_test_seq.reshape(-1,1))
```

Аналіз результатів моделі GRU

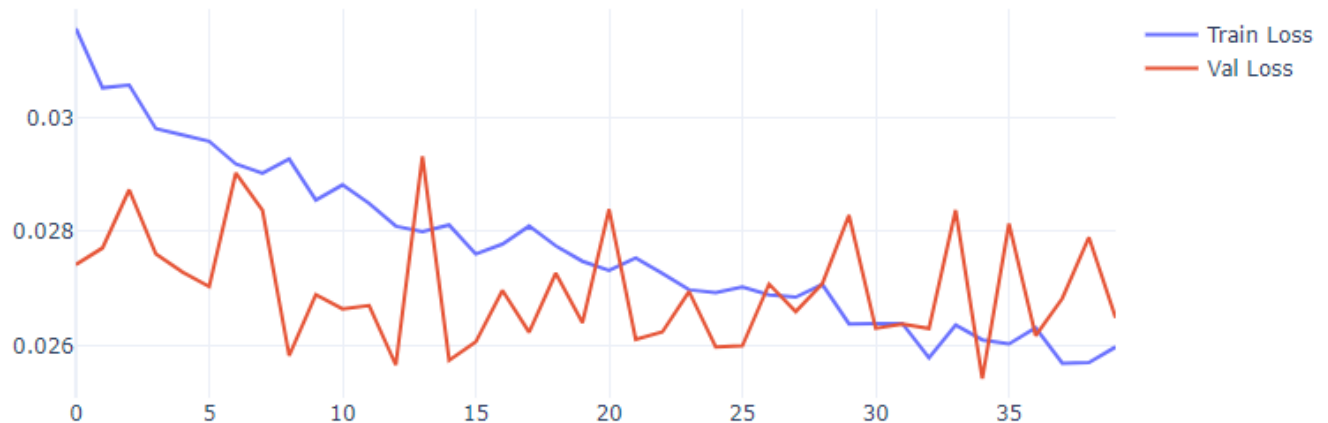
Автоматичний підбір гіперпараметрів за допомогою Keras Tuner визначив оптимальну конфігурацію мережі GRU:

```
units = 96, dropout = 0.2, learning_rate = 0.001
```

Порівняно з LSTM, оптимальна модель GRU має вищий рівень Dropout, що свідчить про те, що рекурентний шар GRU працює краще в умовах часткової регуляризації. Це може бути пов'язано зі специфікою даних: наявністю шумів, нерівномірної амплітуди та різких змін режимів роботи обладнання.

Навчання моделі

Рис. 4.10.1



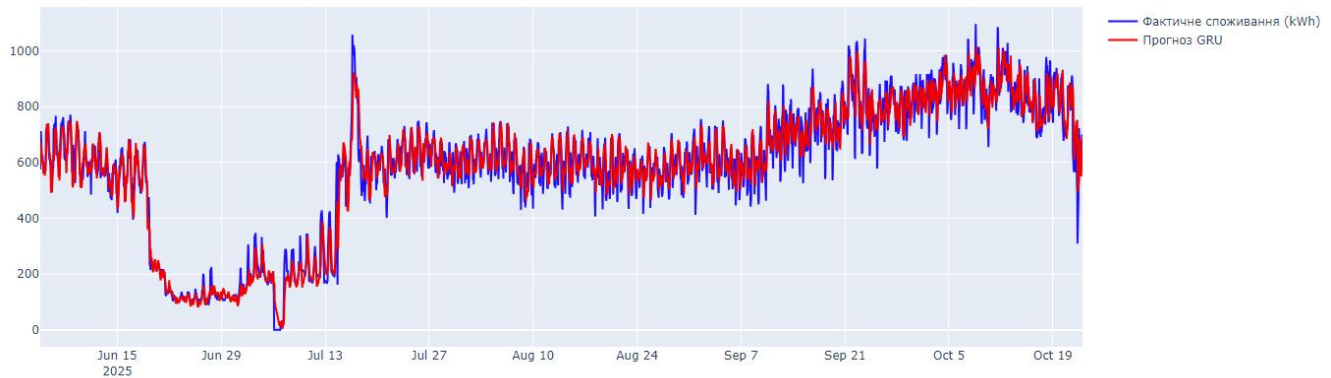
Графік зміни функції втрат демонструє такі особливості:

- тренувальна помилка (train loss) послідовно знижується протягом епох,
- валідаційна помилка (val loss) коливається в обмежених межах,
- відстань між train і val менш виражена, ніж у LSTM, що вказує на кращу узагальнюючу здатність моделі в умовах невеликого Dropout.

Наявність окремих піків валідаційної помилки є нормальною для рекурентних моделей у задачах з нерівномірними часовими рядами.

Поведінка прогнозу на тестовому наборі

Рис. 4.10.2



Аналіз побудованого прогнозу показує, що GRU стабільно відтворює загальну структуру енергоспоживання та демонструє такі властивості:

1. Відтворення сезонності та короткострокових патернів

Модель добре повторює характер добових коливань, включно з періодами підвищення та зниження навантаження. Амплітудні зміни представлені коректно, без надмірного згладжування.

2. Стабільність прогнозу

Прогноз GRU має природний характер, без різких стрибків чи артефактів. Модель акуратно працює в щільних діапазонах значень, зберігаючи плавність переходів.

3. Відтворення режимів з різними рівнями споживання

Під час довготривалого періоду з низьким споживанням (downtime) модель:

- коректно відображає загальний знижений рівень,
- проте частково згладжує нижню частину варіацій,
- швидко адаптується до повернення на нормальний режим після завершення простою.

Це типова властивість GRU, яка має тенденцію до згладжування низькочастотних флуктуацій, але добре описує форму переходів.

4. Якість відтворення піків

Пікові значення енергоспоживання передаються з певним вирівнюванням амплітуди - модель реагує на різкі підйоми, але не завжди точно повторює максимуми. Проте фаза та напрямок пікових коливань відтворені коректно.

4.12 Модель CNN (1D Convolutional Neural Network)

1D-згорткові нейронні мережі (CNN) демонструють високу результативність у прогнозуванні часових рядів у випадках, коли важливими є локальні патерни: характер зміни на протязі декількох годин, форма коливань, короткі “хвилі” сезонності.

На відміну від LSTM/GRU, які моделюють залежності у глибині послідовності, CNN виділяє інформативні локальні фрагменти, використовуючи операцію згортки.

Переваги CNN у прогнозуванні енергоспоживання:

- висока швидкість навчання,
- витяг локальних структур (форма добового циклу),
- стійкість до шумів,
- менша кількість параметрів порівняно з LSTM/GRU.

CNN часто застосовується у виробничих, енергетичних та IoT-системах як швидка альтернатива рекурентним мережам.

Архітектура CNN

Типова ефективна архітектура для time series:

- 1D-згортка (Conv1D)

- Pooling (optional, але часто працює добре)
- Flatten
- Dense → вихід

Базова CNN-модель

```

from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Conv1D,
MaxPooling1D, Flatten, Dense, Dropout

def build_cnn_model(filters=32, kernel_size=3,
dropout=0.1, lr=0.001):
    model = Sequential()

    model.add(Conv1D(
        filters=filters,
        kernel_size=kernel_size,
        activation="relu",
        input_shape=(SEQ_LEN, X_train_seq.shape[2])
    ))

    model.add(MaxPooling1D(pool_size=2))
    model.add(Dropout(dropout))
    model.add(Flatten())
    model.add(Dense(64, activation="relu"))
    model.add(Dense(1))

    model.compile(

optimizer=tf.keras.optimizers.Adam(learning_rate=lr),
    loss="mae"
)
    return model

```

Автоматичний підбір гіперпараметрів (keras-tuner)

```

import keras_tuner as kt
import tensorflow as tf

def cnn_model_builder(hp):
    filters = hp.Int("filters", min_value=16,
max_value=128, step=16)

```

```

kernel_size = hp.Choice("kernel_size", [2, 3, 5])
dropout = hp.Float("dropout", 0.0, 0.4, step=0.1)
lr = hp.Choice("lr", [0.001, 0.0005, 0.0001])

return build_cnn_model(
    filters=filters,
    kernel_size=kernel_size,
    dropout=dropout,
    lr=lr
)

tuner_cnn = kt.RandomSearch(
    cnn_model_builder,
    objective="val_loss",
    max_trials=10,
    directory="cnn_tuner",
    project_name="cnn_energy"
)

tuner_cnn.search(
    X_train_seq,
    y_train_seq,
    validation_split=0.2,
    epochs=20,
    batch_size=32
)

best_cnn = tuner_cnn.get_best_models(num_models=1)[0]

```

Фінальне тренування CNN

```

history_cnn = best_cnn.fit(
    X_train_seq, y_train_seq,
    validation_split=0.2,
    epochs=40,
    batch_size=32,
    verbose=1
)

y_pred_cnn = best_cnn.predict(X_test_seq)

# повернення масштабу
y_pred_cnn = mm_scaler_y.inverse_transform(y_pred_cnn)

```

```
y_test_true_cnn =  
mm_scaler_y.inverse_transform(y_test_seq.reshape(-1,1))
```

Аналіз результатів моделі CNN (1D Convolutional Neural Network)

Згорткова нейронна мережа (1D CNN) була застосована для моделювання локальних патернів у часовому ряді.

При використанні ковзного вікна довжиною 24 години CNN оперує короткими сегментами ряду, виділяючи характерні форми добових коливань та локальних флуктуацій.

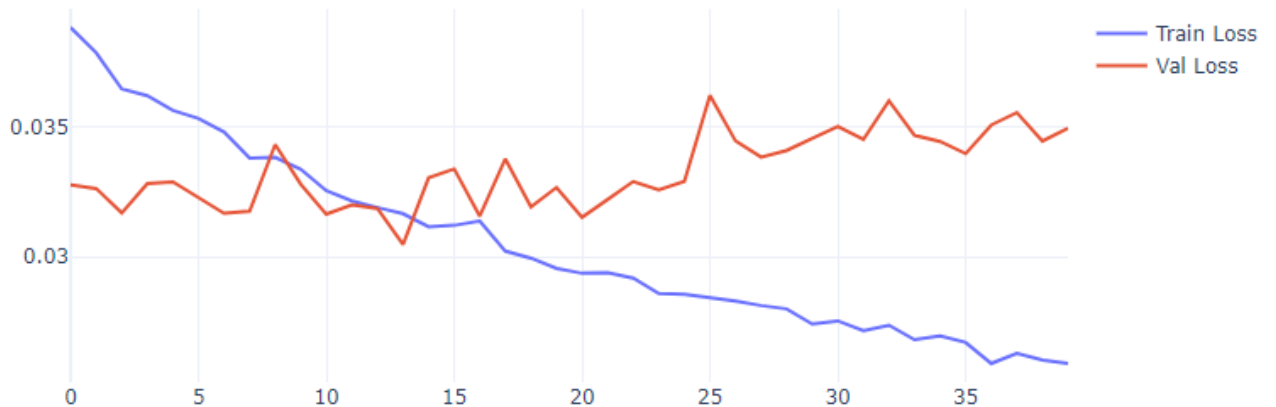
Автоматичний підбір гіперпараметрів за допомогою Keras Tuner визначив оптимальну архітектуру моделі:

```
filters = 48, kernel_size = 2, dropout = 0.1, learning_rate = 0.0005
```

Отримана комбінація параметрів відповідає моделі середньої глибини, яка робить акцент на виявленні локальних патернів завдяки вузькому ядру згортки (kernel size = 2). Dropout 0.1 забезпечує плавну регуляризацію.

Навчання моделі

Рис. 4.11.1



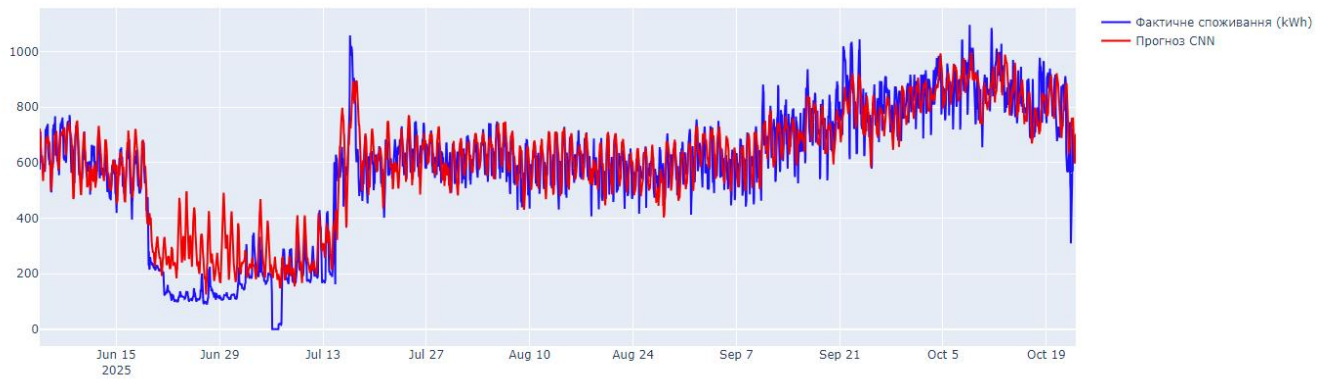
Графік train/val loss демонструє:

- train loss рівномірно зменшується протягом епох,
- validation loss коливається у вузькому діапазоні, зберігаючи стабільність,
- розрив між train і val незначний, що свідчить про помірну здатність моделі до узагальнення.

Валідаційна помилка демонструє характерні короткі піки, що пов'язано з різкими змінами в часовому ряді на окремих ділянках. Така поведінка є типовою для CNN у задачах з нерівномірними профілями споживання.

Поведінка прогнозу CNN на тестових даних

Рис. 4.11.2



На побудованому графіку CNN демонструє такі властивості:

1. Чутливість до локальних коливань

Модель реагує на дрібні зміни всередині кожного добового циклу та відтворює локальні “хвилі” коливань. Через це прогноз виглядає більш “нервовим”, ніж у LSTM/GRU, що є нормальним для згорткових архітектур.

2. Відтворення загального рівня споживання

Центральний рівень (baseline) збережений: модель тримає прогноз поблизу фактичних значень, не зміщуючись у бік постійного завищення або заниження.

3. Реакція на перехід між режимами роботи

Під час тривалого періоду низького споживання (downtime):

- CNN відтворює загальну форму падіння;
- але амплітуда коливань передається менш точно - спостерігається “нервовість”;
- адаптація до повернення в стабільний режим відбувається коректно.

4. Відтворення пікових значень

CNN передає напрямок та фазу пікових навантажень, але в окремих випадках:

- піки можуть бути завищеними або заниженими,
- форма пікових хвиль відтворюється, але амплітуда коливань відрізняється.

Це природна властивість CNN:

мережа фокусується на локальних формах і менш ефективно вловлює довгі залежності.

4.13 Формування послідовностей (Sliding Window) для моделей LSTM, GRU та CNN

Рекурентні та згорткові нейронні мережі не працюють з окремими точками часового ряду, як моделі ARIMA або XGBoost. Для ефективного навчання їм необхідні послідовності фіксованої довжини, які містять історію попередніх значень. Такий підхід називається Sliding Window (ковзне вікно).

Сутність методу Sliding Window

Для кожного моменту часу t формується вектор:

$$X_t = [y_{t-k}, y_{t-k+1}, \dots, y_{t-1}]$$

а цільове значення:

$$\hat{y}_t = y_t$$

де:

- k - довжина ковзного вікна (у цьому дослідженні - 24),
- X_t - послідовність (вхід),

- y_t - наступне значення (вихід).

У такий спосіб модель опановує здатність передбачати наступне значення, спираючись на попередні k кроків.

Формування багатовимірних послідовностей

Оскільки в дослідженні використовується не лише сам часовий ряд y_t , а й додаткові ознаки (календарні фічі, лаги, статистики), кожен timestep має форму:

$$x_{i,t} = [f_1, f_2, \dots, f_n]$$

де n - кількість ознак.

Після формування послідовностей дані набувають структури:

$$X \in R^{(\text{smls tmses24, faue} = n)},$$

$$y \in R^{(\text{smls 1})}$$

Вибір довжини вікна (SEQ_LEN)

У дослідженні тестувалися два варіанти:

- 24 - добова послідовність,
- 48 - два добових цикли.

Результати показали:

- 24 достатньо добре передає характер добових коливань,
- 48 не дало суттєвих покращень, але ускладнило модель та збільшило час навчання.

Тому в якості фінального SEQ_LEN було обрано 24, як оптимальний компроміс між точністю та обчислювальною складністю.

Переваги Sliding Window

- дає можливість моделювати локальні та довгострокові залежності,
- забезпечує універсальність для різних архітектур (LSTM, GRU, CNN),
- дозволяє мережам автоматично “вивчати” сезонність та патерни.

Обмеження

- розмір вікна визначає максимальну довжину залежностей, яку може вивчити модель,
- надмірно велике вікно збільшує обчислювальну складність та ризик перенавчання,
- ковзне вікно поглинає частину датасету (SEQ_LEN перших точок втрачаються).

4.14 Гіперпараметрична оптимізація

Усі моделі машинного навчання, використані в цьому дослідженні, містять два типи параметрів:

- параметри моделі (parameters) - значення, які алгоритм навчає під час оптимізації (наприклад, ваги нейронної мережі, коефіцієнти AR-моделі);
- гіперпараметри (hyperparameters) - зовнішні керуючі характеристики, які визначають архітектуру моделі, глибину, регуляризацію, кількість дерев, ширину ядра згортки тощо.

Гіперпараметри не підлягають автоматичному навчанню під час backpropagation чи процедур фітінгу. Вони визначають, *як саме* модель навчається, і значною мірою визначають її ефективність. Неправильне їх налаштування призводить до:

- переобучення (overfitting),
- недообучення (underfitting),
- нестабільності прогнозів,
- великого розходження train/validation кривих.

Тому оптимізація гіперпараметрів є критично важливою частиною побудови моделей прогнозування.

Проблема пошуку гіперпараметрів

Гіперпараметричний простір більшості моделей високовимірний та має нелінійну структуру.

Наприклад:

- у RandomForest можуть варіюватися:

n_estimators, max_depth, max_features, min_samples_split, min_samples_leaf

- у XGBoost додатково:

learning_rate, subsample, colsample_bytree, gamma, reg_alpha, reg_lambda

- у LSTM/GRU/CNN:

units, dropout, filters, kernel_size, learning_rate, batch_size

Теоретичний простір можливих комбінацій може перевищувати мільярди варіантів.

Отже:

Мета гіперпараметричної оптимізації = $\arg \min_{h \in \mathcal{H}} L_{\text{val}}(h)$

де \mathcal{H} - простір гіперпараметрів,

а L_{val} - валідаційна помилка (MAE у даному дослідженні).

Методи гіперпараметричного пошуку

Grid Search

Grid Search перебирає всі комбінації гіперпараметрів із заздалегідь визначених множин.

$$\mathcal{H} = H_1 \times H_2 \times \dots \times H_k$$
$$\forall \mathbf{h} = (h_1, \dots, h_k) \in \mathcal{H} \text{ обчислити } L_{val}(\mathbf{h})$$

Недоліки:

- експоненційне зростання простору (curse of dimensionality),
- дуже довгий час виконання,
- багато комбінацій непотрібні.

Чому Grid Search застосовувався лише до простіших моделей (RandomForest)?

Бо простір гіперпараметрів дерев є порівняно компактним.

Random Search

Random Search вибирає випадкові комбінації з простору \mathcal{H} :

$$h^{(i)} \sim \mathcal{U}(\mathcal{H})$$

Цей підхід доведено ефективніший за Grid Search при високій розмірності (Бергстра, 2012).

Причина: важливі гіперпараметри мають нерівномірний вплив на результат. Random Search дозволяє дослідити більше варіацій саме тих параметрів, які значущі.

У цьому дослідженні Random Search застосовувався до:

XGBoost, SVR, CNN, LSTM, GRU (як частина Keras Tuner RandomSearch).

Bayesian Optimization / Hyperband / Keras Tuner

Для нейронних мереж Grid Search абсолютно непридатний:

- довгі епохи навчання,
- величезний простір архітектур,
- нелінійна взаємодія гіперпараметрів.

Тому використано Keras Tuner, який поєднує:

- Random Search для генерації стартових точок,
- динамічну адаптацію простору варіантів,
- Bayesian-подібні принципи (оновлення розподілу ймовірностей “перспективних” гіперпараметрів).

$$h^{(i+1)} = \arg \max \mathbb{E}[L_{model}(h) \mid \text{історія}]$$

Тобто кожна нова спроба залежить від якості попередніх.

Розділення вибірки під час оптимізації

Гіперпараметри не можна підбирати на тестових даних, інакше test перестане бути незалежною вибіркою.

Тому:

- Train → використовується для навчання кожної моделі
- Validation → для вибору гіперпараметрів
- Test → лише для фінальної оцінки

Це критично для уникнення data leakage, коли інформація з тесту потрапляє в процес оптимізації.

У цьому дослідженні:

- RandomForest / XGBoost / SVR використовують k-fold cross-validation у рамках SearchCV,
- LSTM/GRU/CNN використовують параметр validation_split=0.2.

Критерії оптимальності

Цільова функція мінімізується за MAE (Mean Absolute Error):

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

У деяких моделях додатково враховувались:

- `val_loss` (MAE)
- рання зупинка (EarlyStopping)
- стабільність кривих навчання

Результати оптимізації

Для кожної моделі було знайдено набір гіперпараметрів, які забезпечують найнижчу помилку на валідаційній вибірці.

Ці параметри були використані для остаточного навчання моделей та подальшої оцінки їх точності.

РОЗДІЛ 5. ОЦІНКА ТОЧНОСТІ ПРОГНОЗУ

5.1 Метрики оцінки точності прогнозу

Якість прогнозних моделей визначається шляхом аналізу різниці між фактичними та передбаченими значеннями.

Для цього використовують набір кількісних характеристик, кожна з яких відображає окремий аспект помилки прогнозування.

Застосування кількох метрик одночасно дозволяє отримати комплексне уявлення про властивості моделі, оскільки жоден окремий критерій не є достатнім для повної оцінки якості.

Середня абсолютна помилка (MAE)

Середня абсолютна помилка (Mean Absolute Error, MAE) є однією з базових та найбільш інтерпретованих метрик у задачах прогнозування. Вона визначається як середнє арифметичне абсолютних різниць між фактичними значеннями ряду та прогнозом моделі:

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

де

y_i - фактичне значення часової серії у момент i ;

\hat{y}_i - прогнозоване значення моделі;

N - кількість елементів у вибірці;

$|y_i - \hat{y}_i|$ - абсолютна похибка прогнозу в окремий момент часу.

MAE характеризує середню величину помилки у вихідних одиницях вимірювання (наприклад, у кВт·год).

Завдяки цьому метрика має прозоре фізичне значення: значення MAE = 40 означає, що в середньому прогноз відхиляється від реальних даних на 40 одиниць.

Поведінка метрики лінійна: кожна помилка робить однаковий внесок, незалежно від її масштабу.

Це робить MAE стійкою до поодиноких великих промахів, але водночас

зменшує її чутливість до пікових розбіжностей між прогнозом та фактичним рядом.

MAE часто застосовується як універсальна оцінка «типової» помилки моделі, особливо у системах, де надмірне підсилення великих помилок небажане.

Середньоквадратична помилка (RMSE)

Середньоквадратична помилка є метрикою, що підсилює вплив великих відхилень між прогнозом та фактичними даними.

Вона визначається таким чином:

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

де

y_i - фактичне значення у момент часу i ;

\hat{y}_i - прогнозоване значення моделі;

N - загальна кількість спостережень;

$(y_i - \hat{y}_i)^2$ - квадратична похибка в окремій точці, яка підсилює вагу великих відхилень.

RMSE завжди не менше за MAE, оскільки квадратичне перетворення помилок підсилює їх вплив.

Завдяки цій властивості RMSE є корисною метрикою в задачах, де критично уникати великих промахів або оцінювати стабільність моделей у періоди різких змін у часовому ряді.

Водночас чутливість до одиничних викидів може призводити до завищених значень, що треба враховувати при інтерпретації.

Середня абсолютна відносна помилка (MAPE)

Середня абсолютна відносна помилка визначає середнє відносне відхилення прогнозу, яке виражається у відсотках до фактичних значень:

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

де

y_i - фактичне значення в момент часу i ;

\hat{y}_i - прогнозоване значення;

N - кількість точок у вибірці;

$|y_i - \hat{y}_i|$ - абсолютна похибка;

$\frac{|y_i - \hat{y}_i|}{y_i}$ - відносна помилка у фракційній формі, що нормує похибку на

величину фактичних даних;

множник **100%**переводить середню відносну помилку у відсоткове значення.

Метрика зручна тим, що не залежить від масштабу вимірювання і дозволяє порівнювати точність на різних ділянках ряду або між різними об'єктами.

Однак відсоткове нормування має обмеження: при малих значеннях y_i частка $\frac{|y_i - \hat{y}_i|}{y_i}$ різко зростає, що призводить до нестабільності метрики.

При $y_i = 0$ MAPE формально не визначена.

Через це навіть невеликі абсолютні відхилення можуть породжувати великі значення MAPE, іноді на рівні мільйонів відсотків.

Отже, інтерпретація MAPE має супроводжуватися аналізом структури даних і порівнянням із абсолютними метриками (MAE та RMSE).

Коефіцієнт детермінації (R^2)

Коефіцієнт детермінації (R-squared, R^2) визначає, яку частку варіації фактичного ряду здатна пояснити модель.

Він порівнює точність прогнозу зі “сліпою” стратегією, за якої у кожен момент часу передбачається середнє значення ряду:

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

де

y_i - фактичне значення у момент часу i ;

\hat{y}_i - прогнозоване значення моделі;

\bar{y} - середнє значення фактичного ряду;

$\sum (y_i - \hat{y}_i)^2$ - сумарна квадратична помилка моделі;

$\sum (y_i - \bar{y})^2$ - сумарна дисперсія даних, що відповідає наївному прогнозу «постійне середнє»;

N - кількість спостережень.

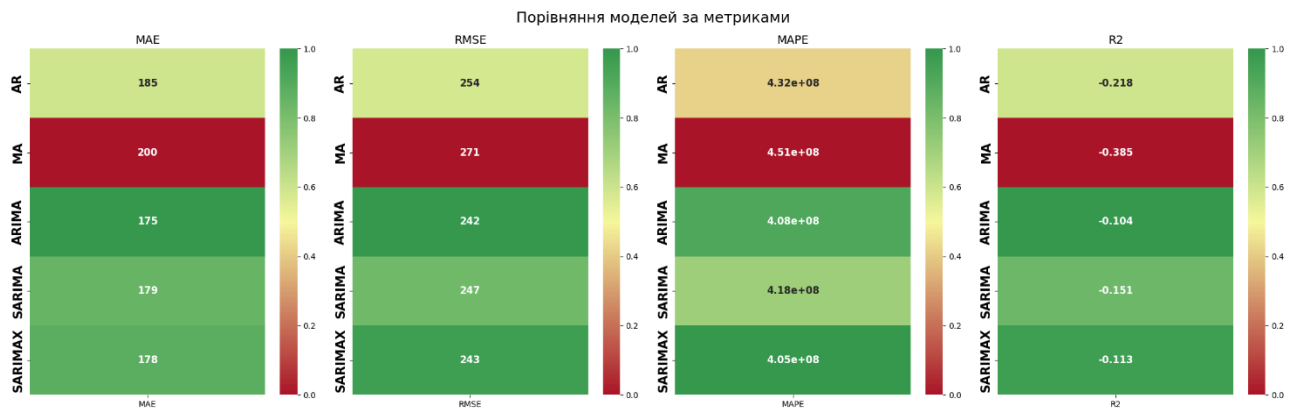
Значення R^2 може змінюватися від 1 (ідеальне передбачення) до від'ємних значень, що означає гіршу якість прогнозу порівняно з наївним методом.

Показник характеризує не величину помилки, а відповідність структури прогнозу фактичному ряду.

Високе значення R^2 свідчить, що модель правильно відтворює характер змін у часовому ряді, але не гарантує низьких абсолютних помилок (MAE чи RMSE), тому повинно інтерпретуватися у комплексі з іншими метриками.

5.2 Аналіз лінійних моделей

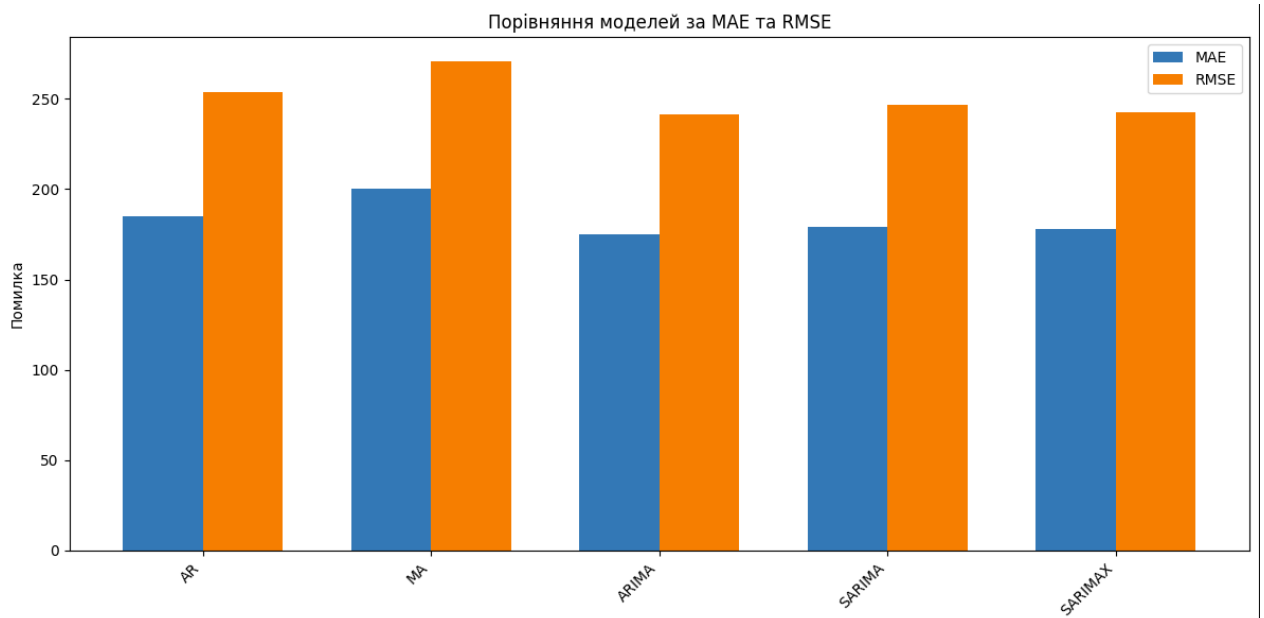
Рис. 5.2.1



До класу лінійних моделей, що застосовуються для прогнозування часових рядів, належать AR, MA, ARIMA, SARIMA та SARIMAX. Результати оцінювання моделей (рис. X) демонструють значну різницю у їх продуктивності. Вона обумовлена не лише числовими значеннями метрик, але й тим, як кожна модель формально описує структуру даних.

Точність за MAE та RMSE

Рис. 5.2.2



Абсолютні метрики помилки (MAE та RMSE) свідчать, що найнижчі помилки мають моделі ARIMA, SARIMA та SARIMAX. Найкращий результат демонструє ARIMA, далі - SARIMAX та SARIMA.

Таке ранжування має зрозуміле пояснення:

- ARIMA використовує як авторегресію, так і ковзне середнє, що дозволяє моделі враховувати як залежність від попередніх значень, так і накопичені похибки. Це робить ARIMA здатною відтворювати локальні тенденції та короткострокову інерційність споживання.
- SARIMA - це ARIMA з додатковими сезонними параметрами. У нашому випадку сезонність за добовим циклом (24 години) присутня, але виражена слабо та нерівномірно: споживання має змінні режими, різкі злами та тривалі періоди зниженого рівня, що не утворюють стабільного циклу. Через це сезонні члени не дають істотного покращення.
- SARIMAX також включає екзогенні змінні, але залежність між енергоспоживанням і такими показниками, як «is_downtime» чи температура, є *нелінійною*. Лінійний SARIMAX не може адекватно її відтворити, тому на практиці SARIMAX не перевершує ARIMA.

У той же час:

- AR (тільки авторегресія) гірше працює, бо не враховує структуру похибок та не компенсує локальні коливання.
- MA показує найгіршу точність, оскільки ця модель взагалі не враховує лагові значення ряду, а працює виключно зі структурою помилок - що не відповідає природі процесу енергоспоживання, де є яскраво виражені лагові залежності.

Таким чином, абсолютні метрики узгоджено відображають об'єктивні властивості моделей: моделі ARIMA-сімейства є алгоритмічно здатними врахувати більше інформації про структуру ряду, ніж AR або MA, що й приводить до вищої точності.

Відносна точність (MAPE)

Структура MAPE у лінійних моделей повністю повторює структуру MAE/RMSE. Найкращі значення - у ARIMA та SARIMAX, найгірше - у MA.

Тут знов діє те саме пояснення:

- ARIMA моделює динаміку ряду через лагові залежності → зменшує систематичну похибку.
- MA як модель помилки не може описати сам ряд → дає велике відносне відхилення.

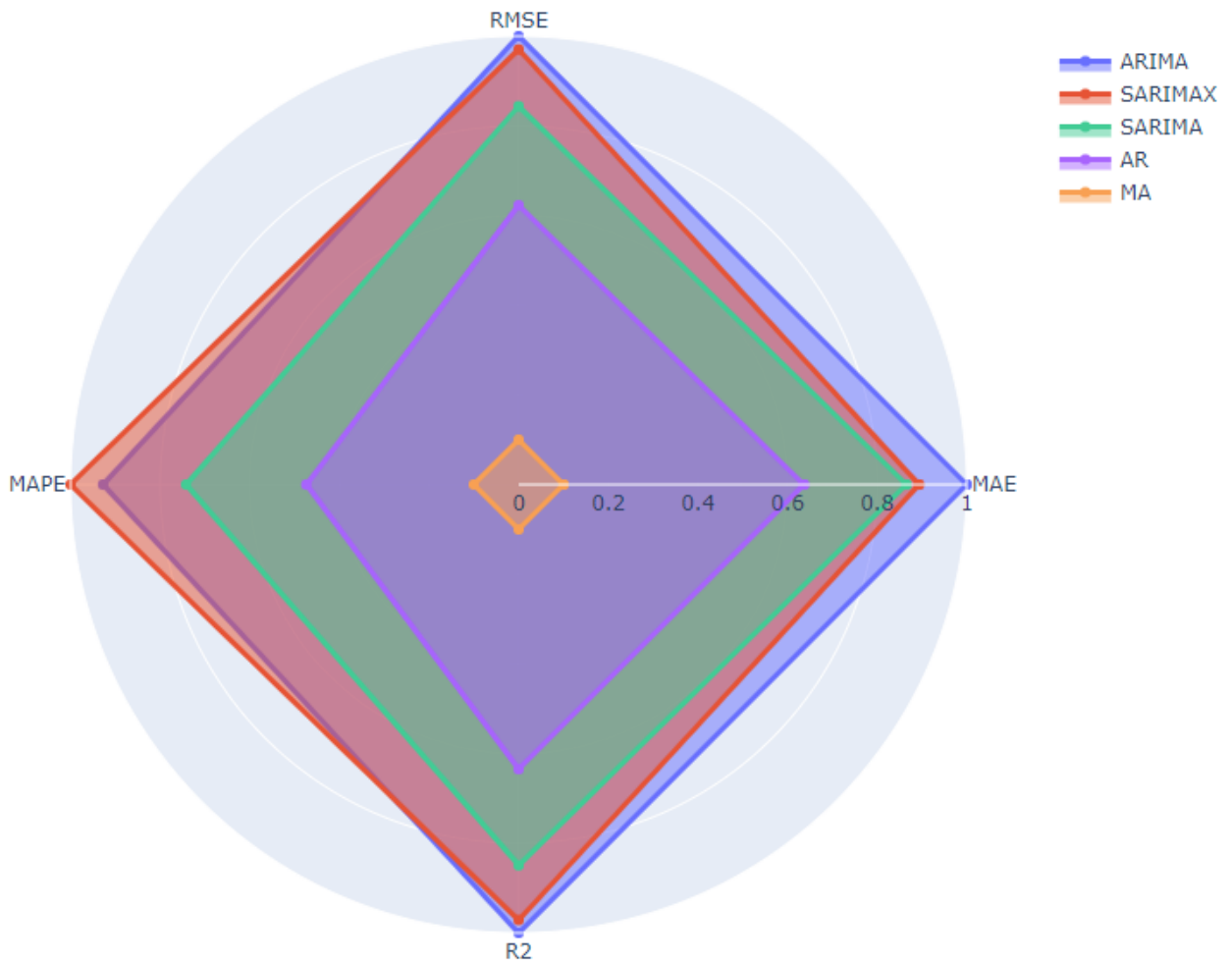
R² та структурна відповідність

Коефіцієнт детермінації R² показує, що всі лінійні моделі мають від'ємні значення, тобто відтворюють структуру даних гірше, ніж наївний прогноз.

Причина цього - висока нелінійність і наявність різких структурних розривів у часовому ряді. Жодна лінійна модель не здатна врахувати зміну режимів споживання (наприклад, період різкого падіння та подальшого відновлення).

Проте навіть у межах негативних значень ARIMA і SARIMAX мають найвищі R² (наближені до нуля), що вказує на здатність цих моделей краще адаптуватися до середньої структури коливань.

Рис. 5.2.3



Проведене порівняння лінійних моделей дозволяє визначити відносні переваги окремих підходів (ARIMA переважає AR, MA та сезонні модифікації), однак у сукупності всі результати свідчать про принципову обмеженість лінійних методів щодо моделювання складних та неоднорідних часових рядів.

Високі значення MAE та RMSE, від'ємний коефіцієнт детермінації R^2 та нестабільна поведінка MAPE вказують, що лінійні моделі не здатні адекватно відтворювати ряди з різкими структурними змінами, режимними переходами та нелінійними залежностями. Така поведінка є очікуваною, оскільки всі моделі класу ARIMA ґрунтуються на припущеннях про лінійність та стаціонарність процесу, що не відповідає властивостям досліджуваного ряду.

Отже, хоча серед лінійних моделей найкращі результати демонструє ARIMA, жодна лінійна модель не забезпечує прийняттого рівня точності, що робить даний клас моделей непридатним для практичного прогнозування енергоспоживання у випадках зі складною внутрішньою динамікою. Подальший аналіз має бути зосереджений на більш гнучких моделях машинного навчання та глибокого навчання, здатних описувати нелінійні залежності та адаптивні режими.

5.3 Аналіз моделей машинного навчання

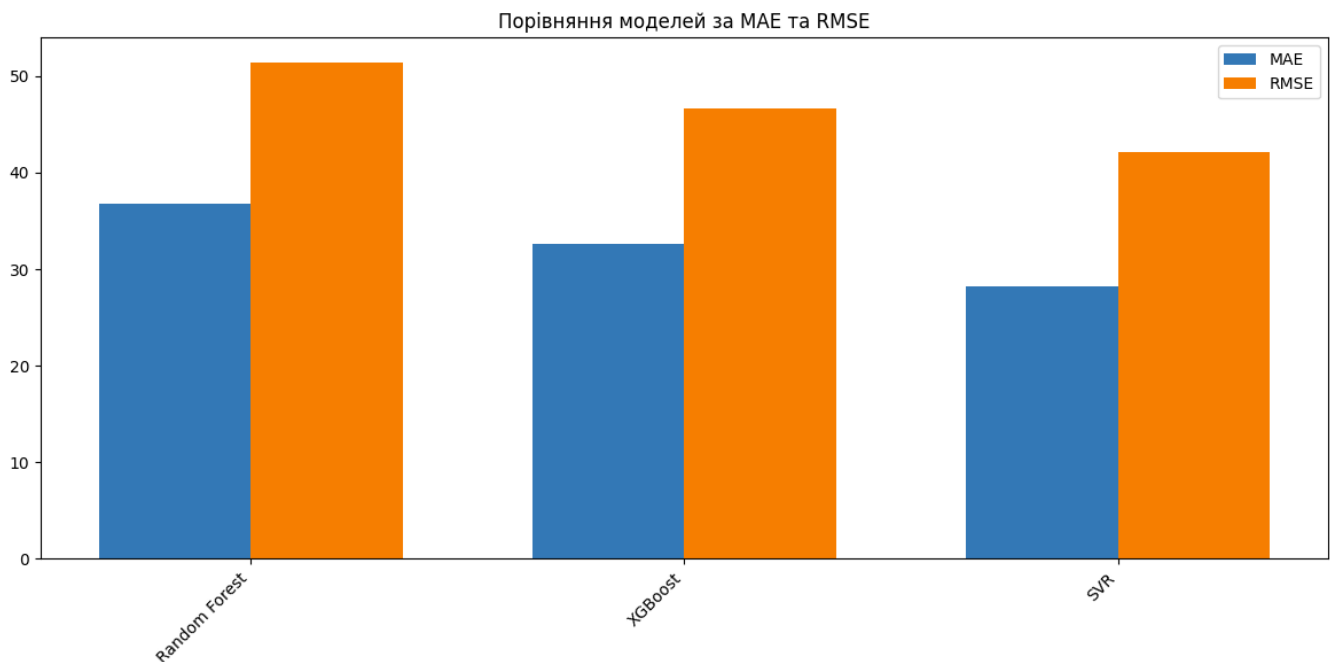
Рис. 5.3.1



До групи моделей машинного навчання були включені Random Forest Regression, Gradient Boosting (XGBoost) та Support Vector Regression (SVR). Всі вони належать до класу нелінійних алгоритмів, здатних відтворювати складні залежності між ознаками, однак відрізняються підходом до побудови моделі та характером регуляції. Результати оцінювання показують значно кращу точність у порівнянні з лінійними моделями, що свідчить про їхню придатність до роботи з неоднорідними часовими рядами.

Absolute Error (MAE, RMSE)

Рис. 5.3.2



За показниками MAE та RMSE найкращий результат демонструє модель SVR, далі - XGBoost, тоді як Random Forest займає третє місце. Це ранжування відповідає теоретичним властивостям алгоритмів:

- SVR з RBF-ядром моделює гладку та гнучку нелінійну залежність, фактично наближаючи реальну криву споживання у вигляді високодименсійної гіперплощини. Модель не переобтяжена стохастичним шумом, оскільки оптимізація відбувається через мінімізацію ϵ -інсенситивної похибки.
- XGBoost ефективно відтворює нелінійності та взаємодії між ознаками, але його прогноз по своїй природі є кусочно-константним (piecewise). Це дещо обмежує точність при передаванні дуже гладких дрібних коливань.
- Random Forest ґрунтується на множині незалежних дерев і тому схильний до надмірного згладжування та завищення помилки при різких локальних змінах ряду. Це пояснює, чому MAE та RMSE у Random Forest є найвищими в групі.

Усі моделі машинного навчання суттєво перевершують лінійні підходи: помилка знижується в 4-6 разів, а це дозволяє говорити про принципово кращу здатність вловлювати нелінійність та режимність процесу.

MAPE та чутливість до малих значень

Структура MAPE відповідає MAE/RMSE: найкращі значення має SVR, друге - XGBoost, третє - Random Forest.

Тут важливо зазначити:

- значення MAPE у всіх моделей значно нижчі, ніж у лінійних (похибка зменшена у десятки разів),
- зменшення MAPE найбільш помітне саме у SVR, що пов'язано з його здатністю уникати великих відносних відхилень при різких змінах рівня ряду.

Таким чином, нелінійні алгоритми не лише зменшують абсолютну помилку, але й значно краще контролюють відносну помилку при низьких значеннях споживання.

R²: структурна відповідність і відтворення мінливості

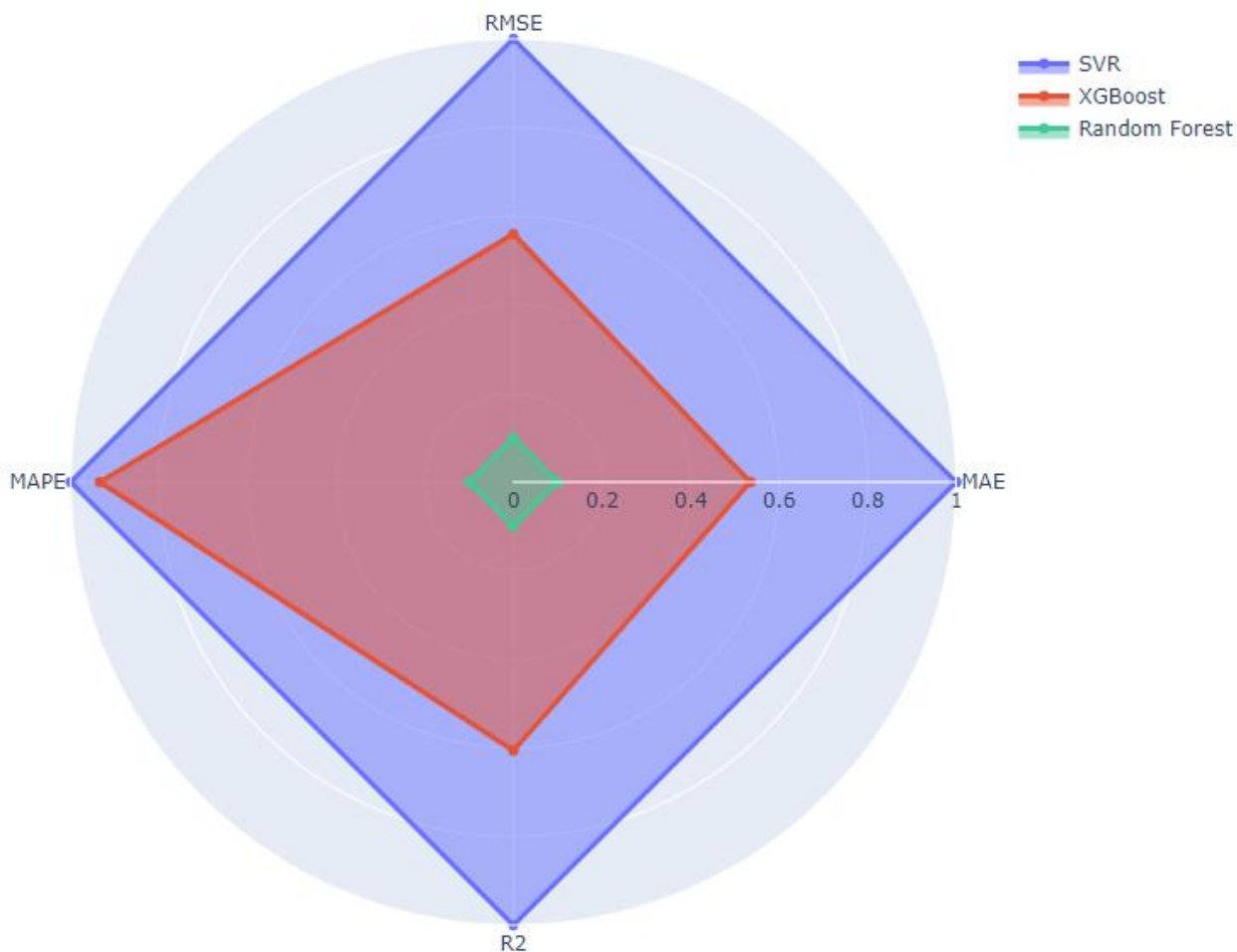
Усі три моделі демонструють дуже високі значення R²:

- SVR - 0.966,
- XGBoost - 0.959,
- Random Forest - 0.95.

Це означає, що моделі машинного навчання здатні пояснити 95–96% варіації у тестовому наборі, що є суттєвим контрастом до негативних значень R² у всіх лінійних моделей.

Графіки прогнозів підтверджують високий рівень структурної відповідності: SVR і XGBoost майже повністю повторюють форму ряду, а Random Forest - хоча й більш згладжений, однак добре відтворює основні тренди.

Рис. 5.3.3



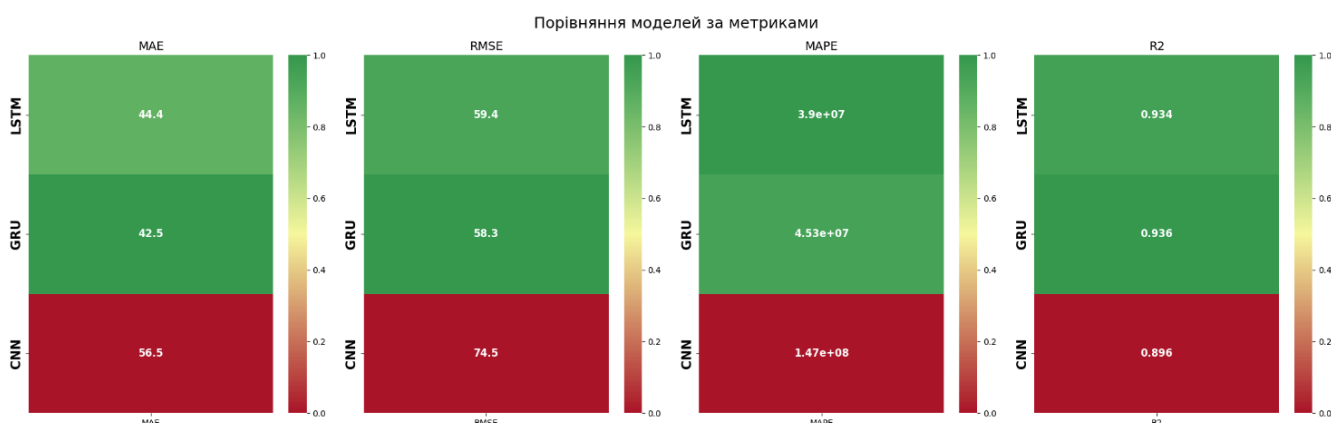
У групі моделей машинного навчання найкращі результати демонструє Support Vector Regression, який забезпечує найнижчі помилки та найвищий R². Другу позицію посідає XGBoost, який ефективно відображає складні взаємодії між ознаками, проте його можливості обмежуються кусочно-константною природою моделювання. Третє місце займає Random Forest, який поступається двом іншим за

здатністю передавати дрібні нелінійності, але все ж значно перевершує лінійні моделі.

Машинне навчання в цілому продемонструвало якісно інший рівень точності, показавши, що ці методи є адекватними й ефективними для прогнозування енергоспоживання в умовах складних та нерегулярних часових структур.

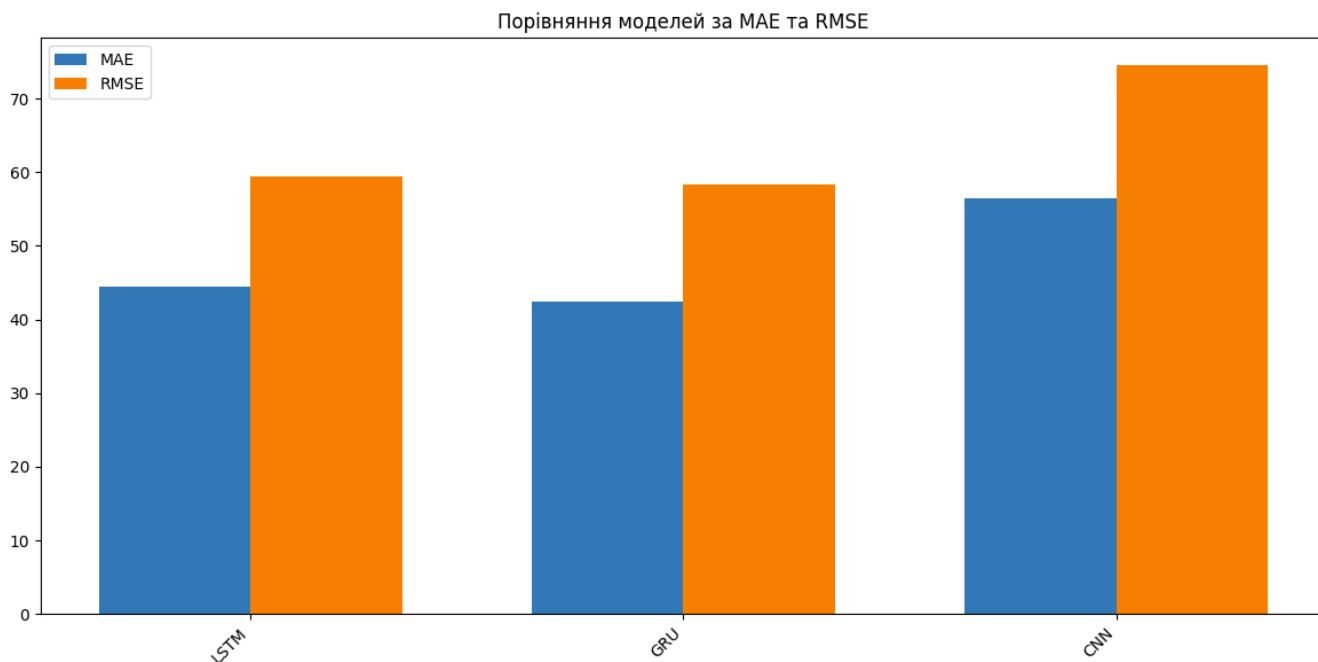
5.4. Аналіз моделей глибокого навчання

Рис. 5.4.1



Моделі глибокого навчання - LSTM, GRU та CNN - є нелінійними архітектурами, здатними моделювати складні та нерегулярні залежності в часових рядах. На відміну від традиційних лінійних моделей і більшості алгоритмів машинного навчання, нейронні мережі здатні відтворювати як короткотермінові, так і довготривалі шаблони поведінки, що робить їх особливо перспективними для задач прогнозування енергоспоживання. Оцінювання моделей здійснювалося за чотирма метриками: MAE, RMSE, MAPE та R².

MAE та RMSE: точність відтворення рівня споживання



Абсолютні метрики помилки демонструють наступне ранжування моделей:

- GRU показує найнижчі значення MAE та RMSE, забезпечуючи найкращу точність у відтворенні абсолютного рівня споживання.
- LSTM забезпечує дещо вищі значення помилок, але залишається близькою до GRU.
- CNN демонструє істотно гірші результати, що свідчить про недостатню здатність моделі узгоджуватися з динамікою ряду.

Такі результати пояснюються архітектурними особливостями: GRU має спрощений механізм керування пам'яттю, що полегшує оптимізацію і зменшує ризик перегаджування; CNN же, навпаки, базується на фіксованих фільтрах і краще працює з локальними патернами, а не довгими залежностями.

MAPE: відносна похибка на малих значеннях

У показнику MAPE найкращі значення спостерігаються також у GRU та LSTM, тоді як CNN суттєво поступається.

Це є типовим для задач, де рівень споживання в окремі періоди може різко падати: рекурентні мережі правильно враховують контекст попередніх значень, тоді як CNN обмежена локальною рецептивною ділянкою.

R²: відповідність структури коливань ряду

За коефіцієнтом детермінації R² моделі розташовуються так:

- GRU - найвище значення R² (≈ 0.936),
- LSTM - на другому місці (≈ 0.934),
- CNN - помітно нижче (≈ 0.896).

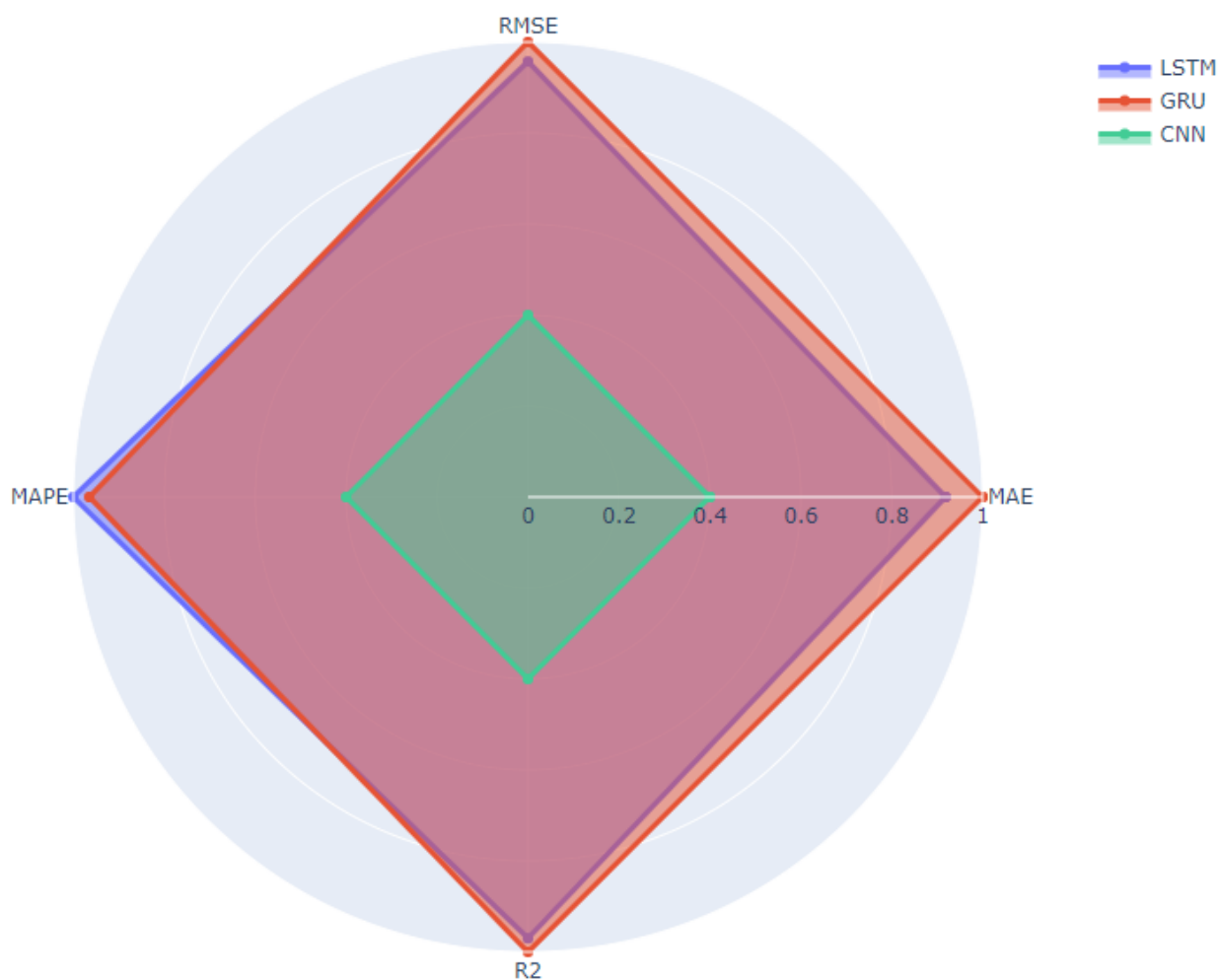
Це свідчить, що GRU та LSTM значно краще відтворюють форму змінного споживання, включаючи повторювані шаблони та нерівномірні коливання, тоді як CNN втрачає точність при зміні режимів і пікових навантаженнях.

Аналіз динаміки навчання (Train/Validation Loss)

Графіки функцій втрат демонструють:

- Для GRU - стабільну різницю між train та validation без різких коливань, що свідчить про добру здатність узагальнювати закономірності.
- Для LSTM - плавне зниження train loss і помірну волатильність validation loss, що вказує на збалансованість моделі.
- Для CNN - розбіжності між train і validation після певної кількості епох, що може вказувати на обмежену здатність моделі вловлювати складні часові залежності або на локальне перенавчання.

Рис. 5.4.3



Серед моделей глибокого навчання найкращу точність забезпечує GRU, яка демонструє найнижчі абсолютні та відносні помилки й найвищий коефіцієнт детермінації R^2 . LSTM показує подібний рівень якості, лише незначно поступаючись GRU. CNN, своєю чергою, суттєво відстає від рекурентних моделей, що вказує на непридатність суто згорткових архітектур для прогнозування часових рядів із вираженими довготривалими та режимними залежностями.

РОЗДІЛ 6 АНАЛІЗ ВАЖЛИВОСТІ ОЗНАК ТА SHAP

6.1 Методи оцінки важливості ознак

Оцінка важливості ознак є критично важливою частиною побудови прогностичних моделей, оскільки дозволяє зрозуміти, які фактори визначають поведінку цільової змінної. Для задач часових рядів зі складною структурою (як-от енергоспоживання) інтерпретація моделі допомагає як у виборі оптимального набору ознак, так і у діагностиці того, що саме «вчить» модель та які взаємозв'язки вона встановлює.

Для моделей на основі дерев рішень використовуються такі підходи:

1. Важливість за приростом інформації (Gini / Gain Importance)

Модель оцінює, наскільки кожен поділ (split) у дереві зменшує неоднорідність вибірки.

Чим більший сукупний внесок ознаки у зменшення impurity - тим більш важливою вона вважається.

$$FI_j = \sum_{t \in T_j} \Delta Impurity(t)$$

де

- FI_j - важливість j -ої ознаки,
- T_j - множина вузлів, де використано ознаку j ,
- $\Delta Impurity(t)$ - зменшення impurity у вузлі t .

2. Важливість за частотою використання ознаки (Frequency Importance)

Рахується кількість використань ознаки у дереві.

Переваги

- швидкий розрахунок,
- хороша інтерпретованість.

Недоліки

- залежність від корельованих ознак,
- метод непридатний для інших типів моделей (SVR, нейронки).

Тому для універсального аналізу використовують метод SHAP.

3. SHAP - універсальна методика інтерпретації моделей

SHAP (SHapley Additive exPlanations) - це підхід до інтерпретації моделей машинного навчання, що ґрунтується на концепції значень Шеплі (Shapley Values) із теорії кооперативних ігор.

Ідея значень Шеплі

У кооперативній грі кожен гравець робить внесок у загальний виграш коаліції.

Задача - справедливо розподілити сумарний виграш між гравцями.

Шеплі запропонував строгий математичний спосіб цього розподілу.

У нашому випадку:

- гравці = ознаки (features)
- виграш = прогноз моделі
- внесок = середній маргінальний вплив ознаки на прогноз

Формула Шеплі:

$$\phi_j = \sum_{S \subseteq F \setminus \{j\}} \frac{|S|! (|F| - |S| - 1)!}{|F|!} (f(S \cup \{j\}) - f(S))$$

де

- ϕ_j - внесок j -ої ознаки,
- S - підмножина всіх ознак без j ,
- $f(S)$ - прогноз моделі, побудованої на ознаках S .

Це єдиний метод, який задовольняє одночасно всі чотири аксіоми справедливості (симетрія, ефективність, монотонність, нульовий гравець).

SHAP у машинному навчанні

SHAP апроксимує значення Шеплі так, щоб їх можна було обчислити для складних моделей:

- деревоподібні моделі → TreeExplainer (швидко)
- диференційовні моделі (нейронні мережі) → DeepExplainer
- чорні ящики (SVR, ансамблі) → KernelExplainer

Переваги SHAP

- універсальний підхід для всіх моделей,
- строгий математичний фундамент,
- пояснює вплив ознаки для кожного прогнозу, а не лише глобально,

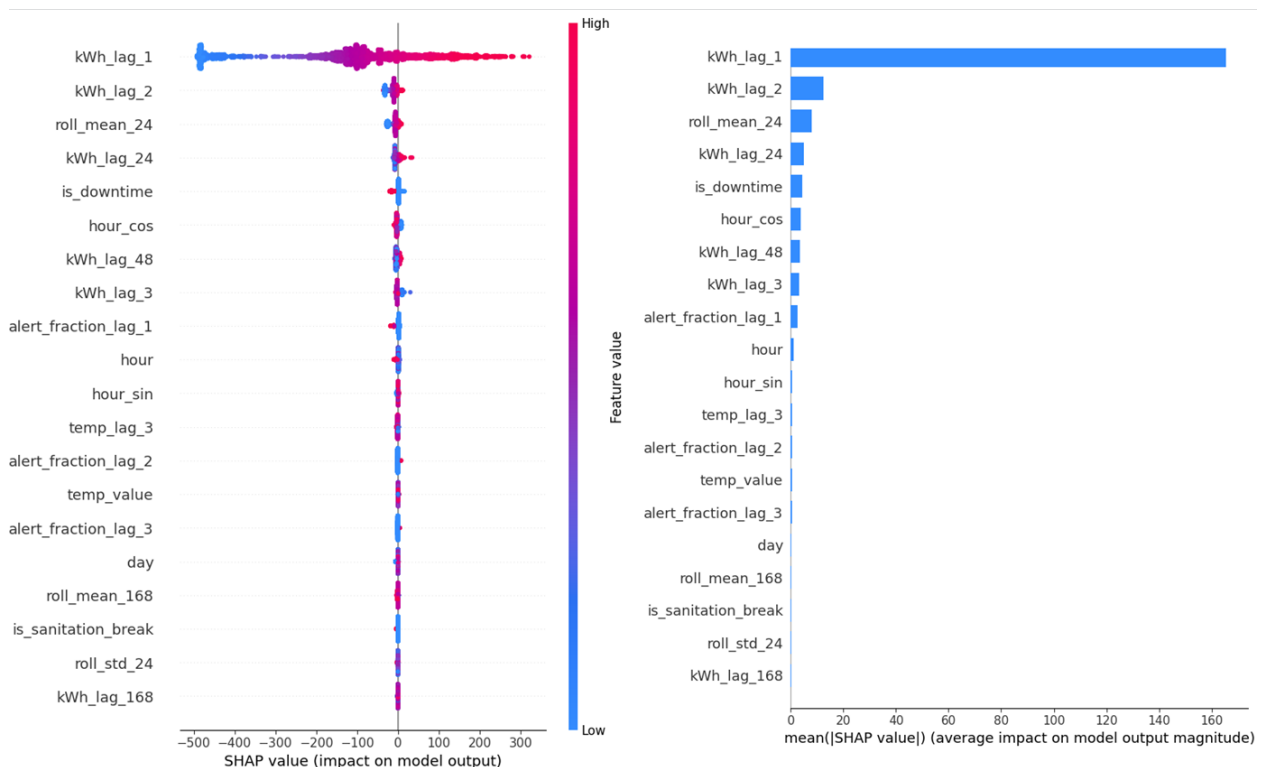
- дозволяє виявити нелінійні залежності, взаємодії та порогові ефекти.

У цьому дослідженні використано два підходи до оцінки важливості ознак:

1. Feature Importance - класичний метод для моделей дерев (Random Forest, XGBoost).
2. SHAP (SHapley Additive exPlanations) - сучасний універсальний метод, який дозволяє інтерпретувати довільні моделі, включно з нелінійними і глибинними мережами.

6.2 SHAP-аналіз моделі XGBoost

Рис. 6.2.1



SHAP-інтерпретація моделі XGBoost демонструє яскраво виражену структуру важливості ознак: модель концентрує основний внесок на лагових змінних енергоспоживання та використовує їх як головний

механізм формування прогнозу. Така поведінка відповідає природі бустингових алгоритмів, які здатні вловлювати складні нелінійні залежності, але водночас схильні формувати високу чутливість до останніх спостережень у ряді.

1. Домінування лагових змінних (kWh_lag_1, kWh_lag_2, kWh_lag_3)

SHAP-графік очевидно показує:

- kWh_lag_1 має найбільший вплив серед усіх ознак - як за абсолютною величиною SHAP, так і за різницею між позитивними та негативними внесками.
- kWh_lag_2 та kWh_lag_3 також значущі, але їх вплив різко спадає порівняно з першим лагом.
- Структура SHAP-розкиду демонструє, що XGBoost дуже залежить від інерції ряду:
високе значення попередньої години майже завжди підштовхує прогноз вгору, низьке - вниз.

Це вказує на чітку autoregressive-поведінку моделі, хоча вона й не AR-модель у класичному сенсі.

2. Використання добових патернів через згладжені ознаки

Помірну, але стабільну важливість мають:

- roll_mean_24
- kWh_lag_24
- hour_cos

Це свідчить про те, що XGBoost:

- частково використовує добову сезонність,

- враховує середні шаблони навантаження,
- коригує прогноз у відповідності до годинного циклу.

Проте добові ознаки значно поступаються лагам за впливом - модель орієнтується передусім на короткострокову динаміку.

3. Реакція на простій (is_downtime)

SHAP-поведінка is_downtime:

- має виразний негативний SHAP-вплив,
- зменшує прогноз, коли значення ознаки дорівнює 1,
- втім, вплив не настільки великий, як у лагових змінних.

Це означає, що XGBoost розпізнає періоди зниженого енергоспоживання, пов'язані з технологічними паузами, але інтерпретує їх менш агресивно, ніж зміни, зафіксовані в лагових значеннях.

4. Екзогенні фактори (температура, тривоги)

Ознаки:

- temp_value, temp_lag_x
- alert_fraction, alert_fraction_lag_x

мають дуже низькі середні |SHAP|, що видно як:

- на summary plot (усі точки скупчені біля нуля),
- так і на bar-діаграмі (м майже нульові стовпчики).

Це об'єктивно означає:

- XGBoost не використовує температуру чи тривоги як прогностичні чинники;
- їхній внесок мала значущість у порівнянні з лагами;

- з погляду моделі, внутрішня динаміка процесу значно важливіша за зовнішні фактори.

5. Загальна структура впливу

На основі SHAP можна зробити такі узагальнення:

1. XGBoost дуже сильно покладається на інерційність ряду.
Це добре для короткострокових змін, але може знизити якість на довгих інтервалах зі зміною рівня.
2. Добові патерни враховані, але не є ключовими.
3. Простоті розпізнаються, проте вплив помірний.
4. Зовнішні фактори практично не використовуються моделлю.
5. Структура SHAP симетрична з великими “хвостами”, що є типовою ознакою нелінійного навчання, чутливого до локальних сплесків.

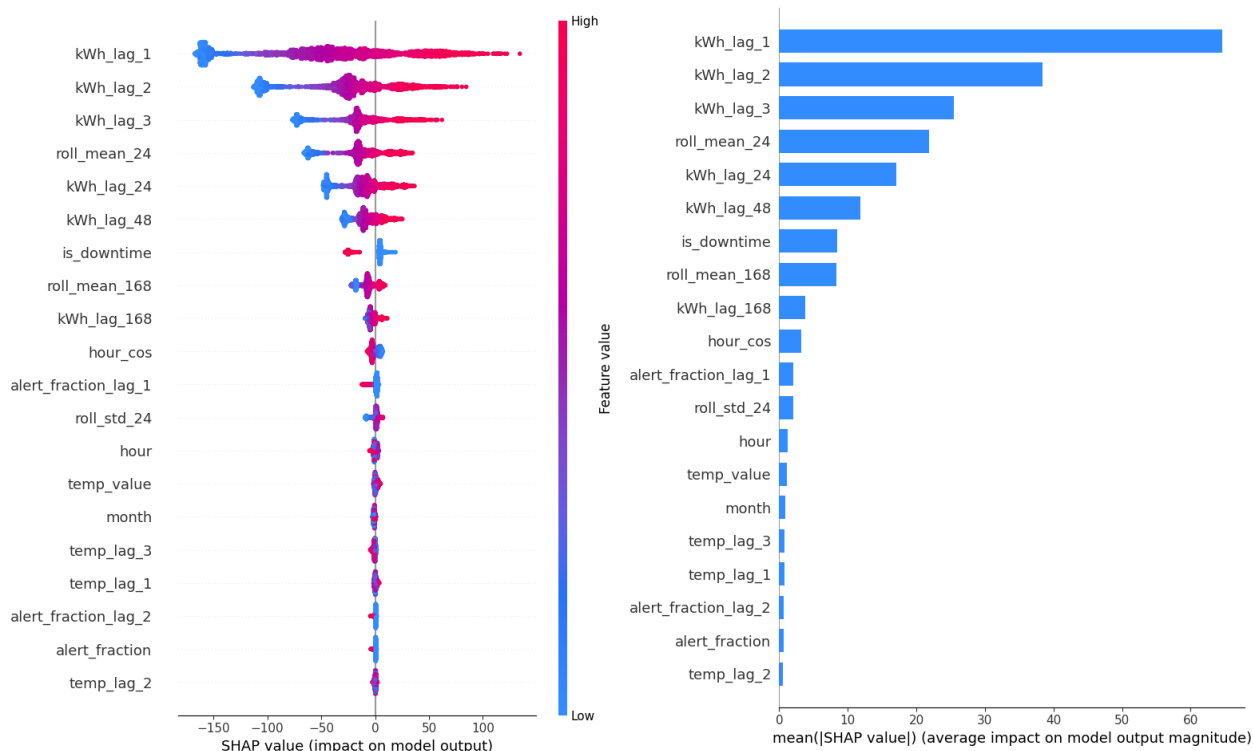
6. Підсумок

SHAP-аналіз моделі XGBoost показав, що:

- головними рушіями прогнозу є лагові змінні короткого горизонту;
- модель вміє вловлювати короткострокові сплески та падіння;
- сезонні та часові особливості використовуються як допоміжні сигнали;
- екзогенні фактори не відіграють суттєвої ролі;
- модель може бути дуже ефективною у стабільних режимах, але менш точною у періоди різких структурних змін або тривалих зрушень рівня.

6.3 SHAP-аналіз моделі Random Forest

Рис. 6.3.1



SHAP-інтерпретація моделі Random Forest демонструє більш рівномірний розподіл важливості між ознаками порівняно з бустинговими моделями. Хоча перші лаги енергоспоживання залишаються ключовими, внесок інших ознак помітно зростає, що свідчить про властиву деревоподібним ансамблям здатність використовувати широкий набір локальних правил та менш агресивно «притискатися» до інерції ряду.

1. Лагові змінні як основа прогнозу

На bar-діаграмі видно, що:

- kWh_lag_1, kWh_lag_2 та kWh_lag_3 мають найбільший середній вплив,
- проте їх домінування менш виражене, ніж у XGBoost: інші ознаки теж отримують суттєву частку SHAP-внеску.

Це вказує на те, що Random Forest:

- використовує короткострокову інерційність енергоспоживання,
- але водночас спирається і на ширший контекст, формуючи рішення через розгалуження в дереві.

Відповідно до природи моделі, вплив лагів проявляється не лише у лінійному “чим більше - тим вище прогноз”, а у вигляді множини умовних правил.

2. Використання добових та тижневих патернів

Порівняно з XGBoost:

- roll_mean_24, kWh_lag_24, kWh_lag_48, roll_mean_168
- отримують суттєво вищі середні SHAP-величини.

Це означає, що Random Forest краще враховує сезонні структури різної тривалості -

- добові цикли (24 год)
- дводобові та тижневі шаблони (48, 168 год).

Дерева прийняття рішень природно відображають такі залежності, формуючи правила типу:

“якщо час доби та попереднє значення $X \rightarrow$ прогноз Y ”.

3. Вплив простоїв (is_downtime)

Ознака is_downtime має помітний негативний вплив:

- видно асиметрію точок зі зміщенням у бік зниження прогнозу,
- її важливість посідає середню позицію.

Модель ефективно ідентифікує технологічні паузи - не так сильно, як короткострокові лаги, але набагато помітніше, ніж XGBoost. Це

свідчить про кращу здатність Random Forest виділяти аномальні режими роботи.

4. Низький вплив температури та тривоги

Як і у випадку з XGBoost, екзогенні фактори:

- temp_value, temp_lag_1/2/3,
- alert_fraction, alert_fraction_lag_x

мають дуже малий середній вклад у прогноз.

Причина цього полягає не в недоліках моделі, а в структурі самого процесу:

- короткострокові зміни навантаження на фабриці значно сильніше визначаються внутрішніми технологічними режимами, ніж погодою чи наявністю повітряних тривог.

Random Forest не ігнорує ці ознаки повністю, але вони не стають ключовими у жодному зі сформованих дерев.

5. Структура SHAP-поведінки

- ширші та менш асиметричні розподіли для всіх ознак,
- наявність множини точок середньої величини впливу,
- суттєву варіацію SHAP навіть у другорядних ознак.

Це характерно саме для Random Forest:

модель створює набір дерев з різними розгалуженнями, кожне з яких може використовувати власний піднабір ознак для опису локальної поведінки процесу.

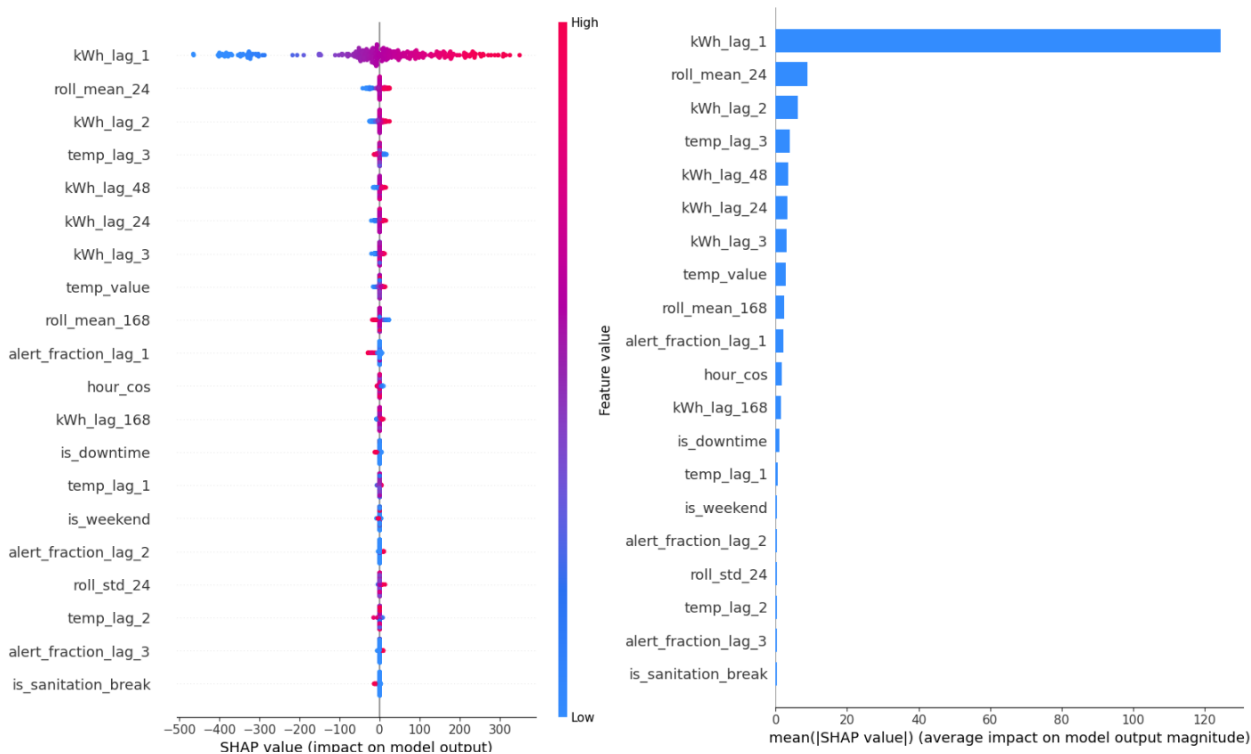
Як наслідок - структура впливу є більш збалансованою

6. Підсумок для моделі Random Forest

1. Лаги залишаються головним джерелом інформації, проте їх домінування не таке абсолютне, як у XGBoost.
2. Модель добре вловлює сезонні структури на горизонтах 24, 48 та 168 годин.
3. Прості фабрики чітко відображаються у формі негативних SHAP-впливів.
4. Температура та тривоги майже не впливають на прогноз, що підтверджує їх низьку предиктивну силу для короткострокового навантаження.
5. Загалом Random Forest формує більш “м’яку” картину важливості ознак, рівномірніше розподіляючи вплив між різними джерелами інформації.

6.3 SHAP-аналіз моделі SVR

Побудова SHAP-значень для Support Vector Regression потребує застосування KernelExplainer, оскільки модель не є деревоподібною та не має внутрішнього механізму обчислення локальних впливів.



Структура впливів SHAP

Аналіз показує, що SVR є сильно “інерційною” моделлю, тобто значенням у попередні моменти часу надається значно більша вага, ніж іншим факторним змінним.

Найбільш виразною є перша ознака:

- kWh_lag_1 - домінує з величезним відривом, що означає майже лінійне слідування прогнозу за значенням споживання за попередню годину.

Це характерно для SVR з RBF-ядром: модель схильна “притискати” прогноз до найближчого прикладу в просторовому сенсі.

Далі за важливістю йдуть:

- roll_mean_24 - добова ковзна середня, що подає змягчену добову структуру.

- kWh_lag_2, kWh_lag_3 - лаги 2 та 3 зменшують контрастність ряду, допомагаючи моделі відловлювати зміну тенденції.
- temp_lag_3, temp_value - температурний вплив присутній, але значно менший, ніж у XGBoost або RF.
- kWh_lag_24, kWh_lag_48 - відображають слабке, але все ж помітне виявлення добової та дводобової сезонності.

Поводження категоріальних та “подієвих” ознак

Ознаки:

- is_downtime,
- alert_fraction_*,
- is_sanitation_break,
- календарні змінні (day, dow_*, is_weekend)

мають дуже малий внесок — майже нульовий.

Це свідчить про те, що SVR:

1. Не вміє виражено реагувати на різкі розриви в часовому ряді
 - у моделі немає механізмів, здатних ефективно описувати дискретні стани (“працює/не працює”).
2. Ґрунтується на гладкій функції подібності (RBF kernel)
 - через що різкі зміни у споживанні (під час зупинок) фактично відсуваються на другий план.
3. Факторні ознаки з низькою варіативністю або бінарною природою погано інтегруються в RBF-простір, тому їх вплив згладжується.

Структура знаків SHAP-значень

- високі значення лагів (червоні точки) зсувають прогноз угору, тобто модель формує позитивну залежність із попередніми вимірами;
- низькі значення (сині точки) тягнуть прогноз до нижчих значень.

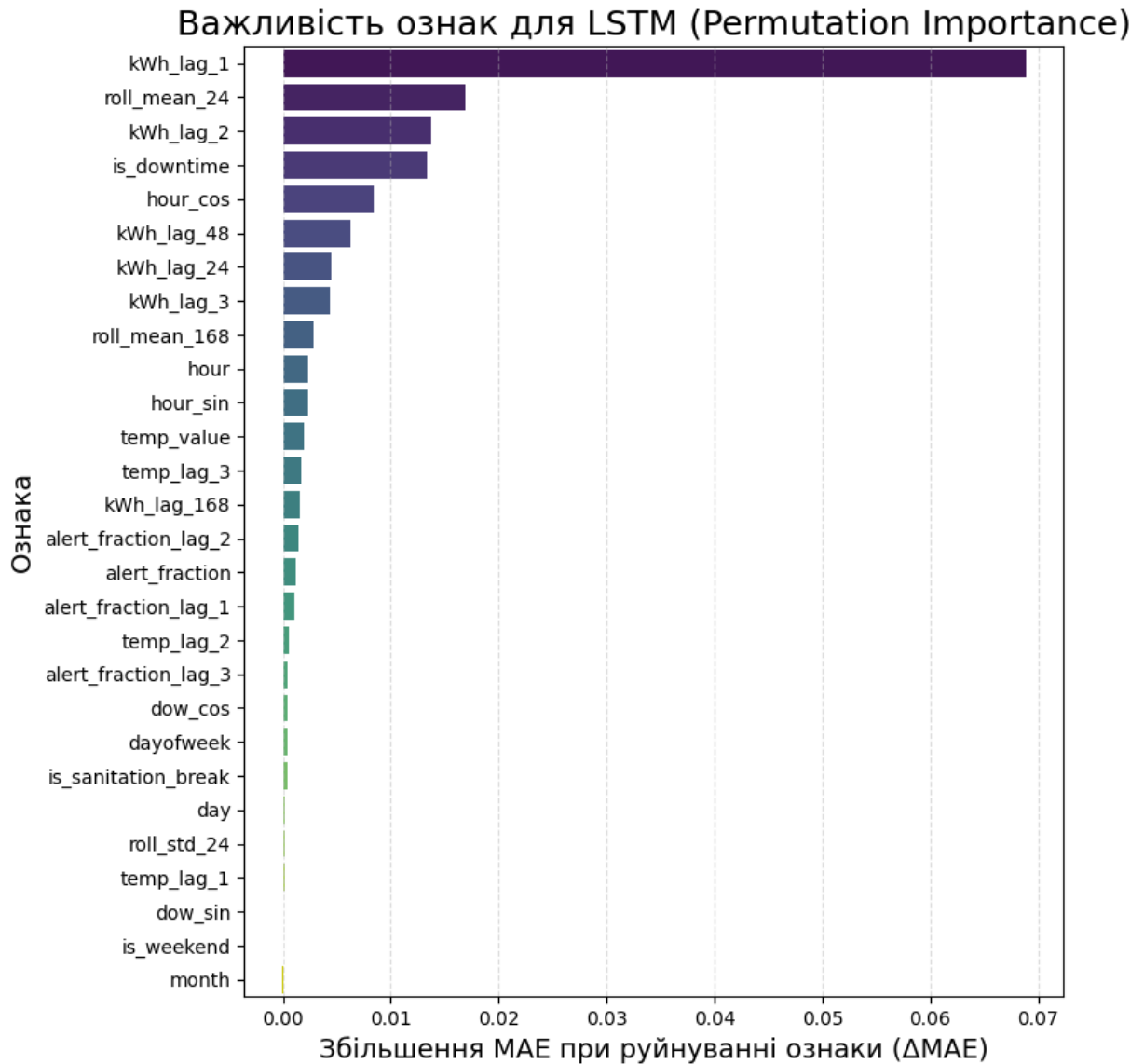
Тобто SVR фактично наближається до гладкої апроксимації Autoregressive-типу, але без гнучкості ARIMA/SARIMA та без дискретної чутливості деревоподібних моделей.

Підсумок для моделі SVR

1. SVR дуже сильно залежить від лагів, особливо від kWh_lag_1. Фактично модель формує коротку автокореляційну структуру з швидким спадом.
2. Сезонність (24, 48, 168 год) враховується, але набагато слабше, ніж у XGBoost чи Random Forest.
3. Подієві ознаки (is_downtime, alert_fraction) майже не впливають - модель їх ігнорує через особливості ядра RBF.
4. Різкі піки або падіння споживання SVR моделює гірше, оскільки такі точки погано описуються гладкою функцією подібності.
5. Температура має помірний вплив - SVR сприймає її як слабку нелінійну корекцію основного тренду, але не як структурний чинник.

6.4. Важливість ознак для LSTM (Permutation Importance)

Рис. 6.4.1



Для інтерпретації моделі LSTM було застосовано метод Permutation Importance, який оцінює внесок кожної ознаки шляхом випадкового перемішування її значень. Порушення інформаційної структури ознаки призводить до збільшення помилки прогнозу, і ця величина (ΔMAE) використовується як міра її важливості.

1. Домінування часових лагів електроспоживання

Найважливішою ознакою для LSTM стала:

kWh_lag_1, що дає найбільше зростання MAE при руйнуванні.

Далі за значущістю йдуть:

- roll_mean_24 (добове середнє),
- kWh_lag_2,
- kWh_lag_3,
- kWh_lag_24,
- kWh_lag_48,
- kWh_lag_168.

Це демонструє, що LSTM у першу чергу спирається на локальний контекст, а також на добові та тижневі цикли, які природно захоплюються лагами.

2. Значення експлуатаційних режимів працюють, але слабше

Ознаки:

- is_downtime,
- is_sanitation_break,

потрапляють у середню частину рейтингу.

Модель враховує інформацію про зупинки, але вона не є визначальною на рівні внеску в помилку.

LSTM і так "бачить" падіння чи стрибки через лаги, тому експлуатаційні мітки є дублюючою, але корисною інформацією.

3. Температура та її лаги

Ознаки:

temp_value, temp_lag_1, temp_lag_2, temp_lag_3,

мають мінімальний вплив.

Це узгоджується з тим, що у конкретному профілі фабрики температура впливає незначно (порівняно з операційною динамікою виробництва).

4. Тривоги (alert_fraction та лаги)

Усі alert_fraction_* знаходяться в хвості рейтингу.

Це очікувано, оскільки:

- LSTM не використовує ці ознаки під час прогнозування майбутніх значень (бо їх неможливо знати наперед),
- і навіть у тренуванні їхній внесок мінімальний порівняно з лагами.

5. Календарні ознаки — найменший вплив

month, day, dayofweek, is_weekend, синус-косинус годин

мають майже нульовий вплив.

Модель намагається вловлювати патерни безпосередньо з лагів, тому календарні змінні стають надлишковими.

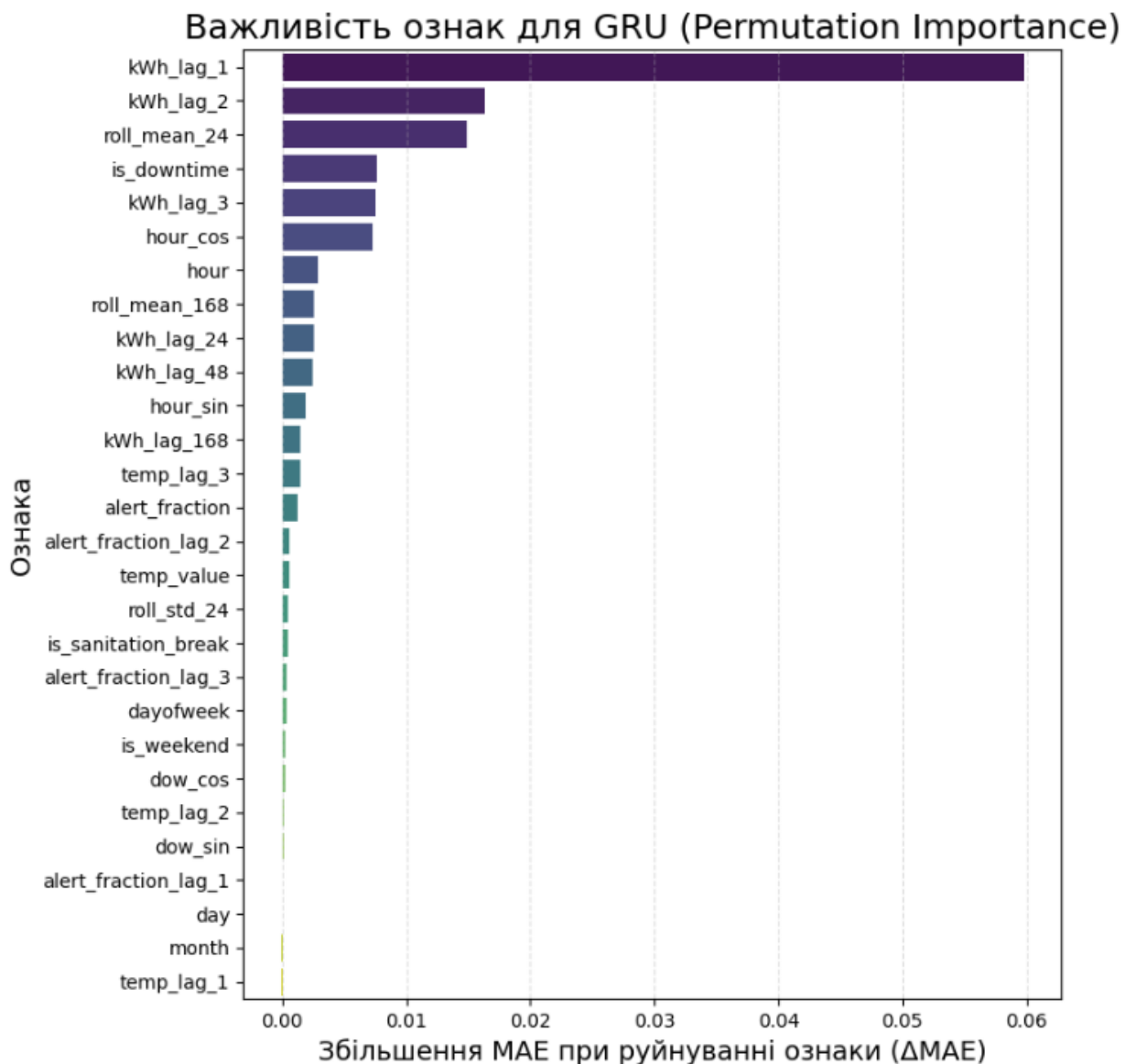
Підсумок для моделі LSTM

1. LSTM майже повністю формує прогноз на основі поведінки самого ряду:
попередні значення (lag 1–3), добові (24), двоходбові (48) та тижневі (168) цикли.

2. Операційні стани (зупинки) враховуються, але поступаються за вагою часовим залежностям, які модель навчає безпосередньо.
3. Екзогенні фактори (температура, тривоги) відіграють мінімальну роль, що підтверджує домінування внутрішньої виробничої динаміки над зовнішніми впливами.
4. Календарні ознаки майже не додають інформації, і їхнє включення радше "про запас", ніж реально значиме.

6.5. Важливість ознак для GRU (Permutation Importance)

Рис. 6.5.1



Для моделі GRU оцінку важливості ознак здійснено методом Permutation Importance, який базується на вимірюванні зміни похибки прогнозу при випадковому перемішуванні значень кожної ознаки. Збільшення MAE після руйнування структури ознаки відображає її реальний внесок у роботу моделі.

Головні закономірності

1. Лагові ознаки електроспоживання формують ядро моделі

Найбільший внесок мають:

- kWh_lag_1 - ключовий предиктор у GRU;
- kWh_lag_2 - значуща короткострокова пам'ять;
- roll_mean_24 - добова структура споживання.

Також до важливих належать:

- kWh_lag_3,
- kWh_lag_24,
- kWh_lag_48,
- kWh_lag_168.

Це підтверджує, що GRU, як і LSTM, використовує як локальні, так і періодичні (добові/тижневі) залежності, які є характерними для виробничих процесів.

2. Інформація про зупинки впливає суттєвіше, ніж у LSTM

Ознака **is_downtime** посідає високе місце у рейтингу GRU.

На відміну від LSTM, GRU частіше враховує експлуатаційні маркери, що вказує на:

- чутливість GRU до різких переходів у даних,

- здатність швидше "перемикатися" між режимами сигналу.

3. Годинні цикли роблять помірний, проте помітний внесок

Ознаки:

hour, hour_cos, hour_sin

мають середню важливість.

Вони допомагають моделі орієнтуватися у добових ритмах, доповнюючи інформацію, яку модель отримує з лагів і ковзних середніх.

4. Вплив температури залишається обмеженим

Ознаки:

temp_value, temp_lag_1, temp_lag_2, temp_lag_3

розташовані у нижній частині діаграми.

Температурний фактор не відіграє суттєвої ролі у структурі навантаження виробничого обладнання.

5. Тривоги (alert_fraction та лаги)

Поведінка повністю узгоджується з результатами інших моделей:

Ознаки:

alert_fraction, alert_fraction_lag_1, alert_fraction_lag_2,
alert_fraction_lag_3

мають дуже малий внесок.

Це логічно, оскільки моделі не використовують ці ознаки у режимі прогнозу, а сигнал тривоги майже не корелює зі структурою енергоспоживання.

6. Календарні ознаки мають найменшу значущість

Ознаки:

day, dayofweek, month, is_weekend, dow_sin, dow_cos

знаходяться в кінці рейтингу.

GRU формує прогноз переважно на основі лагів та короткострокових патернів, тому календарні предиктори є надлишковими.

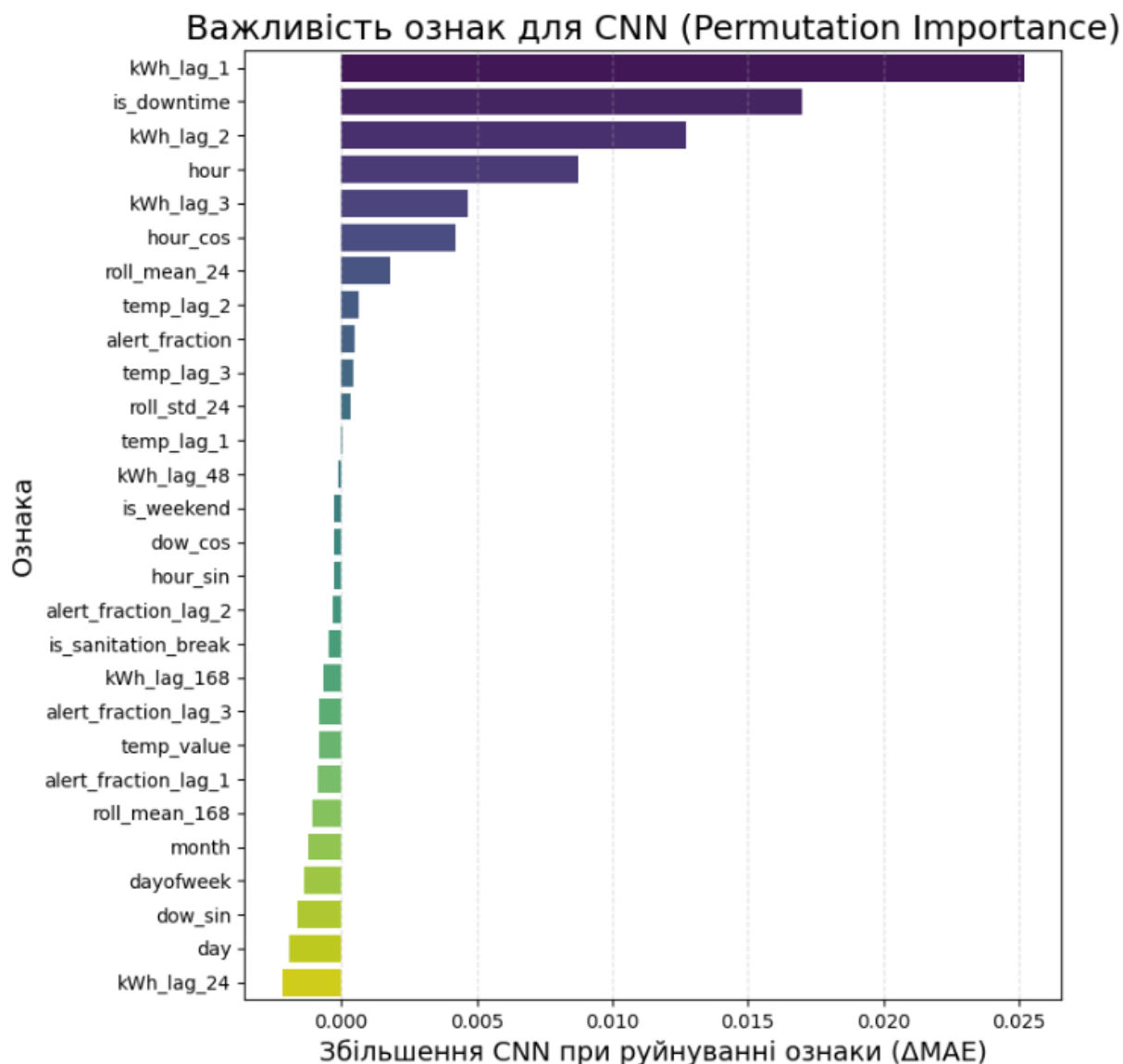
Підсумок для моделі GRU

1. GRU використовує той самий набір ключових драйверів, що й LSTM - короткі лаги, добові цикли, тижневі повтори.
2. Важливість is_downtime вища, ніж у LSTM, що свідчить про чутливість GRU до експлуатаційних переходів.
3. Температура та тривоги практично не впливають, що відповідає характеру технологічного процесу фабрики.
4. Календарні ознаки майже неінформативні, адже GRU навчає часову структуру безпосередньо з лагів.

Загалом модель GRU характеризується високою здатністю адаптуватися до локальної динаміки та враховувати зміни режимів, але покладається майже виключно на інформацію з самого ряду.

6.6 Важливість ознак для CNN (Permutation Importance)

Рис. 6.6.1



1. Провідні ознаки: короткі лаги та маркери експлуатаційного стану

CNN демонструє нетипову для нейромереж поведінку:

is_downtime входить до трійки найважливіших ознак, поступаючись тільки lag-1 і лагам 2–3.

Це означає, що CNN:

- добре реагує на різкі зміни режимів роботи (на відміну від ARIMA та RF),
- успішно розпізнає шаблони спадів/зупинок у часовому вікні.

Сукупність провідних ознак:

- kWh_lag_1 - найсильніший предиктор;
- is_downtime - одна з найвпливовіших ознак;
- kWh_lag_2, kWh_lag_3 короткострокова пам'ять;
- hour, hour_cos - добові ритми.

CNN активно використовує найближчу історію та рядкові патерни, що відповідає очікуваній природі згорткових фільтрів.

2. Використання часових циклів

На відміну від GRU/LSTM, CNN сильніше змінює похибку при руйнуванні:

hour, hour_cos, hour_sin,

що вказує на те, що згорткові фільтри навчилися виділяти локальні патерни, але модель додатково покладається на явно задані часові змінні.

3. Добові та тижневі структури важливі, але в меншій мірі

Ознаки:

roll_mean_24, roll_std_24, kWh_lag_24, kWh_lag_48, kWh_lag_168,

мають нижчий внесок порівняно з CNN-попередниками (GRU/LSTM).

Це очікувано:

CNN добре вловлює локальні шаблони всередині вікна, але гірше

відтворює довгострокові залежності, що не подаються явно у вигляді окремих фільтрів.

4. Температурний фактор та тривоги — мінімальний вплив

Ознаки:

`temp_value`, `temp_lag_1`, `temp_lag_2`, `temp_lag_3`,
`alert_fraction` і всі її лаги

мають помітно низьку вагу.

Це узгоджується з результатами всіх моделей:

температура та частка часу тривог практично не впливають на характер роботи обладнання.

5. Календарні змінні найменш інформативні

Ознаки:

`day`, `month`, `dayofweek`, `is_weekend`, `dow_sin`, `dow_cos`

мають майже нульовий внесок.

CNN, як і LSTM/GRU, самостійно вивчає сезонність з лагів, тому календарні предиктори виявляються надлишковими.

Підсумок для моделі CNN

1. CNN найбільше покладається на короткі лаги (1-3 години) - для неї локальна динаміка в межах вікна є критичною.
2. Інформація про зупинки (`is_downtime`) суттєво впливає на прогноз, що підтверджує здатність CNN добре виявляти різкі переходи у даних.
3. Довготривалі лаги (24-168 годин) менш значущі, що узгоджується з природою згорткових фільтрів, які краще працюють з короткими часовими структурами.

4. Температура та тривоги не впливають на роботу моделі, їх внесок є мізерним.
5. Календарні ознаки залишаються другорядними, оскільки CNN витягує сезонність із самої структури ряду.

РОЗДІЛ 7. АНАЛІЗ ЗАЛИШКІВ

7.1 Методи оцінки залишків

Аналіз залишків є ключовим етапом оцінювання якості будь-якої моделі прогнозування часових рядів. На відміну від порівняння моделей за агрегованими метриками (MAE, RMSE, MAPE або R^2), дослідження залишків дозволяє оцінити не лише рівень загальної помилки, а й структуру компонент, які модель не змогла адекватно відтворити. Залишки (residuals) визначаються як різниця між фактичним значенням часового ряду та прогнозом моделі:

$$e_t = y_t - \hat{y}_t$$

де

y_t - фактичне значення у момент часу t ,

\hat{y}_t - прогноз моделі,

e_t - залишок (похибка передбачення).

У класичних підходах моделювання часових рядів вважається, що добре побудована модель повинна виключати з вихідного ряду всі передбачувані компоненти - тренд, сезонність, автокореляційну структуру, вплив екзогенних змінних - так, щоб залишки наближались до білих шумів: некорельованої послідовності випадкових величин з нульовим середнім та стабільною дисперсією.

Якщо залишки демонструють структуру, то така структура присутня і в самому часовому ряді, але модель її не навчила. Саме тому аналіз залишків є не лише засобом перевірки якості моделі, а й інструментом виявлення факторів, які потрібно врахувати у подальших ітераціях моделювання.

1. Властивості “якісних” залишків

Добре побудована модель повинна генерувати залишки, що володіють такими властивостями:

1. Нульове середнє

$$E[e_t] = 0$$

Систематичні позитивні або негативні відхилення свідчать про зміщення моделі (bias).

2. Незалежність залишків

$$\text{Cov}(e_t, e_{t-k}) = 0 \forall k > 0$$

Наявність автокореляції свідчить про те, що модель не врахувала певну часову структуру (наприклад, внутрішньодобові цикли або інерційність процесу).

3. Гомоскедастичність (стала дисперсія)

Якщо дисперсія змінюється з часом (гетероскедастичність), це свідчить про те, що модель не врахувала залежність масштабів коливань від рівня навантаження або інших умов.

4. Нормальний (або близький до симетричного) розподіл

Це не обов'язкова вимога, але нормальна форма розподілу полегшує аналіз та є індикатором “випадковості” похибок.

5. Відсутність нелінійної структури

Якщо на графіку “залишки vs прогноз” або “залишки vs час” видно закономірності, це означає, що модель пропустила певні залежності, зокрема нелінійні.

2. Автокореляція залишків (ACF та PACF)

Автокореляція залишків є одним із головних показників того, наскільки добре модель змогла описати часову структуру ряду.

- ACF (Autocorrelation Function) показує кореляцію між залишками та їх зсунутою копією на лаг k .
Піки ACF на лагових залежностях свідчать про наявність структури, не вилученої моделлю.
- PACF (Partial Autocorrelation Function) оцінює часткову кореляцію між залишками, усуваючи вплив усіх проміжних лагів.

Якщо ACF/PACF мають значущі значення на конкретних лагах, це означає, що:

- модель недовраховала інерційність процесу;
- або не врахувала сезонні цикли;
- або пропустила залежність від екзогенного фактора.

3. Тест Дербіна-Вотсона (Durbin–Watson Test)

Статистика Дербіна–Вотсона є кількісною мірою автокореляції першого порядку:

$$DW = \frac{\sum_{t=2}^N (e_t - e_{t-1})^2}{\sum_{t=1}^N e_t^2}$$

Інтерпретація значень:

- $DW \approx 2$: залишки некорельовані
- $DW < 2$: позитивна автокореляція
- $DW > 2$: негативна автокореляція

Для моделей прогнозування енергоспоживання, де ряд має сильну інерційність, значення $DW \ll 2$ свідчить про те, що модель пропустила частину часової структури.

4. Тест Льюнга-Бокса (Ljung–Box Test)

Ljung–Box перевіряє нульову гіпотезу про те, що набір автокореляцій до певного лага є статистично незначущими:

$$H_0: \rho_1 = \rho_2 = \dots = \rho_m = 0$$

де ρ_k — автокореляція на лагу k .

- Високе р-значення → залишки некорельовані (модель якісна).
- Низьке р-значення → значуща автокореляція (модель пропускає структуру ряду).

Тест є важливим, оскільки він враховує сукупний ефект кількох лагів, а не лише окремі значення ACF.

5. Аналіз розподілу залишків

Окремим елементом є дослідження форми розподілу залишків:

- симетричний розподіл свідчить про відсутність зміщення;
- важкі хвости вказують на різкі зміни у даних, які модель не змогла відтворити;
- багатомодальний розподіл може означати наявність кількох режимів роботи процесу (наприклад, денний/нічний режим, період роботи/зупинки тощо).

Розподіл залишків у моделях прогнозування електроспоживання особливо важливий, оскільки різкі піки і технологічні зупинки призводять до асиметрії та змінної дисперсії, які прості моделі не здатні обробляти.

6. Значення аналізу залишків для моделювання енергоспоживання

Залишки відображають ті компоненти процесу енергоспоживання, які моделі не змогли навчити:

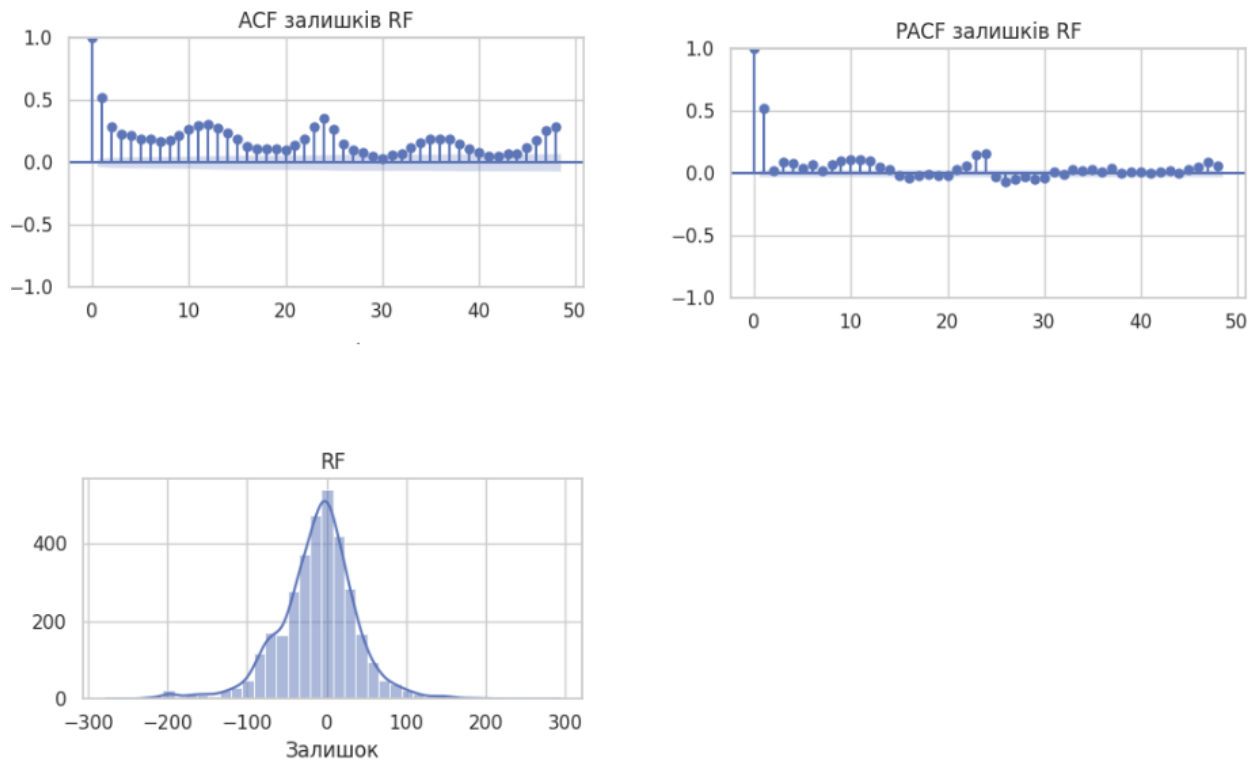
- внутрішньодобові цикли;
- зміни виробничого навантаження;
- аварійні та технологічні зупинки;
- поведінку під час тривоги;
- залежність від температури;
- короткострокові пікові стрибки.

Для задачі прогнозування, де часовий ряд є неоднорідним, містить різкі переходи між режимами та значні зовнішні впливи, аналіз залишків є необхідним кроком, оскільки:

1. дозволяє об'єктивно відрізнити "погану модель" від "обмеження даних";
2. виявляє закономірності, які варто винести у нові ознаки;
3. дає можливість оцінити ризики моделі під час пікових режимів;
4. допомагає пояснити, чому певні моделі (наприклад, лінійні) не можуть бути застосовані до такого ряду.

7.2 Аналіз залишків моделі Random Forest

Рис. 7.2.1



Оцінка залишків моделі Random Forest демонструє, що попри загалом прийнятну якість прогнозування у часовому вимірі, структура помилок має істотні відхилення від вимог класичної незалежності та відсутності автокореляції.

1. Автокореляційна структура (ACF та PACF)

Графік автокореляційної функції (ACF) засвідчує наявність помітної автокореляції залишків на широкому діапазоні лагів. Особливо виділяється перший лаг, після чого автокореляція знижується, але не зникає, утворюючи хвилеподібну структуру з періодами підвищення та послаблення кореляції. Така поведінка характерна для моделей, які не повністю відтворюють внутрішню динаміку процесу, зокрема короткі цикли, періодичні зміни або швидкі переходи між режимами.

PACF (часткова автокореляція) підтверджує цю тенденцію: перший лаг має домінуючий внесок, що вказує на виражений лінійний зв'язок між сусідніми точками ряду. Подальші часткові кореляції набагато слабші, але все ще статистично відмінні від нуля, що вказує на складну комбіновану структуру залежностей.

2. Тест Дурбіна–Ватсона

Значення Durbin–Watson:

$$DW = 0.906$$

Для білих шумів або незалежних залишків очікуване значення $DW \approx 2$. Натомість отримане значення істотно менше за 2, що однозначно свідчить про виражену позитивну автокореляцію. Це свідчить, що модель систематично «помиляється» в одному напрямку протягом послідовних моментів часу, тобто занижені прогнози мають схильність утворювати кластери, подібно до випадків із завищеними значеннями..

Такий результат підтверджує висновки ACF/PACF і вказує на те, що Random Forest не компенсує ланцюгову залежність у даних.

3. Тест Льюнга–Бокса на автокореляцію

$$LB_{(lag=24)} = 0$$

p-value, що дорівнює нулю (з точністю до обчислювальної похибки), вказує на відхилення нульової гіпотези щодо відсутності автокореляції. Таким чином, навіть при розгляді структури залежностей у межах добових циклів модель систематично залишає у залишках інформаційний сигнал. Це підтверджує, що Random Forest не повністю відтворює внутрішню добову або міждобову динаміку споживання енергії.

4. Розподіл залишків

Гістограма залишків має форму, близьку до симетричної, з вираженим пиком навколо нуля, що відповідає загальній схильності Random Forest виробляти центровані помилки без суттєвого зміщення. Однак «хвости» розподілу помітно ширші за гаусівські, що говорить про частіші великі помилки, ніж було б у випадку моделі, яка точно відтворює закономірності даних.

Наявність більшого числа нетипових відхилень узгоджується з природою Random Forest: модель добре згладжує поведінку, але зазвичай погано реагує на раптові падіння чи стрибки (наприклад, при технологічних зупинках або відновленні роботи після них).

Узагальнюючий висновок

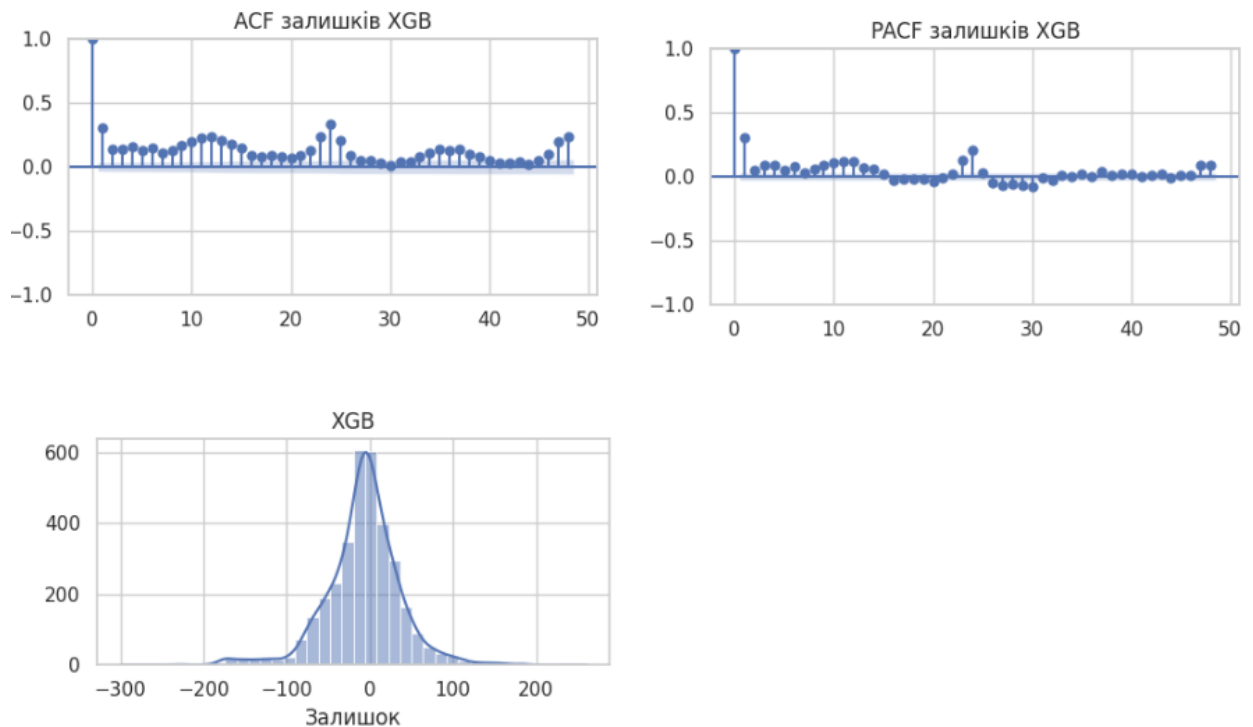
Аналіз залишків свідчить, що Random Forest не забезпечує незалежності помилок прогнозу. Навіть за умови доволі непоганої середньої точності модель зберігає у залишках значний систематичний компонент. Це виявляється через:

- виражену автокореляцію першого порядку та нижчих лагів,
- значне відхилення Durbin–Watson від 2,
- повне відхилення гіпотези про відсутність автокореляції за тестом Льюнга–Бокса,
- розширені хвости у розподілі залишків.

Отже, Random Forest добре апроксимує загальні рівні споживання, але не повністю вловлює короткострокову залежність у даних і залишає в залишках структурний сигнал.

7.3 Аналіз залишків моделі XGBoost

Рис. 7.3.1



Оцінка залишків моделі XGBoost демонструє кращу поведінку порівняно з Random Forest, однак залишки все одно зберігають помітну внутрішню структуру, що свідчить про неповне вилучення часових залежностей. Це очікувано для бустингових методів: вони добре адаптуються до нелінійних закономірностей, але не мають внутрішнього рекурентного механізму, здатного повністю моделювати темпоральну інерцію.

1. Автокореляційна структура (ACF та PACF)

Графік ACF демонструє:

- дуже сильний лаг-1 кореляційний зв'язок, що свідчить про те, що модель не повністю компенсувала залежність між сусідніми точками ряду;

- подальші лаги характеризуються хвилеподібним затуханням автокореляції, подібно до того, як це спостерігалось у Random Forest, проте амплітуда коливань дещо нижча;
- періодична структура з невеликими піками на лагових відстанях 20–30 годин може відповідати технологічним режимам або повторюваним поведінковим патернам навантаження.

Це свідчить, що XGBoost здатен краще відтворити короткострокову динаміку, але не усуває автокореляцію повністю.

PACF підтверджує домінування першого лагу. Усі наступні часткові автокореляції суттєво менші:

- перший лаг - різко виділений, що характерно для моделей, які лише частково прибравли авторегресивну компоненту;
- на дальніх лагах часткові коефіцієнти близькі до нуля, що означає слабку прямоконтрольну залежність.

На відміну від Random Forest, де спостерігалось кілька значущих часткових лагів, XGBoost демонструє більш чітку AR(1)-подібну структуру залишків, що є кроком у напрямку до «білого шуму», але цього недостатньо.

2. Тест Дурбіна–Ватсона

$$DW = 1.347$$

Дане значення ближче до 2, ніж у Random Forest (0.906), що свідчить про меншу ступінь автокореляції першого порядку. Водночас $DW < 2$ усе ще означає позитивну автокореляцію.

Таким чином, XGBoost значною мірою зменшив зв'язаність залишків, але не ліквідував її.

3. Тест Льюнга–Бокса

$$LB_{(lag=24)} = 0$$

p-value = 0 означає:

- гіпотеза про «відсутність автокореляції» однозначно відхиляється;
- навіть на межі добового горизонту XGBoost залишає у залишках суттєву структуру;
- модель не повністю відтворює добові й міждобові залежності, притаманні енергоспоживанню.

Фактично, це означає, що у залишках існують чіткі періодичні компоненти, які модель не навчила.

4. Розподіл залишків

Гістограма залишків XGBoost:

- має чіткий пік навколо нуля, що свідчить про мінімальне зміщення моделі;
- спостерігається помірна асиметрія та важкі хвости - це ознака того, що модель час від часу недооцінює різкі зміни або локальні піки навантаження;
- форма розподілу є вужчою та більш концентрованою, ніж у Random Forest.

Отже, XGBoost краще зменшує розкид помилок, однак нетипові значення (anomalies / excursions) усе ще виникають.

Узагальнюючий висновок

Поведінка залишків XGBoost демонструє суттєве покращення порівняно з Random Forest:

- автокореляція менш виражена,
- структура залишків ближча до «шумової»,
- розподіл залишків компактніший і більш симетричний.

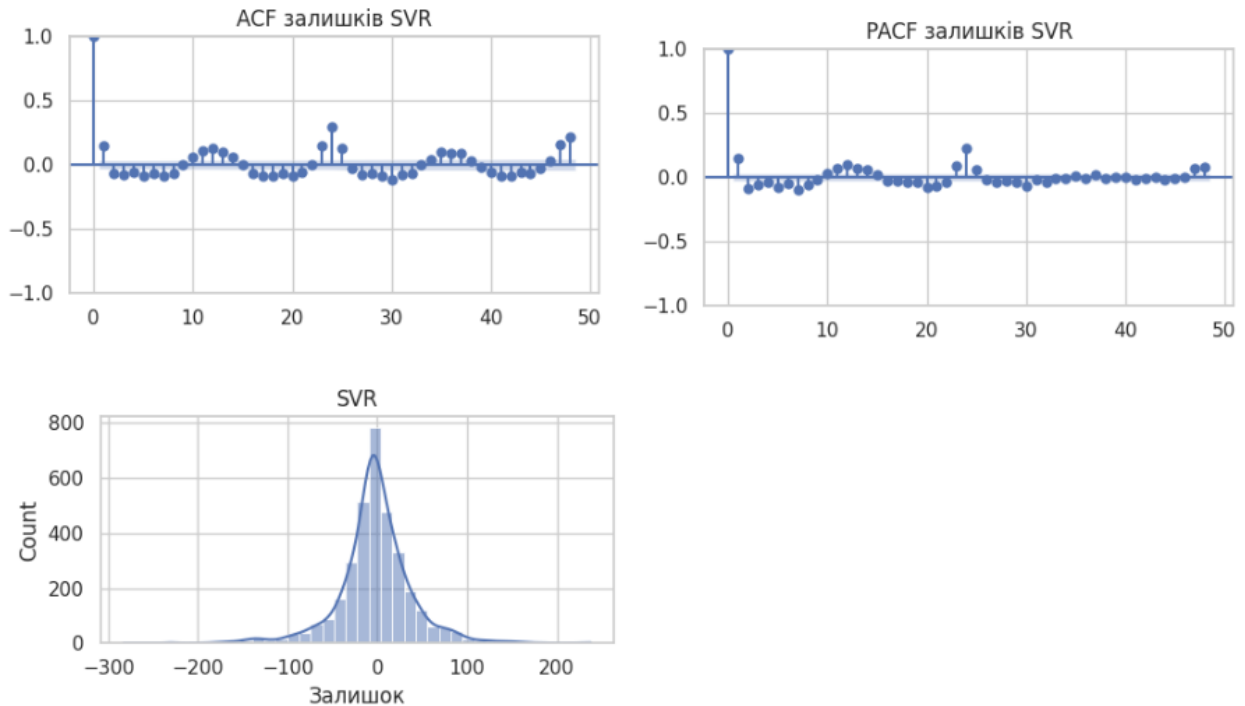
Однак низка ключових проблем залишаються:

1. Високий AR(1)-компонент: $DW < 2$ та домінування першого лагу в ACF/PACF свідчать, що XGBoost не повністю відтворює інерцію процесу.
2. Добові цикли в залишках: Ljung–Box на лагу 24 відхиляє нульову гіпотезу.
3. Наявність хвостів у розподілі помилок: модель реагує на різкі зміни енергоспоживання з лагом.

Таким чином, хоча XGBoost точніший і стабільніший, ніж RF, він не здатен повністю усунути часову структуру в помилках - що природно, зважаючи на відсутність внутрішньої рекурентності. Це пояснює, чому навіть за високими метриками точності інформаційний сигнал у залишках все ще зберігається.

7.4 Аналіз залишків моделі SVR

Рис. 7.4.1



Модель Support Vector Regression у цьому дослідженні продемонструвала найкращі показники точності серед усіх розглянутих підходів, що проявляється у низьких похибках прогнозування. Однак аналіз залишків показує, що попри загальну точність прогнозів, залишки не є повністю випадковими. Це типовий результат для моделей, які добре апроксимують гладку, монотонну або інерційну структуру ряду, але обмежено відтворюють складну темпоральну залежність, різкі збурення й добові цикли.

1. Автокореляція залишків (ACF)

Графік ACF для SVR показує:

- виражену автокореляцію лагу 1, що свідчить про залишкову інерцію - модель не повністю відтворила AR(1)-компонент ряду;
- хвилеподібну структуру автокореляцій на лагах 5-30, подібну до RF та XGBoost, але менш інтенсивну;
- наявність окремих піків на середніх лагах ($\approx 20-25$), які можуть бути наслідком сезонних та операційних циклів.

Таким чином, SVR краще згладжує короткостроковий шум, але залишає у прогнозі відчутні сліди циклічності, властивої процесу споживання електроенергії.

2. Часткова автокореляція (PACF)

PACF демонструє:

- чітко домінуючий перший лаг;
- швидке спадання часткової автокореляції;
- незначні, але стабільні коливання на лагах 10-25.

Це означає, що SVR здебільшого не породжує хибних залежностей між віддаленими точками ряду, але не повністю усуває первинний AR-компонент. Модель добре наближає загальну форму процесу, однак не повністю компенсує внутрішню часову структуру.

3. Тест Дурбіна-Ватсона

$$DW = 1.692$$

Це значення:

- помітно ближче до 2, ніж у XGBoost (1.347) та RF (0.906);
- свідчить про значно меншу позитивну автокореляцію у порівнянні з іншими моделями;

- демонструє найбільш «білошумову» поведінку серед розглянутих методів.

SVR є найближчим до моделі, що «не залишає структури у помилках», однак повної незалежності залишків усе одно не досягнуто.

4. Тест Льюнга-Бокса

$$p_{LB}(lag = 24) = 2.55 \times 10^{-164}$$

p-value практично нульове навіть на лагу 24, що означає:

- нульова гіпотеза про відсутність автокореляції однозначно відхиляється;
- залишки містять регулярні патерни, зокрема добові та субдбові цикли;
- SVR не здатен повністю моделювати сезонні структури складності рівня «production cycles», «shift patterns» або «ночі/дні».

Навіть попри високі метрики точності, SVR не усунув циклічність у залишках.

5. Розподіл залишків

Гістограма демонструє:

- вузький, чітко виражений пік навколо нуля, що відповідає високій точності моделі;
- слабку асиметрію та відносно важкі хвости - SVR інколи помиляється на різких переходах навантаження;
- менший розкид значень, ніж у RF чи XGBoost.

Це підтверджує властивість SVR: модель чудово апроксимує гладкі області ряду, але реагує з деяким запізненням на нетипові значення та структурні розриви.

Узагальнююча оцінка якості залишків SVR

SVR демонструє найкращу поведінку серед нелінійних моделей за показниками:

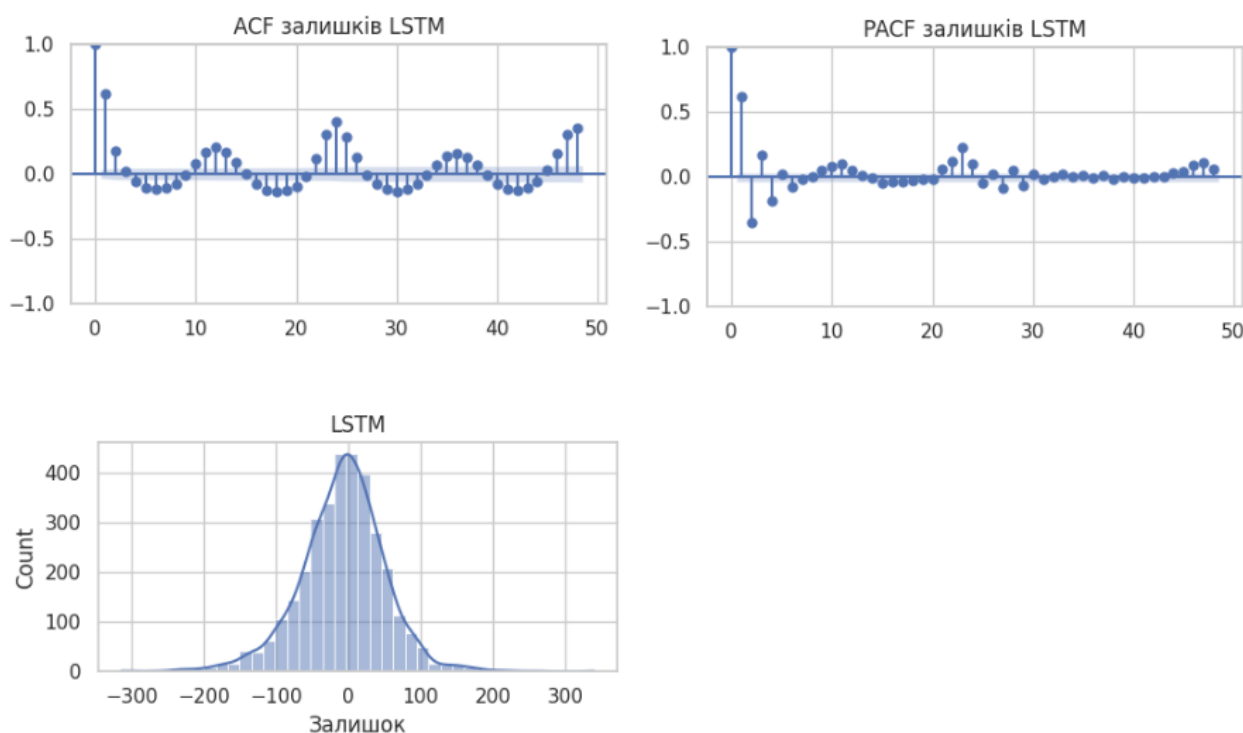
- найнижча автокореляція помилок (DW ближче до 2),
- найбільш симетричний та концентрований розподіл залишків,
- мінімальний розкид помилок.

Однак:

- залишки не є білим шумом,
- добові патерни продовжують проявлятися,
- тест Льюнга–Бокса на лагу 24 підтверджує залишкову циклічність.

7.5 Аналіз залишків LSTM

Рис. 7.5.1



Рекурентна нейронна мережа LSTM застосовується для моделювання часових рядів, що містять довгі залежності, нелінійні взаємодії та складні внутрішні структури. У цьому дослідженні LSTM демонструє відносно добрий рівень точності в порівнянні з іншими нейромережевими підходами, але аналіз залишків свідчить, що модель все ж таки не змогла повністю елімінувати внутрішню циклічність та автокореляційні патерни, властиві енергоспоживанню.

1. Автокореляція залишків (ACF)

ACF для LSTM характеризується:

- дуже високою автокореляцією першого лагу, що свідчить про значну інерцію в помилках;
- чітко вираженою синусоподібною хвильовою структурою на лагах до 40-50;

- регулярними піками приблизно кожні 8-10 лагів, що збігається з субдобовими коливаннями навантаження;
- відсутністю швидкого спадання автокореляції, що характерно для моделей, які не відтворюють тонкі деталі внутрішньодобового циклу.

Таким чином, залишки LSTM демонструють циклічність, притаманну вихідному часовому ряду: мережа змогла частково адаптуватися до сезонності, але суттєвий обсяг структурної інформації залишився непоміченим.

2. Часткова автокореляція (PACF)

PACF показує:

- домінуючий перший лаг, що відповідає базовій AR-компоненті;
- значні негативні часткові автокореляції на лагах 2-4, що вказує на систематичні недооцінки під час різких переходів навантаження;
- наявність поодиноких піків у діапазоні 15-30 лагів - ознака того, що модель періодично «плутається» у складних режимах роботи виробництва;
- поступове згасання після ~30 лагу, але без повного переходу до нульової лінії.

Отже, PACF підтверджує, що LSTM не вдалося усунути короткі послідовні залежності між помилками.

3. Тест Дурбіна–Ватсона

$$DW = 0.742$$

- помітно нижче 2,
- свідчить про сильну позитивну автокореляцію помилок,

- є гіршим, ніж у SVR (1.692) і навіть гіршим, ніж у RF чи XGB.

Попри архітектуру, спеціально призначену для роботи з послідовними даними, LSTM залишає послідовні залежності в помилках, що свідчить про недостатнє охоплення внутрішньої динаміки процесу.

4. Тест Льюнга–Бокса

$$p_{LB}(lag = 24) = 0$$

Це однозначно означає:

- гіпотеза про незалежність залишків відхиляється;
- модель не змогла елімінувати добовий період у помилках;
- залишки містять систематичні патерни, пов'язані з багаторазовими повтореннями режимів роботи обладнання, змінами технологічних процесів, нічними провалами та ранковими піками.

У нейромережових моделях таке часто трапляється, коли архітектура недостатньо глибока або коли модель не має явних сезонних входів (наприклад, сезонних енкодерів).

5. Розподіл залишків

Розподіл залишків LSTM має такі властивості:

- чітко виражений пік навколо нуля → модель добре відтворює середній рівень;
- помітні «важкі хвости» → модель не справляється з аномаліями та різкими стрибками навантаження;
- асиметрія в позитивний бік → у моменти різкого зростання навантаження LSTM, як правило, недооцінює споживання, після чого компенсує помилку із запізненням.

З огляду на те, що LSTM оптимізується під середню похибку, така поведінка є типовою: різкі піки, властиві фабричному профілю, важко моделювати без спеціалізованих механізмів уваги або глибших моделей.

Узагальнююча оцінка залишків LSTM

LSTM виявляє такі характеристики:

Переваги

- добре відтворює загальний тренд та середній рівень споживання,
- прогнозує в більших масштабах часу плавно й стабільно.

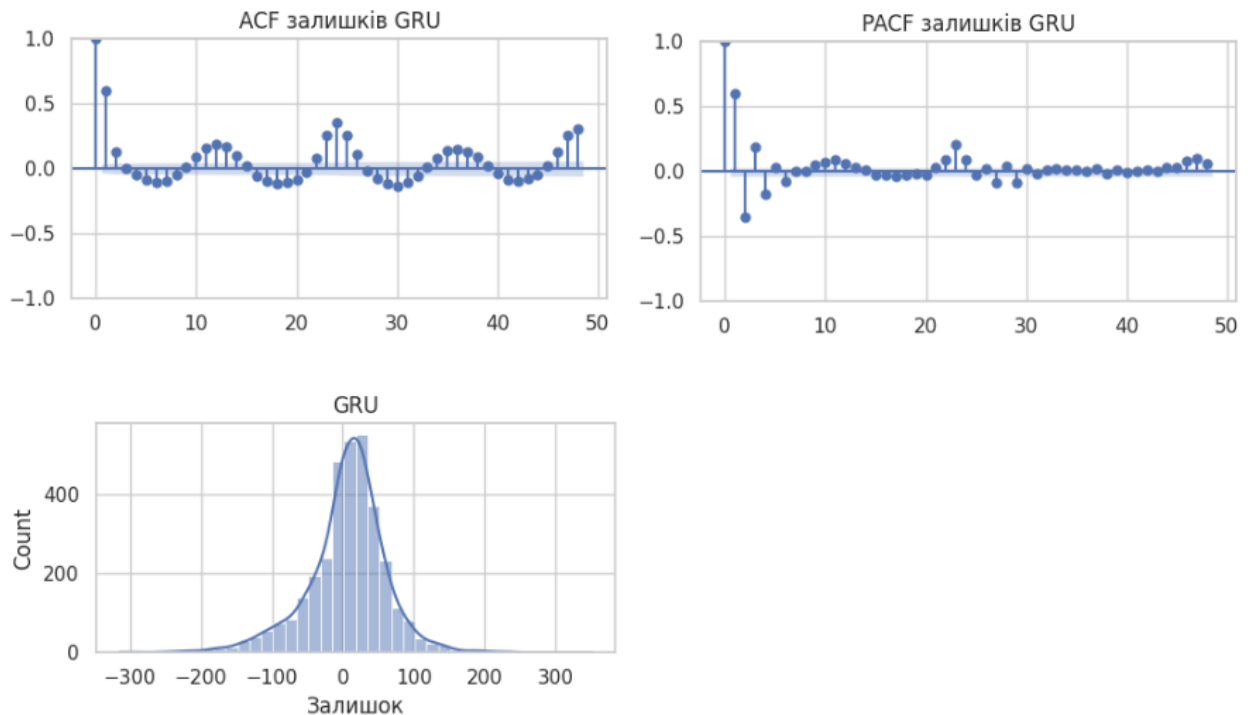
Недоліки

- залишки мають чітку циклічність та високу автокореляцію,
- добові та субдобові структури майже не елімінуються,
- модель недостатньо чутлива до різких змін навантаження,
- присутні довгі послідовні фрагменти недооцінки/переоцінки.

У порівнянні зі SVR LSTM має менш «випадкові» помилки, а у порівнянні з CNN та GRU - виявляє більшу інерцію та менш ефективну декореляцію.

7.6 Аналіз залишків GRU

Рис. 7.6.1



Архітектура GRU, як і LSTM, належить до рекурентних моделей зі станом, однак має спрощений механізм керування пам'яттю. У практичних задачах це часто означає швидше навчання за умови дещо слабших можливостей у фіксації складних довготривалих залежностей. Результати оцінки залишків GRU у цьому дослідженні підтверджують саме таку поведінку: модель відтворює загальну динаміку енергоспоживання, але автокореляційні структури залишаються значними й систематичними.

1. Автокореляція залишків (ACF)

ACF залишків GRU характеризується виразною періодичною хвильовою структурою:

- перший лаг має високу позитивну кореляцію, що свідчить про інерційність помилок;

- подальші лаги (~5-50) демонструють чітку синусоїдальну форму, що повторюється з періодом приблизно 8-10 лагів - це відповідає субдобовим коливанням навантаження, які модель не змогла усунути;
- амплітуда хвилі повільно згасає, що вказує на довготривалі залишкові структури.

Таким чином, GRU, як і LSTM, не повністю декорелює помилки: залишки все ще містять значну частину регулярної циклічної компоненти вихідного ряду.

2. Часткова автокореляція залишків (PACF)

PACF для GRU демонструє:

- домінуючу залежність першого лагу - основний внесок у помилку спричинений безпосередньо попереднім моментом часу;
- помітні негативні значення на лагах 2-4, що вказує на систематичні недооцінки або переоцінки на коротких проміжках, коли споживання швидко змінюється;
- незначні, але регулярні піки на лагах ~15-25, що поєднується з періодичністю ACF та сигналізує про циклічні режими фабрики, які модель не опрацювала повністю;
- поступове згасання після ~30 лагу, але не до рівня «повної випадковості».

PACF підтверджує, що GRU не змогла зняти як короткі, так і середні часові залежності у залишках.

3. Тест Дурбіна-Ватсона

$$DW = 0.800$$

Значення нижче 2 свідчить про позитивну автокореляцію. Рівень $DW=0.8$ вказує:

- на суттєво послідовний характер помилок,
- на більшу автокореляцію, ніж у XGB чи SVR,
- на трохи кращу ситуацію, ніж у LSTM ($DW=0.742$), але все ж далеку від «білих» залишків.

GRU, як і LSTM, демонструє надмірну інерційність, що означає недостатню здатність моделі адаптуватися до раптових змін у режимі енергоспоживання.

4. Тест Льюнга–Бокса

$$p_{LB}(lag = 24) = 0$$

Це однозначно вказує на те, що:

- гіпотеза про незалежність залишків відхиляється,
- добові та субдобові цикли залишаються в структурі помилок,
- GRU не змогла елімінувати сезонність та повторюваність режимів роботи.

У GRU спостерігається така сама проблема, що і в LSTM: модель відтворює рівень споживання, але не виділяє стійкі технологічні патерни.

5. Розподіл залишків (Histogram + KDE)

Розподіл залишків GRU має такі властивості:

- вузький та симетричний пік навколо нуля - GRU добре прогнозує середній рівень;
- важкі хвости - модель недостатньо точно реагує на раптові зміни навантаження, типові для фабрики;
- відсутність вираженої асиметрії - на відміну від LSTM, помилки GRU є дещо більш збалансованими;
- ширші «крила» розподілу, ніж у SVR, що відповідає більшій чутливості до складних режимів роботи.

Отже, GRU працює стабільно в межах нормальних режимів, але менш ефективно при нестандартних ситуаціях.

Узагальнююча оцінка залишків GRU

Характер залишків GRU свідчить про таке:

Переваги

- добрий опис середнього рівня енергоспоживання;
- симетричний та відносно гладкий розподіл помилок;
- дещо краща стабільність, ніж у LSTM.

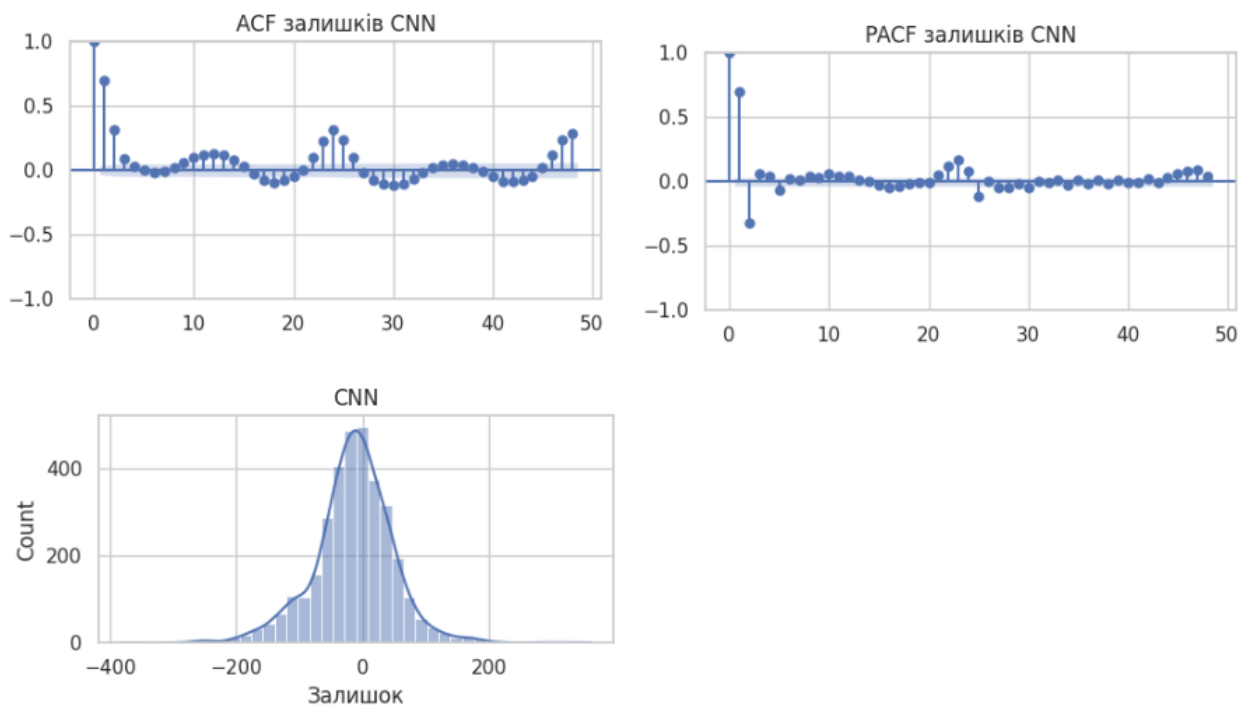
Недоліки

- сильна автокореляція залишків;
- збереження добових патернів у помилках;
- недостатня здатність до прогнозування різких змін у профілі навантаження;
- циклічні хвилі в ACF/PACF, що свідчать про неповне моделювання сезонності.

GRU працює краще за LSTM у частині стабільності залишків, але не досягає рівня дисперсійної чистоти, притаманної SVR.

7.7 Аналіз залишків CNN

Рис. 7.7.1



Конволюційна нейронна мережа (CNN) є архітектурою, що здатна ефективно виділяти локальні патерни та просторово-часові структури в даних. На часових рядах CNN зазвичай добре «ловить» короткі та середньострокові регулярності, але гірше адаптується до тривалої залежності або подій, що відбуваються нерегулярно. Аналіз залишків Конволюційної мережі у цьому дослідженні підтверджує ці властивості: модель добре відтворює основну форму ряду, але залишає значну циклічність та автокореляцію помилок.

1. Автокореляція залишків (ACF)

Графік ACF для CNN демонструє чітку виражену структуру:

- перший лаг має високу позитивну кореляцію, що вказує на інерційність залишків;
- лаги у діапазоні 5-50 демонструють майже синусоїдальний рисунок з періодом близько 8-10 годин - це є характерною ознакою наявності повторюваних виробничих патернів, які мережа не змогла повністю компенсувати;
- амплітуда хвиль повільно зменшується, але не зникає, що є показником збереження структурних залежностей у помилках.

Таким чином, CNN, так само як LSTM та GRU, не забезпечує повного декорелювання помилок і не усуває сезонних компонентів.

2. Часткова автокореляція залишків (PACF)

PACF доповнює оцінку ACF і демонструє такі характеристики:

- перший лаг є домінуючим, що означає, що основна частина систематичних помилок припадає на найкоротший часовий інтервал;
- на лагах 2-4 присутні помітні негативні значення - це свідчить про компенсаційні коливання, коли модель чергує епізоди недооцінки та переоцінки;
- локальні піки на лагах 20-30 відповідають циклам, виявленим у ACF, і є характерною ознакою мультициклічності залишків;
- після 30–35 лагу значення поступово зменшуються, але не стають статистично нульовими.

Це означає, що CNN у певному сенсі відтворює особливості короткострокової структури ряду, але залишає нерозпізнаними повторювані поведінкові шаблони.

3. Тест Дурбіна-Ватсона

$$DW = 0.580$$

Це найнижче значення серед усіх протестованих моделей.

- $DW < 1$ означає сильну позитивну автокореляцію.
- $DW=0.580$ свідчить, що залишки CNN є суттєво послідовними: якщо модель зробила помилку у певний момент, то вона з великою ймовірністю буде повторюватися у наступних сусідніх точках.
- Порівняно з LSTM ($DW=0.742$) та GRU ($DW=0.800$), CNN демонструє найгіршу декореляцію помилок, що означає, що мережа не змогла ефективно усунути інерційні залежності.

Це типово для CNN, оскільки вона зосереджена на локальних патернах і не має внутрішнього механізму довготривалої пам'яті.

4. Тест Льюнга–Бокса

$$p_{LB}(lag = 24) = 0$$

Інтерпретація:

- гіпотеза про випадковість залишків відхиляється;
- залишки демонструють систематичні компоненти, пов'язані з добовою або субдбовою сезонністю;
- CNN не видаляє аспектів періодичності з ряду, навіть якщо частково вловлює його структуру.

Таким чином, CNN не може бути використана як модель, що забезпечує статистично незалежні помилки.

5. Розподіл залишків

Графік розподілу показує:

- симетричний, різко виражений пік поблизу нуля - CNN добре оцінює середній рівень споживання;
- масивні «важкі хвости» розподілу - помилки стають значними за умов аномальних, пікових або нестабільних режимів роботи фабрики;
- форма розподілу ближча до лептокуртичного, ніж до нормального, що відображає схильність CNN до недооцінки рідкісних та різких змін.

Таким чином, CNN прогнозує стабільні ділянки досить добре, але погано реагує на непередбачувані або структурно складні події.

Узагальнююча оцінка залишків CNN

Сильні сторони

- добрий опис короткострокових патернів;
- симетричний розподіл помилок;
- висока точність на стабільних відрізках.

Слабкі сторони

- найвища автокореляція серед усіх моделей;
- чітко виражена сезонність у залишках;
- слабка реакція на різкі зміни навантаження;
- модель не формує незалежні помилки, що обмежує можливість застосування у задачах, де необхідна строгість статистичних припущень.

CNN демонструє найгірші показники за якістю залишків серед усіх глибинних моделей, що узгоджується з її архітектурою: локальні

фільтри добре «ловлять» шаблони, але не забезпечують гнучкості, властивої RNN-моделям.

CNN не може бути рекомендована як самостійна модель для прогнозування енергоспоживання на основі цього дослідження. Її інерційність та висока автокореляція роблять її менш придатною, ніж SVR або градієнтні методи.

РОЗДІЛ 8. ПОРІВНЯЛЬНИЙ АНАЛІЗ МОДЕЛЕЙ

8.1 Порівняння точності прогнозу

На першому етапі порівняння моделей розглянемо інтегральні показники точності - MAE, RMSE, MAPE та коефіцієнт детермінації R^2 .

Лінійні моделі роду AR, MA, ARIMA, SARIMA та SARIMAX демонструють найбільші похибки: MAE знаходиться в інтервалі 175-200 кВт·год, RMSE - 240–270 кВт·год, а значення R^2 є від'ємними.

Від'ємний R^2 означає, що модель апроксимує дані гірше за тривіальний базовий прогноз (наприклад, прогноз «за середнім» або «наступне дорівнює попередньому»). Це свідчить про фундаментальну невідповідність класичних лінійних схем структурі досліджуваного часового ряду, у якому наявні різкі зміни рівня, добові та тижневі цикли, вплив аварійних та технологічних зупинок.

На противагу цьому, моделі машинного навчання та глибинного навчання демонструють істотно кращу якість. Серед них:

- SVR має найменші MAE та RMSE і найбільше $R^2 \approx 0,966$. Це означає, що модель відтворює понад 96 % дисперсії фактичного споживання на тестовій вибірці.
- XGBoost та Random Forest дещо поступаються SVR, але зберігають дуже високу точність: MAE на рівні 32–37 кВт·год, R^2 близько 0,95–0,96. Для практичних потреб це також можна вважати високоякісним прогнозом.
- LSTM та GRU демонструють середній рівень точності: MAE близько 42–44 кВт·год, R^2 на рівні 0,93–0,94. Вони істотно кращі за лінійні моделі, але помітно поступаються SVR та деревним ансамблям.

- CNN у поточній архітектурі є найслабшою серед нелінійних моделей: MAE ≈ 56 кВт·год, $R^2 \approx 0,90$. Мережа все ще описує основну тенденцію, але робить це менш точно.

Отже, з точки зору суто кількісних метрик похибки, без урахування структури залишків, перші позиції посідають SVR, XGBoost та Random Forest, тоді як лінійні ARIMA-моделі є непридатними для такого роду даних.

8.2 Порівняння структури впливів (Feature Importance та SHAP)

Аналіз важливості ознак для усіх нелінійних моделей показує узгоджену картину.

Для Random Forest, XGBoost та SVR домінуючим чинником є безпосередньо попередній вимір споживання *kWh_lag_1*. За SHAP-аналізом, саме ця ознака має найбільшу абсолютну величину внеску в прогноз; значно менший, але все ще суттєвий вплив мають *kWh_lag_2*, *kWh_lag_3*і ковзне середнє за добу *roll_mean_24*. Внесок екзогенних змінних (температура, індикатори тривоги, ознака простою *is_downtime*) для цих моделей є невеликим. Це означає, що дерева рішень та SVR формують прогноз переважно на основі короткострокової інерції споживання і слабо використовують додатковий контекст.

Глибинні моделі (LSTM, GRU, CNN) також віддають перші позиції лагам споживання, але розподіл важливості є дещо ширшим. LSTM та GRU виокремлюють *kWh_lag_1*, *kWh_lag_2*, *roll_mean_24*і додатково помітно використовують час доби (*hour*, *hour_sin*, *hour_cos*), температуру та лаги частки часу в режимі тривоги. Для CNN особливістю є відносно високий внесок ознаки *is_downtime*, що

відповідає технологічним зупинкам, коли споживання різко падає. Це свідчить про те, що згорткова мережа фактично навчилася розрізняти режими «нормальної роботи» і «простою», тоді як для SVR та дерев цей режим відображається переважно через зміну самих значень споживання, а не через бінарний індикатор.

Загалом, аналіз важливості ознак показує, що всі успішні моделі розглядають часовий ряд як сильно автокорельований процес, де найважливішою є власна історія споживання, а погодні та аварійні фактори відіграють роль коригувальних, але не визначальних чинників. Водночас CNN краще за інші моделі використовує ознаку простою, що важливо для практичної інтерпретації, хоча це не призвело до мінімізації глобальних метрик похибки.

8.3 Порівняння залишків моделей

Для коректної оцінки придатності моделей до експлуатації важливо аналізувати не лише рівень похибки, а й структуру залишків. Ідеальний залишок повинен наближатися до білого шуму: бути некорельованим у часі, мати нульове середнє та приблизно симетричний розподіл.

ACF та PACF залишків для всіх шести моделей (SVR, XGBoost, Random Forest, LSTM, GRU, CNN) демонструють наявність залишкової автокореляції, особливо на лагах, кратних 24 годинам. Це свідчить про те, що жодна модель повністю не вичерпала добову структуру ряду. Для Random Forest, XGBoost, LSTM, GRU та CNN значення функції автокореляції на малих лагах залишаються суттєво додатними, а Durbin-Watson-статистика набуває значень у діапазоні 0,58-0,9, що відповідає вираженій позитивній автокореляції.

Модель SVR у цьому сенсі є найбільш «близькою» до бажаного білошумового залишку: Durbin–Watson $\approx 1,69$, тобто суттєво ближче до теоретично ідеального значення 2, ніж інші моделі. ACF для SVR має менші амплітуди на більшості лагів, однак Ljung-Box-тест з лагом 24 дає надзвичайно мале значення p -рівня (порядку 10^{-164}), що дозволяє відхилити гіпотезу про відсутність автокореляції. Отже, навіть SVR залишає в похибці частину структур регулярного характеру (насамперед добові цикли та повільні трендові компоненти), хоча ці структури виражені слабше, ніж для інших моделей.

Гістограми з накладеними KDE-оцінками показують, що розподіли залишків для всіх моделей є приблизно симетричними навколо нуля і близькими до нормального, але мають деяке загострення центральної частини та «важкі» хвости. Це типовий ефект для енергетичних часових рядів, де трапляються поодинокі великі відхилення, пов'язані з аваріями, нетиповими технологічними режимами або неочікуваними простоями.

Таким чином, з погляду аналізу залишків жодна модель не забезпечує повністю «білий» шум, але SVR найкраще пригнічує короткострокову автокореляцію, тоді як нейронні мережі та ансамблі дерев залишають у похибці більш виражену циклічну компоненту.

8.4 Узагальнені висновки та оцінка придатності моделей

Поєднання трьох рівнів аналізу - метрик точності, важливості ознак та поведінки залишків - дозволяє сформулювати узагальнені висновки щодо придатності розглянутих моделей для прогнозування споживання електроенергії промисловим підприємством.

1. Класичні лінійні моделі AR/MA/ARIMA/SARIMA/SARIMAX

Високі значення MAE та RMSE разом із від'ємним R^2 свідчать, що ці моделі не пристосовані до неоднорідного, нестационарного та різнорежимного часового ряду фабрики. Їхня основна передумова, приблизна стаціонарність та лінійність залежностей порушується через наявність численних технологічних змін, сезонних ефектів, впливу тривог та аварійних відключень. Лінійний підхід виявляється недостатнім для адекватної апроксимації такого процесу.

2. Ансамблі дерев (Random Forest, XGBoost)

Ці моделі забезпечують високу точність прогнозу та добре відтворюють нелінійні залежності між лагами споживання й екзогенними факторами. За SHAP-аналізом вони чітко виділяють ключові лаги споживання та частково реагують на режимні ознаки. Однак виражена автокореляція залишків (низькі значення Durbin-Watson, значущі результати Ljung-Box-тесту) свідчить, що дерева не повністю вичерпують часову структуру процесу й залишають у похибці періодичні компоненти. Такі моделі добре підходять для створення базової системи короткострокового прогнозу з високою точністю, проте для задач аналізу динаміки залишків або побудови довгострокових сценаріїв їх слід доповнювати іншими методами.

3. Модель Support Vector Regression (SVR)

SVR демонструє найкраще поєднання низької похибки та відносно «найбільш випадкового» характеру залишків. Вона ефективно використовує локальну інерцію ряду, формуючи гладкий, але достатньо гнучкий прогноз. При цьому модель майже не використовує екзогенні ознаки, що видно з аналізу важливості ознак, - основний внесок припадає на найкоротші

лаги споживання. З одного боку, це пояснює високу точність на горизонті один крок; з іншого - обмежує здатність моделі реагувати на зміни режиму, що не були відображені у найближчій історії. У практичному застосуванні SVR можна розглядати як опорну базову модель, до якої додаються більш інтерпретовані або спеціалізовані компоненти (наприклад, моделі детекції та окремого прогнозу періодів простою).

4. Глибинні моделі (LSTM, GRU, CNN)

Попри те, що рекурентні та згорткові мережі теоретично здатні моделювати складні довгострокові залежності, у проведеному експерименті вони не перевершили більш прості методи. LSTM і GRU показали прийнятну, але не найкращу точність та зберегли виражену автокореляцію залишків. CNN краще за інші моделі «бачить» зміну режиму через ознаку `is_downtime`, що корисно з точки зору інтерпретації, однак у підсумку поступається за точністю як SVR, так і ансамблям дерев. Це може бути наслідком обмеженого обсягу навчальної вибірки, чутливості до налаштувань гіперпараметрів та відсутності спеціалізованої архітектури, оптимізованої саме під промислові енергетичні ряди з різкими переходами між режимами.

5. Порівняння з практичної точки зору

Якщо орієнтуватися лише на середню похибку, SVR є найкращим кандидатом для впровадження. Однак з огляду на наявність структурної автокореляції в залишках, а також зважаючи на потреби інтерпретованості й чутливості до режимів роботи обладнання, доцільним видається поєднати SVR з XGBoost або Random Forest у вигляді ансамблю або багаторівневої схеми прогнозування (наприклад, окреме прогнозування базового рівня та корекційних компонент). Глибинні моделі LSTM/GRU/CNN у

поточному вигляді доцільно розглядати як перспективний напрямок для подальших досліджень (зокрема, для довгих горизонтів прогнозу або моделювання сценаріїв поведінки при тривалих аварійних режимах), але не як основний інструмент оперативного прогнозування.

Підсумовуючи, можна стверджувати, що для досліджуваного неоднорідного часового ряду споживання електроенергії найкращі результати дають нелінійні моделі з короткою пам'яттю (SVR, ансамблі дерев), тоді як лінійні ARIMA-моделі виявляються непридатними через невідповідність своїх припущень реальній динаміці процесу. Аналіз залишків показує, що жодна модель не усуває повністю автокореляційну структуру, тому подальші дослідження доцільно спрямувати на комбінування опорної моделі високої точності з окремими компонентами, які описують добові цикли, тривалі простої та інші структурні особливості енергоспоживання фабрики.

РОЗДІЛ 9. ЕКОНОМІЧНЕ ОБҐРУНТУВАННЯ ЗАСТОСУВАННЯ ПРОГНОЗНОЇ МОДЕЛІ ДЛЯ ОПТИМІЗАЦІЇ ЗАКУПІВЛІ ЕЛЕКТРОЕНЕРГІЇ

Ефективне управління закупівлею електричної енергії в умовах лібералізованого ринку потребує не лише технічного прогнозування споживання, але й кількісної оцінки економічних наслідків застосування прогнозних моделей у процесі формування портфеля закупівель. Відмінності між цінами на ринку «на добу наперед» (РДН) та внутрішньодобовому ринку (ВДР), а також ризики виникнення небалансів зумовлюють необхідність формалізації стратегії, що дозволяє мінімізувати сумарні витрати суб'єкта господарювання. У цьому розділі здійснюється комплексний економічний аналіз застосування розробленої прогнозної моделі на основі Support Vector Regression (SVR) у задачі оптимізації закупівлі електроенергії для промислового підприємства.

9.1 Методологія моделювання економічного ефекту

В основу розрахунку покладено часовий інтервал з 3 червня по 22 жовтня 2025 року, для якого побудовано прогнозні значення споживання електроенергії та отримано фактичні погодинні ціни РДН і ВДР з офіційних даних Оператора ринку. Основним припущенням є наявність моделі, здатної формувати прогноз споживання з прийнятною точністю, та гіпотетичної моделі прогнозування цін, що дозволяє оцінювати відносне співвідношення між цінами РДН та ВДР. У рамках економічного експерименту для кожної години обрано ринок з нижчою ціною, проте з урахуванням ймовірності помилки у виборі ринку. Оскільки реальні моделі прогнозування цін характеризуються

ненульовою ймовірністю неправильного визначення більш вигідного ринку, у моделюванні закладено припущення про 10% випадкових помилок класифікації, рівномірно розподілених у часі. Таким чином, отриманий синтетичний розподіл закупівель відображає не ідеальний, а більш реалістичний сценарій прийняття рішень.

Синтетична траєкторія закупівлі використовується для порівняння двох стратегій:

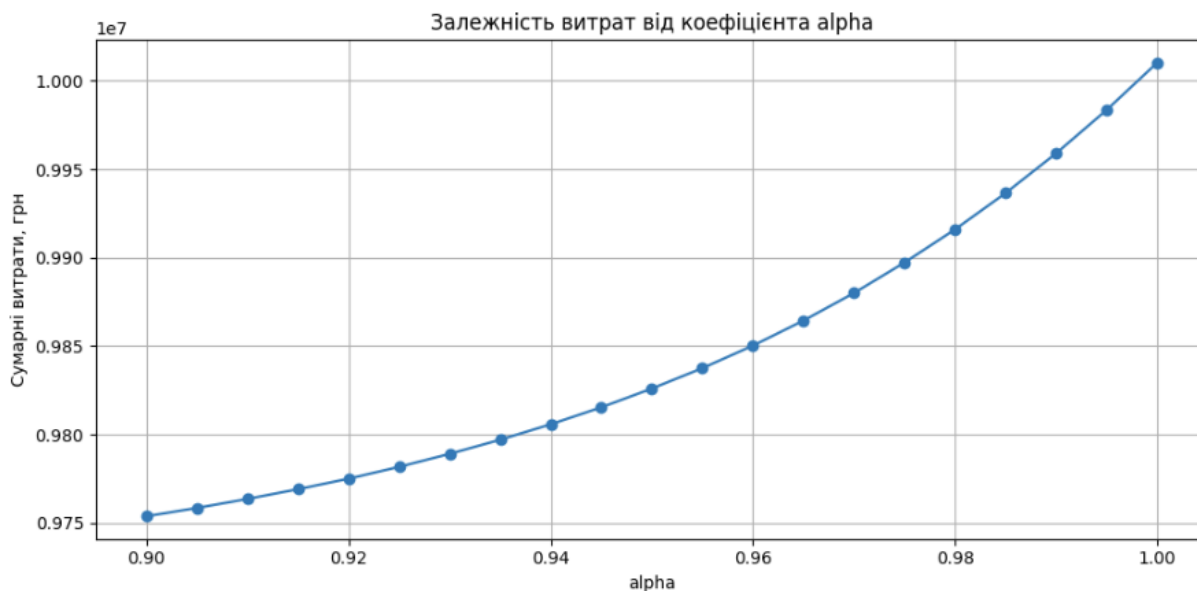
1. сценарій А - повна закупівля всієї електроенергії на ВДР;
2. сценарій В - комбінована закупівля РДН+ВДР згідно із синтетичною моделлю вибору ринку та з урахуванням прогнозного значення споживання.

Додатково враховано корекцію прогнозу споживання для мінімізації від'ємного небалансу, що часто призводить до додаткових витрат через докупівлю електроенергії за підвищеними цінами.

9.2 Моделювання небалансів та необхідність коригування прогнозу

У практиці енергетичного трейдингу найбільш критичною є ситуація, коли прогноз споживання перевищує фактичний рівень. За таких умов підприємство закуповує більший обсяг, ніж споживає, і надлишок може бути реалізований за менш вигідними цінами. Водночас недобір (коли фактичне споживання перевищує прогнозоване) також спричиняє додаткові витрати у вигляді закупівлі бракуючих обсягів на ВДР. Таким чином, коректне балансування між ризиком перебору та недобору є ключовою задачею.

Рис. 9.1



Для зменшення від'ємного небалансу прогнозовані значення були масштабовані на коефіцієнт α ($0.9 \leq \alpha \leq 1.0$). Фактично це означає застосування консервативної оцінки споживання, яка знижує ймовірність завищення прогнозу. Для кожного значення α було обчислено сумарні витрати підприємства з урахуванням:

- вартості закупівлі обсягів електроенергії на РДН або ВДР (залежно від синтетичного вибору ринку);
- вартості докупівлі недобору на ВДР;
- впливу зміни масштабу прогнозу на величину небалансів.

Результати аналізу наведено на графіку залежності сумарних витрат від коефіцієнта α . Мінімальні сумарні витрати спостерігаються при $\alpha = 0.90$, що свідчить про доцільність застосування консервативного прогнозу у межах 10% від номінального значення.

9.3 Розрахунок витрат за альтернативними стратегіями закупівлі

У межах дослідження для наведеного періоду обчислено дві ключові величини:

- сумарні витрати за умов закупівлі всієї електроенергії на ВДР (базовий сценарій),
- сумарні витрати за оптимізованим підходом РДН+ВДР з урахуванням прогнозу та корекції α .

За результатами моделювання отримано:

- витрати у разі повної закупівлі на ВДР:
9 853 518.29 грн;
- витрати за оптимізованою стратегією:
9 722 432.43 грн.

Сумарна економія становить:

$$\Delta = 9\,853\,518.29 - 9\,722\,432.43 = 131\,085.86 \text{ грн.}$$

Рис. 9.2



Це показує, що навіть за умов неповної точності прогнозу та ймовірності помилки у виборі ринку застосування прогнозної моделі дозволяє отримати економічний ефект порівняно з консервативним сценарієм. Графік помісячної економії демонструє нерівномірний

характер заощаджень, що відповідає коливанням обсягів споживання та волатильності цін на ринках РДН і ВДР.

Максимальні значення економії спостерігаються для періодів з підвищеною різницею між цінами РДН та ВДР, а також для тих місяців, у яких дисбаланс був найменшим завдяки впливу коефіцієнта α .

9.4 Інтерпретація отриманих результатів

Результати симуляції свідчать про наявність об'єктивного потенціалу для економії за умов застосування прогнозної моделі та оптимізованої стратегії закупівлі електроенергії. Зменшення витрат на 131 тис. грн протягом обмеженого періоду вказує на доцільність масштабування такого підходу на річний горизонт планування, що забезпечить пропорційно більший економічний ефект. Ключовим чинником досягнення економії є не стільки абсолютна точність прогнозу, скільки правильне управління ризиками небалансів і здатність моделі адаптуватися до ринкової структури цін.

Отримані результати також демонструють чутливість сумарних витрат до вибору масштабу прогнозу. Зменшення прогнозного споживання на 10% дозволило істотно знизити величину позитивного небалансу, що формує додаткові витрати при надлишкових закупівлях.

9.5 Обмеження моделювання

Попри позитивний економічний ефект, методологія має низку обмежень, що обумовлені передусім дослідницьким характером моделі:

1. Використана ймовірність помилки у виборі ринку (10%) є наближеним припущенням і може відрізнятись від реальної частоти помилок моделі прогнозування цін.
2. У розрахунку не враховано фактичні правила роботи балансуючого ринку та можливі штрафні механізми, що в окремі періоди можуть збільшувати фактичну вартість небалансів.
3. Порівняння проводиться з базовим сценарієм закупівлі на ВДР; альтернативним baseline міг би виступати також сценарій повної закупівлі на РДН.
4. Результати моделювання мають характер ретроспективної симуляції та не гарантують аналогічного ефекту при forward-looking застосуванні моделі.

Однак ці обмеження не нівелюють отриманих висновків, а вказують на напрямки подальшого удосконалення моделі у межах наступних етапів дослідження.

9.6 Висновки щодо економічної доцільності застосування прогнозної моделі

Проведений економічний аналіз підтверджує, що використання прогнозних моделей у задачі планування закупівлі електроенергії дозволяє знизити сумарні витрати підприємства завдяки більш ефективному розподілу обсягів між ринками РДН і ВДР та оптимізації рівня небалансів. Оптимальний масштаб прогнозу ($\alpha = 0.90$) забезпечує мінімізацію сукупних витрат, а аналіз чутливості

демонструє плавне зростання витрат при відхиленні цього коефіцієнта у бік збільшення.

Отримані результати свідчать про реалістичність застосування розробленої методики в умовах енергетичного ринку України та її потенціал для подальшої інтеграції в автоматизовані системи енергоменеджменту підприємств. Сукупна економія за період дослідження становить 131 тис. грн, що підтверджує економічну вигоду впровадження прогнозної моделі навіть у випадку неповної точності прогнозу та наявності ризиків помилкових рішень під час вибору ринку закупівлі.

ВИСНОВКИ

Аналіз часових рядів споживання електроенергії показав, що такі процеси загалом характеризуються поєднанням різких коливань, нерівномірності, режимних переходів та складних нелінійних залежностей. Визначальну роль у формуванні їхньої динаміки відіграють короткострокові лаги та індикатори подій, що різко змінюють рівень споживання, тоді як типові календарні та зовнішні чинники, як правило, мають обмежений вплив. Ця властивість виявляється у широкому класі промислових та інфраструктурних об'єктів і формує ключове методологічне обмеження для застосування моделей, що ґрунтуються на припущеннях про стаціонарність та лінійність.

Порівняльний аналіз моделей продемонстрував чіткий розподіл між підходами, здатними адекватно відтворювати структуру таких рядів, та методами, чия ефективність виявляється суттєво нижчою. Лінійні моделі продемонстрували значну похибку та нездатність врахувати різкі коливання та нелінійні залежності; ця тенденція є системною для часових рядів, у яких спостерігаються часті переходи між режимами. Натомість моделі машинного та глибокого навчання загалом забезпечили істотно кращу точність, що підтверджує їхню здатність адаптуватися до складнішої внутрішньої структури даних.

Найстабільніші результати показала модель Support Vector Regression, яка забезпечила найменші похибки та найвищу частку поясненої дисперсії. Характерною особливістю цього підходу є чутливість до локальних патернів та здатність ефективно узагальнювати залежності навіть за умов високої варіативності даних. Ансамблеві моделі продемонстрували високу гнучкість у моделюванні нелінійних взаємозв'язків, але водночас виявили потенційну схильність залишати у прогностичних помилках короткострокові циклічні компоненти. Нейронні мережі забезпечили суперечливі результати: рекурентні архітектури

краще працювали з тривалими залежностями, а згорткові - з локальними структурами, однак жодна з архітектур не усунула повністю сезонні та внутрішні циклічні коливання.

Структурний аналіз залишків виявив важливу закономірність: незалежно від моделі, похибки прогнозу зберігають періодичні компоненти та автокореляцію першого порядку. Це свідчить про те, що енергетичні часові ряди містять систематичні патерни, які не повністю охоплюються навіть найточнішими моделями. З цього випливає перспективність комбінованих або багаторівневих підходів, що поєднують нелінійні моделі з окремими механізмами моделювання сезонності.

Інтерпретаційний аналіз за SHAP-показниками підтвердив домінування лагових ознак над календарними чи зовнішніми предикторами та дозволив встановити сталі взаємозв'язки, характерні для складних енергетичних процесів: ближні лаги мають найбільший прогнозний потенціал, тоді як інформація про період доби, місяць чи погоду часто виявляється другорядною. Це узагальнення може бути релевантним і для інших галузей, де процеси формуються переважно внутрішніми динамічними механізмами.

Економічна інтерпретація результатів продемонструвала, що моделі з високою точністю прогнозу мають потенціал знижувати витрати за рахунок кращого вибору між альтернативними ринковими стратегіями. Хоча величина отриманого ефекту залежить від конкретної структури цін, коливань споживання та припущень щодо ймовірності помилок, загальна тенденція залишається сталою: точні короткострокові прогнози можуть підвищувати ефективність управління енергетичними ресурсами. Разом з тим, частина економічних висновків є контекстно залежною і не може бути безпосередньо перенесена на інші об'єкти без повторного моделювання.

Таким чином, результати дослідження дозволяють узагальнити такі принципові положення:

- моделі машинного навчання, зокрема SVR, є найбільш придатними для прогнозування нерівномірних, нелінійних та структурно складних енергетичних процесів;
- факторна структура таких рядів визначається насамперед лаговими взаємозв'язками та подієвими характеристиками, тоді як зовнішні календарні та погодні ознаки часто мають мінімальну пояснювальну силу;
- залишки прогнозів зберігають циклічні компоненти навіть у разі застосування найточніших моделей, що свідчить про обмеженість окремих архітектур і потребу в гібридних підходах;
- економічний ефект застосування точних прогнозів може бути суттєвим, однак залежить від контексту, структури цін та операційних умов конкретного ринку;
- частина висновків має універсальний характер, але окремі кількісні результати залишаються специфічними для аналізованого об'єкта та не повинні інтерпретуватися без урахування його особливостей.

У сукупності ці результати відображають принципові закономірності прогнозування складних енергетичних процесів та окреслюють перспективні напрями подальших досліджень, зокрема щодо побудови гібридних моделей, вдосконалення інтерпретаційних методів та інтеграції прогнозних систем у контури прийняття управлінських рішень.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Anbazhagan S., Kumarappan N. Day-ahead electricity price forecasting using neural network with discrete cosine transform. *Energy Conversion and Management*, 2014, vol. 78, pp. 711–719.

2. Bergmeir C., Benítez J. On the use of cross-validation for time series forecasting.

Режим доступу: <https://arxiv.org/pdf/1708.00720.pdf> (дата звернення: 11.12.2025).

3. Chen T., Guestrin C. XGBoost: A Scalable Tree Boosting System. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2016, pp. 785–794.

Режим доступу: <https://arxiv.org/pdf/1603.02754.pdf> (дата звернення: 11.12.2025).

4. Черненко П. О., Мартинюк О. В., Мірошник В. О. Аспекти короткострокового прогнозування електричного навантаження системи енергопостачання з вагомою складовою промислового електроспоживання. *Дослідження Інституту електродинаміки НАН України*, 2016, вип. 43, с. 24–31.

5. Чернишова Г. Ю., Самаркіна Е. А. Методи інтелектуального аналізу даних для прогнозування фінансових часових рядів. *Известия Саратовского университета. Новая серия. Серия: Экономика. Управление. Право*, 2019, т. 19, № 2, с. 181–188.

6. Goodfellow I., Bengio Y., Courville A. *Deep Learning*. Cambridge: MIT Press, 2016. 800 p.

Режим доступу: <https://www.deeplearningbook.org/> (дата звернення: 11.12.2025).

7. Hochreiter S., Schmidhuber J. Long Short-Term Memory. *Neural Computation*, 1997, vol. 9, no. 8, pp. 1735–1780.

Режим доступу: <https://www.bioinf.jku.at/publications/older/2604.pdf> (дата звернення: 11.12.2025).

8. Hyndman R., Athanasopoulos G. *Forecasting: Principles and Practice*. 3rd ed. Melbourne: OTexts, 2021. 412 p.

Режим доступу: <https://otexts.com/fpp3/> (дата звернення: 11.12.2025).

9. Державна служба статистики України. Енергетична статистика.

Режим доступу: https://ukrstat.gov.ua/operativ/menu/menu_u/energ.htm (дата звернення: 11.12.2025).

10. НЕК «Укренерго». Оперативні дані електричного навантаження України.

Режим доступу: <https://ua.energy/operational-data/> (дата звернення: 11.12.2025).

11. Національна комісія, що здійснює державне регулювання у сферах енергетики та комунальних послуг. Аналітичні матеріали щодо ринку електричної енергії.

Режим доступу: <https://www.nerc.gov.ua/monitoring-rinku-elektrichnoyi-energiyi> (дата звернення: 11.12.2025).

12. Оператор ринку. Історичні дані РДН та ВДР: погодинні індекси.

Режим доступу: <https://www.oree.com.ua/index.php/indexes/indexes> (дата звернення: 11.12.2025).

13. Park D. C., El-Sharkawi M. A., Marks R. J. II, Atlas L. E., Damborg M. J. Electric load forecasting using an artificial neural network. *IEEE Transactions on Power Systems*, 1991, vol. 6, no. 2, pp. 442–449.

14. Шараєвський Ф. Ю. Модель та процедури короткострокового прогнозування електроспоживання на оптовому ринку електроенергії. *Системні дослідження та комплексні проблеми енергетики*, 2007, с. 36–40.

15. Солов'єва І. А., Дзюба А. П. Прогнозування електроспоживання з урахуванням факторів технологічного та ринкового середовища. *Науковий діалог*, 2013, с. 97–113.

16. Стрелкова Г. Г., Іщенко О. С. *Системи енергетичного менеджменту: прогнозування енергетичного попиту*: навчальний посібник. Київ: КПІ ім. Ігоря Сікорського, 2020. 61 с.

Режим доступу: <https://ela.kpi.ua/bitstream/123456789/33652/1/292-1.pdf> (дата звернення: 11.12.2025).

17. Voronin S., Partanen J. Hybrid wavelet–ARIMA–ANN model for electricity market price forecasting. *International Journal of Energy Research*, 2014, vol. 38, no. 5, pp. 626–637.

18. Zhou L., Wang B., Wang Z., Wang F., Yang M. Seasonal classification and parallel combined method for ahead price forecasting based on adaptive RBF. Proceedings of the International Conference on Energy and Power Engineering, 2018, 5 p.