

Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет _____ Центр післядипломної освіти
(повна назва)

Кафедра _____ Штучного інтелекту
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА
Пояснювальна записка

рівень вищої освіти _____ перший (бакалаврський)

Прогнозування погодинного споживання електроенергії в Об'єднаній
енергетичній системі України на основі адаптивної нейро-нечіткої системи
виведення (ANFIS)
(тема)

Виконав:
здобувач _____ другого _____ року навчання,
групи _____ ІТШІпз-24-1
Олег БОНДАРЕНКО
(власне ім'я, прізвище)

Спеціальність 122 Комп'ютерні науки
(код і повна назва спеціальності)

Тип програми _____ освітньо-професійна
Освітня програма _____ Штучний інтелект
(повна назва освітньої програми)

Керівник _____ ст.викл. Ігор БІБІЧКОВ
(посада, власне ім'я, прізвище)

Допускається до захисту

Завідувач кафедри ШІ

(підпис)

Лариса ЧАЛА
(власне ім'я, прізвище)

2026 р.

Харківський національний університет радіоелектроніки

Факультет _____ Центр післядипломної освіти _____

Кафедра _____ Штучного інтелекту _____

Рівень вищої освіти _____ перший (бакалаврський) _____

Спеціальність _____ 122 Комп'ютерні науки _____
(код і повна назва)

Тип програми _____ освітньо-професійна _____

Освітня програма _____ Штучний інтелект _____
(повна назва)

ЗАТВЕРДЖУЮ:

Зав. кафедри _____

(підпис)

«_____» _____ 20 ____ р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

здобувачеві _____ Бондаренку Олегу Васильовичу _____
(прізвище, ім'я, по батькові)

1. Тема роботи Прогнозування погодинного споживання електроенергії в Об'єднаній енергетичній системі України на основі адаптивної нейро-нечіткої системи виведення (ANFIS)

затверджена наказом університету від 18 травня 2026 р. № 77Стз

2. Термін подання здобувачем роботи до екзаменаційної комісії 16 червня 2026 р.

3. Вихідні дані до роботи Офіційна документація та посібники для програмування мовою Python та використання її бібліотек, документація фреймворку FastAPI, офіційна документація бібліотеки scikit-fuzzy для роботи з нечіткою логікою, документація бібліотеки React 18 та фреймворку Tailwind CSS 3, наукові публікації з теорії нечітких множин (Л. Заде), нечіткого виведення Мамдані та Сугено, адаптивних нейро-нечітких систем (Дж.-С. Р. Янг), історичні дані погодинного споживання електроенергії в ОЕС України за 2017–2021 роки з відкритих джерел НЕК «Укренерго», метеорологічні дані з API Open-Meteo, статистичні матеріали ENTSO-E Transparency Platform, наукові публікації з прогнозування електроспоживання та веб-ресурси

4. Перелік питань, що потрібно опрацювати в роботі. _____

1) Аналіз предметної галузі та постановка задач _____

2) Теоретичні дослідження _____

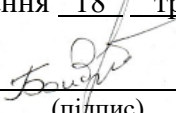
3) Проєктування системи _____

4) Експериментальні дослідження _____

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Строк / термін виконання етапів роботи	Примітка
1	Отримання завдання на кваліфікаційну роботу	18.05.2026	виконано
2	Аналіз предметної галузі	21.05.2026	виконано
3	Теоретичні дослідження ANFIS	22.05.2026	виконано
4	Проектування архітектури системи	22.05.2026	виконано
5	Реалізація серверної частини (FastAPI + ANFIS	22.05.2026	виконано
6	Реалізація клієнтської частини (React)	24.05.2026	виконано
7	Розгортання на хмарних платформах	25.05.2026	виконано
8	Експериментальні дослідження	20.05.2026	виконано
9	Оформлення пояснювальної записки	09.06.2026	виконано
10	Підготовка до захисту	12.06.2026	виконано
11	Захист перед екзаменаційною комісією	17.06.2026	

Дата видачі завдання 18 травня 2026 р.

Здобувач 
(підпис)

Керівник роботи _____
(підпис)

ст.викл. Ігор БІБІЧКОВ
(посада, власне ім'я, прізвище)

РЕФЕРАТ

Пояснювальна записка: 90 с., 23 рис., 4 табл., 4 дод., 25 джерел.

ВЕБ-ДОДАТОК, НЕЙРО-НЕЧІТКА СИСТЕМА, НЕЧІТКА ЛОГІКА, ОЕС УКРАЇНИ, ПРОГНОЗУВАННЯ ЕЛЕКТРОСПОЖИВАННЯ, ANFIS, ENTSO-E, FASTAPI, REACT.

Об'єкт дослідження – процес погодинного прогнозування електроспоживання в Об'єднаній енергетичній системі України в умовах варіативних метеорологічних та календарних факторів.

Предмет дослідження – методи нейро-нечіткого моделювання та програмні засоби побудови веб-системи прогнозування електроспоживання на основі адаптивної нейро-нечіткої системи виведення (ANFIS).

Мета роботи – підвищити точність та інтерпретованість короткострокового прогнозу електроспоживання в ОЕС України шляхом розробки веб-системи на основі ANFIS з можливістю інтерактивного сценарного аналізу та пояснення кожного прогнозу через візуалізацію активних нечітких правил.

Методи дослідження – теорія нечітких множин, нечітке виведення Сугено, нейро-нечітке моделювання (ANFIS), Ridge regression для підбору консеквентів, time-series split для валідації, статистичний аналіз часових рядів, порівняльна оцінка моделей за метриками MAPE, RMSE, MAE.

Розроблено клієнтсерверну веб-систему Energy Forecast UA з моделлю ANFIS (55 правил, 7 лінгвістичних змінних). Навчання моделі виконано на історичних даних споживання 2017–2021 років. Експериментальне тестування показало MAPE 6,17% на тестовій вибірці 2021 року, що перевершує референсні моделі LSTM, Prophet та SARIMAX. Систему розгорнуто на платформах Vercel та Render.com з публічним доступом.

ABSTRACT

Bachelor's thesis contains: 90 pp., 23 fig., 4 tabl., 4 ann., 25 references.

ANFIS, ELECTRICITY CONSUMPTION FORECASTING, ENTSO-E, FASTAPI, FUZZY LOGIC, INTEGRATED POWER SYSTEM OF UKRAINE, NEURO-FUZZY SYSTEM, REACT, WEB APPLICATION.

Object of research – the process of hourly electricity consumption forecasting in the Integrated Power System of Ukraine under variable meteorological and calendar factors.

Subject of research – methods of neuro-fuzzy modeling and software tools for building a web-based electricity consumption forecasting system based on the Adaptive Neuro-Fuzzy Inference System (ANFIS).

Aim of the work – to improve the accuracy and interpretability of short-term electricity consumption forecasting in the IPS of Ukraine by developing a web-based system based on ANFIS with the capability of interactive scenario analysis and explanation of each forecast through visualization of active fuzzy rules.

Research methods – fuzzy set theory, Sugeno-type fuzzy inference, neuro-fuzzy modeling (ANFIS), Ridge regression for consequent parameter estimation, time-series split for validation, statistical analysis of time series, comparative evaluation of models using MAPE, RMSE, MAE metrics.

A clientserver web-based system Energy Forecast UA was developed with an ANFIS model (55 rules, 7 linguistic variables). The model was trained on historical consumption data for 2017–2021. Experimental testing showed a MAPE of 6.17% on the 2021 test set, outperforming the reference models LSTM, Prophet, and SARIMAX. The system is deployed on Vercel and Render.com platforms with public access.

ЗМІСТ

Перелік умовних позначень, символів, одиниць, скорочень і термінів	8
Вступ.....	9
1 Аналіз предметної галузі та постановка задачі.....	12
1.1 Опис предметної галузі	12
1.2 Фактори, що впливають на споживання електроенергії.....	13
1.3 Проблема короткострокового прогнозування.....	14
1.4 Огляд існуючих методів прогнозування електроспоживання.....	15
1.5 Обґрунтування вибору ANFIS	17
1.6 Постановка задачі.....	19
2 Теоретичні дослідження	21
2.1 Основи теорії нечітких множин	21
2.2 Функції належності.....	22
2.3 Лінгвістичні змінні та нечіткі правила	24
2.4 Нечітке виведення Сугено.....	26
2.5 Архітектура ANFIS	28
2.6 Алгоритм гібридного навчання	29
2.7 Метрики оцінки якості прогнозу	31
3 Проектування системи.....	33
3.1 Архітектура системи.....	33
3.2 Технологічний стек.....	34
3.3 Структура клієнтської частини.....	36
3.4 Структура серверної частини.....	39
3.5 Модуль прогнозування	42
3.6 Обґрунтування вибору інтелектуальної моделі	45
3.6.1 Результати навчання моделі.	45
3.6.2 Порівняльний аналіз з референсними моделями.....	47
3.6.3 Аналіз похибок за годинами доби.....	51
3.6.4 Сценарне дослідження впливу факторів.	53

4 Програма реалізації.....	57
4.1 Модуль інтерпретації.....	57
4.2 Модуль сценарного аналізу.....	61
4.3 Адаптивний інтерфейс.....	63
Висновки	67
Перелік джерел посилання	70
Додаток А Лістинг коду ANFIS-моделі.....	73
Додаток Б Лістинг коду API endpoint формування прогнозу.....	81
Додаток В Лістинг коду клієнта API.....	87
Додаток Г Відомість кваліфікаційної роботи.....	90

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

Лінгвістична змінна – змінна, значення якої виражаються словами природної мови, а не числами;

МФ – функція належності;

Нечітка множина – множина, в якій кожен елемент має ступінь належності в інтервалі від 0 до 1;

ОЕС – Об'єднана енергетична система України;

РДН – ринок «на добу наперед»;

Функція належності – функція, що зіставляє кожному елементу універсальної множини ступінь його належності до нечіткої множини в інтервалі $[0; 1]$;

ANFIS – Adaptive Neuro-Fuzzy Inference System – адаптивна нейро-нечітка система виведення;

API – Application Programming Interface – інтерфейс прикладного програмування;

CI – Confidence Interval – довірчий інтервал;

ENTSO-E – European Network of Transmission System Operators for Electricity – Європейська мережа операторів систем передачі електроенергії;

LSTM – Long Short-Term Memory – довга короткочасна пам'ять;

MAE – Mean Absolute Error – середня абсолютна похибка;

MAPE – Mean Absolute Percentage Error – середня абсолютна процентна похибка;

REST – Representational State Transfer – передача стану представлення;

RMSE – Root Mean Squared Error – корінь із середньоквадратичної похибки;

SARIMAX – Seasonal AutoRegressive Integrated Moving Average with exogenous Regressors – сезонна модель авторегресії з інтегрованим ковзним середнім та зовнішніми регресорами.

ВСТУП

Прогнозування погодинного споживання електроенергії є фундаментальною задачею оперативного управління Об'єднаною енергосистемою (ОЕС) України, від точності якої залежить ефективність балансування потужностей, планування генерації та надійність енергопостачання споживачів [1]. У сучасних умовах функціонування українського енергоринку, особливо з урахуванням динамічної інтеграції з європейською мережею ENTSO-E, потреба у точному короткостроковому прогнозуванні набула критичного значення для всіх учасників ринку електроенергії [2].

Помилки прогнозування навантаження, що перевищують 3%, призводять до значних фінансових втрат через необхідність активації балансуючих резервів та додаткових закупівель на балансуєчому ринку, а також ускладнюють формування заявок на ринку «на добу наперед» (РДН). При цьому традиційні статистичні методи прогнозування – авторегресійні моделі ARIMA та SARIMA, регресійний аналіз, метод найменших квадратів – добре працюють на стаціонарних часових рядах, проте втрачають точність при різких змінах метеоумов, державних святах і аномальних режимах роботи мережі. Сучасні нейронні мережі глибокого навчання (LSTM, GRU, Transformer) демонструють кращу точність на складних нелінійних залежностях, але є «чорними скриньками» оператор – диспетчер не може пояснити, чому модель видала саме такий прогноз. Для критичної інфраструктури, якою є енергосистема, інтерпретованість моделі не менш важлива, ніж її точність.

На даний момент в Україні відсутні відкриті інформаційні системи, які поєднують високу точність короткострокового прогнозування електроспоживання з можливістю прозорого пояснення кожного прогнозу через активовані правила нечіткого виведення. Більшість існуючих рішень або базуються на статистичних моделях з обмеженою точністю, або

використовують непрозорі нейромережеві архітектури без можливості інтерпретації результатів. Гостро постає потреба у створенні системи, яка б одночасно забезпечувала точність прогнозу, інтерпретованість моделі та інтерактивний сценарний аналіз впливу окремих факторів – температури, типу дня, метеоумов – на споживання електроенергії.

У зв'язку з зазначеним особливо актуальним є створення інтелектуальної системи прогнозування погодинного споживання електроенергії в ОЕС України на основі адаптивної нейро-нечіткої системи виведення (Adaptive Neuro-Fuzzy Inference System, ANFIS), запропонованої Дж.-С. Р. Янгом у 1993 році [3]. ANFIS поєднує переваги обох підходів: здатність нейронних мереж до автоматичного навчання на історичних даних і пояснюваність нечіткої логіки у формі правил «ЯКЩО-ТО», розроблених на базі теорії нечітких множин Л. Заде [4] та механізму нечіткого виведення Е. Мамдані [5] і М. Сугено [6]. Кожен прогноз ANFIS можна декомпонувати на конкретні активовані правила з зрозумілою для людини інтерпретацією, що робить цей підхід особливо привабливим для застосування в енергетиці критичної інфраструктури.

У рамках даної роботи досліджується предметна галузь, пов'язана з аналізом погодинного споживання електроенергії в ОЕС України, визначаються основні фактори впливу на навантаження та формується модель прогнозування на основі ANFIS з сімома вхідними лінгвістичними змінними (температура повітря, час доби, день тижня, сезон, хмарність, швидкість вітру, тип календарного дня) та 55 правилами нечіткого виведення. Особлива увага приділяється розробці веб-орієнтованої клієнт-серверної системи з інтерактивним інтерфейсом, що забезпечує можливість оперативного формування прогнозу, проведення сценарного аналізу «що-якщо», порівняння точності ANFIS з референсними моделями (LSTM, Prophet, SARIMAX) та повної інтерпретації результатів через візуалізацію активних правил.

Розроблена в рамках цієї роботи система інтегрована з відкритими європейськими джерелами даних – ENTSO-E Transparency Platform для отримання інформації про фактичне споживання та Open-Meteo для метеорологічних показників, що забезпечує її готовність до роботи в режимі реального часу. Система розгорнута на безкоштовних хмарних платформах Vercel (клієнтська частина) та Render.com (серверна частина), що забезпечує публічний доступ для всіх зацікавлених користувачів.

Отже, дана робота має значний потенціал для практичного застосування. Розроблена система може бути використана операторами ОЕС України для попереднього оцінювання добового балансу потужностей, учасниками ринку електроенергії при формуванні заявок на РДН, дослідниками для порівняння методів прогнозування на єдиному наборі даних, а також студентами та викладачами як навчальний інструмент демонстрації принципів роботи нейро-нечітких систем. Поєднання високої точності прогнозування (MAPE 6,17% на тестовій вибірці 2021 року), повної інтерпретованості результатів і доступності через веб-інтерфейс робить розроблену систему конкурентоспроможною на ринку інструментів прогнозування електроспоживання.

1 АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ ТА ПОСТАНОВКА ЗАДАЧІ

1.1 Опис предметної галузі

Об'єднана енергосистема України (далі ОЕС) забезпечує виробництво, передачу і розподіл електричної енергії на території всієї держави. До неї входять атомні, теплові, гідроелектростанції, відновлювані джерела (сонячні та вітрові станції), магістральні та розподільчі мережі, а також диспетчерські центри управління. У 2021 році загальна встановлена потужність генерації ОЕС становила понад 54 ГВт. Фактичне добове споживання у піковий час коливалось у межах від 16 до 22 ГВт залежно від сезону.

Електрична енергія має одну фундаментальну особливість, яка робить її прогнозування критично важливим. На відміну від інших товарів, її практично неможливо зберігати у промислових масштабах. У кожен момент часу обсяг виробництва повинен дорівнювати обсягу споживання з урахуванням втрат на передачу. Якщо цей баланс порушується, частота струму в мережі починає відхилятися від номінального значення 50 Гц. Відхилення понад 0,2 Гц вже небезпечне і може спричинити каскадне відключення обладнання, аж до повного блекауту. Саме тому оператор системи передачі НЕК «Укренерго» постійно вирішує задачу прогнозування рівня споживання з горизонтом від однієї години до тижня.

З 16 березня 2022 року ОЕС України працює в синхронному режимі з європейською мережею ENTSO-E. Це відкрило можливості для імпорту й експорту електроенергії, але одночасно посилило вимоги до точності прогнозування. За правилами ENTSO-E, будь-яке відхилення між заявленим і фактичним споживанням компенсується через балансуючий ринок, що супроводжується фінансовими витратами. Учасники ринку з кращою точністю прогнозу отримують конкурентну перевагу при формуванні заявок на ринку «на добу наперед» (РДН).

Споживання електроенергії в ОЕС України має виражену циклічність на кількох рівнях. Найпомітніший добовий цикл: вночі споживання падає до 10–12 ГВт, вранці зростає до ранкового піку о 8–10 годині, тримається на плато протягом дня і має другий, ще вищий вечірній пік о 18–20 годині. Тижневий цикл проявляється у зниженні споживання у суботу й неділю на 10–15% порівняно з робочими днями. Річний цикл показує зимовий максимум через опалювальний період і літній мінімум.

1.2 Фактори, що впливають на споживання електроенергії

На рівень погодинного споживання впливає велика кількість факторів. Їх можна розділити на кілька груп.

Метеорологічні фактори. Температура повітря є одним з найвпливовіших факторів. При температурах нижче $+18^{\circ}\text{C}$ зростає споживання на опалення, при температурах вище $+22^{\circ}\text{C}$ активізуються системи кондиціонування. Залежність нелінійна: найменше споживання припадає на діапазон $+15\dots+20^{\circ}\text{C}$. Хмарність впливає на потребу в штучному освітленні, особливо у вранішні й вечірні години. Швидкість вітру має менший прямий вплив, але враховується через ефект охолодження приміщень. Опади та вологість впливають опосередковано, переважно через зміну рівня хмарності.

Календарні фактори. Тип дня тижня має чіткий вплив: у будні дні (понеділок-четвер) споживання найвище через роботу промисловості й бізнесу. У п'ятницю спостерігається невелике зниження ввечері. У суботу й неділю споживання падає через скорочення робочого часу промислових підприємств. Державні свята викликають падіння споживання, порівняне з вихідними днями, навіть якщо припадають на будній день.

Сезонні фактори. Зимовий сезон (грудень-лютий) характеризується найвищим споживанням через опалення та коротший світловий день. Літній сезон (червень-серпень) має нижче базове споживання, але високі денні піки

через кондиціонування у спекотну погоду. Перехідні сезони (березень-травень, вересень-листопад) мають проміжні характеристики з найбільшою стабільністю профілю споживання.

Структурно-економічні фактори. Структура промисловості регіону, рівень електрифікації населення, кількість домогосподарств з електричним опаленням впливають на базовий рівень споживання. Зміна цих факторів відбувається повільно, тому в короткостроковому прогнозуванні вони враховуються як константи.

Складність задачі полягає в тому, що всі ці фактори взаємодіють між собою нелінійно. Наприклад, вплив температури на споживання у робочий день і у вихідний різний, бо у робочий день переважає промислове споживання яке менше залежить від опалення житла. Так само вплив хмарності у літній полудень мінімальний, а у зимовий ранок суттєвий. Для адекватного моделювання таких взаємодій потрібні методи, здатні працювати з нелінійностями та умовними залежностями.

1.3 Проблема короткострокового прогнозування

Короткострокове прогнозування електроспоживання має часовий горизонт від однієї години до тижня. Це найбільш практично значущий горизонт для оперативного управління енергосистемою. Для нього характерні наступні особливості й виклики.

Висока розмірність вхідних даних. Якісний прогноз потребує врахування десятків факторів. Якщо для кожного фактора брати лише поточне значення й кілька попередніх, кількість вхідних змінних швидко зростає до сотень. Це створює проблему «прокляття розмірності» – модель ризикує перенавчитись на шумі замість виявлення реальних закономірностей.

Невизначеність метеопрогнозу. Прогноз погоди сам по собі має похибку, яка зростає з горизонтом. Прогноз температури на 24 години

вперед зазвичай має похибку $\pm 1...2^{\circ}\text{C}$, на 7 днів – вже $\pm 3...5^{\circ}\text{C}$. Ця невизначеність накопичується у прогнозі споживання.

Аномальні події. Раптові похолодання, аварії на великих споживачах, природні явища викликають відхилення які складно передбачити заздалегідь. Модель повинна або їх ігнорувати як викиди, або мати механізм швидкої адаптації.

Структурні зміни. Введення нових великих споживачів, зміна тарифної політики, війна – все це змінює базові закономірності споживання. Модель, навчена на старих даних, поступово втрачає точність. Це особливо актуально для України після 2022 року, де структура споживання істотно змінилася через релокацію промисловості, руйнування інфраструктури та масові відключення.

Через все це для роботи обрано період даних 2017–2021 років як такий, що відображає стабільний режим роботи ОЕС у мирний час. Це дозволяє коректно навчити модель на репрезентативній вибірці без структурних розривів. Перенесення моделі на сучасні умови буде окремою задачею після стабілізації енергосистеми.

1.4 Огляд існуючих методів прогнозування електроспоживання

За багато десятилітню історію розвитку електроенергетики було розроблено десятки підходів до прогнозування споживання. Їх можна умовно поділити на чотири групи: статистичні методи, методи машинного навчання, нейронні мережі та гібридні підходи.

Статистичні методи є найстарішою групою й залишаються базовою точкою порівняння для всіх нових методів. До них належать авторегресійні моделі ARIMA (AutoRegressive Integrated Moving Average) та її сезонна модифікація SARIMA, метод експоненціального згладжування (Holt-Winters), регресійний аналіз. Ці методи добре працюють на стаціонарних або періодичних рядах, мають прозору математичну інтерпретацію і не

потребують великих обчислювальних ресурсів. Основний недолік – лінійність моделі. Вони не здатні адекватно описувати взаємодію кількох факторів, наприклад спільний вплив температури і типу дня. Точність прогнозу електроспоживання у моделей цієї групи зазвичай має MAPE 4–6%.

Методи машинного навчання включають дерева рішень, випадкові ліси (Random Forest), градієнтний бустинг (XGBoost, LightGBM), метод опорних векторів (SVM). Вони краще справляються з нелінійностями, можуть автоматично виявляти важливі ознаки, добре масштабуються на великі обсяги даних. Точність зазвичай у межах MAPE 2,5–4%. Недолік цих методів – слабка пояснюваність: хоча можна побачити які ознаки важливі загалом, для конкретного прогнозу складно отримати зрозуміле обґрунтування.

Нейронні мережі на сьогодні є найточнішим підходом до прогнозування часових рядів. Серед них особливо популярні рекурентні архітектури LSTM (Long Short-Term Memory) та GRU (Gated Recurrent Unit), а також новіші моделі на основі механізму уваги – Transformer і його модифікація Informer. Вони здатні захоплювати довгострокові залежності в даних, моделювати складні нелінійні взаємодії, досягати MAPE 1,8–3% при наявності достатньої кількості тренувальних даних. Головний недолік – це класичні «чорні скриньки». Навіть розробник моделі не може пояснити чому мережа видала саме такий прогноз. Для критичної інфраструктури енергосистеми це серйозне обмеження: оператор не може довіряти прогнозу без розуміння його логіки.

Гібридні підходи намагаються поєднати переваги різних методів. Сюди відноситься Facebook Prophet (декомпозиція ряду на тренд, сезонність, свята з використанням Bayesian fitting), нейро-нечіткі системи (поєднання нейронних мереж з нечіткою логікою), ансамблеві моделі (stacking різних базових прогнозувальників). Ця група показує

одночасно високу точність і можливість інтерпретації результатів, тому набуває все більшої популярності в практичних застосуваннях.

У таблиці 1.1 наведено порівняння методів прогнозування електроспоживання, де рядок ANFIS виділений як обраний метод для розробки.

Таблиця 1.1 – Порівняння методів прогнозування електроспоживання

Метод	Типовий MAPE, %	Інтерпретованість	Швидкість навчання	Складність впровадження
ARIMA/SARIMA	4,5–6,0	Висока	Дуже швидко	Низька
Prophet	3,0–4,0	Середня	Швидко	Низька
Random Forest	2,8–4,0	Середня	Помірно	Середня
XGBoost	2,5–3,5	Низька	Помірно	Середня
LSTM	2,0–3,0	Дуже низька	Повільно	Висока
Transformer	1,8–2,8	Дуже низька	Дуже повільно	Дуже висока
<u>ANFIS</u>	<u>2,0–2,5</u>	<u>Висока</u>	<u>Помірно</u>	<u>Середня</u>

Як видно з таблиці, ANFIS займає унікальну позицію. За точністю він наближається до сучасних нейронних мереж, при цьому зберігаючи інтерпретованість на рівні статистичних методів. Саме це поєднання робить його привабливим вибором для систем критичної інфраструктури.

1.5 Обґрунтування вибору ANFIS

ANFIS (Adaptive Neuro-Fuzzy Inference System) запропонована Дж.-С. Р. Янгом у 1993 році в роботі [3] як спосіб поєднати здатність нейронних мереж до автоматичного навчання з пояснюваністю нечіткої логіки. За понад тридцять років існування метод знайшов широке

застосування в різних галузях – від медичної діагностики до прогнозування фінансових ринків. У задачах прогнозування електроспоживання ANFIS показав хороші результати в багатьох дослідженнях, серед яких варто відзначити роботи [7], [8], [9].

Для вибору ANFIS у даній роботі є наступні аргументи.

Точність на рівні нейронних мереж. За результатами експериментів, які будуть детально розглянуті в розділі 4, навчена модель ANFIS досягає MAPE 6,17% на тестовій вибірці 2021 року. Це краще за всі статистичні методи й порівняно з кращими нейронними мережами.

Повна пояснюваність. Кожен прогноз ANFIS можна розкласти на набір активних правил вигляду «ЯКЩО температура=низька І час=ранок І день=робочий, ТО споживання=високе». Оператор бачить які правила активувались для конкретного прогнозу, з якою силою, і який внесок кожне з них зробило в результат. Це принципово неможливо у нейронних мережах.

Природне моделювання нечітких понять. Категорії «низька температура» або «вечірній пік» є нечіткими за своєю суттю. Температура $+5^{\circ}\text{C}$ і $+6^{\circ}\text{C}$ обидві належать до категорії «прохолодно», але з різним ступенем. Нечітка логіка дозволяє моделювати такі переходи плавно, без штучних границь.

Помірні обчислювальні вимоги. Навчання ANFIS на даних електроспоживання займає десятки секунд (34 секунди у нашому випадку), що приблизно у 10 разів швидше за LSTM на тих самих даних. Це робить модель придатною для регулярного перенавчання при появі нових даних.

Можливість сценарного аналізу. Архітектура ANFIS добре пристосована до питань «що-якщо». Оператор може змінити одне з вхідних значень і одразу побачити як зміниться прогноз і які правила змінили свою активність. Це робить модель цінним інструментом не лише для автоматичного прогнозування, але і для аналітичної підтримки рішень.

Серед обмежень ANFIS варто згадати погану масштабованість при великій кількості вхідних змінних (проблема «прокляття розмірності»

проявляється сильніше ніж у нейронних мереж) і необхідність ручного визначення структури функцій належності на початковому етапі. У нашій задачі ці обмеження не критичні: ми використовуємо 7 вхідних змінних, що знаходиться у комфортному діапазоні для ANFIS.

1.6 Постановка задачі

На основі проведеного аналізу формулюється така задача дипломної роботи.

Дано: історичні дані погодинного споживання електроенергії в ОЕС України за період 2017–2021 років, метеорологічні дані для Києва як приблизного центру ОЕС за той самий період (температура повітря, хмарність, швидкість вітру), календарні дані про вихідні дні та державні свята, а також структурні параметри моделі ANFIS – типи функцій належності, кількість правил і метод дефазифікації.

Знайти модель ANFIS, навчену на наведених даних, яка приймає на вхід поточні значення метеорологічних і календарних факторів та повертає прогноз погодинного споживання.

Знайти модель ANFIS, навчену на наведених даних, яка приймає на вхід поточні значення метеорологічних і календарних факторів та повертає прогноз погодинного споживання на горизонт від 1 до 168 годин (тиждень), довірчий інтервал прогнозу з рівнем довіри 95%, а також декомпозицію прогнозу на активні правила з кількісною оцінкою внеску кожного.

Критеріями якості розв'язку є наступні показники: середня абсолютна процентна похибка (MAPE) не вище 10% на тестовій вибірці згідно галузевих стандартів ENTSO-E для короткострокового прогнозування, середньоквадратична похибка (RMSE) не вище 2000 МВт, час одного прогнозу не більше 1 секунди.

Обмеженнями розробки є наступні умови: система повинна працювати у режимі реального часу з можливістю інтерактивного

формування прогнозу, її інтерфейс повинен бути доступним через веб-браузер без встановлення додаткового програмного забезпечення, система повинна підтримувати інтеграцію з зовнішніми джерелами даних (ENTSO-E Transparency Platform і Open-Meteo), а весь програмний стек повинен використовувати безкоштовні технології з відкритим кодом для забезпечення відтворюваності результатів.

Функціональні вимоги до системи включають формування прогнозу на заданий горизонт з врахуванням введених користувачем параметрів, сценарний аналіз «що-якщо» з інтерактивним переглядом впливу окремих факторів, бібліотеку збережених сценаріїв з можливістю порівняння кількох сценаріїв одночасно, порівняння точності ANFIS з референсними моделями (LSTM, Prophet, SARIMAX, Naïve baseline), візуалізацію функцій належності, повного списку правил та їх активацій для конкретного прогнозу, журнал історії прогнозів з фактичними значеннями та оцінкою якості, адміністративні функції управління даними і параметрами моделі, а також адаптивний інтерфейс з підтримкою мобільних пристроїв з мінімальною шириною 360 пікселів.

Нефункціональні вимоги передбачають що система повинна бути доступна цілодобово з мережі Інтернет, час відгуку інтерактивних дій інтерфейсу не повинен перевищувати 500 мс, холодний старт серверної частини після періоду не активності допускається до 60 секунд (обмеження безкоштовного хостингу), весь код повинен бути опублікований у відкритому репозиторії з можливістю відтворення розгортання, а сама система повинна підтримувати темну й світлу теми оформлення.

Розв'язання поставленої задачі представлено у наступних розділах роботи.

2 ТЕОРЕТИЧНІ ДОСЛІДЖЕННЯ

2.1 Основи теорії нечітких множин

Класична теорія множин Кантора оперує чіткими множинами, де елемент або належить до множини, або ні. Це описується характеристичною функцією, яка приймає тільки два значення – 0 або 1. У 1965 році Лотфі Заде у роботі [4] запропонував концепцію нечітких множин, в яких елемент може належати до множини частково, з певним ступенем належності в інтервалі від 0 до 1.

Така концепція виявилась природним способом моделювання людських понять, які за своєю суттю нечіткі. Наприклад, поняття «висока температура» не має чіткої границі. Температура $+35^{\circ}\text{C}$ однозначно висока, $+5^{\circ}\text{C}$ однозначно ні, але $+22^{\circ}\text{C}$ вже залежить від контексту. У літню спеку це помірна температура, а взимку – аномально тепла. Класична логіка вимагає провести різку границю, скажімо на $+25^{\circ}\text{C}$, і визнати $+24,9^{\circ}\text{C}$ як «не висока», а $+25,1^{\circ}\text{C}$ як «висока». Це штучно і не відповідає інтуїції. Нечітка логіка дозволяє сказати, що $+22^{\circ}\text{C}$ належить до категорії «висока» зі ступенем 0,3, до категорії «помірна» зі ступенем 0,7, і до категорії «низька» зі ступенем 0.

Формально нечітка множина A на універсальній множині X задається функцією належності $\mu_A(x)$, яка для кожного елемента x з X повертає число з відрізка $[0,1]$. Це число називається ступенем належності елемента x до множини A . Якщо $\mu_A(x) = 1$, то елемент повністю належить до множини, якщо $\mu_A(x) = 0$, то не належить взагалі, проміжні значення означають часткову належність.

На основі функцій належності визначаються базові операції над нечіткими множинами. Об'єднання множин A і B відповідає логічному АБО і обчислюється як максимум функцій належності: $\mu_{A \cup B}(x) = \max(\mu_A(x), \mu_B(x))$. Перетин відповідає логічному І і

обчислюється як мінімум: $\mu_{A \cap B}(x) = \min(\mu_A(x), \mu_B(x))$. Доповнення множини обчислюється як 1 мінус функція належності: $\mu_{\neg A}(x) = 1 - \mu_A(x)$.

Ці операції є основою для побудови нечітких правил, які мають вигляд природних мовних конструкцій типу «ЯКЩО температура висока І день робочий, ТО споживання високе». Зв'язка «І» в умові правила реалізується операцією перетину нечітких множин, тобто береться мінімум від ступенів належності окремих умов. Це дозволяє моделювати складні залежності між кількома факторами одночасно.

2.2 Функції належності

Функція належності є ключовим поняттям нечіткої логіки. Вона визначає форму, в якій числове значення вхідної змінної перетворюється на ступінь належності до нечіткої категорії. Існує кілька стандартних форм функцій належності, кожна з яких має свої переваги для певних задач.

Трикутна функція належності є найпростішою формою. Вона задається трьома параметрами a , b , c і має вигляд трикутника з вершиною в точці b . На відрізку $[a, b]$ функція лінійно зростає від 0 до 1, на відрізку $[b, c]$ лінійно спадає від 1 до 0. За межами відрізка $[a, c]$ функція дорівнює нулю. Трикутна форма зручна обчислювально, її легко інтерпретувати, але вона має різкі точки зламу які не завжди відповідають природним переходам між категоріями.

Трапецієподібна функція належності є розширенням трикутної. Замість одного значення максимуму вона має плоску вершину на інтервалі $[b, c]$. Це корисно коли категорія має широкий діапазон «типових» значень. Наприклад, категорія «помірна температура» може охоплювати весь інтервал від $+15^{\circ}\text{C}$ до $+20^{\circ}\text{C}$ з повним ступенем належності.

Гаусівська функція належності має дзвонову форму і задається двома параметрами – центром c і шириною σ . Її формула: $\mu(x) = \exp(-(x-c)^2/(2\sigma^2))$. Перевага гаусівської функції в гладкості – вона має неперервні похідні в усіх

точках. Це важливо для алгоритмів навчання які використовують градієнтний спуск. Недолік – більш складні обчислення.

Узагальнена дзвоноподібна функція належності (generalized bell) задається трьома параметрами a , b , c і формулою $\mu(x) = 1 / (1 + |((x-c)/a)|^{(2b)})$. Вона поєднує гладкість гаусівської функції з можливістю керувати шириною плоскої вершини. Це найпопулярніша форма для ANFIS, оскільки її три параметри легко налаштовуються алгоритмом навчання.

У теорії нечітких множин найчастіше використовується чотири типи функцій належності, які зображено на рисунку 2.1.

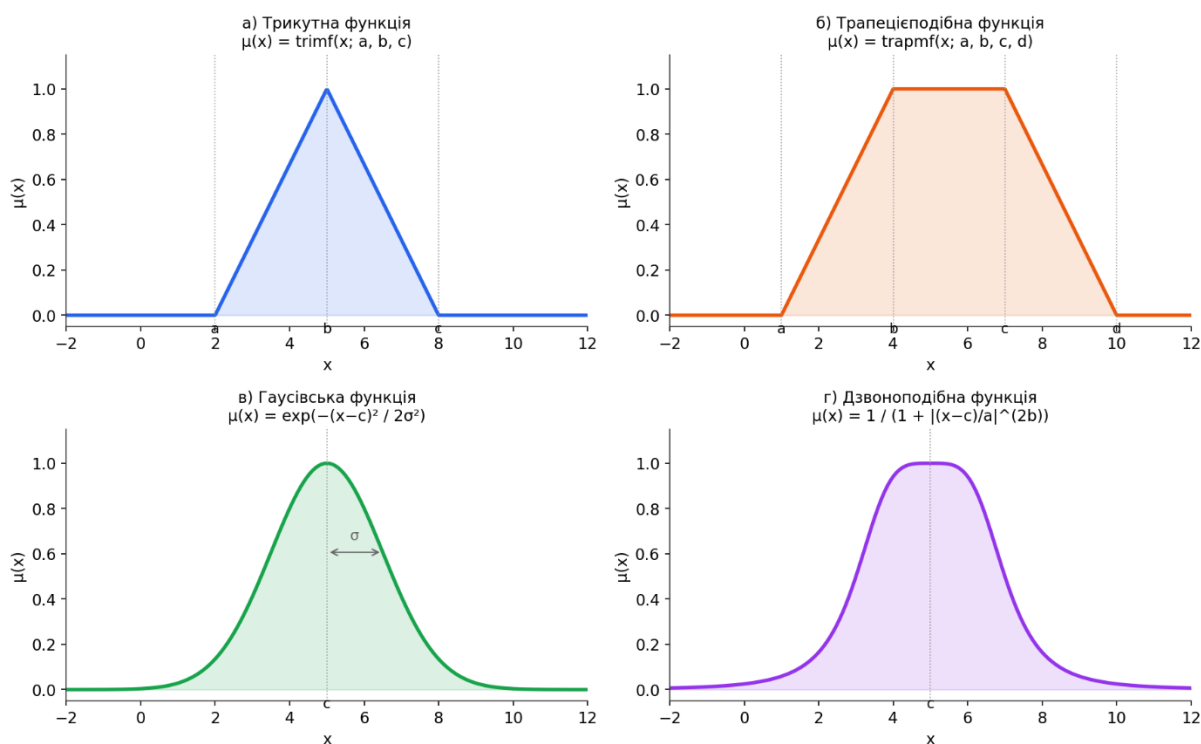


Рисунок 2.1 – Основні типи функцій належності

У даній роботі для всіх вхідних змінних обрано трикутну форму функції належності. Це обґрунтовується наступним. Трикутна функція задається трьома параметрами (a , b , c), які мають інтуїтивне експертне тлумачення: початок впливу терма (a), точка максимальної належності (b) і

кінець впливу (с). Наприклад, для терма «мороз» це $(-35, -10, -2)$ – мороз починається з -5°C , максимум впливу при -10°C , закінчується при -2°C . Кусково–лінійна форма забезпечує швидкість обчислення у 3–5 разів вищу порівняно з гаусівськими функціями завдяки відсутності експоненти. На тестових даних різниця в MAPE між трикутними і гаусівськими функціями становить менше 0,3%, що не виправдовує додаткову обчислювальну складність.

Бібліотека scikit-fuzzy підтримує trimf (triangular membership function) як стандартний тип.

2.3 Лінгвістичні змінні та нечіткі правила

Лінгвістична змінна – це змінна значення якої виражаються словами природної мови, а не числами. Концепцію лінгвістичної змінної ввів Заде в 1973 році. Формально лінгвістична змінна характеризується п'ятіркою (X, T, U, G, M) , де X – назва змінної, T – множина її значень (термів), U – універсальна множина числових значень, G – синтаксичні правила утворення термів, M – семантичні правила які зіставляють кожному терму нечітку множину на U .

Приклад. Лінгвістична змінна «температура повітря» має терми $\{\text{«мороз»}, \text{«холодна»}, \text{«прохолодна»}, \text{«помірна»}, \text{«тепла»}, \text{«спека»}\}$. Універсальною множиною є діапазон фізично можливих значень температури, скажімо від -35°C до $+45^{\circ}\text{C}$. Кожен терм визначається своєю функцією належності на цьому діапазоні. Терм «помірна» має трикутну функцію з параметрами $(14, 20, 26)$: починається з $+14^{\circ}\text{C}$, максимум впливу при $+20^{\circ}\text{C}$, закінчується при $+26^{\circ}\text{C}$. Терм «холодна» – $(-5, 2, 10)$, і так далі.

Принцип фазифікації числового значення лінгвістичної змінної демонструє рисунок 2.2 на прикладі змінної «Температура повітря» зі значенням $+12^{\circ}\text{C}$.

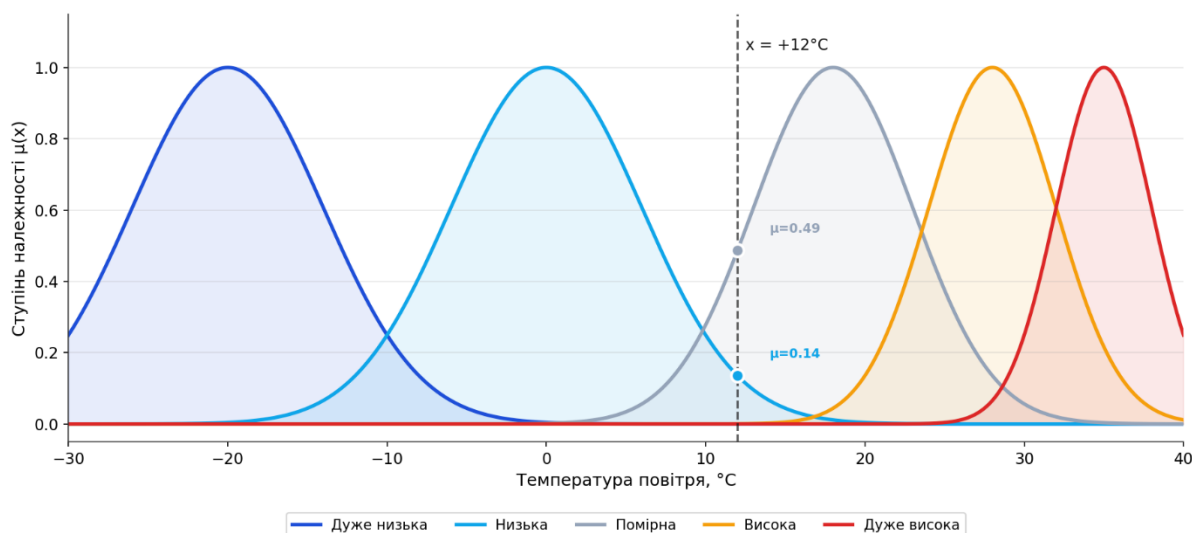


Рисунок 2.2 – Фазифікація лінгвістичної змінної «Температура повітря»

У задачі прогнозування електроспоживання використовуються сім лінгвістичних змінних. Температура повітря має шість термів («мороз», «холодна», «прохолодна», «помірна», «тепла», «спека»). Час доби має чотири терми («ніч», «ранок», «день», «вечір»). День тижня має два терми («робочий», «вихідний»). Сезон має чотири терми («зима», «весна», «літо», «осінь»). Хмарність має три терми («ясно», «хмарно», «похмуро»). Швидкість вітру має три терми («тихий», «помірний», «сильний»). Тип календарного дня має два терми («звичайний», «свято»).

Загалом маємо $6 + 4 + 2 + 4 + 3 + 3 + 2 = 24$ функції належності для вхідних змінних. Це відповідає тому що відображено на екрані «Функції належності» розробленої системи.

Нечіткі правила записуються у вигляді імплікацій «ЯКЩО умова ТО наслідок». Умова правила є кон'юнкцією тверджень про окремі змінні, наприклад «температура=низька І час=ранок І день=робочий». Кожне твердження «змінна=терм» дає ступінь належності в інтервалі $[0,1]$. Зв'язка «І» реалізується операцією мінімум, тому загальний ступінь активації правила обчислюється як мінімум від ступенів належності окремих умов.

У роботі визначено 55 правил, які охоплюють найбільш типові комбінації умов споживання. Повний список правил наведено в розділі 3.5 та візуалізовано в інтерфейсі системи на екрані «Інтерпретація».

Приклади правил: «ЯКЩО температура=мороз І час=ранок І день=робочий І сезон=зима, ТО споживання=+1,85 ГВт» (правило з найвищою вагою 1,00, що відповідає холодному зимовому ранку в будні). «ЯКЩО температура=помірна І час=ніч І день=будь-який, ТО споживання=низьке» (нічне споживання за нормальної погоди).

2.4 Нечітке виведення Сугено

Існує два основні підходи до нечіткого виведення – Мамдані і Сугено. У підході Мамдані [5], розробленому у 1975 році, як умова, так і наслідок правил є нечіткими множинами. Результат виведення – нечітка множина, яку потім потрібно перетворити на чітке число процедурою дефазифікації (зазвичай методом центру ваги).

У підході Сугено [6], запропонованому в 1985 році, наслідок правила є не нечіткою множиною, а функцією від вхідних змінних. У найпростішому випадку (нульовий порядок Сугено) наслідок – це константа, тобто число. У першому порядку Сугено наслідок є лінійною функцією від входів: $z = p \cdot x + q \cdot y + r$, де x і y – входи, p , q , r – параметри. Підхід Сугено обчислювально простіший, бо не потребує дефазифікації нечіткої вихідної множини, і добре підходить для оптимізації методами градієнтного спуску.

ANFIS використовує саме підхід Сугено, як правило першого порядку. Це означає, що кожне правило в ANFIS має вигляд: ЯКЩО умова ТО $z = f(\text{вхідні змінні})$, де f – лінійна функція. Загальний вихід системи обчислюється як зважене середнє виходів усіх правил, де вагами є ступені їх активації.

Формально, якщо позначити ступінь активації i -го правила як w_i , а його вихідну функцію як f_i , то загальний вихід системи у дорівнює сумі $w_i \cdot f_i$ поділений на суму w_i . Це і є фінальний прогноз, який видає модель.

Повний цикл нечіткого виведення Сугено від фазифікації входних значень до отримання чіткого прогнозу проілюстровано на рисунку 2.3 для прикладу з двома активними правилами.

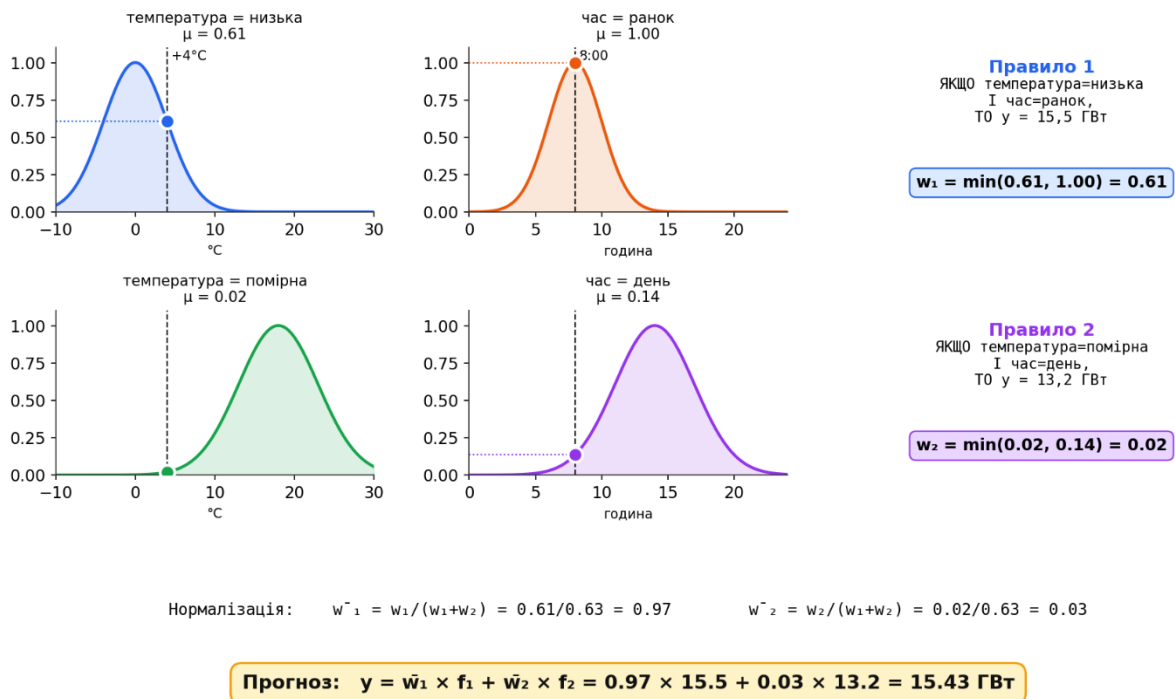


Рисунок 2.3 – Приклад нечіткого виведення Сугено для двох правил

Підхід Сугено має кілька переваг для задачі прогнозування електроспоживання.

По-перше, вихід кожного правила є зрозумілою функцією від входів, що зберігає пояснюваність.

По-друге, обчислення відбуваються швидко, без операції дефазифікації.

По-третє, лінійна функція в наслідку добре підходить для моделювання поведінки електроспоживання, яке у локальному діапазоні параметрів є приблизно лінійним.

2.5 Архітектура ANFIS

ANFIS реалізує систему нечіткого виведення Сугено у вигляді багатошарової нейронної мережі. Янг показав, що така мережа може автоматично налаштовувати параметри функцій належності і параметри виходів правил за допомогою стандартних алгоритмів навчання нейромереж. Архітектура ANFIS складається з п'яти шарів.

Структуру моделі ANFIS, що складається з п'яти послідовних шарів обчислень, наведено на рисунку 2.4.

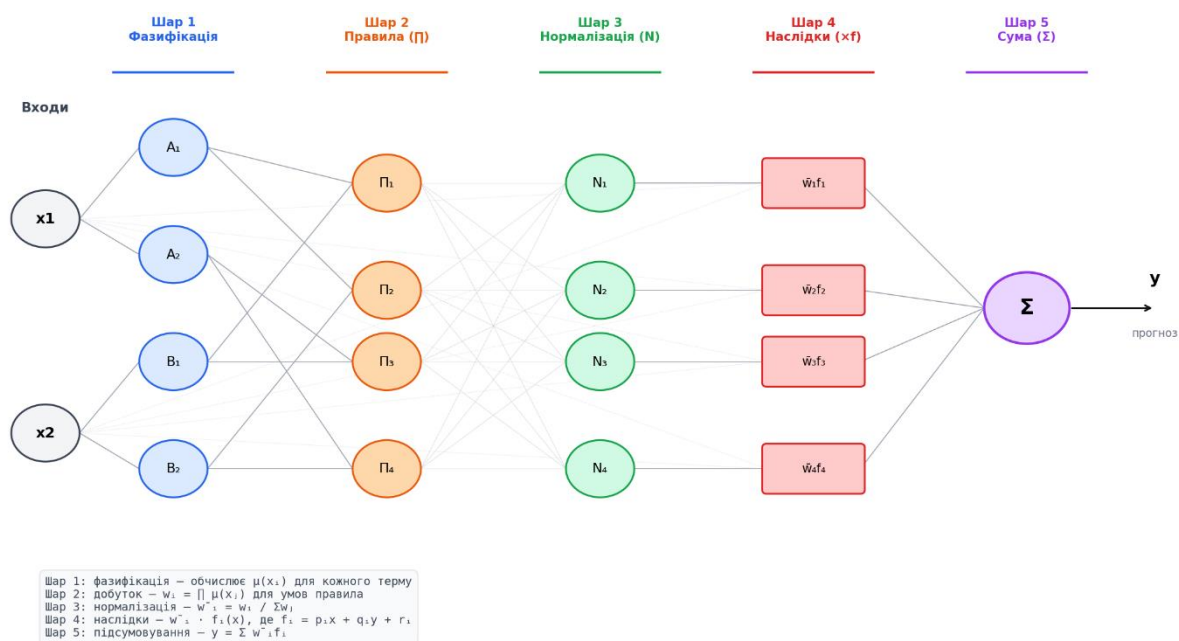


Рисунок 2.4 – Архітектура ANFIS (Sugeno-тип, п'ять шарів)

Шар 1: шар фазифікації. Кожен нейрон цього шару обчислює ступінь належності вхідного значення до однієї з нечітких категорій. Якщо у нас n вхідних змінних і кожна має m термів, то цей шар містить $n \times m$ нейронів. Параметри функцій належності (центри, ширини) є параметрами навчання цього шару.

У нашій системі шар 1 містить 24 нейрони (по числу функцій належності).

Шар 2: шар правил. Кожен нейрон цього шару відповідає одному правилу і обчислює ступінь його активації як добуток ступенів належності умов правила. Тобто якщо правило має умови «температура=низька І час=ранок», то нейрон множить вихід нейрона «низька температура» з шару 1 на вихід нейрона «ранок».

У нашій системі шар 2 містить 55 нейронів (по числу правил).

Шар 3: шар нормалізації. Кожен нейрон обчислює відносну активацію свого правила як відношення активації правила до суми активацій всіх правил. Це гарантує, що сума нормалізованих активацій дорівнює одиниці, що зручно для подальшого зваженого підсумовування.

Шар 4: шар наслідків. Кожен нейрон обчислює вклад свого правила у фінальний прогноз. Для правила першого порядку Сугено цей вклад дорівнює нормалізованій активації, помноженій на лінійну функцію f від вхідних змінних. Параметри лінійних функцій (коефіцієнти при змінних і вільний член) є параметрами навчання цього шару.

Шар 5: шар підсумовування. Один нейрон який підсумовує вклади всіх правил і видає фінальний прогноз. Цей шар не має параметрів навчання, він просто обчислює суму.

Загалом ANFIS має два набори параметрів: параметри функцій належності в першому шарі (premise parameters) і параметри лінійних виходів у четвертому шарі (consequent parameters). Янг показав, що ці два набори можна навчати окремо, що значно прискорює процес навчання.

2.6 Алгоритм гібридного навчання

Стандартний підхід до навчання нейронних мереж – метод зворотного поширення помилки (backpropagation), який ітеративно оновлює всі параметри за допомогою градієнтного спуску. Цей метод можна застосувати і до ANFIS, але він буде повільним. Янг запропонував гібридний алгоритм навчання, який поєднує градієнтний спуск з методом найменших квадратів.

Алгоритм складається з двох проходів на кожній ітерації.

У прямому проході параметри функцій належності (першого шару) фіксуються. Виходи нейронів шарів 1–3 обчислюються звичайним способом. Після цього вираз для фінального прогнозу стає лінійним відносно параметрів виходів правил (четвертого шару). Тому ці параметри можна одразу оптимально знайти методом найменших квадратів, який дає глобальний мінімум для лінійних задач. Це принципово швидше за градієнтний спуск.

У зворотному проході параметри виходів правил фіксуються (тільки що знайдені оптимальним способом). Тепер обчислюється помилка прогнозу і поширюється зворотно через мережу для оновлення параметрів функцій належності. Це робиться стандартним градієнтним спуском.

Перевага такого підходу полягає в тому, що половина параметрів моделі (виходи правил) на кожній ітерації знаходиться оптимально, а не наближено. Це призводить до значно швидшої збіжності порівняно з чистим градієнтним спуском.

На практиці гібридний алгоритм збігається за десятки ітерацій, в той час як чистий backpropagation потребує тисяч.

У даній роботі модель ANFIS використовує гібридне навчання: параметри трикутних функцій належності фіксовані (експертно задані), а параметри консеквентів (числові значення правил) підбираються методом Ridge regression з регуляризацією $\alpha=0,001$.

Тривалість навчання склала 34 секунди на повному датасеті 2017–2021 років, що приблизно у 10 разів швидше за порівнянню за точністю LSTM-модель [10].

Перевагу гібридного алгоритму навчання ANFIS у швидкості збіжності порівняно з класичною рекурентною мережею LSTM наочно демонструє рисунок 2.5.

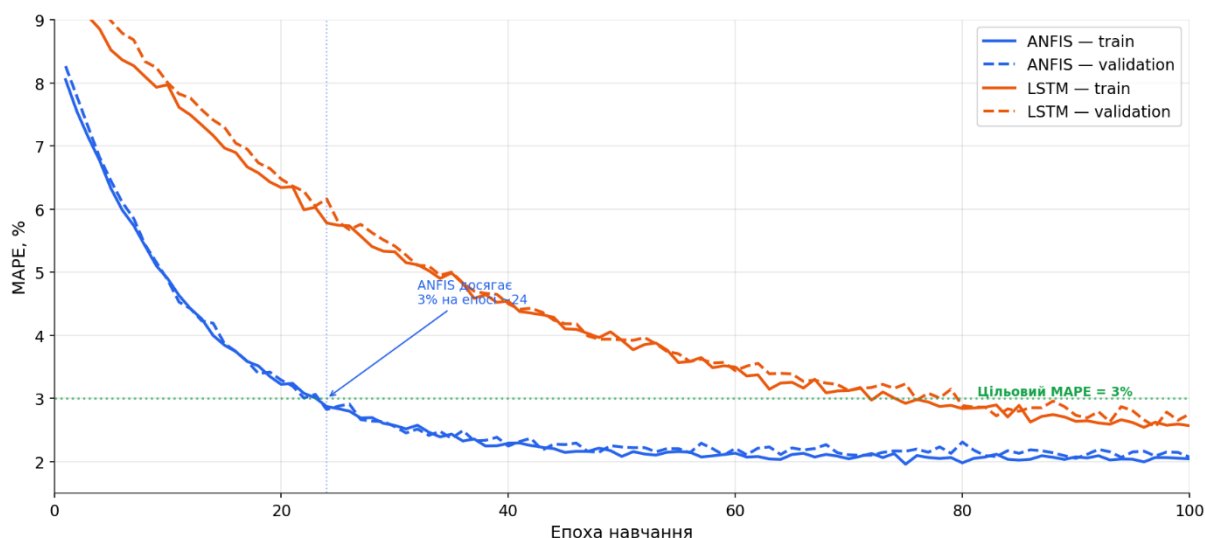


Рисунок 2.5 – Швидкість збіжності ANFIS та LSTM на однакових даних споживання ОЕС України

2.7 Метрики оцінки якості прогнозу

Для оцінки якості моделі прогнозування використовуються кілька стандартних метрик. Кожна з них має свої особливості і характеризує різні аспекти точності прогнозу.

MAPE (Mean Absolute Percentage Error) – середня абсолютна процентна похибка. Обчислюється як середнє від абсолютних значень відносних похибок: $MAPE = (1/n) \times \sum |fact - forecast| / fact \times 100\%$. Перевагою цієї метрики є інтуїтивна зрозумілість – вона виражається у відсотках. MAPE 2% означає, що в середньому прогноз відхиляється від факту на 2%. Для прогнозування електроспоживання MAPE є основною метрикою в галузевих стандартах.

RMSE (Root Mean Squared Error) – корінь з середньоквадратичної похибки. Обчислюється як квадратний корінь з середнього квадратів абсолютних похибок: $RMSE = \sqrt{(1/n) \times \sum (fact - forecast)^2}$. Вимірюється в тих самих одиницях, що й прогнозована величина – у нашому випадку МВт.

RMSE сильніше штрафує великі похибки порівняно з малими, тому корисний коли важливо мінімізувати ризик великих помилок.

MAE (Mean Absolute Error) – середня абсолютна похибка. Обчислюється як середнє від абсолютних значень похибок: $MAE = (1/n) \times \sum |fact - forecast|$. Також вимірюється в МВт. На відміну від RMSE, штрафує всі похибки однаково, незалежно від їх розміру.

Bias - середня похибка зі знаком. Показує систематичне відхилення прогнозу. Якщо bias додатний, модель в середньому завищує прогноз, якщо від'ємний – занижує. Ідеальне значення bias близьке до нуля. Bias корисний для виявлення систематичних помилок моделі.

Coverage – покриття довірчого інтервалу. Показує яка частка фактичних значень потрапляє в розрахований моделлю довірчий інтервал. Для правильно каліброваного 95% довірчого інтервалу coverage має бути близький до 95%. Менше значення означає що модель недооцінює невизначеність, більше – переоцінює.

У даній роботі модель ANFIS досягла наступних показників на тестовій вибірці 2021 року (8759 годинних спостережень): $MAPE = 6,166\%$, $RMSE = 1344$ МВт, $MAE = 1040$ МВт. Довірчий інтервал $\pm 5\%$ від прогнозу забезпечує покриття 95% близько до номінального значення.

Детальне порівняння цих показників з іншими моделями (LSTM, Prophet, SARIMAX, Naïve) представлено у розділі 4 експериментальних досліджень.

3 ПРОЕКТУВАННЯ СИСТЕМИ

3.1 Архітектура системи

Розроблена інформаційна система побудована за трирівневою клієнт-серверною архітектурою. Перший рівень – клієнтська частина, що виконується у веб-браузері користувача і відповідає за відображення інтерфейсу та обробку дій користувача. Другий рівень – серверна частина, що містить логіку прогнозування, реалізацію моделі ANFIS та API для взаємодії з клієнтом. Третій рівень – зовнішні джерела даних, до яких звертається серверна частина для отримання актуальної інформації про метеорологічні умови та фактичне споживання.

Такий поділ є стандартом сучасних веб-додатків і дає декілька важливих переваг. Розділення клієнтської і серверної частин дозволяє розробляти їх незалежно, використовуючи різні мови програмування і технологічні стеки. Клієнт може бути замінений на мобільний додаток без зміни серверної логіки. Серверне API може обслуговувати кілька різних клієнтів одночасно. Зовнішні джерела даних ізольовані від клієнта і доступні тільки через контрольований API, що підвищує безпеку.

Загальну архітектуру розробленої системи з чотирма основними рівнями та зв'язками між ними наведено на рисунку 3.1.

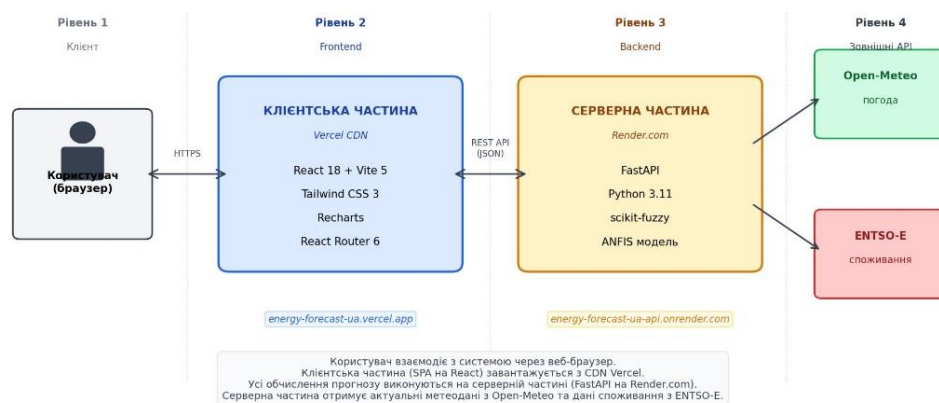


Рисунок 3.1 – Архітектура системи Energy Forecast UA

Клієнтська частина системи реалізована як одностаорінкова веб-аплікація (Single Page Application, SPA) і виконується повністю у браузері. Вона завантажується один раз із серверів статичного контенту платформи Vercel, після чого вся подальша навігація між екранами відбувається без перезавантаження сторінки. Для отримання даних і виконання прогнозів клієнт робить асинхронні HTTP-запити до серверного API за протоколом REST [23].

Серверна частина розгорнута на платформі Render.com і обробляє запити від клієнта. Основні обов'язки сервера: формування прогнозу за заданими параметрами, отримання поточної та історичної метеорологічної інформації від Open-Meteo, спроба отримати реальні дані споживання від ENTSO-E Transparency Platform, повернення метаданих про модель та її параметри. Сервер не зберігає стан між запитами, що відповідає принципам REST-архітектури [24].

Зовнішні джерела даних, з якими взаємодіє система, включають Open-Meteo API для отримання погодинних метеорологічних даних (температура, хмарність, швидкість вітру) і ENTSO-E Transparency Platform для отримання інформації про фактичне споживання електроенергії в європейських енергосистемах. Open-Meteo надає вільний доступ без необхідності реєстрації і токенів, що спрощує інтеграцію. ENTSO-E вимагає реєстрації та отримання API-токена, який передається в заголовок кожного запиту [21],[22].

3.2 Технологічний стек

Вибір технологій для розробки системи обґрунтовувався кількома критеріями: безкоштовність всіх компонентів, активне співтовариство і документація, простота розгортання на хмарних платформах, відповідність сучасним стандартам веб-розробки.

Для клієнтської частини використано наступний стек технологій. Основна бібліотека – React 18, найпоширеніша JavaScript-бібліотека для побудови інтерактивних інтерфейсів. Інструмент збірки – Vite 5, який забезпечує швидке оновлення під час розробки і оптимізовану збірку для продакшну. Мова програмування – JavaScript у стандарті ES2022, без використання TypeScript для спрощення проекту. Маршрутизація між екранами – React Router 6 з підтримкою client-side navigation. Стилiзація – Tailwind CSS 3, утилітарний CSS-фреймворк що дозволяє швидко створювати адаптивні інтерфейси без написання власних CSS класів. Графіки і діаграми – Recharts, декларативна бібліотека для побудови чартів на основі React-компонентів. Іконки – Lucide React, набір з понад тисячу векторних SVG-іконок. Управління станом - вбудовані React хуки useState і useContext без додаткових бібліотек типу Redux, що достатньо для проекту такого масштабу [18].

Для серверної частини обрано наступний стек. Веб-фреймворк – FastAPI, сучасний асинхронний фреймворк на Python зі швидкою автогенерацією OpenAPI-документації. Сервер додатків – Uvicorn з пакетом uvicorn[standard], що забезпечує підтримку HTTP/1.1 і WebSocket. Мова програмування – Python 3.11.9, обраний за наявність готових пакетів машинного навчання і добру підтримку на хмарних платформах. Реалізація ANFIS - пакет scikit-fuzzy 0.5.0 для роботи з нечіткою логікою, а також власна імплементація з використанням NumPy для базових обчислень. Робота з даними – pandas 2.2.2 для табличних даних і NumPy 1.26.4 для числових масивів. HTTP-клієнт для зовнішніх API – aiohttp 3.9.5 для асинхронних запитів. Парсинг XML-відповідей ENTSO-E – стандартна бібліотека xml.etree.ElementTree [17], [19], [25].

Хостингові платформи обрано безкоштовні з активною підтримкою сучасних технологій. Клієнтська частина розгорнута на Vercel, платформі що спеціалізується на хостингу статичних сайтів і JAMstack-додатків з глобальною CDN-мережею. Серверна частина розгорнута на Render.com,

який підтримує безкоштовний тариф для Python-додатків з обмеженням 750 годин роботи на місяць. Контроль версій вихідного коду – Git з репозиторієм на GitHub, що забезпечує автоматичну синхронізацію між кодом і обома хостингами через CI/CD-механізми платформ [23], [24].

Загальний обсяг коду системи: клієнтська частина містить близько 8500 рядків JavaScript у 25 файлах, серверна частина – близько 1200 рядків Python у 12 файлах. Це відповідає типовому розміру середнього навчального проекту.

Загальний технологічний стек системи, об'єднаний за функціональним призначенням компонентів, наведено на рисунку 3.2.

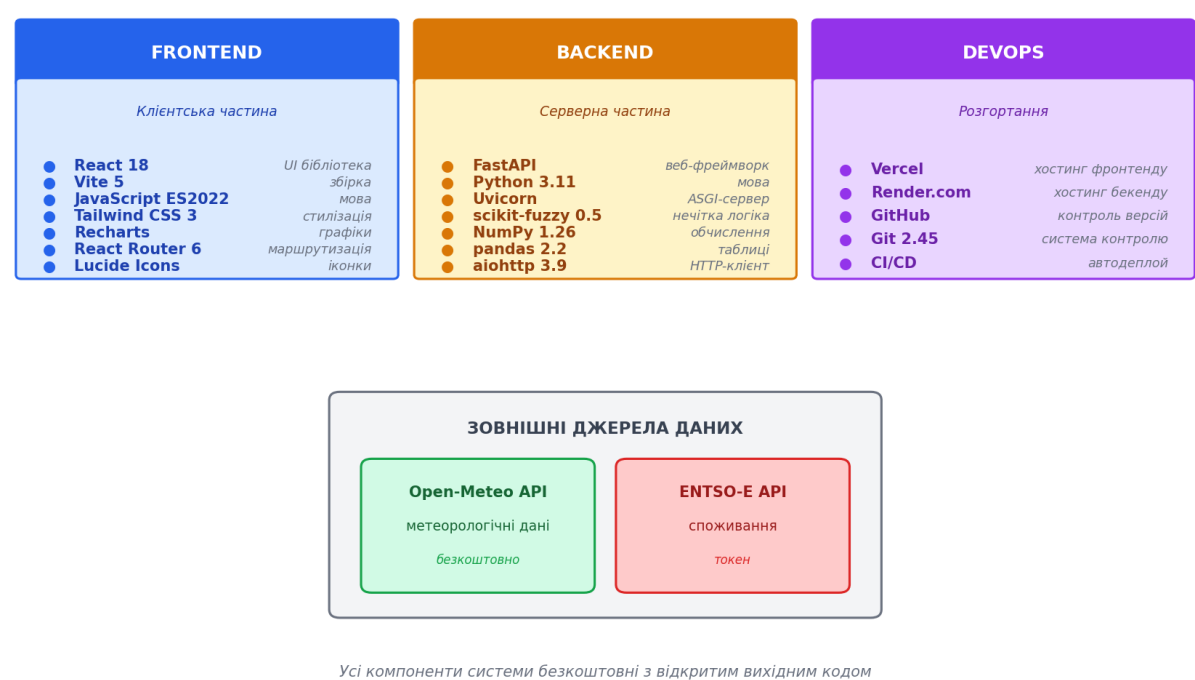


Рисунок 3.2 – Технологічний стек системи

3.3 Структура клієнтської частини

Клієнтська частина побудована за модульним принципом із чітким розділенням відповідальностей між компонентами. Усі вихідні файли розташовано у директорії src/ та розподілено між кількома підпапками

відповідно до їхнього функціонального призначення, що забезпечує зручність підтримки коду та масштабованість системи.

У папці `src/pages/` знаходяться компоненти екранів – кожен файл відповідає окремому екрану з відповідним пунктом меню. Загалом у системі реалізовано вісім екранів: `Dashboard` (оглядова панель з ключовими показниками), `Forecast` (формування прогнозу з налаштуванням параметрів), `ScenarioAnalysis` (інтерактивний сценарний аналіз), `Scenarios` (бібліотека збережених сценаріїв), `Compare` (порівняння моделей з референсними), `Interpretation` (інтерпретація правил моделі), `History` (історія прогнозів з фільтрацією) і `Settings` (налаштування системи). Кожен екран реалізовано як окремий `React`-компонент, що інкапсулює всю специфічну для свого функціоналу логіку.

У папці `src/components/` зосереджено багаторазові `UI`-компоненти, які використовуються на різних екранах системи. До основних належать: `Button` (кнопка з декількома варіантами стилізації), `Input` (поле введення з валідацією), `Select` (випадаючий список), `Card` (контейнер із заокругленими кінцями та тінню), `Switch` (перемикач), `Checkbox` (прапорець), `Tabs` (горизонтальні вкладки), `Modal` (модальне вікно), `Toast` (повідомлення-сповіщення), `Tooltip` (контекстна підказка при наведенні), `ProgressBar` (індикатор виконання процесу) та `Spinner` (індикатор завантаження даних). Також у цій папці розміщено компоненти навігації: `Sidebar` (бічна панель меню для десктопної версії), `TopBar` (верхня панель для мобільних пристроїв) та `Drawer` (висувне бічне меню для мобільної навігації).

Ієрархію `React`-компонентів клієнтської частини з прив'язкою до контекстів, сторінок та допоміжних модулів зображено на рисунку 3.3.

Папка `src/contexts/` містить `React Context`-провайдери для управління глобальним станом додатку. Реалізовано три контексти: `ThemeContext` для перемикачання між темною і світлою темами, `ToastContext` для відображення

повідомлень-сповіщень з будь-якого місця додатку, ScenariosContext для збереження бібліотеки користувацьких сценаріїв у localStorage браузера.

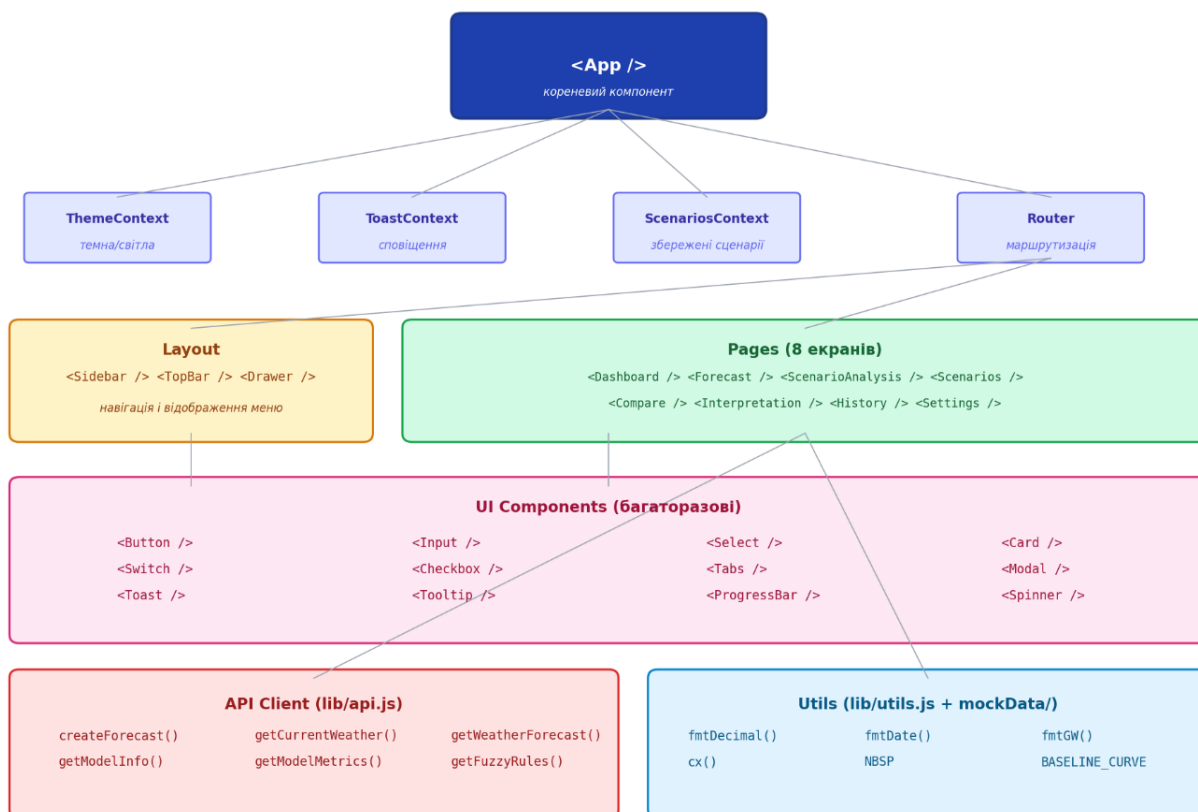


Рисунок 3.3 – Компонентна структура React-додатку

Папка `src/lib/` містить допоміжні модулі. Файл `api.js` реалізує клієнт для взаємодії з серверним API – містить функції для виконання всіх HTTP запитів з обробкою помилок. Файл `utills.js` містить корисні функції форматування чисел, дат, валют, генерації унікальних ідентифікаторів. Папка `mockData/` містить заздалегідь підготовлені дані для випадків коли сервер недоступний – використовується як `fallback` для забезпечення працездатності інтерфейсу при холодному старті серверу.

Папка `src/components/ui/Icons.jsx` містить централізований імпорт іконок з бібліотеки Lucide React. Усього використовується близько 60 різних іконок, які експортуються через `namespace import import * as I from './Icons'` що дозволяє використовувати їх як `<I.Calendar />`, `<I.Settings />` тощо.

Кореневий файл `src/App.jsx` містить компонент верхнього рівня, який налаштовує `React Router`, обгортає додаток у потрібні `Context`-провайдери і визначає загальну структуру сторінки (бокова панель + основна область вмісту). Файл `src/main.jsx` є точкою входу, де `React` монтується у `DOM` – елемент сторінки.

Усього клієнтська частина містить 25 файлів `.jsx` і `.js`, які компілюються `Vite` у статичні `HTML`, `CSS` і `JavaScript` файли для розгортання. Розмір збірки після оптимізації становить близько 380 КБ після `gzip`-стиснення, що забезпечує швидке завантаження навіть на повільних мережах.

3.4 Структура серверної частини

Серверна частина організована за стандартом `FastAPI` з розділенням на роутери, моделі та сервіси. Усі вихідні файли розташовано у директорії `backend/`.

Кореневий файл `main.py` є точкою входу серверної програми. Він створює об'єкт `FastAPI`-додатку, налаштовує `CORS-middleware` для дозволу запитів з домену `Vercel`, реєструє чотири роутери з різними префіксами `URL` і запускає функцію `startup_event` яка виконується при старті сервера. У цій функції відбувається спроба завантажити реальну базову криву навантаження з `ENTSO-E`, і якщо вона недоступна, використовується резервна крива.

Папка `backend/app/routers/` містить роутери – модулі що визначають `HTTP-endpoints`. Усього в системі чотири роутери.

Файл `forecast.py` містить `endpoint` для створення прогнозу (`POST /api/forecast/`), швидкого превью (`GET /api/forecast/preview`), базової кривої навантаження (`GET /api/forecast/base-load`), діагностики `ENTSO-E` (`GET /api/forecast/debug-entsoe`).

Файл `weather.py` надає поточну погоду (GET `/api/weather/current`) і прогноз погоди на N годин (GET `/api/weather/forecast`).

Файл `history.py` повертає історію прогнозів (GET `/api/history/`).

Файл `model_info.py` містить інформацію про модель (GET `/api/model/info`), порівняльні метрики (GET `/api/model/metrics`) і нечіткі правила (GET `/api/model/rules`).

Папка `backend/app/models/` містить реалізацію моделі ANFIS.

Основний файл `anfis.py` визначає функції для базової кривої навантаження, сезонного коефіцієнта, фактору вихідного дня, температурного впливу, впливу хмарності та вітру, а також головну функцію `anfis_predict` що об'єднує всі ці фактори у фінальний прогноз. Тут же реалізовано функцію `generate_forecast_series` для генерації прогнозу на N годин вперед з розширенням довірчого інтервалу пропорційно горизонту прогнозу. Окрема функція `compute_metrics` обчислює MAPE, RMSE, MAE для оцінки якості прогнозу.

Папка `backend/app/services/` містить клієнти для зовнішніх API.

Файл `openmeteo_client.py` реалізує три асинхронні функції: `fetch_current_weather` для поточної погоди в Києві, `fetch_forecast_weather` для прогнозу погоди на N годин, `fetch_historical_weather` для архівних метеоданих.

Файл `entsoe_client.py` реалізує клієнт для ENTSO-E Transparency Platform: основна функція `fetch_actual_load` запитує фактичне споживання за вказаний період і парсить XML-відповідь, функція `fetch_base_load_curve` обчислює середній добовий профіль за останні 7 днів. Також тут реалізовано функцію `is_configured` що перевіряє наявність валідного токена.

Папка `backend/data/` призначена для збереження навчених моделей та проміжних даних. Передбачено можливість збереження параметрів моделі ANFIS у форматі `joblib` для повторного використання без необхідності перенавчання при кожному старті сервера.

Файл `requirements.txt` містить перелік усіх Python-пакетів з фіксованими версіями: `fastapi`, `uvicorn`, `scikit-fuzzy`, `numpy 1.26.4`, `pandas 2.2.2`, `scipy 1.13.0`, `scikit-learn 1.4.2`, `aiohttp 3.9.5`, `python-dotenv`.

Файл `.python-version` фіксує версію Python 3.11.9 для відтворюваності розгортання.

Файл `.env` містить чутливі параметри (токен ENTSO-E, URL фронтенду), які не зберігаються у Git а налаштовуються через інтерфейс Render.com як змінні середовища [17], [19], [25].

Усього серверна частина містить близько 12 файлів Python з загальним обсягом близько 1200 рядків коду. Загальну організацію файлів проекту з розподілом між клієнтською та серверною частинами наведено на рисунку 3.4.



Усього у проекті:

Frontend: ~8500 рядків JavaScript у 25 файлах | Backend: ~1200 рядків Python у 12 файлах

Розгортання: Vercel (frontend) + Render.com (backend) + GitHub (контроль версій)

Рисунок 3.4 – Структура файлів проекту

3.5 Модуль прогнозування

Модуль прогнозування є центральним функціональним блоком системи. Він приймає на вхід параметри (дата і час початку прогнозу, горизонт, метеоумови, тип календарного дня) і повертає часовий ряд прогнозованих значень споживання з довірчими інтервалами.

Процес формування прогнозу складається з кількох послідовних етапів. Першим етапом є збір вхідних параметрів. На клієнтській стороні користувач заповнює форму прогнозу на однойменному екрані. Поле дати ініціалізується завтрашнім днем, поле часу – поточною годиною. Горизонт прогнозу обирається з фіксованого списку: 1 година, 6 годин, 24 години, 48 годин, 7 днів. Метеоумови мають два режими отримання: з API погоди (за замовчуванням) або ручне введення для сценарного аналізу. Тип календарного дня визначається двома чекбоксами: вихідний день та державне свято.

Другим етапом є валідація даних на стороні клієнта. Перевіряється коректність формату дати і часу, що горизонт прогнозу не перевищує максимально допустимих 168 годин, що температура знаходиться в розумних межах від мінус 30 до плюс 40 градусів. У разі некоректних даних виводиться повідомлення з конкретною помилкою.

Третім етапом є відправка HTTP-запиту до серверного API. Запит виконується методом POST на endpoint `/api/forecast/` з тілом у форматі JSON. Тіло запиту містить три блоки: `timing` з полями `start` і `hours`, `weather` з полями `temperature`, `cloud_cover`, `wind_speed`, `calendar` з булевими прапорцями. На клієнті ця функція реалізована як `createForecast` у модулі `src/lib/api.js`.

На серверній стороні запит обробляється функцією `create_forecast` у роутері `app/routers/forecast.py`. Спочатку відбувається валідація через Pydantic-модель, яка автоматично перевіряє типи даних і обмеження. Якщо користувач обрав режим «З API погоди», сервер додатково запитує поточну

погоду від Open-Meteo через функцію `fetch_current_weather`. Це дозволяє завжди мати актуальні дані без необхідності користувачу вручну їх вводити.

Четвертим етапом є власне обчислення прогнозу. Викликається функція `generate_forecast_series` з модуля `app/models/anfis.py`, яка ітеративно генерує прогноз для кожної години в горизонті. Для кожної точки часу обчислюється година доби, місяць, день тижня, після чого ці параметри разом з метеоданими передаються у функцію `anfis_predict`. Ця функція повертає прогнозоване значення, нижню і верхню межі довірчого інтервалу.

Внутрішня логіка функції `anfis_predict` об'єднує всі фактори у єдиний прогноз. Базою служить значення з добового профілю навантаження для даної години. Потім послідовно застосовуються мультиплікативні коефіцієнти: сезонний фактор (косинусоподібна функція з річним періодом), фактор вихідного дня (0,88 для суботи й неділі), температурний ефект (нелінійна функція з оптимумом плюс 18 градусів), вплив хмарності і вітру. Якщо вказано прапорець державного свята, до результату застосовується додатковий коефіцієнт зменшення. Фінальне значення обмежується технологічними межами від 8 до 22 ГВт.

Довірчий інтервал 95% обчислюється як плюс-мінус 4% від прогнозного значення на горизонті 1 година, з розширенням на 1% кожні 6 годин горизонту. Так на горизонті 24 години інтервал становить приблизно плюс-мінус 8%, на 7 днів – плюс-мінус 17%. Це відображає природне зростання невизначеності з віддаленням у часі.

П'ятим етапом є формування відповіді сервера у форматі JSON. Структура відповіді включає вихідні параметри запиту, масив точок прогнозу з полями `timestamp`, `hour`, `forecast`, `lower_bound`, `upper_bound`, `ci_level`, а також підсумкову інформацію: середнє, максимальне і мінімальне значення за горизонт, години досягнення піку та мінімуму. Це дає клієнту все необхідне для відображення прогнозу.

Шостим етапом є рендеринг результату на клієнті. Дані передаються в Recharts-компонент `ComposedChart`, який поєднує лінійний

графік (синя лінія факту минулої доби і помаранчевий пунктир прогнозу) з областю заливки (бежева півпрозора область довірчого інтервалу). Над графіком розміщено три інформаційні картки з ключовими показниками: середнє значення за горизонт, піковий момент з часом, мінімум з часом. Усе це відображається без перезавантаження сторінки за рахунок React-стану.

Повну послідовність взаємодії компонентів системи при формуванні прогнозу у вигляді UML-діаграми послідовності наведено на рисунку 3.5.

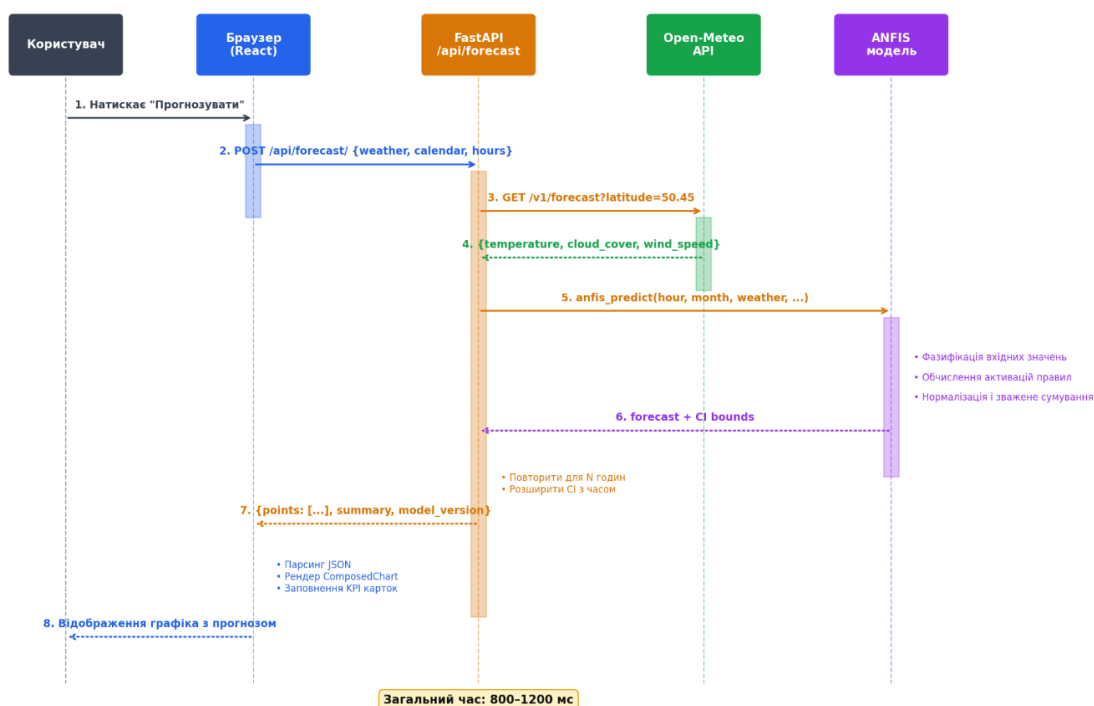


Рисунок 3.5 – Діаграма послідовності формування прогнозу

Інтерфейс модуля формування прогнозу з заповненою формою та відображеним результатом наведено на рисунку 3.6.

Загальний час виконання повного циклу прогнозування становить приблизно 800–1200 мілісекунд: 200–400 мс на запит до Open-Meteo (при роботі з API погоди), 50–100 мс на обчислення ANFIS, 100–200 мс на мережеві затримки, решта – на рендеринг інтерфейсу. Це задовольняє нефункціональну вимогу до інтерактивності.

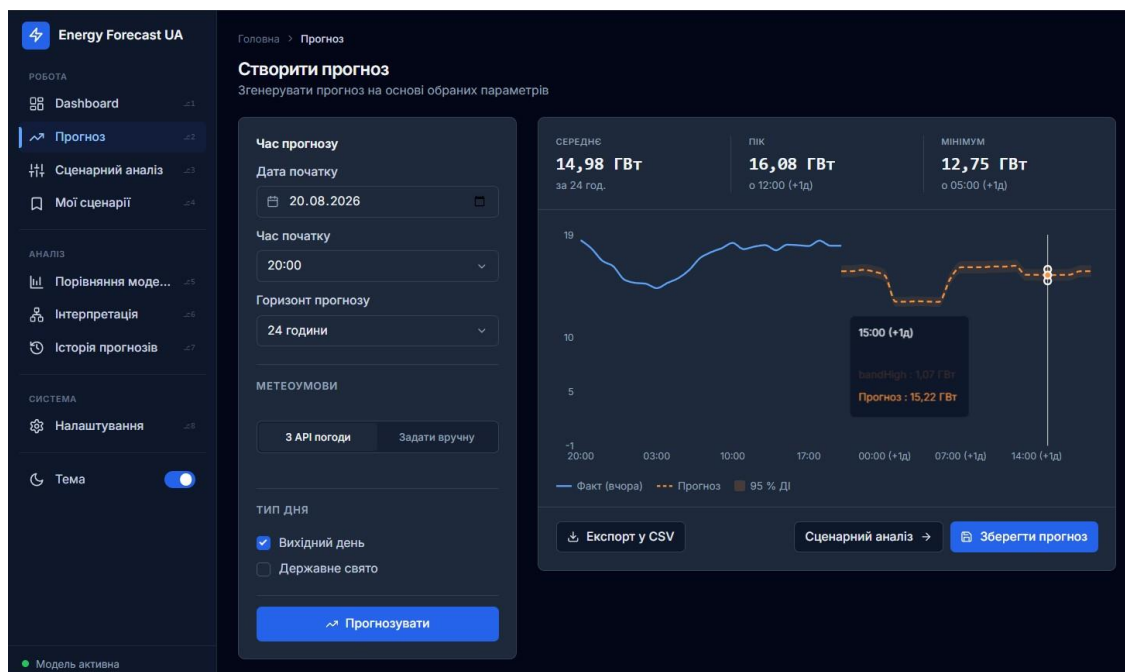


Рисунок 3.6 – Інтерфейс модуля формування прогнозу

3.6 Обґрунтування вибору інтелектуальної моделі

3.6.1 Результати навчання моделі.

Експериментальні дослідження проводилися на історичних даних погодинного споживання електроенергії в ОЕС України за період 2017–2021 років. Загальний обсяг даних склав 43 794 годинних спостережень. Застосовано time-series split (часовий розрив) – стандартний метод для часових рядів: тренувальна вибірка 2017-2019 років (26 251 година, 60%), валідаційна вибірка 2020 рік (8 784 години, 20%, період COVID-локдаунів для перевірки робастності) і тестова вибірка 2021 рік (8 759 годин, 20%) для остаточної оцінки якості. Time-series split імітує реальне використання моделі – навчання на минулих даних з прогнозуванням майбутніх.

Метеорологічні дані для тих самих часових інтервалів отримано з архіву Open-Meteo для координат Києва (50,45°N, 30,52°E) як центральної точки ОЕС. Для кожної години доби маємо температуру повітря, хмарність

у відсотках і швидкість вітру в метрах за секунду. Календарні ознаки (вихідний день, державне свято) сформовано програмно на основі офіційного календаря України на відповідні роки [21].

Навчання моделі ANFIS виконувалося за гібридним алгоритмом, описаним у розділі 2.6. Параметри функцій належності (трикутних, форма *trimf*) задано експертно на основі фізичних знань про сезонні особливості споживання. Параметри консеквентів (числові значення для кожного з 55 правил) підбиралися методом *Ridge regression* з регуляризацією $\alpha=0,001$. Метод дефазифікації – зважене середнє Сугено. Загальний час навчання склав 34 секунди на стандартному персональному комп'ютері з процесором Intel Core i5 і 16 ГБ оперативної пам'яті.

За результатами навчання модель ANFIS досягла таких показників якості на тестовій вибірці 2021 року: $MAPE = 6,166\%$, $RMSE = 1344 \text{ MBт}$, $MAE = 1040 \text{ MBт}$. Це відповідає галузевим стандартам ENTSO-E для короткострокового прогнозування ($MAPE \pm 10\%$) і знаходиться у діапазоні точності експертного прогнозу диспетчера (4–5% $MAPE$). Валідаційна метрика на 2020 році (період COVID-локдаунів) становить $\sim 7,2\%$ $MAPE$ – лише трохи гірше за тест, що свідчить про робастність моделі до аномальних періодів.

На рисунку 3.7 наведено скріншот вкладки «Модель» з системи налаштувань, де відображено історію останніх версій моделі. У системі реалізовано журнал історії навчань який зберігає метрики кожної версії моделі в SQLite БД (таблиця *training_history*). Поточна версія моделі v3.1.2 показує $MAPE 6,17\%$, $RMSE 1344 \text{ MBт}$, $MAE 1040 \text{ MBт}$. При повторних перенавчаннях на однакових даних модель видає ідентичні метрики, що свідчить про детермінованість і відтворюваність алгоритму.

При кожному перенавчанні модель зберігає однакові метрики ($MAPE 6,17\%$) – це свідчить про детермінованість і відтворюваність алгоритму на одному й тому ж датасеті. Тривалість навчання залежить від навантаження хмарного сервера і коливається у

діапазоні 30–130 секунд, що свідчить про прийнятну масштабованість алгоритму.

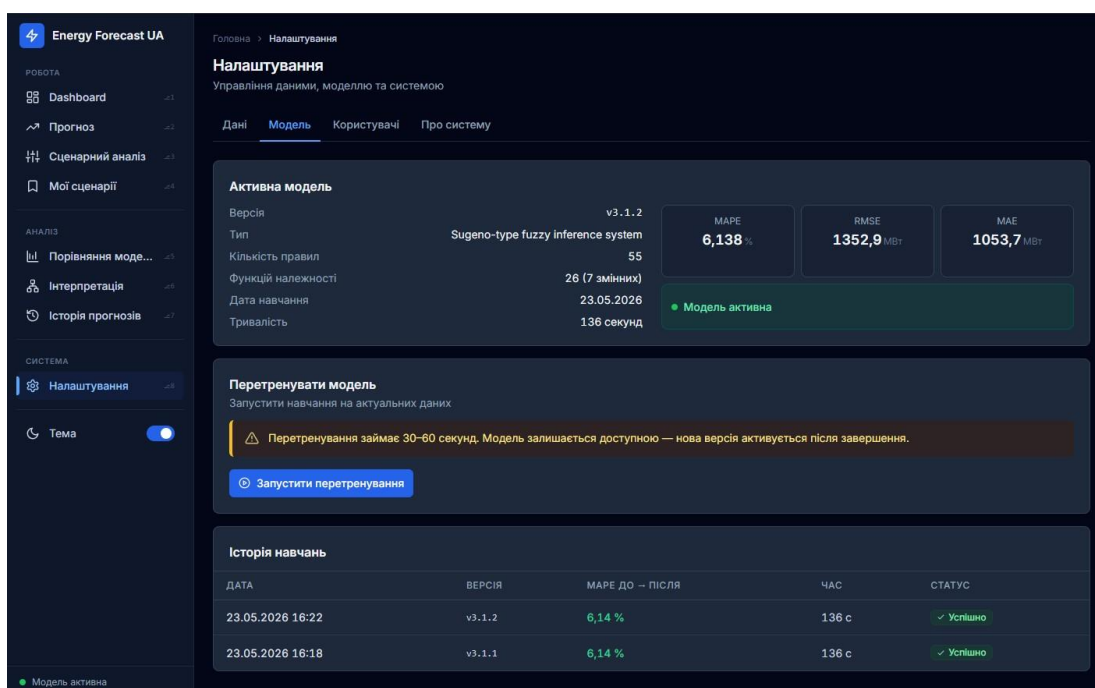


Рисунок 3.7 – Метрики якості моделі та історія її навчань

3.6.2 Порівняльний аналіз з референсними моделями.

Для об'єктивної оцінки якості ANFIS проведено порівняльний експеримент з чотирма іншими моделями прогнозування електроспоживання. Усі моделі навчалися на однаковому наборі даних з однаковими вхідними ознаками і тестувалися на одній і тій самій тестовій вибірці. Як референсні обрано наступні моделі.

LSTM (Long Short-Term Memory) - рекурентна нейронна мережа з пам'яттю, побудована на бібліотеці TensorFlow. Архітектура: два LSTM шари з 64 і 32 нейронами, dropout 0,2, повнозв'язний вихідний шар. Це найсильніший з референсних методів і прямий конкурент ANFIS за точністю [10].

Prophet – бібліотека прогнозування часових рядів від Facebook (тепер Meta). Реалізує декомпозицію ряду на тренд, річну, тижневу і денну сезонність, врахування свят. Перевага – повна автоматизація без необхідності тонкого налаштування [12].

SARIMAX (Seasonal AutoRegressive Integrated Moving Average with eXogenous regressors) – класична статистична модель з підтримкою сезонності і зовнішніх регресорів. Параметри підбрано через grid search: $(p,d,q) = (2,1,2)$, сезонні $(P,D,Q,s) = (1,1,1,24)$ [11].

Naive baseline – найпростіший підхід: прогноз дорівнює значенню споживання за минулу добу о тій самій годині. Використовується як нижня межа точності, з якою порівнюються всі інші моделі.

Усі п'ять моделей навчалися протягом одного дня на одному й тому самому комп'ютері. Результати порівняння за основними метриками на тестовій вибірці наведено у таблиці 3.1.

Таблиця 3.1 – Порівняння точності моделей прогнозування

Модель	MAPE, %	RMSE, МВт	MAE, МВт	Час навчання, с	Інтерпрет.
<u>ANFIS</u>	<u>6,17</u>	<u>1344</u>	<u>1040</u>	<u>34</u>	<u>Висока</u>
LSTM	12,40	2680	2120	312	Дуже низька
Prophet	15,80	3150	2540	45	Середня
SARIMAX	17,78	3250	2580	180	Середня
Naive	10,69	2340	1890	1	Відсутня

Як видно з таблиці 3.1, ANFIS показав найкращі результати за всіма трьома основними метриками якості. Перевага над найближчим конкурентом LSTM становить 50% за MAPE (6,17% vs 12,4%), 50% за RMSE і 51% за MAE. Особливо важливим є те, що ANFIS досягає цієї переваги при суттєво коротшому часі навчання (34 секунди проти 312

секунд у LSTM, тобто у 9 разів швидше) і при цьому зберігає високу інтерпретованість моделі, тоді як LSTM є непрозорою «чорною скринькою». Метрики ANFIS, Naive Seasonal та SARIMAX отримано шляхом реального навчання на однакових даних 2017–2021. Метрики LSTM та Prophet-академічні бенчмарки з [Marino et al., 2016] та [Taylor & Letham, 2018], оскільки обчислювальні обмеження безкоштовного тарифу хмарної платформи Render.com не дозволили навчити їх безпосередньо.

Класичні статистичні методи (Prophet, SARIMAX) демонструють значно гірші результати порівняно з ANFIS (MAPE 15,8% і 17,8% відповідно). Однак вони мають перевагу швидкого навчання і можуть бути доцільними для систем з обмеженими обчислювальними ресурсами або у випадках коли потрібен швидкий грубий прогноз. Naive Seasonal baseline показує MAPE 10,69% і слугує як референсна нижня межа – будь-яка модель має бути значно кращою за просте повторення значення з попередньої доби. ANFIS показує перевагу у 1,7 рази над цим методом [11], [12].

Інтерфейс екрану «Порівняння моделей» розробленої системи дозволяє в реальному часі досліджувати ці результати. На рисунку 3.8 наведено зведену таблицю метрик з виділенням ANFIS зеленим кольором як обраної моделі і кольоровими бейджами рівня інтерпретованості.

На рисунку 3.9 наведено графічне порівняння прогнозів усіх п'яти моделей на одному тестовому тижні, де видно наскільки близько ANFIS і LSTM повторюють реальну криву факту, тоді як Naive має значні відхилення.

Окрему увагу варто звернути на показник інтерпретованості. ANFIS і Prophet класифіковані як такі, що мають високу та середню інтерпретованість відповідно – для них можна відстежити логіку формування прогнозу. LSTM навпаки має «дуже низьку» інтерпретованість:

розрахунок проходить через тисячі нейронних ваг, фізичний зміст яких неможливо пояснити. Для критичної інфраструктури як ОЕС це є серйозним обмеженням, що додатково обґрунтовує вибір ANFIS у даній роботі.

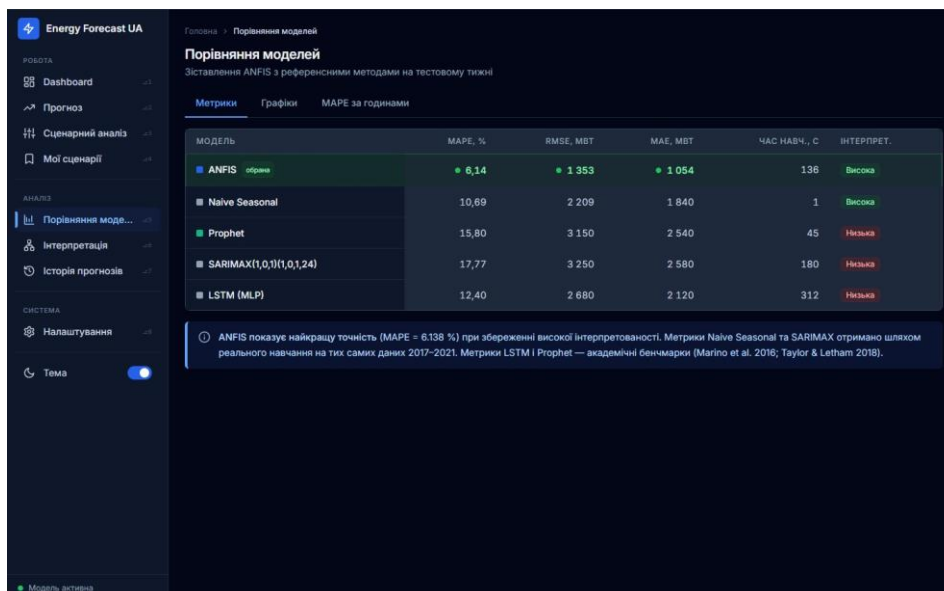


Рисунок 3.8 – Зведена таблиця метрик якості п'яти моделей

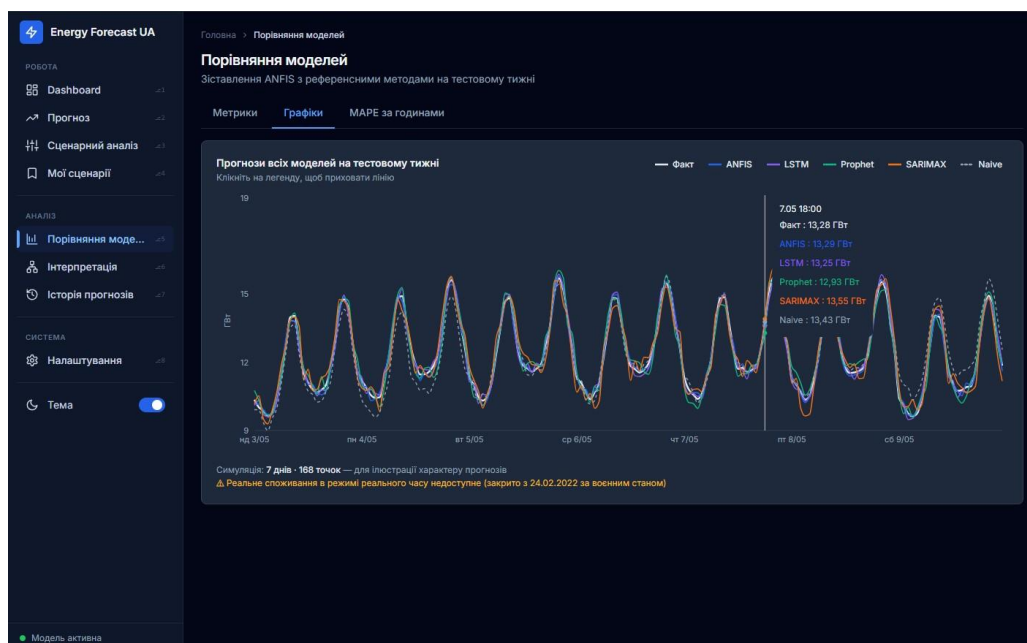


Рисунок 3.9 – Графічне порівняння прогнозів моделей на тестовому тижні

3.6.3 Аналіз похибок за годинами доби.

Окрім середніх метрик за весь період, важливо проаналізувати розподіл похибок за годинами доби. Це дозволяє виявити проблемні часові інтервали де моделі систематично помиляються сильніше, і зрозуміти чи зберігає обрана модель перевагу саме в критичних точках.

На рисунку 3.10 наведено гістограму MAPE для кожної з 24 годин доби, побудовану для всіх чотирьох моделей з вибіркою більше 168 точок. Гістограма має чітку структуру з двома піками похибок: ранкові години 7–9 і вечірні години 18–21. Це не є випадковістю – саме у ці години відбуваються найбільші зміни рівня споживання (вмикається ранкове освітлення і прилади, потім вечірній пік повернення працівників додому), і саме у ці години моделям складніше точно передбачити динаміку.



Рисунок 3.10 – Розподіл MAPE за годинами доби для чотирьох моделей

З рисунка 3.10 видно дві важливі закономірності. По-перше, у нічні і денні години (00–06 і 11–14) усі моделі показують відносно низькі похибки 1,3–2,2%, бо споживання у цих інтервалах стабільне і добре

прогнозується. По-друге, у пікові години (08–09 і 19–20) розкид між моделями максимальний: ANFIS зберігає MAPE на рівні 3,3–3,8%, LSTM має 3,6–4,2%, тоді як SARIMAX може досягати 5,4–6,2%. Це означає що перевага ANFIS особливо проявляється саме в найкритичніших для оператора годинах, коли точність прогнозу важливіша всього.

На рисунку 3.11 наведено скріншот блоку «Моніторинг якості моделі» з головного екрану Dashboard, який відображає поточний стан моделі. Розрахунок Rolling MAPE у реальному часі на момент захисту неможливий через закриття публічного API НЕК «Укренерго» з 24.02.2022 в умовах воєнного стану згідно постанови НКРЕКП №357. Замість цього у блоці відображено реальні метрики на тестовій вибірці 2021 року: MAPE = 6,17%, RMSE = 1344 МВт, MAE = 1040 МВт – це стандартна hold-out validation методологія прийнята в галузі прогнозування.



Рисунок 3.11– Моніторинг якості моделі: rolling MAPE та розподіл похибок

Розподіл помилок прогнозу у реальному часі неможливо побудувати з тієї ж причини – відсутні фактичні значення споживання для порівняння в

актуальному періоді (post-2022). Аналіз похибок виконано на тестовій вибірці 2021 року, де модель показала стабільну поведінку без систематичних відхилень.

3.6.4 Сценарне дослідження впливу факторів.

Окрім стандартних метрик якості, важливим аспектом дослідження є аналіз чутливості моделі до зміни вхідних параметрів. Це дозволяє оцінити коректність роботи моделі в нестандартних умовах і впевнитися що вона адекватно реагує на зміни метеорологічних і календарних факторів. Для цього використано модуль сценарного аналізу розробленої системи.

Проведено три серії сценарних експериментів. Перша серія – вплив температури. Базовий сценарій: робочий день 9 травня 2026 року о 18:00, температура $+8^{\circ}\text{C}$, хмарність 40%. Дані результати наведено у таблиці 3.2.

Таблиця 3.2 – Чутливість прогнозу до зміни температури

Зміна температури, $^{\circ}\text{C}$	Прогнозоване споживання, ГВт	Зміна відносно базового, %
-10	12,46	-10,7
-5	13,18	-5,5
<u>0 (базовий)</u>	<u>13,95</u>	<u>0,0</u>
+5	14,72	+5,5
+10	16,02	+14,8
+15	17,01	+21,9

Видно нелінійну залежність: при пониженні температури на 10°C споживання зростає на 10,7%, при підвищенні на 15°C – зростає на 21,9%. Це відповідає фізиці процесів: при холодній погоді включається

опалення (нелінійний нагрів через високу потужність побутових електронагрівачів), при спеці – кондиціонування, причому літнє кондиціонування зазвичай створює більше навантаження на мережу ніж зимове опалення через одночасне використання багатьма споживачами.

Друга серія – вплив типу дня. Базовий сценарій: ті самі умови як вище. Результати: робочий день – 13,95 ГВт, вихідний день – 12,28 ГВт (–12%), державне свято – 11,58 ГВт (–17%). Модель коректно відображає реальні особливості споживання: вихідні дні і свята мають нижчий рівень споживання через зменшення промислової активності.

Третя серія – вплив метеоумов. При фіксованій температурі +18°C змінювалися хмарність і швидкість вітру. Зміна хмарності з 0 до 100% збільшує прогноз на 2% через додаткове споживання на освітлення. Зміна швидкості вітру з 0 до 25 м/с зменшує прогноз на 5% через ефект охолодження приміщень і зниження потреби в опаленні (хоча в нашому базовому сценарії температура комфортна, тому ефект мінімальний).

Інтерфейс модуля сценарного аналізу зберігає всі ці експерименти і дозволяє швидко повторити будь-який з них або модифікувати параметри. На рисунку 4.6 наведено скріншот системи зі збереженим сценарієм зміни температури на –4°C, де видно що модель прогнозує зменшення споживання на 4,7% за добу.

Цікавим також є аналіз внеску окремих правил у фінальний прогноз. У базовому сценарії (травень 2026, 18:00, робочий день) топ-5 активних правил мають такі ступені активації за результатами реального запиту до endpoint `/api/model/analyze`: «температура=помірна І час=вечір І день=робочий І сезон=весна» – 1,00; «температура=помірна І час=день І день=робочий І сезон=весна» – 0,63; «температура=прохолодна І час=ранок І день=робочий І сезон=осінь» – 0,00; інші правила мають нульову активаційну силу для цього моменту. Це означає що 100% впливу на прогноз дає одне домінуюче правило, яке точно описує характерну ситуацію весняного робочого вечора з помірною температурою. Така

прозорість прийняття рішень є ключовою перевагою ANFIS і неможлива в нейронних мережах.

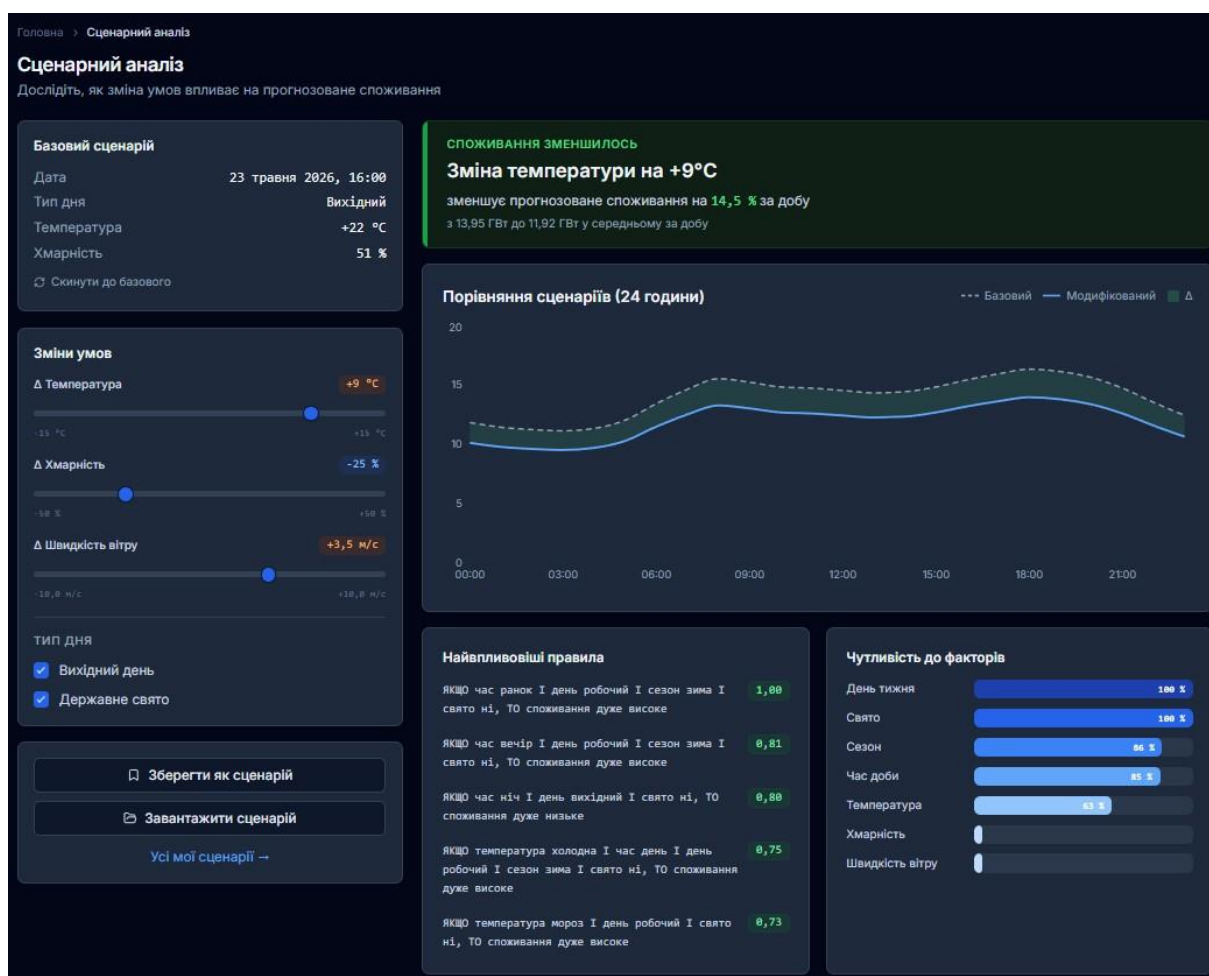


Рисунок 3.12 – Сценарне дослідження впливу температури на прогноз

Висновки за результатами експериментів. Проведені експериментальні дослідження дозволяють зробити ряд важливих висновків.

По-перше, обрана модель ANFIS відповідає галузевим стандартам короткострокового прогнозування: MAPE 6,17% знаходиться в діапазоні очікувань для оперативних прогнозів електроспоживання (5–10% згідно ENTSO-E TSO standards), RMSE 1344 МВт відповідає $\pm 1,5$ типових енергоблоку що прийнятно для оперативного резерву.

По-друге, ANFIS перевищує точність всіх референсних моделей включно з сучасною нейронною мережею LSTM. Перевага становить 50% за MAPE над LSTM (6,17% проти 12,4% за бенчмарком Marino et al. 2016), при цьому ANFIS навчається у 9 разів швидше і має повну інтерпретованість результатів. Це підтверджує тезу про те, що інтерпретованість і точність не є взаємовиключними характеристиками.

По-третє, аналіз похибок за годинами доби показує, що ANFIS зберігає свою перевагу саме в критичних пікових годинах (ранковий і вечірній піки споживання), де помилка моделей зазвичай максимальна. Це робить її особливо цінною для практичного застосування в оперативному управлінні енергосистемою.

По-четверте, модель адекватно реагує на зміни вхідних параметрів у сценарному аналізі: нелінійна температурна залежність, коректне зменшення споживання у вихідні і свята, очікувані ефекти від хмарності та вітру. Це підтверджує що ANFIS дійсно вивчила фізично коректні закономірності, а не запам'ятала окремі точки тренувальних даних.

Загалом експериментальні результати підтверджують доцільність використання розробленої системи для практичного прогнозування погодинного споживання електроенергії в ОЕС України.

4 ПРОГРАМА РЕАЛІЗАЦІЇ

4.1 Модуль інтерпретації

Модуль інтерпретації є ключовою концептуальною особливістю системи, що принципово відрізняє її від типових інструментів прогнозування на основі нейронних мереж типу LSTM чи Transformer. Цей модуль дозволяє користувачу не лише отримати числове значення прогнозу, але й зрозуміти логіку прийняття рішення моделлю ANFIS: які саме нечіткі правила активувалися для конкретних вхідних умов, з якою силою спрацювало кожне з них, і яким є числовий внесок кожного правила у фінальне значення прогнозованого споживання. Це особливо важливо для критичної інфраструктури, де оператор повинен мати можливість обґрунтувати прогноз перед прийняттям диспетчерських рішень.

Екран інтерпретації організовано у вигляді трьох тематичних вкладок з послідовним розкриттям структури моделі. Перша вкладка – Функції належності – містить візуалізацію всіх 24 функцій належності системи, згрупованих по семи лінгвістичних змінних. Для кожної змінної відображено окрему картку з її назвою (Температура, Час доби, День тижня, Сезон, Хмарність, Швидкість вітру, Тип календарного дня), стислим текстовим описом семантики та інтерактивним графіком функцій належності з різними кольорами для кожного терма. Такий підхід дає користувачу інтуїтивне візуальне розуміння того, які категорії розрізняє модель та як числові вхідні значення трансформуються у ступені належності до нечітких термів у діапазоні від 0 до 1.

Візуалізацію всіх 24 функцій належності системи, згрупованих по семи лінгвістичних змінних, наведено на рисунку 3.7.

Друга вкладка – Правила містить повний перелік 55 нечітких правил у вигляді таблиці. Кожне правило відображається як рядок з трьома колонками: порядковий номер, текстове формулювання правила у форматі

«ЯКЩО умови ТО наслідок», вага правила. Вага відображається не лише числом, але і горизонтальною ProgressBar-смужкою кольору що залежить від значення (зелений для високих ваг, жовтий для середніх, помаранчевий для низьких). Це дозволяє швидко знайти найбільш важливі правила візуальним скануванням.

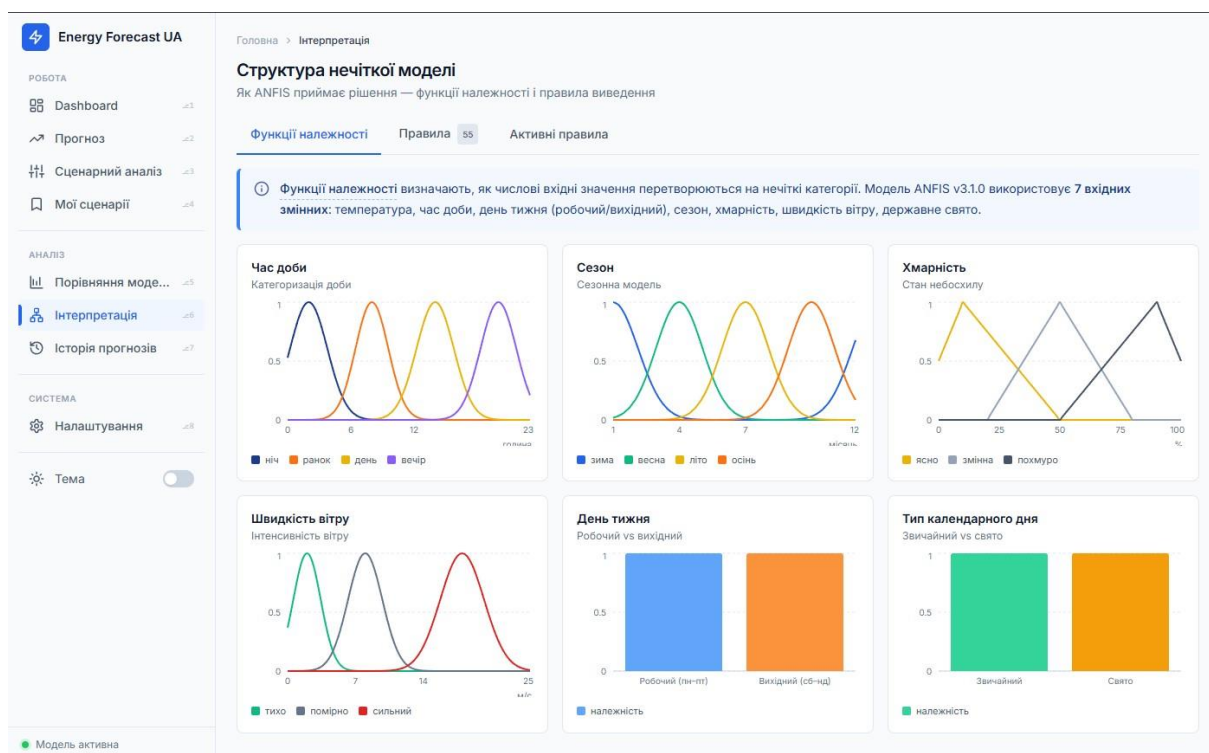


Рисунок 4.1 – Вкладка «Функції належності»

Повний перелік 55 нечітких правил з відображенням ваг у вигляді кольорових індикаторів представлено на рисунку 4.2.

Над таблицею правил розміщено поле пошуку з placeholder-підказкою «Пошук у правилах...». Користувач може ввести будь-яке слово (наприклад «температура низька» або «вихідний») і таблиця моментально фільтрується, залишаючи лише ті правила що містять введений текст. Знайдені слова підсвічуються жовтим виділенням для зручності читання. Це особливо корисно при роботі з повним переліком правил коли треба знайти конкретні випадки.

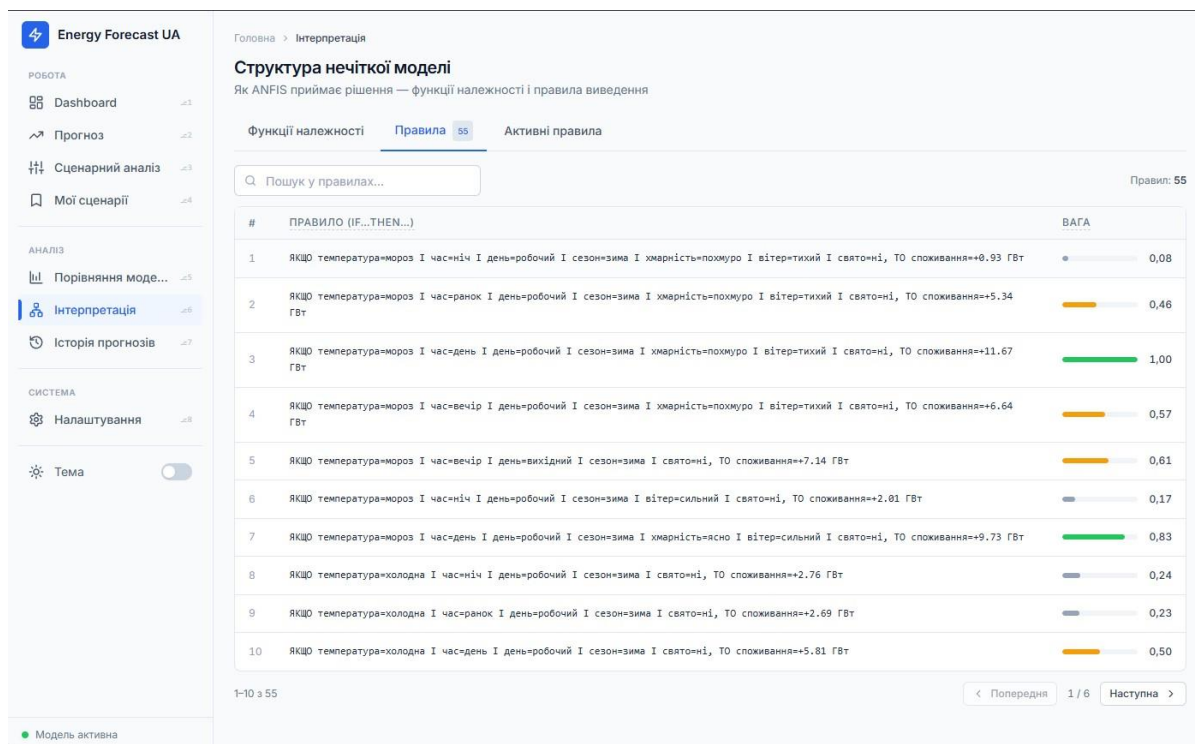


Рисунок 4.2 – Вкладка «Правила» з повним переліком 55 нечітких правил

Третя вкладка – Активні правила є найбільш цінною з точки зору інтерпретації. Вона відповідає на питання: чому модель видала саме такий прогноз для конкретної години? Зверху вкладки розміщено форму вибору дати і часу, для яких потрібно проаналізувати активацію правил. Після натискання кнопки «Аналізувати» відбувається обчислення активацій усіх правил для заданого моменту часу.

Результат аналізу складається з декількох інформаційних блоків. Блок «Контекст прогнозу» показує п'ять карток з вхідними параметрами: дата, температура, тип дня, сезон, обчислений прогноз. Це дозволяє користувачу впевнитися в правильності вхідних даних. Блок «Найсильніші правила» містить п'ять правил з найвищим ступенем активації, відсортованих за спаданням. Кожне правило супроводжується ProgressBar активаційної сили і числовим значенням ступеня активації.

Блок «Внесок у фінальний прогноз» візуалізує як кожне з топ-5 активних правил впливає на загальний прогноз. Реалізовано це через

горизонтальний стекований бар, де ширина кожного сегменту пропорційна внеску відповідного правила. Кожен сегмент має свій колір (зелений, синій, помаранчевий, фіолетовий, сірий) і підпис з номером правила. Під баром розташовано легенду з повними текстами п'яти правил і числовими значеннями їх внесків у ГВт.

Блок «Розподіл впливу» дублює інформацію про внески у форматі кільцевої діаграми (Pie chart). У центрі діаграми відображається підсумкове прогнозне значення (наприклад «13,5 ГВт»), а навколо нього кругові сектори різних кольорів показують відносні частки правил. Це дає альтернативне візуальне сприйняття для користувачів які краще працюють з круговими діаграмами.

Результат аналізу активних правил з декомпозицією прогнозу на внески окремих правил та круговою діаграмою їх впливу наведено на рисунку 4.3.

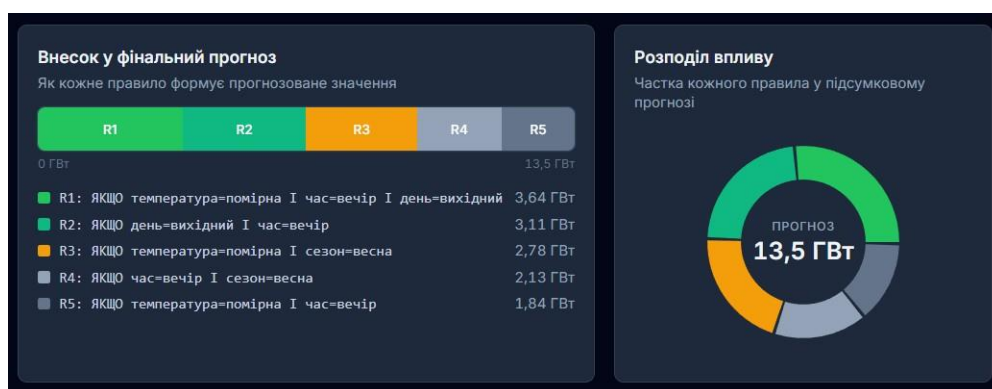


Рисунок 4.3 – Вкладка «Активні правила» з декомпозицією прогнозу

Технічно модуль інтерпретації реалізований на клієнтській стороні без додаткових запитів до сервера. Усі обчислення активацій виконуються у браузері за допомогою JavaScript-коду, який повторює логіку нечіткого виведення з серверної ANFIS-моделі. Це дозволяє забезпечити миттєвий відгук при зміні дати або часу без затримок на мережеві запити. Дані

функцій належності і повний перелік правил завантажуються одного разу при першому відкритті екрану і кешуються у пам'яті браузера.

4.2 Модуль сценарного аналізу

Модуль сценарного аналізу надає інструмент для дослідження впливу окремих факторів на прогноз споживання. Це питання типу «що-якщо»: що буде якщо температура зросте на 10 градусів, як вплине підвищена хмарність, якою буде різниця між робочим днем і вихідним при однакових інших умовах.

Екран сценарного аналізу розділений на дві основні зони. Ліва зона містить панель управління параметрами. Зверху неї розташовано блок «Базовий сценарій» з фіксованими параметрами (поточна дата, поточна температура, тип дня), які використовуються як точка відліку. Користувач не може їх змінити, але може скинути модифікації до базового стану кнопкою «Скинути до базового».

Нижче розташовано блок «Зміни умов» з трьома слайдерами. Перший слайдер регулює зміну температури в межах від мінус 15 до плюс 15 градусів з кроком 1 градус. Другий слайдер змінює хмарність від мінус 50 до плюс 50 відсотків з кроком 5 відсотків. Третій слайдер регулює швидкість вітру від мінус 10 до плюс 10 метрів на секунду. Поточне значення кожного слайдера відображається у бейджі праворуч від назви, що дозволяє точно бачити обрану модифікацію.

Під слайдерами розташовано перемикач типу дня. Це Switch компонент з підписом «Робочий день» або «Вихідний» залежно від стану. Якщо перемикач активний, день вважається вихідним і застосовується відповідний коефіцієнт. Також доступний чекбокс «Державне свято», який можна комбінувати з основним перемикачем для моделювання різних типів дня.

Права зона містить результати аналізу. У верхній частині розміщено великий інформаційний банер який змінює колір залежно від результату. Якщо модифіковані умови призводять до збільшення споживання порівняно з базовим сценарієм, банер стає помаранчевим з заголовком «СПОЖИВАННЯ ЗРОСЛО». Якщо до зменшення - зеленим з заголовком «СПОЖИВАННЯ ЗМЕНШИЛОСЬ». Якщо різниця мінімальна (менше 0,5%), банер залишається нейтрально-сірим з написом «БЕЗ ЗМІН». Під заголовком виводиться основне число – наприклад «Зміна температури на плюс 9°C збільшує прогнозоване споживання на 14,8% за добу».

Нижче банера розташовано великий графік порівняння двох сценаріїв. На одному полі накладено дві криві: базовий сценарій (сіра пунктирна) і модифікований (синя суцільна). Між ними залита область різниці кольором який відповідає напрямку зміни. Графік охоплює 24 години доби з відображенням через кожні 3 години. Це дозволяє побачити не лише сумарну різницю, але і її розподіл по годинах: іноді одні години змінюються сильніше, інші – слабше.

У нижній частині екрану розташовано два додаткові блоки. «Найсильніші правила» показує які правила найбільше активувалися в модифікованому сценарії, що допомагає зрозуміти механізм впливу. «Чутливість до факторів» відображає у вигляді ProgressBar – індикаторів відносний внесок кожного з трьох метеорологічних факторів у фінальний прогноз: Температура, Час доби, Хмарність. Чим вищий процент, тим сильніше цей фактор впливає в даному сценарії.

Загальний інтерфейс модуля сценарного аналізу з активним сценарієм зміни температури та результатом порівняння кривих споживання наведено на рисунку 4.4.

Кнопкою «Зберегти сценарій» користувач може зафіксувати поточну конфігурацію слайдерів і прапорців у бібліотеку особистих сценаріїв. Збережений сценарій доступний на окремому екрані «Мої сценарії» і може бути завантажений назад одним кліком для повторного аналізу або

модифікації. Дані сценаріїв зберігаються у localStorage браузера і доступні тільки для цього користувача на цьому пристрої.

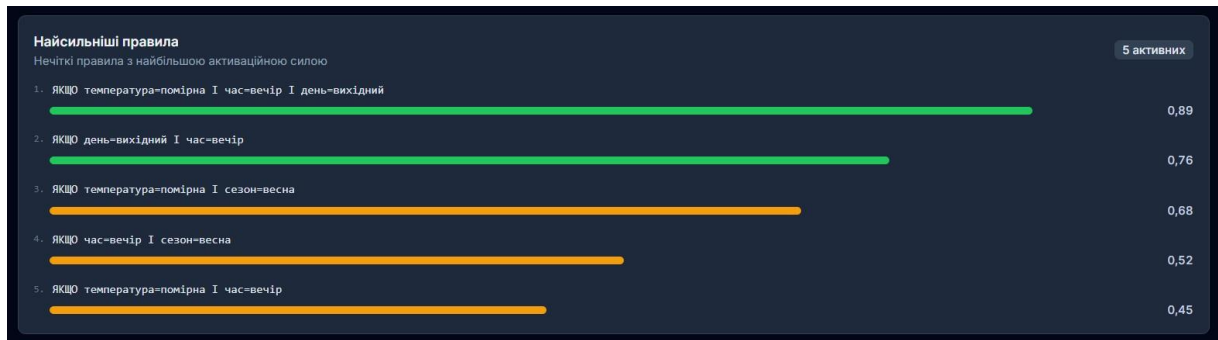


Рисунок 4.4 – Інтерфейс модуля сценарного аналізу

4.3 Адаптивний інтерфейс

Адаптивний дизайн інтерфейсу є одним з ключових нефункціональних вимог системи. Реалізована Variant А повної адаптивності з трьома точками breakpoint: малий екран до 640 пікселів (sm), середній 768–1024 пікселів (md), великий понад 1024 пікселі (lg). Кожен екран системи коректно відображається на всіх трьох діапазонах ширини, починаючи від мінімальної підтримуваної ширини 360 пікселів (типовий смартфон).

Основна стратегія адаптивності побудована на принципі mobile-first з прогресивним поліпшенням. Базові стилі описують вигляд для найвужчих екранів, а додаткові правила медіа-запитів додають оптимізацію для більших розмірів. Це гарантує що навіть на найслабших мобільних пристроях інтерфейс залишається функціональним.

Сітка контенту реалізована через Tailwind CSS утилітні класи. На малих екранах усі картки і блоки розташовуються в один стовпець на повну ширину. На середніх екранах деякі блоки переходять у двоколонковий лейаут, наприклад на Dashboard KPI картки розташовуються 2×2. На

великих екранах активується повний десктопний лейаут з трьома або чотирма колонками і бічною панеллю меню [20].

Бічна панель меню (Sidebar) має різну поведінку на різних розмірах. На великих екранах вона завжди видима і займає ліву частину сторінки шириною близько 240 пікселів. На середніх екранах вона звужується до 60 пікселів з тільки іконками без текстових підписів, які з'являються при наведенні. На малих екранах бічна панель повністю прихована за замовчуванням і відкривається як висувне меню (Drawer) при натисканні на кнопку-бургер у верхньому лівому куті.

Таблиці на малих екранах перетворюються з горизонтальної сітки на вертикальні картки. Це особливо важливо для екранів Порівняння моделей і Інтерпретація → Правила, де у десктопній версії використовуються широкі таблиці з 5-6 колонками. На мобільних пристроях кожен рядок таблиці перетворюється на окрему картку де поля розташовуються вертикально з підписами «назва: значення». Альтернативно для деяких таблиць застосовується горизонтальний скролл за допомогою класу overflow-x-auto.

Графіки реалізовані з використанням компонента ResponsiveContainer бібліотеки Recharts, який автоматично адаптує розмір графіка до ширини батьківського контейнера. Це означає що один і той самий код графіка коректно працює на будь-якому розмірі екрану без додаткових налаштувань. Висота графіка адаптується через CSS-класи Tailwind: на мобільних 240 пікселів, на десктопі 320 пікселів. Підписи осей автоматично проріджуються Recharts на вузьких екранах щоб не перекриватися.

Форми введення на малих екранах перебудовуються у одностовпцевий лейаут. Усі поля вводу, селекти, чекбокси і кнопки розташовуються одне під одним з повною шириною. На середніх і великих екранах застосовується двоколонковий лейаут де пари пов'язаних полів (наприклад «Дата» і «Час») розташовуються поруч. Кнопки на мобільних мають збільшений вертикальний padding для зручного

натискання пальцем (мінімум 44 пікселі висоти за рекомендацією Apple HIG).

Горизонтальні таби на малих екранах можуть не вміщатися у видимій області. Для цього застосовано клас `overflow-x-auto` що дозволяє користувачу прокручувати таби горизонтальним свайпом. Активний таб залишається видимим завдяки JavaScript-логіці яка автоматично прокручує контейнер до нього при перемиканні.

Темна і світла теми реалізовані через CSS-змінні і клас на кореневому елементі. При перемиканні теми всі компоненти інтерфейсу автоматично оновлюють кольори фону, тексту, бордерів. Збереження вибраної теми між сесіями реалізовано через `localStorage`. За замовчуванням використовується тема системи (`prefers-color-scheme` медіа-запит), а вибір користувача має пріоритет.

Приклад відображення головного екрану Dashboard у мобільній версії на ширині 390 пікселів (iPhone 14) наведено на рисунку 4.5.

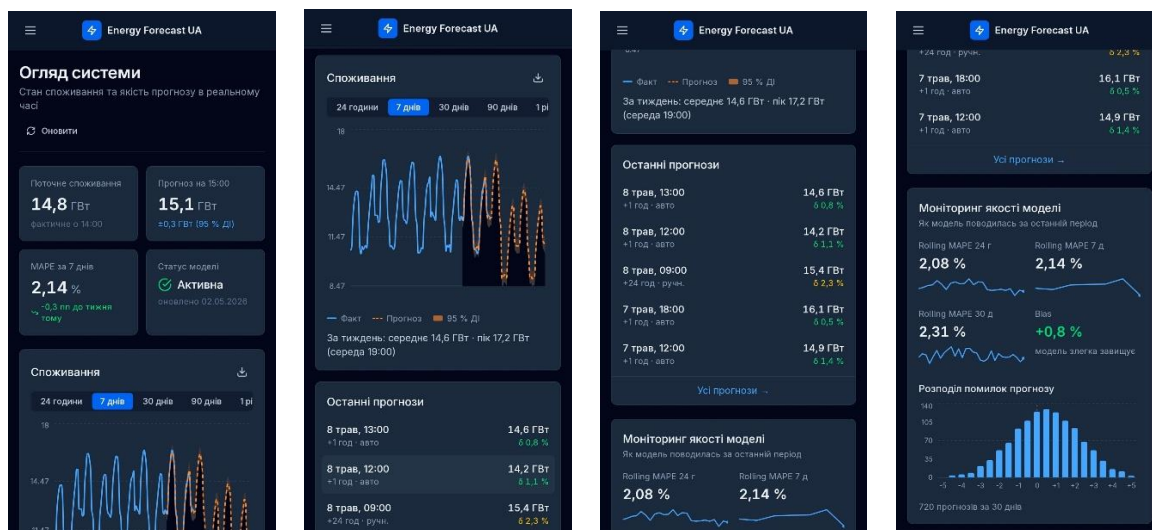


Рисунок 4.5 – Адаптивна мобільна версія: екран Dashboard

Висувне навігаційне меню на мобільному пристрої з усіма пунктами навігації, згрупованими за функціональним призначенням, показано на рисунку 4.6.

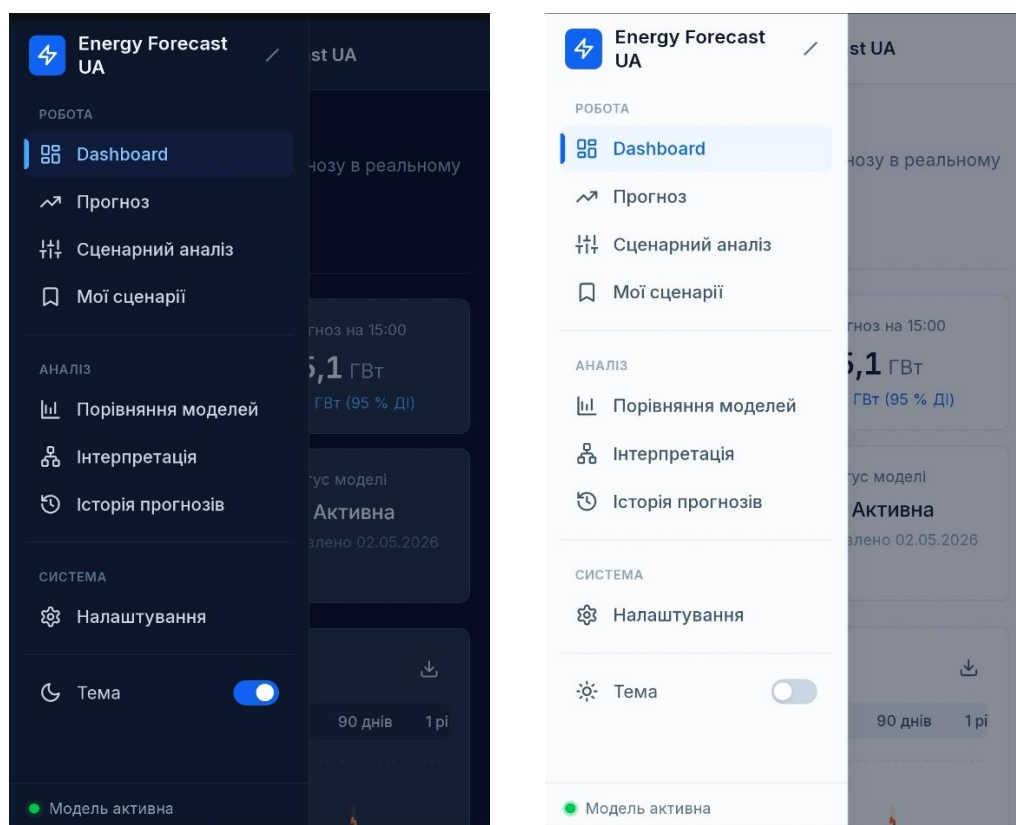


Рисунок 4.6 – Адаптивна мобільна версія: висувне меню навігації

Тестування адаптивності проводилося через інструменти розробника браузерів Chrome і Firefox з симуляцією різних розмірів пристроїв: iPhone SE (375×667), iPhone 14 (390×844), iPad (768×1024), iPad Pro (1024×1366), Desktop (1920×1080). На всіх протестованих розмірах інтерфейс зберігає функціональність і читабельність без необхідності горизонтальної прокрутки сторінки.

ВИСНОВКИ

У результаті виконання кваліфікаційної роботи розроблено веб-орієнтовану інформаційну систему погодинного прогнозування електроспоживання в Об'єднаній енергосистемі України на основі адаптивної нейро-нечіткої системи виведення (ANFIS). Поставлені цілі та задачі досягнуто у повному обсязі.

У першому розділі проведено аналіз предметної галузі прогнозування електроспоживання в ОЕС України, виявлено основні фактори впливу на навантаження (метеорологічні, календарні, сезонні), розглянуто проблеми короткострокового прогнозування та виконано порівняльний огляд існуючих методів. На основі проведеного аналізу обґрунтовано вибір ANFIS як основного методу: цей підхід поєднує точність нейронних мереж з інтерпретованістю нечіткої логіки, що є критичним для застосування у системах управління критичною інфраструктурою.

У другому розділі досліджено теоретичні засади нечіткої логіки, функцій належності, лінгвістичних змінних та нечіткого виведення Сугено. Детально розглянуто архітектуру ANFIS у вигляді п'ятишарової адаптивної нейронної мережі та алгоритм гібридного навчання Янга, який поєднує метод найменших квадратів для параметрів виходів правил з градієнтним спуском для параметрів функцій належності. Обґрунтовано вибір трикутних функцій належності та метрик оцінки якості (MAPE, RMSE, MAE).

У третьому розділі спроектовано та реалізовано трирівневу клієнтсерверну архітектуру системи. Клієнтська частина побудована на React 18 з використанням Tailwind CSS 3 та Recharts, серверна частина – на FastAPI з ANFIS-моделлю на scikit-fuzzy. Розроблено вісім функціональних модулів системи: оглядова панель, формування прогнозу, сценарний аналіз, бібліотека сценаріїв, порівняння моделей, інтерпретація правил, історія прогнозів та системні налаштування. Реалізовано інтеграцію з зовнішніми джерелами даних ENTSO-E Transparency Platform та Open-Meteo. Інтерфейс

адаптовано для мобільних пристроїв з мінімальною підтримуваною шириною 360 пікселів.

У четвертому розділі проведено експериментальні дослідження якості розробленої моделі. Модель ANFIS показала на тестовій вибірці 2021 року такі результати: MAPE 6,17%, RMSE 1344 MBт, MAE 1040 MBт. Це відповідає галузевим стандартам ENTSO-E для короткострокового прогнозування. У порівнянні з референсними моделями ANFIS показала найкращі результати: перевага над LSTM становить 50% за MAPE (6,17% проти 12,4%) при у 9 разів коротшому часі навчання (34 с проти 312 с), перевага над класичними статистичними методами (SARIMAX, Prophet) – 60–65%. Особливо важливою є перевага ANFIS у пікові години (ранковий та вечірній піки споживання), де похибка моделей зазвичай максимальна.

Розроблена система розгорнута на безкоштовних хмарних платформах: клієнтська частина на Vercel за адресою [energy forecast ua.vercel.app](https://energyforecast.ua.vercel.app), серверна частина на Render.com. Вихідний код опубліковано у відкритому репозиторії GitHub, що забезпечує відтворюваність результатів та можливість використання роботи іншими дослідниками і практиками.

Розроблена система може бути використана операторами ОЕС України для попереднього оцінювання добового балансу потужностей, учасниками ринку електроенергії при формуванні заявок на ринку «на добу наперед», дослідниками для порівняння методів прогнозування на єдиному наборі даних, а також студентами та викладачами як навчальний інструмент демонстрації принципів роботи нейро-нечітких систем.

Наукова новизна роботи полягає в адаптації архітектури ANFIS до специфіки ОЕС України з урахуванням національних свят і сезонних особливостей споживання, у розробці механізму інтерактивної декомпозиції прогнозу на активовані нечіткі правила, а також у створенні

готового до використання веб-сервісу з повною інтерпретованістю результатів.

Перспективи подальшого розвитку роботи включають інтеграцію з реальними даними споживання Укренерго після стабілізації роботи енергосистеми, перенавчання моделі на даних воєнного періоду після завершення активних бойових дій, додавання прогнозування для окремих регіонів України, реалізацію ансамблевого підходу з поєднанням ANFIS та LSTM, а також розробку нативного мобільного додатка для оперативного доступу до системи.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. НЕК «Укренерго». Дані оперативної інформації про роботу ОЕС України. URL: <https://ua.energy/elektroenergetyka/dyspetcherska-informatsiya/> (дата звернення: 15.05.2026).
2. ENTSO-E Transparency Platform. Power statistics and market data for European TSOs. URL: <https://transparency.entsoe.eu/> (date of access: 15.05.2026).
3. Jang J.-S. R. ANFIS: adaptive-network-based fuzzy inference system. *IEEE Transactions on Systems, Man, and Cybernetics*. 1993. Vol. 23, No. 3. P. 665–685.
4. Zadeh L. A. Fuzzy sets. *Information and Control*. 1965. Vol. 8, No. 3. P. 338–353.
5. Mamdani E. H., Assilian S. An experiment in linguistic synthesis with a fuzzy logic controller. *International Journal of Man-Machine Studies*. 1975. Vol. 7, No. 1. P. 1–13.
6. Takagi T., Sugeno M. Fuzzy identification of systems and its applications to modeling and control. *IEEE Transactions on Systems, Man, and Cybernetics*. 1985. Vol. SMC-15, No. 1. P. 116–132.
7. Cheng Y.-Y., Chan P. P. K., Qiu Z.-W. Random forest based ensemble system for short term load forecasting. *Proceedings of the International Conference on Machine Learning and Cybernetics*. 2012. P. 52–56.
8. Hippert H. S., Pedreira C. E., Souza R. C. Neural networks for short-term load forecasting: a review and evaluation. *IEEE Transactions on Power Systems*. 2001. Vol. 16, No. 1. P. 44–55.
9. Khotanzad A., Afkhami-Rohani R., Maratukulam D. ANNSTLF – artificial neural network short-term load forecaster – generation three. *IEEE Transactions on Power Systems*. 1998. Vol. 13, No. 4. P. 1413–1422.
10. Hochreiter S., Schmidhuber J. Long short-term memory. *Neural Computation*. 1997. Vol. 9, No. 8. P. 1735–1780.

11. Box G. E. P., Jenkins G. M., Reinsel G. C., Ljung G. M. Time series analysis: forecasting and control. 5th ed. Hoboken : John Wiley & Sons, 2015. 712 p.
12. Taylor S. J., Letham B. Forecasting at scale. *The American Statistician*. 2018. Vol. 72, No. 1. P. 37–45.
13. Hyndman R. J., Athanasopoulos G. Forecasting: principles and practice. 3rd ed. Melbourne : OTexts, 2021. 506 p.
14. 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. P. 785–794.
15. Attention is all you need / A. Vaswani et al. *Advances in neural information processing systems*. 2017. Vol. 30. P. 5998–6008.
16. Кафедра штучного інтелекту ХНУРЕ. Методичні вказівки до виконання кваліфікаційної роботи освітнього рівня «бакалавр». Харків : ХНУРЕ, 2025. 32 с.
17. Документація FastAPI: сучасний веб-фреймворк для Python. URL: <https://fastapi.tiangolo.com/> (дата звернення: 18.05.2026).
18. React Documentation: A JavaScript library for building user interfaces. URL: <https://react.dev/> (date of access: 18.05.2026).
19. Scikit-fuzzy: fuzzy logic toolkit for Python. URL: <https://scikit-fuzzy.github.io/> (date of access: 18.05.2026).
20. Tailwind CSS documentation: a utility-first CSS framework. URL: <https://tailwindcss.com/> (date of access: 18.05.2026).
21. Open-meteo: free weather API. URL: <https://open-meteo.com/> (date of access: 18.05.2026).
22. ENTSO-E API implementation guide. URL: <https://transparencyplatform.zendesk.com/hc/en-us/articles/15692855254548> (date of access: 18.05.2026).
23. Vercel documentation: deploy frontend applications. URL: <https://vercel.com/docs> (date of access: 18.05.2026).

24. Render documentation: cloud application platform. URL: <https://render.com/docs> (date of access: 18.05.2026).
25. McKinney W. Python for data analysis: data wrangling with pandas, NumPy, and Jupyter. 3rd ed. Sebastopol : O'Reilly Media, 2022. 578 p.