

Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет комп'ютерної інженерії та управління
(повна назва)

Кафедра електронних обчислювальних машин
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА
Пояснювальна записка

Рівень вищої освіти другий (магістерський)

Модель штучної неймережі для підтримки
мікроклімату у великих приміщеннях

(тема)

Виконав:

здобувач 2 року навчання,

групи СПМ-23-5

Владислав ПЛУГІН

(власне ім'я, прізвище)

Спеціальність 123 «Комп'ютерна інженерія»

(код і повна назва спеціальності)

Тип програми освітньо-наукова

(освітньо-професійна або освітньо-наукова)

Освітня програма Системне програмування

(повна назва освітньої програми)

Керівник: доц. Олексій ПІСКАРЬОВ

(посада, прізвище, ініціали)

Допускається до захисту

Зав. кафедри ЕОМ

(підпис)

Андрій КОВАЛЕНКО

(прізвище, ініціали)

2025 р.

Харківський національний університет радіоелектроніки

Факультет _____ комп'ютерної інженерії та управління _____

Кафедра _____ електронних обчислювальних машин _____

Рівень вищої освіти _____ другий (магістерський) _____

Спеціальність _____ 123 «Комп'ютерна інженерія» _____
(код і повна назва)

Тип програми _____ освітньо-наукова _____
(освітньо-професійна або освітньо-наукова)

Освітня програма _____ Системне програмування _____
(повна назва)

ЗАТВЕРДЖУЮ:

Зав. кафедри _____
(підпис)

“ _____ ” _____ 20__ р.

ЗАВДАННЯ

НА КВАЛІФІКАЦІЙНУ РОБОТУ

здобувачеві _____ Плугіну Владиславу В`ячеславовичу _____
(прізвище, ім'я, по батькові)

1. Тема роботи Модель штучної нейромережі для підтримки мікроклімату
у великих приміщеннях

затверджена наказом по університету від “ 21 ” квітня 2025 р. № 296 Ст

2. Термін подання студентом роботи до екзаменаційної комісії 16 червня 2025 р.

3. Вхідні дані до роботи методи підтримки мікроклімату, критерії якості методів
підтримки, сучасні нейромережі, методи машинного навчання,
параметрів мікроклімату: температура, вологість, концентрація CO₂, повітрообмін
дослідження моделі.

4. Перелік питань, що потрібно опрацювати у роботі _____

1) огляд існуючих методів та моделей підтримки мікроклімату

2) аналіз нейромереж

3) методи машинного навчання

4) дослідження моделі

5) висновки

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій

Слайд презентація – 20 слайдів

6. Консультанти розділів роботи (заповнюється за наявності консультантів згідно з наказом, зазначеним у п.1)

Найменування розділу	Консультант (посада, прізвище, ім'я, по батькові)	Позначка консультанта про виконання розділу	
		підпис	дата

КАЛЕНДАРНИЙ ПЛАН

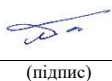
№	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1	Огляд моделей підтримки мікроклімату	22.04.25-1.05.25	
2	Аналіз нейромереж	02.05.25-10.05.25	
3	Методи машинного навчання	11.05.25-15.05.25	
4	Дослідження моделі	16.05.25-19.05.24	
5	Оформлення матеріалів кваліфікаційної роботи	20.05.25-31.05.25	
6	Подання кваліфікаційної роботи керівникові та її попередній захист	01.06.25-07.06.25	
7	Подання кваліфікаційної роботи на рецензування	08.06.25-12.06.25	

Дата видачі завдання “ 22 ” квітня 2025 р.

Здобувач


(підпис)

Керівник роботи


(підпис)

доц. Олексій ПИСКАРЬОВ

(посада, власне ім'я, прізвище)

РЕФЕРАТ

Пояснювальна записка кваліфікаційної роботи: 72 с., 13 рис., 1 табл., 3 дод., 16 джерел.

МІКРОКЛІМАТ, ШТУЧНА НЕЙРОМЕРЕЖА, ПРОГНОЗУВАННЯ ПАРАМЕТРІВ, ВЕЛИКІ ПРИМІЩЕННЯ, АДАПТИВНЕ РЕГУЛЮВАННЯ, МАШИННЕ НАВЧАННЯ, ІНТЕЛЕКТУАЛЬНА СИСТЕМА, СЕНСОРИ, ОПТИМІЗАЦІЯ, РЕГУЛЯЦІЯ, КОНТРОЛЬ.

Метою цієї роботи є розробка моделі штучної нейромережі для інтелектуальної підтримки мікроклімату у великих приміщеннях шляхом прогнозування й адаптивного регулювання температури, вологості та концентрації CO₂ на основі сенсорних даних і змінних умов експлуатації.

У межах кваліфікаційної роботи проведено аналіз методів і моделей керування мікрокліматом, розглянуто підходи «білого», «чорного» й «сірого» ящика, а також сучасні нейромережеві технології, активаційні функції та методи машинного навчання. Окремо розглянуто ансамблеві моделі, зокрема XGBoost, як ефективний інструмент для точного прогнозування.

Розроблена нейромережева модель забезпечує інтеграцію сенсорної інформації, масштабованість і підтримку адаптивного регулювання з можливістю безперервного навчання. Вона успішно відтворює динаміку мікрокліматичних параметрів, що підтверджується високою точністю прогнозування.

Запропонована система є перспективною для впровадження в автоматизовані комплекси управління мікрокліматом, сприяє енергоефективності та може масштабуватись і адаптуватись до різних будівель і джерел даних.

ABSTRACT

Master's thesis: 72 pages, 13 figures, 1 tabl, 3 appendices, 16 sources.

MICROCLIMATE, ARTIFICIAL NEURAL NETWORK, PARAMETER PREDICTION, LARGE ROOMS, ADAPTIVE CONTROL, MACHINE LEARNING, INTELLIGENT SYSTEM, SENSORS, OPTIMIZATION, REGULATION, CONTROL.

The purpose of this work is to develop an artificial neural network model for intelligent microclimate support in large rooms by predicting and adaptively controlling temperature, humidity, and CO₂ concentration based on sensor data and variable operating conditions.

The qualification work analyzes methods and models of microclimate control, considers white, black, and gray box approaches, as well as modern neural network technologies, activation functions, and machine learning methods. Ensemble models, in particular XGBoost, are considered separately as an effective tool for accurate forecasting.

The developed neural network model provides integration of sensor information, scalability, and support for adaptive control with the possibility of continuous learning. It successfully reproduces the dynamics of microclimatic parameters, which is confirmed by the high accuracy of forecasting.

The proposed system is promising for implementation in automated microclimate control systems, promotes energy efficiency, and can be scaled and adapted to different buildings and data sources.

ЗМІСТ

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ	7
ВСТУП	8
1 АНАЛІЗ МЕТОДІВ ТА МОДЕЛЕЙ ПІДТРИМКИ МІКРОКІМАТУ	10
1.1 Моделі управління мікрокліматом	10
1.1.1 Моделі «білого ящика»	10
1.1.2 Моделі «чорного ящика»	11
1.1.3 Моделі «сірого ящика»	11
2 АНАЛІЗ НЕЙРОМЕРЕЖ	15
2.1 Історія нейромереж	15
2.2 Штучні нейронні мережі	16
2.3 Функції активації	19
3 МЕТОДИ МАШИННОГО НАВЧАННЯ	21
3.1 Ансамблеві методи	22
3.2 Модель XGBoost - extreme gradient boosting	25
4 ДОСЛІДЖЕННЯ МОДЕЛІ ШТУЧНОЇ НЕЙРОМЕРЕЖІ	37
4.2 Порівняння прогнозованих та фактичних значень	43
ВИСНОВКИ	52
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ	54
ДОДАТОК А Графічний матеріал кваліфікаційної роботи	56
ДОДАТОК Б Наукові публікації за темою кваліфікаційної роботи	67
ДОДАТОК В Фрагмент коду на PYTHON 3.8, використання моделі XGBOOST ДЛЯ прогнозування концентрації CO ₂	72

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ

- АНС – адаптивна нейромережева система
- ВМІ – внутрішнє мікрокліматичне інформаційне середовище
- ГСІ – гібридне сховище інформації
- ЗМІ – зовнішнє мікрокліматичне інформаційне середовище
- СКМ – система керування мікрокліматом
- ANN – штучна нейронна мережа (англ., Artificial Neural Network)
- CO₂ – концентрація вуглекислого газу
- CSV – Comma-Separated Values
- DBMS – Database Management System
- DNN – Deep Neural Network
- HVAC – Heating, Ventilation and Air Conditioning
- MAE – середня абсолютна похибка (англ., Mean Absolute Error)
- ML – машинне навчання (англ., Machine Learning)
- MSE – середньоквадратична похибка (англ., Mean Squared Error)
- PMV – передбачуваний середній індекс тепловідчуття (англ., Predicted Mean Vote)
- RHM – відносна вологість повітря, (англ. Relative Humidity Measurement)
- RMSE – корінь середньоквадратичної похибки (англ., Root Mean Squared Error)
- TMPin – внутрішня температура повітря (англ., Temperature Inside)
- TMPout – зовнішня температура повітря (англ., Temperature Outside)

ВСТУП

В умовах стрімкого зростання масштабів урбанізації, підвищених вимог до енергоефективності будівель та погіршення якості навколишнього середовища, особливої актуальності набуває проблема створення адаптивних систем підтримки мікроклімату у великих приміщеннях. Нестабільність зовнішніх погодних умов, різноманітність архітектурних рішень та висока щільність перебування людей зумовлюють складність забезпечення стабільних та комфортних параметрів мікроклімату, таких як температура, вологість, концентрація CO₂ та повітрообмін. Традиційні системи автоматизації часто ґрунтуються на фіксованих алгоритмах регулювання, що не враховують змінні умови експлуатації й не здатні адаптуватися до динаміки внутрішнього середовища, що значно знижує ефективність їх функціонування.

Актуальність теми обумовлена необхідністю впровадження інтелектуальних методів керування, здатних забезпечувати оптимальні мікрокліматичні умови в режимі реального часу. Основними аспектами, що визначають значущість даного дослідження, є зростаюча роль здорового середовища у підвищенні продуктивності праці, енергозбереження як ключовий фактор у проектуванні інженерних систем будівель, а також потреба у мінімізації втручання людини в процес контролю мікроклімату. Запровадження штучних нейромереж у систему керування відкриває нові перспективи для реалізації самонавчальних моделей, здатних передбачати поведінку середовища та приймати оптимальні рішення щодо параметрів регулювання на основі історичних і поточних даних.

Одним із перспективних напрямів вирішення поставленої проблеми є побудова моделі штучної нейромережі, що інтегрується в комп'ютерну систему керування та виконує прогнозування ключових параметрів мікроклімату з наступною адаптацією дій виконавчих механізмів. Поєднання

такого підходу з гібридним зберіганням даних дозволяє ефективно обробляти великі обсяги сенсорної інформації, зберігаючи оперативний доступ до найбільш релевантних записів.

Крім того, доцільність роботи пов'язана з тим, що вона передбачає розробку та оптимізацію програмних компонентів низького рівня для взаємодії з апаратною частиною системи керування, моделювання обчислювальних процесів у реальному часі, впровадження алгоритмів машинного навчання та організацію ефективного зберігання й обробки даних

Таким чином, розвиток автоматизованих інтелектуальних систем керування мікрокліматом має стати основою для формування комфортного, безпечного та енергозберігаючого середовища у приміщеннях різного призначення.

1 АНАЛІЗ МЕТОДІВ ТА МОДЕЛЕЙ ПІДТРИМКИ МІКРОКЛІМАТУ

1.1 Моделі управління мікрокліматом

Мікроклімат залежить від параметрів оточуючого середовища, таких як температура повітря, середня температура, швидкість повітря й вологість, а також від людських факторів, зокрема рівня фізичної активності та термічного опору одягу [1]. Нечітка модель для оцінки умов повітря в приміщеннях, має дозволяти визначення оптимальних значень температури та вологості повітря, й враховувати індивідуальні особливості людини [2]. Усі моделі управління мікрокліматом в приміщенні можна умовно поділити на три основні класи (рисунок 1.1) [2].

Моделі управління мікрокліматом		
<p>"Білий ящик"</p> <ul style="list-style-type: none"> • Класичні диференційні рівняння • Модель Фангера • Модель PD • Модель PS • Модель TS • Адаптивні моделі 	<p>"Чорний ящик"</p> <ul style="list-style-type: none"> • ПД-регулятор • Нейроні мережі • Моделі нечіткої логіки • Мультиагентні моделі • Генетичні алгоритми 	<p>"Сірий ящик"</p>

Рисунок 1.1 – Класифікація моделей управління мікрокліматом

1.1.1 Моделі «білого ящика»

Модель «білого ящика» передбачає повне знання всіх вхідних та вихідних даних, а також усіх зовнішніх й внутрішніх параметрів. Вона складається з кількох складних диференціальних рівнянь, де

використовуються численні коефіцієнти, що враховують геометрію будівлі, теплові властивості огорожувальних конструкцій тощо. Усі параметри та коефіцієнти таких моделей відомі й можуть бути обчислені або виміряні. Однак ці моделі є громіздкими, вимагають значних обчислювальних ресурсів та часу. Інший тип моделей зосереджений на чутливості людини до теплових та вологісних умов, базуючись на показниках комфортності, таких як PMV і PDD. Третій тип моделей ґрунтується на електротепловій аналогії, що виникає через аналогію між рівнянням електропровідності та рівнянням теплопровідності [3].

1.1.2 Моделі «чорного ящика»

Моделі «чорного ящика» не використовують явне застосування фізичних принципів при їх побудові. Вони базуються лише на вхідних і вихідних параметрах, без детального опису внутрішніх механізмів. Така модель є апроксимацією, що відображає спостережувані зв'язки між параметрами. До таких моделей належать різноманітні підходи, зокрема моделі на основі нейронних мереж та нечіткої логіки [3].

1.1.3 Моделі «сірого ящика»

Моделі «сірого ящика» частково ґрунтуються на фізичних принципах. Вони не містять повного набору відомих параметрів, які можна обчислити або виміряти. Така модель є гібридною, поєднуючи риси моделей першого та другого типу.

В процесі аналізу встановлено, що найбільш поширеними є моделі, засновані на показниках комфортності (PMV/PDD), а також моделі, розроблені за допомогою нечіткої логіки (рисунок 1.2).

Для визначення комфортних умов моделі використовують різні набори даних. Вибір конкретної моделі розрахунку та набору даних може залежати

від конкретних завдань і умов їх застосування.

1.2 Аналіз переваг й недоліків існуючих моделей

Проведемо короткий аналіз переваг та недоліків існуючих моделей. Класичні диференціальні рівняння теплового балансу [2, 3] мають наочну розрахункову схему, що забезпечує високу надійність й точність результатів, а також можливість коригування формул та врахування додаткових чинників, таких як вологість та інші. Однак, такі моделі мають складність й громіздкість обчислень, а також потребують ідентифікації великої кількості параметрів.

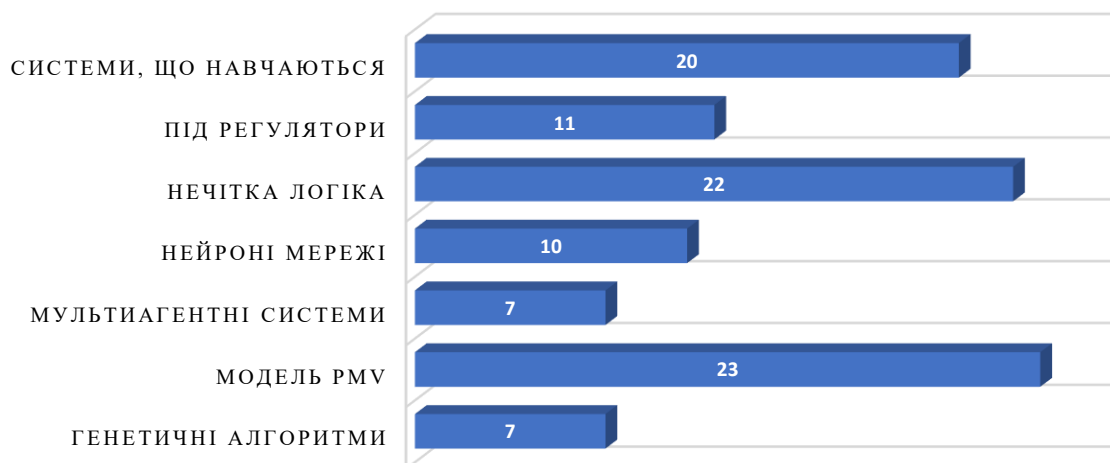


Рисунок 1.2 – Порівняння застосування методів управління мікрокліматом

Модель РМВ [3] використовує велику кількість параметрів, враховуючи метаболічний фактор та умови наявності одягу. Однак вона не може бути застосована при коливаннях декількох змінних одночасно, оскільки основні параметри повинні залишатися в заданих інтервалах.

Моделі ET-DISC та SET [3] імітують систему терморегуляції людського організму, однак не враховують адаптацію організму до змінних умов. Вони використовують фіксовані фактори, що обмежує їх гнучкість у реальних умовах.

Моделі TS, PD, PS [3] дозволяють визначити комфортні умови для факторів, що впливають на якість життя в приміщенні. Однак вони є моделями локального комфорту, мають невисоку точність і можуть бути складними для інтерпретації результатів.

Адаптивна модель [3] враховує зміни зовнішнього клімату і має хорошу стійкість. Однак її реалізація є складною та трудомісткою.

RC-модель [3] характеризується простотою, оскільки система лінійних диференціальних рівнянь дозволяє враховувати взаємозв'язок процесів теплопередачі і вологості. Однак її точність обчислень не є високою, і виникають труднощі при обліку параметрів зовнішнього середовища, таких як сонячне випромінювання, вітер та інші фактори.

PID управління [3] має високу швидкодію й забезпечує точне утримання заданої температури, однак виникають труднощі при підборі коефіцієнтів моделі, що може призвести до низької точності. Крім того, налаштування системи при такому управлінні є трудомістким, а при великих відхиленнях від номінальних значень параметрів можуть виникати суттєві помилки.

Управління, засноване на нечіткій логіці [2, 3], відзначається високою точністю, стійкістю та швидкістю, що дозволяє ефективно керувати нелінійними системами з динамічно змінюваними параметрами. Однак цей підхід має свої недоліки, зокрема складність у налаштуванні та розробці бази нечітких правил, а також значне ускладнення обчислень, яке спостерігається при збільшенні кількості вхідних змінних.

Управління на основі штучних нейронних мереж [3, 4] дозволяє обробляти велику кількість вхідних даних й забезпечує надійне прогнозування, при цьому не вимагаючи попередньої інформації про будівлю або кліматичні умови. Однак для досягнення якісного прогнозу необхідно мати велику кількість експериментальних даних, а також враховувати тривалість процесу навчання, що може бути значним обмеженням.

Нейро-нечіткі системи управління [5] дозволяють ефективно управляти

складними нелінійними динамічними об'єктами та розробляти нелінійні закони управління для таких систем. Однак цей підхід має обмеження, зокрема можна використовувати лише обмежену кількість вхідних змінних, а також процес навчання є досить тривалим.

Мультиагентні системи управління [5] відзначаються гнучкістю, оскільки агенти можуть бути доповнені та модифіковані відповідно до змін умов. Вони також мають здатність до самовідновлення та високу стійкість до збоїв. Однак велика кількість розрізнених агентів може ускладнити управління системою та зробити її менш керованою.

Генетичні алгоритми [4, 5] є ефективними для вирішення задач оптимізації великої розмірності, особливо коли вихідні дані не мають чіткої впорядкованості. Однак їх застосування супроводжується труднощами в пошуку точного глобального оптимуму, а також потребує тривалого часу для виконання алгоритму та складності при розробці програмного коду.

Система управління мікрокліматом може прогнозувати його стан за допомогою різних моделей, здатних виявляти зміни параметрів як зовнішнього, так і внутрішнього середовища. Однак деякі з цих моделей обмежуються врахуванням лише кількох параметрів, що впливають на комфорт, і тому мають вузьку область застосування. Найбільш поширеною завдяки своїй гнучкості та здатності враховувати широкий спектр факторів є модель PMV, яку буде використовуватися в подальшому.

Отже, для ефективного управління мікрокліматом у приміщеннях важливо використовувати моделі, які здатні враховувати широкий спектр параметрів, що впливають на комфорт.

2 АНАЛІЗ НЕЙРОМЕРЕЖ

Останнім часом штучні нейронні мережі набули широкої популярності, й про них досить часто згадується в різних контекстах, що може створити враження, ніби йдеться про зовсім нову технологію. Однак насправді концепція нейронних мереж виникла значно раніше, й сучасна хвиля інтересу є результатом тривалого періоду розвитку та вдосконалення цих моделей.

2.1 Історія нейромереж

У 1943 році нейрофізіолог Воррен Маккалок та математик Волтер Пітс опублікували дослідження, в якому описали, як можуть функціонувати нейрони мозку. Це стало однією з перших робіт, що привернули увагу до ідеї нейронних мереж і штучного інтелекту. Вони також створили базову електричну схему для демонстрації своїх припущень. У 1949 році Дональд Хебб розвинув ці ідеї, запропонувавши, що нейронні зв'язки зміцнюються з кожною активацією, сформулювавши принцип, відомий як Хеббівське навчання. У 1950-х розвиток комп'ютерної техніки дозволив перевірити ці теорії на практиці. Першу спробу реалізації штучної нейронної мережі здійснив дослідник ІВМ Натаніель Рочестер, хоча вона й не увінчалася успіхом. Проте вже у 1954 році в Массачусетському технологічному інституті була створена перша мережа, заснована на принципах Хебба. У 1960–1980-х роках інтерес до нейромереж зменшився, але починаючи з середини 1980-х відбулося його відновлення завдяки зростанню обчислювальних потужностей і доступності великих обсягів даних. Подальші наукові розробки та технологічні досягнення сприяли утвердженню нейронних мереж як одного з ключових напрямів у сфері штучного інтелекту [4, 5].

2.2 Штучні нейронні мережі

Штучні нейронні мережі є обчислювальними системами, що імітують біологічну структуру мозку і мають здатність до навчання. Вони складаються з великої кількості окремих елементів – нейронів, що виконують обчислення на основі вхідних даних, а результат їх обробки може передаватися іншим нейронам. Однією з основних особливостей таких мереж є здатність до самонавчання, що дозволяє системі знаходити закономірності і зв'язки між вхідними та вихідними даними без необхідності чітко визначеного алгоритму. Це дає можливість мережам виконувати складні завдання, такі як класифікація, прогнозування чи оптимізація, самостійно вивчаючи патерни в даних.

Нейронна мережа працює завдяки паралельній обробці даних, що дозволяє ефективно обробляти великі обсяги інформації, а також адаптуватися до змінних умов. Вхідні сигнали можуть надходити до нейронів різними способами, залежно від архітектури мережі: сигнали можуть надходити лише до вхідних нейронів, на всі нейрони одночасно або передаватися між нейронами різних рівнів (що забезпечує зворотний зв'язок).

Якщо порівнювати штучну нейронну мережу з біологічним мозком, то кожен нейрон у мережі має структуру, аналогічну біологічному нейрону, з такими компонентами, як дендрити (які отримують вхідні сигнали) і аксон (який передає вихідні сигнали). Важливу роль у нейронних мережах відіграють синапси – з'єднання між нейронами, через які передаються сигнали. Кожен синапс має свою "вагу", яка визначає силу впливу одного нейрона на інший. Змінюючи ваги синапсів, нейронна мережа може змінювати своє поведіння і адаптуватися до нових умов або даних. Це робить її потужним інструментом для розв'язання задач, що вимагають гнучкості і здатності до навчання [6].

Основні елементи структури нейронної мережі включають кілька

ключових компонентів, що взаємодіють один з одним для здійснення обчислень і навчання. Першим елементом є штучні нейрони, які є спрощеною моделлю біологічного нейрона і виконують роль обробки інформації, отриманої від інших нейронів або ззовні.

Наступним елементом є синапс, що виступає як з'єднання між виходом одного нейрона та входом іншого. В синапсі відбувається процес посилення або ослаблення сигналу, що передається між нейронами, з урахуванням ваги цього зв'язку.

Сигнал це значення, що передається через синапс й яке може бути змінене на різних етапах обробки, в залежності від ваги і функції активації.

Суматор це функціональний блок нейрона, який здійснює складання всіх вхідних сигналів, помножених на відповідні їм ваги. Це дозволяє нейрону зважувати вплив кожного входу на кінцевий результат.

Останнім елементом є функція активації, яка визначає, чи буде нейрон активним і, відповідно, передавати сигнал далі. Функція активації вводить нелінійність у модель, що дозволяє нейронним мережам вирішувати складні задачі, що не піддаються лінійним методам обробки. Вибір функції активації має значний вплив на ефективність і здатність нейронної мережі до навчання.

Синапси є важливим компонентом нейронної мережі, оскільки вони здійснюють зв'язок між нейронами. Вони передають сигнал від одного нейрона до іншого, множачи вхідний сигнал на число, яке характеризує силу зв'язку — вага синапсу. Ці ваги можуть бути як позитивними, так і негативними. Позитивні ваги вважаються збудливими, оскільки вони підвищують ймовірність активації наступного нейрона, тоді як негативні ваги є гальмуючими, і зменшують цю ймовірність.

Суматор відіграє важливу роль у процесі обробки сигналів. Він здійснює додавання всіх вхідних сигналів, що надходять через синаптичні зв'язки від інших нейронів та зовнішніх вхідних сигналів. Це дозволяє нейрону зважено оцінювати вплив кожного з цих сигналів.

Далі, перетворювач реалізує функцію активації нейрона. Це функція

одного аргументу, який є виходом суматора, і перетворює його в деяку вихідну величину нейрона. Функція активації може бути різною в залежності від конкретної реалізації нейронної мережі, і визначає, чи буде нейрон активний або неактивний. Таким чином, нейрон виконує скалярну функцію векторного аргументу [7].

Структуру штучного нейрона показано на рисунку 2.1.

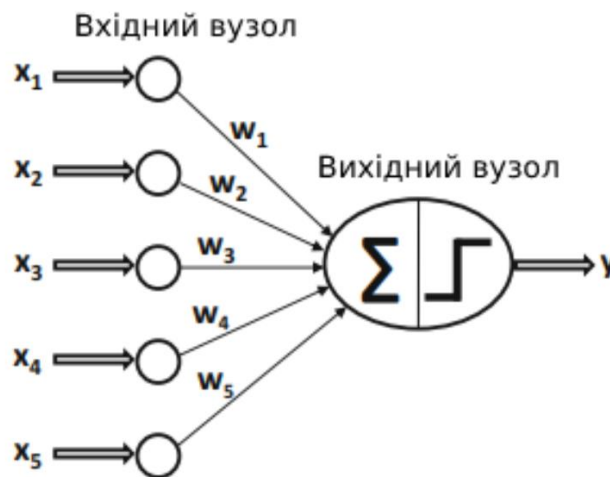


Рисунок 2.1 - Структура штучного нейрона

Вхідні сигнали x та вагові коефіцієнти w можуть бути дійсними числами, що дозволяє нейронній мережі обробляти широкий спектр інформації та виконувати складні задачі з високою точністю. Вихід нейрона, у свою чергу, залежить від виду функції активації і може бути як дійсним числом, так і цілим числом, залежно від того, яку задачу нейронна мережа повинна вирішити.

Нейронні мережі мають низку ключових переваг, які роблять їх ефективними для управління мікрокліматом. Вони демонструють високу адаптивність, здатні підлаштовуватись до змін середовища та, за потреби, перенавчатися в реальному часі. Це дозволяє підтримувати стабільну роботу навіть за динамічних умов, хоча надмірна чутливість може спричинити нестійкість.

Мережі стійкі до шуму, самостійно визначають значущі сигнали без

потреби в попередній фільтрації. Вони ефективно працюють при відсутності чітких моделей, виявляючи приховані закономірності у складних або слабо формалізованих системах. Завдяки децентралізованому зберіганню інформації, такі моделі мають високу відмовостійкість й зберігають працездатність навіть при часткових пошкодженнях.

Їх нелінійна структура дозволяє точно моделювати складні взаємозв'язки між параметрами, що є критично важливим у задачах, де класичні методи не дають точних результатів. Для досягнення поставленої мети необхідно класифікувати вхідні сигнали на певні класи, тому важливо звернути увагу на нейронні мережі, які використовуються для кластеризації. Після процесу навчання така мережа здатна визначати, до якого конкретного класу належить вхідний сигнал. У випадку, коли вхідний сигнал не відповідає жодному з уже визначених класів, мережа може сигналізувати про це. Це може бути індикатором того, що вхідний сигнал містить нові, невідомі дані, які не були представлені в навчальній вибірці. Таким чином, нейронні мережі для кластеризації не лише допомагають організувати дані в класи, але й здатні виявляти аномальні або нові дані, які потребують подальшого аналізу [7].

2.3 Функції активації

Активаційна функція відіграє важливу роль у визначенні реакції нейрона на вхідні сигнали. Вона аналізує зважену суму сигналів і вирішує, чи варто активувати нейрон. Оскільки вихідні значення можуть бути в широкому діапазоні, функція активації забезпечує необхідну межу для прийняття рішення про подальшу обробку сигналу.

Існує кілька основних типів функцій активації (рисунок 2.2), серед яких: порогова функція, лінійна функція, сигмоїдальна функція, гіперболічний тангенс та ReLU (Rectified Linear Unit).

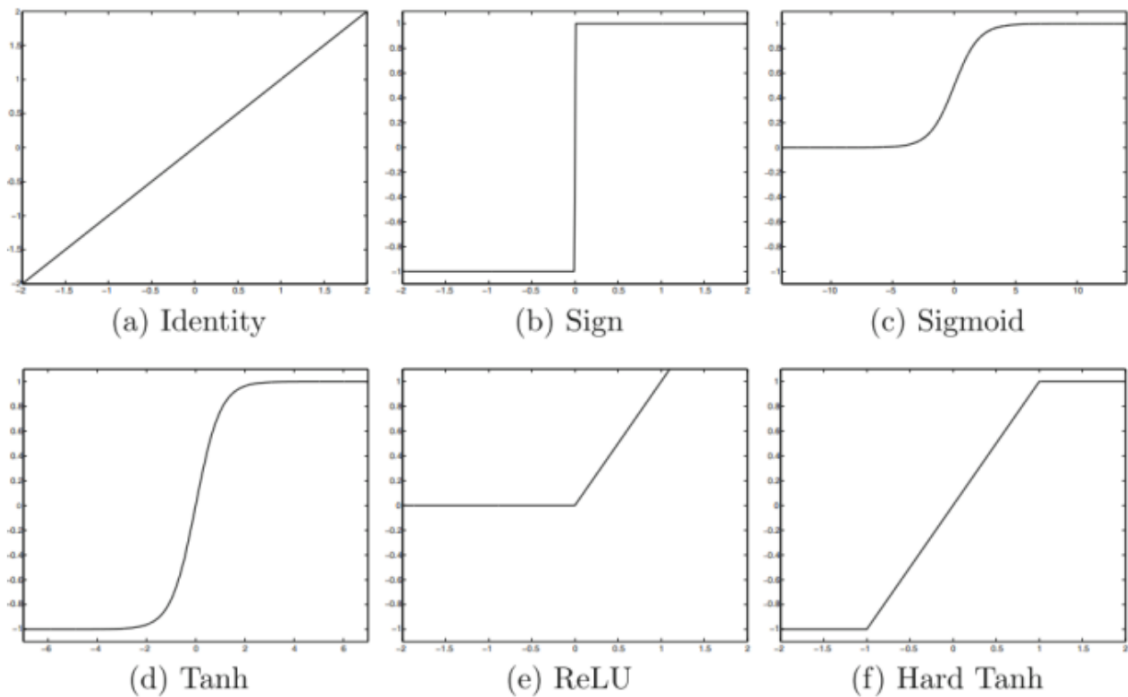


Рисунок 2.2 – Функції активації

Для задач класифікації доцільно використовувати сигмоїдальну функцію активації, оскільки вона забезпечує плавний перехід між 0 і 1, що зручно для ймовірнісних оцінок. Її форма сприяє чіткому розмежуванню класів і кращому навчанню.

Штучні нейронні мережі можуть реалізовуватись апаратно або програмно. Програмна реалізація простіша у використанні та гнучкіша в налаштуванні, хоча поступається за швидкістю. Нейронні мережі здатні до самонавчання, адаптації, працюють зі складними задачами навіть при відсутності чітких закономірностей [7]. Ще однією важливою функцією є апроксимація складних функціональних залежностей — тут нейронні мережі показують здатність до моделювання навіть дуже складних, нелінійних систем. В аналітичних задачах нейронні мережі допомагають витягати значущі закономірності зі складних структурованих і неструктурованих масивів даних, сприяючи оптимізації процесів у промисловості, логістиці, бізнесі й науці.

3 МЕТОДИ МАШИННОГО НАВЧАННЯ

Методи машинного навчання охоплюють широкий спектр підходів, що дозволяють комп'ютерним системам виявляти закономірності в даних та використовувати їх для прийняття рішень або прогнозування. Тобто необхідно побудувати модель, здатну узагальнювати знання з наданих прикладів й застосовувати їх до нових, раніше невідомих даних.

Залежно від структури навчальних даних та поставленої задачі, машинне навчання поділяється на кілька основних напрямів. Навчання з учителем передбачає наявність пар «вхід–вихід», де модель вивчає залежність між вхідними ознаками та цільовою змінною, що дозволяє вирішувати задачі класифікації та регресії. У випадку навчання без учителя модель аналізує лише вхідні дані, без чітко визначеного результату, з метою виявлення прихованих структур, кластерів або аномалій у даних. Напівкероване навчання поєднує переваги обох підходів й використовується, коли анотовані дані частково доступні, а також дозволяє ефективно використовувати велику кількість неанотованої інформації [8].

У сфері машинного навчання виділяють кілька основних груп моделей: класичні моделі з навчанням під наглядом або без нього, ансамблеві підходи, моделі з підкріпленням, а також нейронні мережі й глибокі архітектури. Класичні моделі з навчанням під наглядом зазвичай застосовуються до задач класифікації, де необхідно передбачити певну ознаку об'єкта, або до задач регресії, що передбачають визначення числового значення. У свою чергу, моделі без учителя орієнтовані на задачі кластеризації, які передбачають пошук подібних об'єктів і групування їх у кластери, а також на пошук закономірностей і зменшення розмірності даних.

Таким чином, моделі машинного навчання класифікуються за принципом організації навчання і характером завдань, які вони вирішують. Моделі з навчанням під наглядом орієнтовані на задачі прогнозування, тоді

як без нагляду – на виявлення внутрішньої структури даних. Такий поділ забезпечує гнучкість і широкі можливості для застосування методів машинного навчання в різних сферах аналізу та обробки інформації.

3.1 Ансамблеві методи

У ситуаціях, коли складне питання ставиться великій кількості випадкових людей, а їх відповіді агрегуються — за певних умов така сукупна думка може виявитися точнішою, ніж відповідь окремого експерта. Це явище називається колективним розумом. Подібний принцип використовується й у машинному навчанні: коли поєднуються прогнози групи моделей (класифікаторів або регресорів), у деяких випадках результат виявляється точнішим, ніж прогноз найкращої з окремих моделей. Такий підхід називається ансамблевим навчанням. Наприклад, можна створити групу моделей на основі дерев рішень, кожне з яких тренується на випадковому піднаборі даних, а остаточне рішення приймається більшістю голосів усіх дерев. Цей підхід називається випадковим лісом (random forest) й на сьогоднішній день вважається одним з найефективніших алгоритмів машинного навчання. Ансамблеві моделі особливо доцільні, коли вже побудовано кілька добре працюючих моделей, та виникає потреба поєднати їх для отримання ще точнішого прогнозу. Розглянемо найбільш поширені підходи до ансамблевого навчання, зокрема беггінг, бустінг, стекінг, випадкові ліси та інші варіанти.

Беггінг, що розшифровується як bootstrap aggregating, передбачає створення кількох моделей шляхом випадкового формування навчальних підмножин з використанням методу бутстрепа. Для кожної такої підмножини навчається окрема базова модель, після чого їхні прогнози об'єднуються для формування остаточного результату. Кожну з моделей прийнято називати слабким учнем або базовою моделлю. Метод бутстрепа базується на випадковому виборі з поверненням, тобто одна й та сама точка може

потрапити в підмножину кілька разів. Такий підхід ґрунтується на припущенні, що обрані вибірки є незалежними і належать до одного розподілу. Незалежність означає, що кожна вибірка формується без впливу попередніх, а однаковий розподіл передбачає, що всі вибірки відображають ту саму сукупність даних.

Прикладом моделі, що базується на принципах беггінгу, є метод випадкового лісу (Random Forest). Він створює ансамбль дерев рішень, кожне з яких навчається на випадковій підмножині даних та ознак, обраних з повного навчального набору. У випадку класифікації кінцевий прогноз визначається більшістю голосів, а для регресії — як середнє значення всіх прогнозів дерев. Такий підхід дозволяє зменшити дисперсію результатів, знизити ризик перенавчання, а також забезпечити стійкість до відсутніх значень у даних, оскільки кожна модель використовує лише частину записів і ознак. Завдяки можливості паралельного навчання прискорюється обчислювальний процес. Проте недоліком такого методу є високе споживання пам'яті й обчислювальних ресурсів, що пов'язано з необхідністю одночасної обробки великої кількості моделей.

Ансамблева модель, відома як стекінг (stacking), або стекове узагальнення, передбачає використання окремої моделі для об'єднання результатів кількох інших моделей. У процесі її реалізації спочатку кожна з моделей першого рівня формує власний прогноз. Далі ці прогнози слугують вхідними даними для іншої моделі — мета-учня або змішувача, — яка навчається на основі таких виходів і генерує фінальний прогноз. Такий підхід дозволяє враховувати особливості кожного прогнозатора та будувати більш узагальнене рішення, що може поліпшити точність у порівнянні з використанням окремих моделей.

Навчання моделі стекінгу передбачає розбиття початкового набору даних на два піднабори. Перший із них використовується для тренування моделей першого рівня, які формують прогнози для прикладів із другого, так званого утриманого піднабору. Важливо, що дані з утриманого піднабору не

використовуються під час навчання моделей першого рівня. Прогнози, отримані від кожної з таких моделей, об'єднуються у вхідні характеристики нового навчального набору, у якому зберігається також значення цільової змінної. На цьому наборі далі навчається модель змішувача, що виконує остаточне передбачення на основі узагальнених прогнозів першого рівня.

У такий спосіб можна сформувати багаторівневий ансамбль, у якому використовуються різні моделі змішувачів, наприклад, одна з яких побудована на випадковому лісі, а інша — на методі опорних векторів. Для цього навчальний набір може бути поділений на три частини. Перша частина призначена для навчання моделей першого рівня. Друга частина використовується для створення нового набору, сформованого з прогнозів першого рівня, що слугує основою для тренування моделей другого рівня. Третя частина застосовується для побудови ще одного набору — на основі прогнозів другого рівня, — який використовується для навчання моделей третього рівня. Остаточне передбачення для нового прикладу формується шляхом послідовного проходження через усі рівні моделі [8, 9].

Стекінг застосовується як до моделей класифікації, так і до моделей регресії, причому процес для обох типів аналогічний. У випадку регресії спочатку формуються вхідні ознаки для мета-моделі шляхом генерації прогнозів базовими моделями, серед яких можуть бути модель випадкового лісу, XGBoost або нейронна мережа.

Перевагою стекінгу є те, що він дозволяє об'єднувати різні моделі з різними гіпотезами, поєднуючи їх сильні сторони та досягаючи вищої точності, що робить його популярним на змаганнях з машинного навчання. Проте його недоліком є висока обчислювальна складність і тривалість навчання, що може ускладнити застосування в системах, які потребують оперативного прийняття рішень у реальному часі.

Бустінг (Boosting) — це підхід, за якого кілька моделей навчаються послідовно, при цьому кожна наступна модель концентрується на тих прикладах, які були неправильно класифіковані або передбачені

попередньою. Суть полягає в тому, щоб посилити точність прогнозу, надаючи більшу вагу прикладам із помилками, що дозволяє поступово зменшувати похибку.

Серед найпоширеніших моделей, які реалізують цю концепцію, можна виділити AdaBoost, Gradient Boosting, CatBoost, LightGBM та XGBoost — кожна з них має свої особливості й оптимізації, проте всі засновані на одній ідеї покращення результату за рахунок поетапного навчання та корекції помилок.

3.2 Модель XGBoost - extreme gradient boosting

У процесі подальших експериментальних досліджень було прийнято рішення зосередити увагу на моделі XGBoost (eXtreme Gradient Boosting) [10], результати якої будуть порівнюватися з показниками інших моделей нейронних мереж. XGBoost вважається одним із найефективніших інструментів машинного навчання для обробки табличних даних. Її здатність успішно моделювати складні нелінійні залежності забезпечила широке визнання як у промислових застосуваннях, так і в наукових дослідженнях.

XGBoost по суті базується на концепції градієнтного бустінгу, але вирізняється впровадженням регуляризації для контролю складності моделі, що дозволяє зменшити ризик перенавчання. Завдяки цьому його часто називають регуляризованим градієнтним бустінгом, що ілюструє один із ключових чинників його високої ефективності. Щоб глибше зрозуміти архітектуру XGBoost, необхідно звернутись до базових понять, які лежать в його основі — зокрема, регресійних дерев та самої ідеї градієнтного бустінгу.

Регресійне дерево є базовою моделлю, яка застосовується для задач прогнозування числових значень. Воно формує рішення через послідовне розгалуження на основі логічних умов, що дає змогу захоплювати нелінійні залежності між ознаками та цільовою змінною. Однією з переваг таких дерев є інтерпретованість: структура дерева прозора і зрозуміла для людини.

Водночас ці моделі мають суттєві недоліки — вони нестійкі до змін у даних, схильні до перенавчання та не володіють повноцінною ймовірнісною основою, що обмежує їх застосування в умовах невизначеності.

Основна неформальна схема побудови точної моделі регресійного дерева передбачає створення кореневого вузла, визначення оптимального значення розбиття для кожної ознаки з метою максимізації певного критерію, після чого перевіряється виконання умов зупинки. Якщо критерії зупинки задоволені — побудову припиняють, в іншому випадку продовжують створювати нові вузли.

Навчання регресійного дерева залежить від кількох ключових параметрів. Глибина дерева визначає кількість рівнів і контролює складність моделі: чим глибше дерево, тим більша ймовірність перенавчання, тому зазвичай обмеження встановлюють у межах 4–8 рівнів. Кількість вузлів або листків також впливає на структуру дерева — цей параметр подібний до глибини, однак є асиметричним і може призвести до незбалансованої архітектури. Важливим критерієм є мінімальна кількість спостережень, необхідних для розбиття: вона визначає, скільки даних має бути у вузлі, щоб дозволити подальше розгалуження. Крім того, мінімальне покращення встановлює поріг, який нове розбиття повинно перевищити, щоб бути прийнятим у структуру дерева. Усі ці параметри разом формують правила, що забезпечують баланс між складністю дерева та його здатністю до генералізації.

Відсічення гілок дерева (tree pruning) застосовується після завершення навчання моделі з метою зменшення перенавчання. Зокрема, метод відсічення з оптимальною складністю (cost complexity pruning) поступово спрощує структуру дерева, видаляючи листові вузли. На кожному кроці обчислюється показник, що враховує як точність моделі, так і її складність, і на основі цього визначається доцільність подальшого спрощення. Це дозволяє досягти компромісу між здатністю моделі до узагальнення та її адаптацією до навчальних даних [11].

Алгоритм Gradient Boosting реалізує ідею поступового вдосконалення ансамблю слабких прогнозуючих моделей — зазвичай дерев регресії — де кожна наступна модель спрямована на виправлення помилок, допущених попередніми. Таким чином, формуються точніші прогнози, що в сукупності дозволяє побудувати потужного узагальнюючого прогнозатора. Цей підхід забезпечує гнучкість та високу точність у задачах як класифікації, так і регресії.

$$\hat{F}(x) = \sum_{i=1}^M \gamma_i h_i(x). \quad (3.1)$$

Остаточний прогноз у градієнтному бустінгу визначається як зважена сума всіх побудованих дерев M , де кожне дерево враховується з певним коефіцієнтом стиснення λ . Цей підхід дозволяє поступово вдосконалювати модель, зберігаючи при цьому контроль над її складністю та запобігаючи перенаванчання.

Однією з інтуїтивних концепцій градієнтного бустінгу є аналогія з градієнтним спуском (рисунок 3.1), що широко застосовується для мінімізації функції втрат у задачах оптимізації.

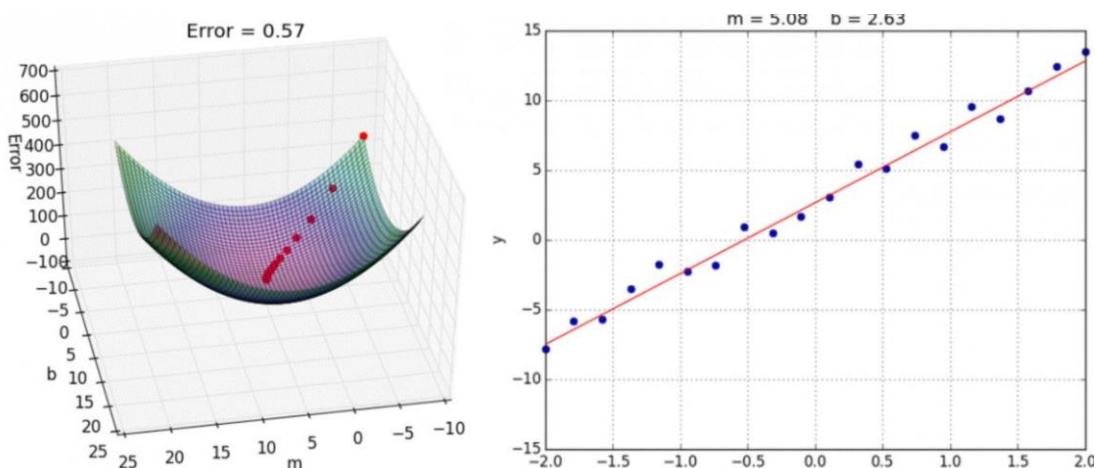


Рисунок 3.1 – Принцип градієнтного спуску для лінійної регресії

Наприклад, у випадку простої лінійної регресії (метод найменших квадратів, OLS), від'ємний градієнт функції втрат вказує напрямок, у якому необхідно коригувати параметри моделі (такі як нахил чи зсув), щоб зменшити похибку. У градієнтному бустінгу така логіка переноситься на побудову дерев: кожне наступне дерево будується таким чином, щоб скоригувати помилки попередніх, орієнтуючись на напрямок градієнта функції втрат, що робить цей підхід надзвичайно потужним і гнучким.

Представити це математично у вигляді наступного виразу для функції втрат, яка визначає міру відхилення прогнозованих значень від фактичних:

$$L = \sum_i (y_i - \hat{y}_i)^2, \quad (3.2)$$

де y_i - фактичне значення цільової змінної для i -го прикладу;

\hat{y}_i — прогнозоване значення.

Ця функція втрат є стандартною для задач регресії та називається сумою квадратів залишків (residual sum of squares, RSS). Її мінімізація є основною метою алгоритму градієнтного бустінгу при побудові моделі, що максимально точно наближає залежність між вхідними ознаками та цільовою змінною.

Для мінімізації функції втрат обчислюється її похідна за параметрами моделі - зокрема, за кутовим коефіцієнтом і вільним членом. Потім до кожного параметра додається протилежне за знаком значення цієї похідної, зважене на коефіцієнт швидкості навчання (learning rate). Такий крок здійснюється ітеративно: на кожному етапі параметри моделі коригуються у напрямку зменшення помилки, доки не буде досягнуто збіжності або прийняттого рівня точності. Цей процес дозволяє поступово наблизитися до оптимального набору параметрів, які забезпечують мінімальне значення функції втрат і, відповідно, високу якість прогнозування.

$$\theta^{(m+1)} = \theta^{(m)} - learningRate * \frac{\partial L}{\partial \theta^{(m)}}. \quad (3.3)$$

Оновлення параметрів моделі, зокрема нахилу та перетину, базується на використанні похідної функції втрат за відповідними змінними. В даному контексті сума квадратів залишків виражається через передбачення моделі. Якщо взяти похідну функції втрат безпосередньо за прогнозованими значеннями, то можна отримати напрямок, у якому слід коригувати ці передбачення. Ітеративне оновлення прогнозів за цим градієнтом дозволяє поступово зменшити залишки, тобто різницю між фактичними й передбаченими значеннями, забезпечуючи точнішу апроксимацію цільової функції. Такий підхід є основою градієнтного бустінгу [12], де на кожному кроці модель навчається компенсувати помилки попередніх прогнозів:

$$\hat{y}_i^{(m+1)} = \hat{y}_i^{(m)} - learningRate * \frac{\partial L}{\partial \hat{y}_i^{(m)}}. \quad (3.4)$$

У цьому підході існує принципове обмеження: для прямого обчислення втрат і відповідного оновлення прогнозів необхідно знати справжнє значення цільової змінної y . Це унеможливорює застосування методу без наявності реальних даних. Вихід із цієї ситуації забезпечують послідовно побудовані дерева регресії, які виступають як слабкі моделі. Замість прямого використання аналітичного градієнта, на кожному етапі будується дерево, яке навчається прогнозувати значення градієнтів втрат для поточного стану моделі. Це дозволяє алгоритму застосовуватись до нових, ще не бачених даних, забезпечуючи узагальнення.

Оскільки сам напрямок (градієнт) визначає лише вектор руху до мінімуму, постає питання вибору оптимального розміру кроку в цьому напрямку. Цей параметр визначає вплив нової слабкої моделі на загальний прогноз. Один із поширених методів визначення цього параметра — лінійний пошук, що дозволяє адаптивно знаходити оптимальне значення для кожної

ітерації з урахуванням мінімізації функції втрат.

Функцію втрат у випадку градієнтного бустингу можна представити у вигляді суми внесків усіх дерев, де кожне дерево додає корекцію до попереднього прогнозу. Завдяки математичному перетворенню, цю функцію вдається виразити через параметри дерева, такі як значення у листках та структура розбиття, що дозволяє сформулювати рівняння для оцінки якості дерева з точки зору зменшення втрат. Такий підхід дає змогу оптимізувати побудову дерева, спрямовуючи її на мінімізацію функції втрат на кожній ітерації навчання.

$$L(f_m) = - \sum_{j=1}^T \frac{G_j^2}{n_j}, \quad (3.5)$$

де T - кількість листових вузлів у дереві;

G_j^2 - сума градієнтів у листі;

j - кількість вибірок у цьому листовому вузлі.

Якість дерева можна оцінити через функцію втрат, де кожен листовий вузол робить внесок у загальну функцію втрат пропорційно. Сума цих значень по всіх листах дозволяє отримати значення, яке оптимізується на кожній ітерації побудови дерева в алгоритмі XGBoost.

З огляду на обчислювальну складність повного перебору всіх можливих структур дерева, у XGBoost замість цього застосовується підхід локальної оптимізації, коли функція втрат мінімізується на кожному етапі побудови дерева, а саме — під час додавання нових гілок. Це досягається шляхом оцінювання приросту виграшу (gain), який отримується при розбитті вузла за певною ознакою. Розрахунок приросту базується на другому порядку апроксимації функції втрат із використанням градієнтів, що дозволяє значно точніше та ефективніше обирати розбиття. Таким чином, модель не просто підганяє залишки, як у класичному градієнтному бустингу, а виконує оптимізацію, що забезпечує кращу узагальнювальну здатність та

керованість складністю дерева:

$$Gain = \frac{G_{jL}^2}{n_{jL}} + \frac{G_{jR}^2}{n_{jR}} - \frac{(G_{jL} + G_{jR})^2}{n_{jL} + n_{jR}}, \quad (3.6)$$

де G_{jL} та n_{jL} - сума градієнтів й кількість вибірок для лівого боку розбиття;

G_{jR} та n_{jR} - сума градієнтів й кількість вибірок для правого боку розбиття.

Це й становить суть градієнтного бустінгу: алгоритм починає з побудови початкового прогнозу, який мінімізує функцію втрат. У задачах регресії цей прогноз зазвичай дорівнює середньому значенню цільової змінної у навчальному наборі. Якщо ж завданням є класифікація, то початковим прогнозом є логарифм відношення ймовірностей для бінарного класу, що забезпечує стартову точку для подальшого уточнення прогнозу шляхом додавання нових дерев:

$$F_0(x) = \operatorname{argmin} \sum_{j=1}^n L(\gamma_j, \gamma). \quad (3.7)$$

На наступному етапі необхідно обчислити від'ємний градієнт функції втрат по відношенню до поточних прогнозів моделі. Цей градієнт відображає напрямок, у якому потрібно скорегувати прогноз для зменшення помилки. Далі будується регресійне дерево, яке навчається апроксимувати ці градієнти. Його завдання — відтворити напрямки помилок, на основі яких модель зможе уточнити прогноз й зробити його більш точним навіть на нових, ще не бачених даних:

$$r_{jm} = - \left[\frac{\partial L(y_i, F(x_i))}{\partial F(x_i)} \right]. \quad (3.8)$$

Отже, коли побудована структура дерева регресії, залишається визначити єдине числове значення для кожного листа, яке мінімізує загальні втрати. Це досягається шляхом взяття похідної функції втрат за гаммою (параметром, що відповідає значенню у листі) й прирівнювання її до нуля. У результаті оптимальним значенням для прогнозу в кожному листковому вузлі стає середнє значення залишків (градієнтів) у цьому листі, що забезпечує найкращу локальну апроксимацію цільової функції:

$$W_{jm} = -\frac{G_{jm}}{n_{jm}}. \quad (3.9)$$

Нове дерево регресії додається до початкового прогнозу, причому його внесок зменшується шляхом множення на визначений коефіцієнт швидкості навчання. Далі обчислюється від'ємний градієнт вже оновленої моделі, і цей процес повторюється задану кількість разів, кожного разу уточнюючи прогноз шляхом додавання нових дерев, які компенсують помилки попередніх. Такий послідовний підхід дозволяє поступово зменшувати втрати й покращувати загальну точність моделі.

$$F_m(x) = F_{m-1}(x) + \gamma_m h_m(x). \quad (3.10)$$

Алгоритм XGBoost — метод, створений Генці Чженом та Карлосом Гестеріном у 2016 році, позиціонується як оптимізована бібліотека розподіленого градієнтного бустінгу, орієнтована на високу продуктивність, гнучкість і кросплатформенність. На відміну від класичного градієнтного бустінгу, XGBoost включає низку вдосконалень, зокрема реалізацію підсилення на основі методу Ньютонa, механізми регуляризації для контролю складності моделі, вбудовану обробку пропущених значень із урахуванням розрідженості даних, застосування зважених квантильних ескізів для оптимізації поділу вузлів, підтримку користувацьких функцій

втрат, а також можливість паралельного виконання, що забезпечує значне пришвидшення навчання.

XGBoost ґрунтується на структурі градієнтного підсилення, в якій слабкими учнями виступають регресійні дерева, однак у ньому реалізовано низку важливих інновацій. Класичний підхід до градієнтного бустінгу розглядає процес навчання як градієнтний спуск, де напрямок обчислюється за допомогою першої похідної (градієнта функції втрат), а розмір кроку визначається методом лінійного пошуку. Основна відмінність XGBoost полягає в тому, що замість звичайного градієнтного спуску він застосовує метод Ньютона—Рафсона, який передбачає використання не лише градієнтів, а й других похідних для досягнення точнішої оцінки як напрямку, так й величини кроку зменшення функції втрат. Такий підхід дозволяє точніше та ефективніше наближатися до глобального мінімуму, покращуючи якість моделі та прискорюючи її збіжність:

$$x^{(i+1)} = x^{(i)} - p_m \nabla f(x^i). \quad (3.11)$$

При використанні методу Ньютона оновлення відбувається з урахуванням не лише першої, а й другої похідної функції втрат. Це дозволяє враховувати кривизну функції в околі поточної точки, що забезпечує більш точний напрямок і величину кроку під час мінімізації [12, 13]. Формально, оновлення параметра відбувається за формулою:

$$x^{(i+1)} = x^{(i)} - \frac{\nabla f(x^i)}{Hess(x^i)}, \quad (3.12)$$

де *Hess* (скорочено від Hessian) відноситься до Гессіана — тобто другої похідної функції втрат по відношенню до параметра x .

У контексті XGBoost *Hess* використовується для уточнення напрямку та величини кроку оновлення, дозволяючи моделі більш точно та стабільно

мінімізувати функцію втрат, враховуючи кривизну її поверхні.

Таким чином, оновлення параметрів у XGBoost враховує як першу, так і другу похідну функції втрат, що забезпечує значно точнішу оцінку напряму оптимізації. Завдяки використанню розкладу Тейлора другого порядку для локального наближення функції втрат, модель отримує не лише інформацію про напрям зміни (градієнт), але й про її кривизну (Гессіан), що дозволяє ефективніше будувати нові дерева.

Це наближення виглядає так:

$$L(f_m) \approx \sum_{j=1}^{T_m} \left[G_{jm} w_{jm} + \frac{1}{2} (H_{jm} w_{jm}^2) \right] + \Omega(f_m), \quad (3.13)$$

де T_m - це кількість листків у дереві;

G_{jm} - та - це сума градієнту в області;

H_{jm} та - це сума гессіана в області;

w_{jm} - сума ваги листка.

Це формулювання функції втрат, побудоване на градієнті та Гессіані, стає основою для оптимізації в кожному новому дереві. Однією з ключових переваг такого підходу є те, що оптимізація залежить лише від першої та другої похідної для кожного з прикладів. Завдяки цьому механізму XGBoost може легко адаптуватися до різних функцій втрат, включаючи нестандартні або спеціалізовані, оскільки достатньо лише надати відповідні вирази для градієнта та другої похідної. Це забезпечує високу гнучкість й дозволяє використовувати XGBoost у широкому спектрі задач — від регресії та класифікації до специфічних доменних застосувань.

Щоб уникнути проблем із використанням пам'яті при сортуванні великих наборів неперервних ознак, XGBoost застосовує ефективний підхід до знаходження приблизних точок розбиття. Замість повного аналізу всіх можливих значень, модель формує кандидатські розбиття, ґрунтуючись на процентилях розподілу ознаки. Це означає, що неперервна ознака

поділяється на рівні інтервали за кількістю спостережень — наприклад, можна розглянути 10 квантилів і аналізувати зміни лише на межах кожного десятого процентиля, як у випадку значень від 10 до 90 для набору від 1 до 100. Такий підхід суттєво зменшує обчислювальне навантаження і дозволяє отримати якісне наближення. Крім того, XGBoost враховує зміну суми ознак між точками та перевіряє, чи зросла кількість елементів під новою точкою розбиття на фіксовану величину порівняно з попередньою. Це дозволяє уникнути надмірного дроблення і забезпечити стабільну обчислювальну ефективність при побудові дерева.

Якщо значення ознак дуже схожі, немає сенсу виконувати розбиття між ними — доцільніше зосередитись на тих частинах даних, де модель робить найбільші помилки. У цьому контексті гессіан виступає як ваговий коефіцієнт, що відображає ступінь невірності прогнозу: чим вищий гессіан, тим вагомішим є внесок спостереження у втрати. Відповідно, перший 10-квантиль визначається не за 10% кількості значень ознаки, а за 10% сумарного гессіану, що дозволяє розумно локалізувати зони з найбільш проблемними прогнозами.

Ще однією важливою відмінністю XGBoost є спосіб роботи з відсутніми значеннями. Кожен вузол дерева містить не лише точку розбиття, а й базовий напрямок фільтрації для пропущених значень. За замовчуванням вони спрямовуються вліво, але модель може навчитися автоматично обирати оптимальний напрям. Для цього вона порівнює приріст приросту втрат, коли всі відсутні значення направляються вліво, та коли — вправо. Обирається варіант із меншим приростом, після чого цей напрямок закріплюється за вузлом. Таким чином, навіть якщо значення ознаки відсутні, цього достатньо для прогнозування, адже розрахунки базуються на градієнтах та гессіанах, а не на самих значеннях ознак.

Алгоритм XGBoost передбачає налаштування численних гіперпараметрів, що дозволяє адаптувати його до конкретних умов задачі та структури даних. Параметр `scale_pos_weight` регулює баланс між

позитивними і негативними прикладами, що особливо важливо для несбалансованих вибірок. Параметр `min_child_weight` визначає мінімальну суму ваг екземплярів у дочірньому вузлі; чим більше це значення, тим обережніше модель виконує поділ. Параметр `eta`, що відповідає за зменшення кроку навчання, дозволяє зробити підсилення більш плавним, зменшуючи ризик перенавчання. Значення `gamma` задає поріг для зменшення функції втрат, необхідне для подальшого поділу вузла, й також впливає на рівень консервативності дерева. Параметр `subsample` визначає частку навчальних даних, що використовуються на кожній ітерації; обмеження цього значення дозволяє зменшити варіативність і ризик перенавчання. Підвибірка ознак у XGBoost реалізується через параметри родини `colsample_by`, що відповідають за вибірку ознак при побудові дерева, на кожному рівні або на рівні окремого вузла. Їх спільне використання дозволяє значно зменшити розмірність кожної ітерації пошуку оптимального поділу.

Регуляризація реалізована через L_2 та L_1 , які дозволяють керувати складністю моделі шляхом штрафування великих ваг. Нарешті, `tree_method` задає стратегію побудови дерева — від точного повного перегляду можливих розділень до оптимізованого гібридного методу, який балансує точність і швидкість. Саме гнучке налаштування всіх цих параметрів дає XGBoost змогу ефективно вирішувати широкий спектр задач регресії та класифікації.

4 ДОСЛІДЖЕННЯ МОДЕЛІ ШТУЧНОЇ НЕЙРОМЕРЕЖІ

Для дослідження запропонованої моделі нейромережі використовувалося дані з великого приміщення площею 1320 м², що обладнане системами моніторингу та керування мікрокліматом й яке відповідає сучасним вимогам автоматизованого управління параметрами внутрішнього середовища. Приміщення оснащено сучасними датчиками для вимірювання ключових мікрокліматичних параметрів, включаючи температуру, відносну вологість, рівень CO₂, швидкість та напрямок повітряних потоків, сонячну радіацію та рівень опадів. Робота системи обігріву відстежувалася на основі історії її активності, що зберігалася у контролері керування мікрокліматом. Усі отримані дані використовувалися для аналізу динаміки змін кліматичних умов у приміщенні та їх впливу на регуляційні процеси. Отримані параметри були використані як вхідні змінні у моделі прогнозування мікроклімату приміщення, що дозволило здійснювати точний аналіз та контроль його внутрішнього середовища.

4.1 Побудова моделі машинного навчання

Для побудови моделі машинного навчання XGBoost використовували Python 3.8 та бібліотеку XGBoost. Для налаштування гіперпараметрів моделі було застосовано Optuna, відкритий фреймворк для оптимізації параметрів машинного навчання [11]. Гіперпараметри шукалися за допомогою алгоритму Tree-structured Parzen Estimator (TPE) [13], що дозволяло ефективно досліджувати простір можливих значень. Основною метою оптимізації було мінімізувати середньоквадратичну помилку (MSE) на валідаційному наборі даних під час 5-кратної крос-валідації, що забезпечувало підбір найбільш оптимальних параметрів моделі.

Використана у цьому дослідженні модель XGBoost є вдосконаленим

фреймворком градієнтного бустингу, що застосовує ансамблеве навчання на основі дерев рішень [12].

Функція втрат $L(\Theta)$, вираз (3.2), оцінює різницю між фактичними та прогнозованими значеннями, що дозволяє моделі поступово покращувати свої передбачення.

Регуляризаційний член $\Omega(\Theta)$ контролює складність моделі, допомагаючи уникнути перенавчання. На кожному етапі навчання модель оновлюється за допомогою методу градієнтного бустингу, навчаючись на залишкових помилках попереднього кроку:

Використовуючи цей підхід, XGBoost послідовно будує кілька CART (Classification and Regression Trees), кожне з яких аналізує помилки попереднього дерева та вносить коригування. Цей процес повторюється T разів, що дозволяє кожному дереву навчатися на різних аспектах даних, покращуючи загальну точність моделі.

Фінальне прогнозоване значення визначається як сума прогнозів усіх дерев, помножених на оптимальні вагові коефіцієнти, встановлені під час навчання (рисунок 4.1).

У цьому дослідженні гіперпараметри моделі були налаштовані з урахуванням її продуктивності. Початкове значення швидкості навчання було встановлено на рівні 0.1, що допомагало уникнути занадто швидких оновлень і зменшувало ризик перенавчання. Максимальна глибина дерев була обмежена 5, щоб забезпечити належний баланс між точністю та узагальнювальною здатністю. Кількість дерев рішень була встановлена на 100, що дозволяло моделі навчатися на різних патернах у даних.

Завдяки цим параметрам навчання XGBoost проводилося таким чином, щоб мінімізувати похибку прогнозування, забезпечуючи при цьому узагальнені висновки. Такий підхід дозволяє ефективно використовувати XGBoost для прогнозування мікроклімату у великих приміщеннях, зокрема для оптимізації керування температурою, вологістю та концентрацією CO₂ у реальному часі.

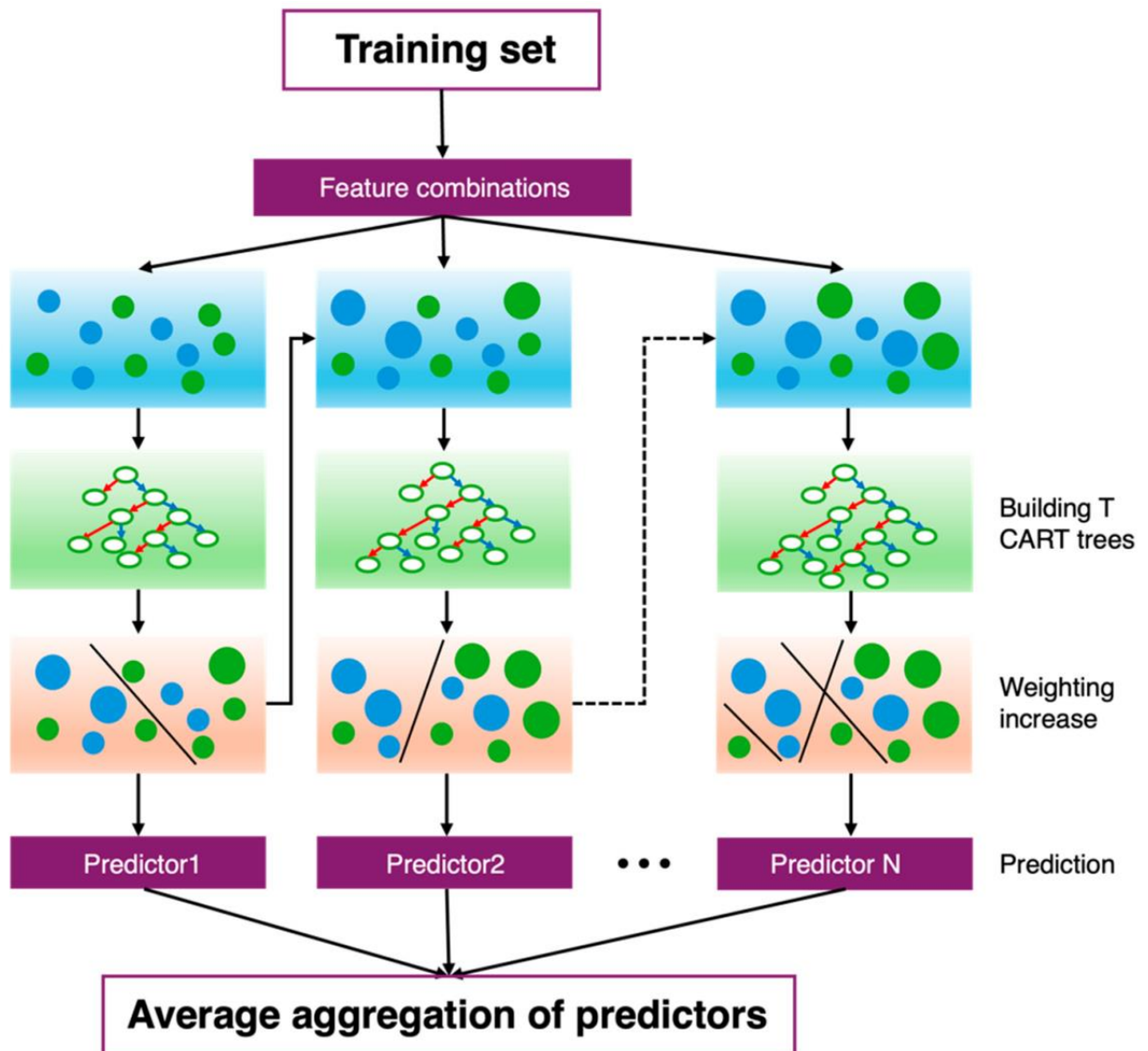


Рисунок 4.1 - Схематичне зображення використання моделі XGBoost для моделювання мікроклімату

Рисунок 4.1 ілюструє принцип роботи моделі XGBoost, зокрема її структуру та механізм навчання. XGBoost використовує ансамблеве навчання на основі градієнтного бустингу, створюючи послідовність дерев рішень Classification and Regression Trees (CART). Кожне наступне дерево навчається на залишкових помилках попередніх дерев, поступово покращуючи прогноз. Схема демонструє, що на кожному етапі додається нове дерево, яке враховує вагові коефіцієнти попередніх рішень та коригує передбачені значення. Процес навчання продовжується, доки не буде досягнуто заданої кількості ітерацій або мінімізації функції втрат. Кінцеве

прогнозне значення отримується шляхом сумування внесків усіх дерев, помножених на оптимальні ваги, визначені в ході навчання. Це дозволяє XGBoost досягати високої точності, зберігаючи при цьому низьку ймовірність перенавчання завдяки використанню регуляризації. Завдяки такій структурі модель XGBoost добре підходить для прогнозування параметрів мікроклімату у великих приміщеннях, оскільки може ефективно виявляти складні залежності між змінними та адаптуватися до різних умов експлуатації.

Для оцінки продуктивності моделей, побудованих у цьому дослідженні, було розраховано середньоквадратичну похибку (RMSE), коефіцієнт детермінації (R^2) та залишкову передбачувальну девіацію (RPD) як для навчального, так і для тестового наборів даних. Ці метрики широко використовуються для оцінювання точності регресійних моделей.

Середньоквадратична похибка (RMSE) визначається як квадратний корінь із середнього значення квадратів різниць між фактичними та прогнозованими значеннями:

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2}, \quad (4.1)$$

де n - кількість спостережень,

y_i - фактичні значення,

\hat{y}_i - прогнозовані значення моделі

Коефіцієнт детермінації R^2 є важливим показником якості моделі, який демонструє, наскільки добре модель пояснює варіабельність даних:

$$R^2 = 1 - \frac{\sum (y_i - \hat{y}_i)^2}{\sum (y_i - \bar{y})^2}, \quad (4.2)$$

де \bar{y} - середнє значення фактичних спостережень.

Значення R^2 змінюється у діапазоні від 0 до 1. Чим ближче значення R^2 до 1, тим краще модель відображає закономірності у даних і пояснює їх варіативність. Навпаки, значення, близькі до 0, вказують на слабку пояснювальну здатність моделі та високу ступінь невизначеності у прогнозах [13].

Високе значення R^2 свідчить про те, що модель успішно враховує зв'язки між вхідними параметрами та прогнозованими значеннями, що є критично важливим для точного керування мікрокліматом у великих приміщеннях. Однак сам коефіцієнт детермінації не дає повної картини, тому його слід аналізувати у поєднанні з іншими метриками, такими як RMSE і RPD, для отримання повної оцінки продуктивності моделі.

Залишкова передбачувальна девіація (RPD) визначається як відношення стандартного відхилення фактичних значень до стандартного відхилення помилок моделі. Ця метрика відображає здатність моделі точно передбачати варіативність даних:

$$RPD = \frac{\sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \bar{y})^2}}{RMSE}, \quad (4.3)$$

Вищі значення RPD вказують на кращу прогностичну ефективність моделі. Типові інтерпретації значень RPD такі:

- RPD < 1 – модель вважається непридатною для прогнозування варіабельності даних;
- RPD \approx 1 – середній рівень передбачувальної здатності;
- RPD > 1 – модель є корисною для прогнозування;
- RPD > 2 – висока точність прогнозування [13].

Ця метрика є особливо важливою для оцінки ефективності моделей, застосованих у прогнозуванні мікрокліматичних умов у великих приміщеннях. Високе значення RPD свідчить про те, що модель здатна точно передбачати зміну параметрів середовища, що має вирішальне значення для

автоматизованих систем управління.

Було оцінено ефективність моделі штучного інтелекту XGBoost для прогнозування внутрішньої температури (TMPin), відносної вологості (RHM) та концентрації вуглекислого газу (CO₂) на 30 хвилин наперед.

Для оцінки продуктивності моделей застосовувалися метрики RMSE і R², причому середньоквадратичні похибки для навчального та тестового наборів даних позначалися як RMSET та RMSEP, а коефіцієнти детермінації—як R_T² та R_P² відповідно (таблиця 4.1).

Таблиця 4.1 – Параметри продуктивності моделі прогнозування XGBoost

Параметр	RMSET	R _T ²	RMSEP	R _P ²	RPD
TMPin	0.5226	0.9753	0.5471	0.9724	6.0197
RHM	3.1256	0.9682	3.2789	0.9656	5.3878
CO ₂	8.5519	0.9940	9.5919	0.9929	11.8464

Окрім цього, було досліджено значення RPD на тестовому наборі даних для оцінки точності прогнозування. Високі значення RPD свідчать про надійність моделі у передбаченні змін мікроклімату, що має вирішальне значення для ефективного керування великими приміщеннями.

Результати дослідження показали, що для прогнозування внутрішньої температури (TMPin) модель XGBoost продемонструвала найнижчі значення RMSE для обох наборів даних та найвищі значення R² - 0.9753 для навчального набору й 0.9724 для тестового. Аналогічно, для відносної вологості (RHM) модель XGBoost мала найнижчі значення RMSE та найвищі R² — 0.9682 для навчального набору і 0.9656 для тестового. Щодо прогнозування концентрації CO₂, модель XGBoost значно перевершила всі інші підходи, забезпечивши найнижчі RMSE (8.5519 для навчального набору та 9.519 для тестового) й найвищі R² (0.9940 та 0.9929, відповідно).

Отримані результати свідчать, що продуктивність XGBoost значно перевищує попередні дослідження у сфері прогнозування мікроклімату

великих приміщень та пов'язаних з ним технологічних процесів.

4.2 Порівняння прогнозованих та фактичних значень

Значення RPD для кожного параметра показали, що моделі MLR, SVM та ANN також демонструють хороші результати, оскільки всі мали $RPD > 3.8$, що свідчить про відмінну передбачуваність варіацій даних. Однак XGBoost продемонстрував значно вищі значення RPD:

- 6.0197 для внутрішньої температури (TMPin),
- 5.3878 для відносної вологості (RHМ),
- 11.8464 для концентрації CO₂.

Ці значення значно перевищують показники інших моделей, що вказує на найвищу точність і надійність XGBoost для прогнозування мікроклімату у великих приміщеннях. Додатковий аналіз, заснований на графічному зіставленні фактичних та прогнозованих значень для кожного параметра (рисунок 4.2), показав, що чим ближче передбачені значення до лінії 1:1, тим точніше модель здатна прогнозувати параметри мікроклімату.

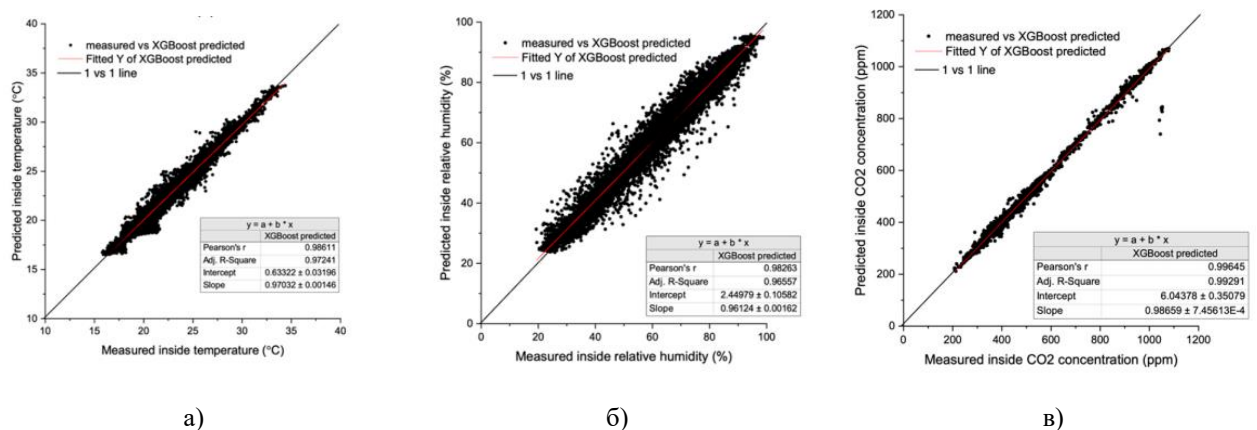


Рисунок 4.2 - Порівняння прогнозованих та фактичних значень внутрішньої температури приміщення TMPin (а), відносної вологості приміщення RHМ (б) та значень концентрації CO₂ (в) у приміщенні

Візуальне порівняння дозволяє зробити висновок про стабільність й точність XGBoost у передбаченні ключових екологічних параметрів, що ще раз підтверджує її перевагу над іншими підходами.

Результати підтверджують, що XGBoost виявилася стабільнішою, ніж інші моделі.

4.2 Дослідження взаємозв'язку між характеристиками моделі та особливостями формування параметрів мікроклімату.

У цьому дослідженні проаналізовано значення вхідних параметрів у прогнозуванні внутрішньої температури приміщення (TMPin) на 30 хвилин наперед за допомогою моделі XGBoost (рисунок 4.3). Порівнюючи значення F-оцінки кожного параметра, було виявлено, що TMPin (поточна температура) має найвищу значущість із показником 778, що свідчить про її вирішальну роль у визначенні температури через 30 хвилин. Сонячна радіація (SRD) посіла друге місце з показником 533, що вказує на її значний вплив на зміну температури у приміщенні. Графік показує оцінку значущості параметрів, що визначає, наскільки часто кожен параметр використовується в розгалуженнях дерев рішень моделі. Найважливішими факторами виявилися TMPin (778) та SRD (533), тоді як HEAT (3), WDR (15) і RHM (31) мали мінімальний вплив [13].

Аналіз показав, що найбільший вплив на зміну температури має поточне її значення (TMPin), а також такі чинники, як час доби (DATETIME), опади (RAIN), зовнішня температура (TMPout) та накопичена сонячна радіація (CSR), що підтверджує важливість сонячного випромінювання. Водночас стан обігріву (HEAT), напрямок вітру (WDR) та вологість (RHM) мають незначний вплив. Модель XGBoost продемонструвала, що TMPin є ключовим чинником прогнозу, а SRD і DATETIME взаємодіють на глибших рівнях дерева.

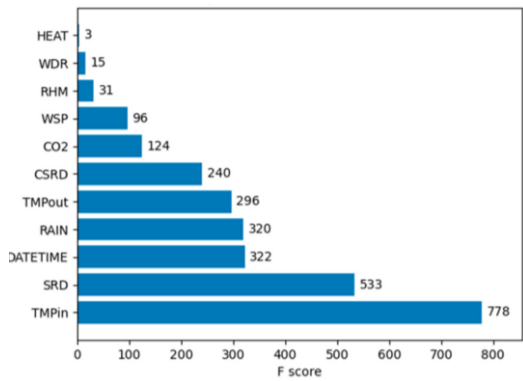


Рисунок 4.3 - Аналіз впливу параметрів моделі XGBoost на прогнозування внутрішньої температури

Зростання температури прогнозується при збільшенні сонячного випромінювання й часу, що підтверджує здатність моделі виявляти фізичні закономірності мікроклімату.

Дерево рішень XGBoost, побудоване для прогнозування TMPin (рисунок 4.4), демонструє логіку прийняття рішень моделі на основі основних параметрів. На кореневому вузлі розміщено TMPin, що підтверджує його провідну роль у прогнозі. У глибших рівнях структури дерева враховуються SRD та DATETIME, які відіграють важливу роль у зростанні температури в короткостроковій перспективі [14].

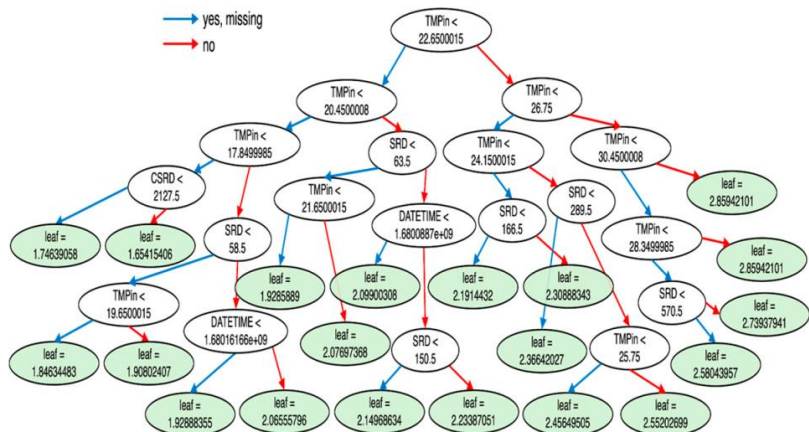


Рисунок 4.4 - Структура дерева рішень моделі XGBoost для прогнозування внутрішньої температури

Аналіз графіків свідчить, що модель XGBoost адекватно відображає природні закономірності зміни температури: підвищення сонячної радіації (SRD) супроводжується зростанням температури, а час доби (DATETIME) істотно впливає на загальний тепловий режим. Водночас вплив таких параметрів, як обігрів (HEAT) і вологість (RHM), виявляється незначним для короткострокового прогнозу. Це підтверджує здатність моделі точно виявляти ключові чинники, що формують внутрішню температуру, та забезпечувати надійне передбачення її зміни в межах короткого періоду.

У контексті прогнозування відносної вологості повітря на найближчі 30 хвилин модель також продемонструвала ефективність. Результати аналізу важливості параметрів (за графіком F- оцінки) показали, що провідним чинником прогнозу є поточна вологість (RHM) із найвищим значенням 776. Значний вплив також виявлено у параметра сонячної радіації (SRD), чий показник дорівнює 512, що підтверджує її суттєву роль у регулюванні мікроклімату в приміщенні.

У прогнозуванні відносної вологості повітря у приміщенні (RHM) на 30 хвилин наперед за допомогою моделі XGBoost було проведено аналіз відносної важливості кожного вхідного параметра (рисунок 4.5).

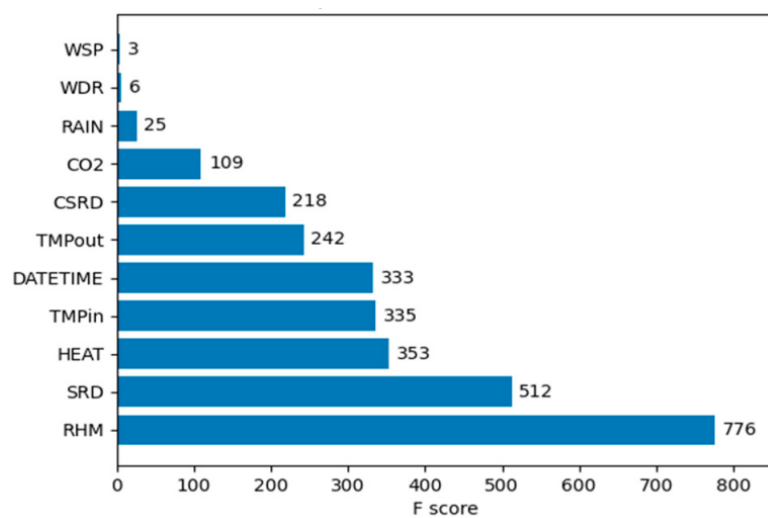


Рисунок 4.5 - Аналіз впливу параметрів моделі XGBoost на прогнозування відносної вологості

Аналіз значень F-оцінки для прогнозування вологості показав, що після параметрів RHM і SRD, помірний вплив на зміну вологості мають такі фактори, як стан системи обігріву (HEAT), поточна температура в приміщенні (TMPin) та час доби (DATETIME), кожен із яких демонструє значення вище 333. Це вказує на їхню помірну, але все ж суттєву роль у формуванні мікроклімату. Натомість такі параметри, як швидкість вітру (WSP), його напрямок (WDR) та наявність опадів (RAIN), продемонстрували дуже низькі значення, що підтверджує їхній мінімальний вплив у короткостроковому горизонті прогнозу. Так само, як і у випадку з температурою, домінуючими чинниками залишаються сам прогнозований параметр (RHM) та сонячна радіація (SRD), що вказує на ключову роль випромінювання не лише в регуляції температурного режиму, а й у динаміці вологості в приміщенні.

Дерево рішень XGBoost, побудоване для прогнозування відносної вологості RHM (рисунок 4.6), демонструє логіку ухвалення рішень на основі ключових змінних.

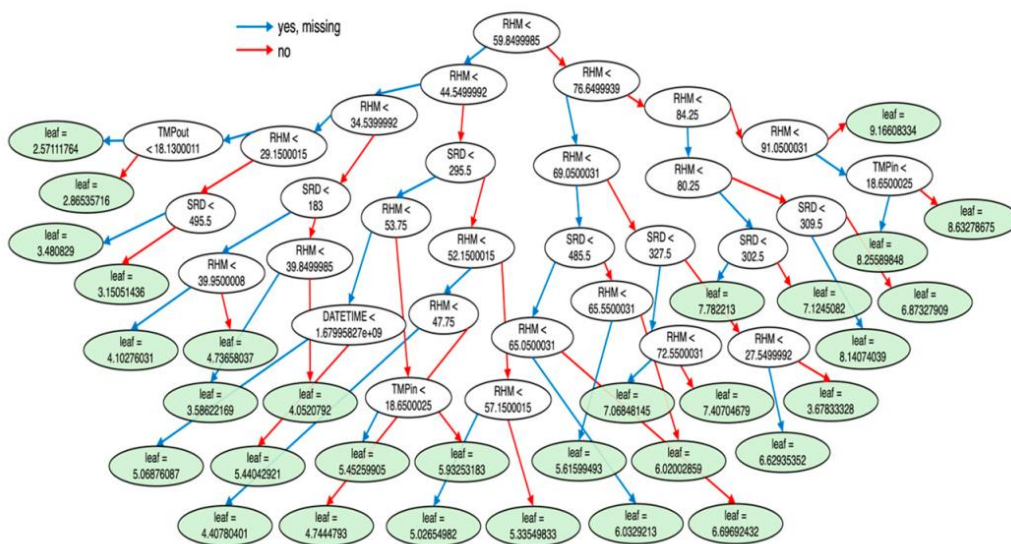


Рисунок 4.6 - Структура дерева рішень моделі XGBoost для прогнозування відносної вологості

На його вершині розташовано параметр RHM, що свідчить про його провідну роль у моделюванні. На нижчих рівнях дерева помітна участь таких факторів, як сонячна радіація (SRD), внутрішня (TMPin) та зовнішня температура (TMPout), що вказує на їх вагомий вплив на рівень вологості. Порівняно з деревом рішень для прогнозування температури, структура цієї моделі є значно складнішою, що свідчить про більшу багатофакторність та складнішу взаємодію параметрів при прогнозуванні динаміки вологості повітря.

Результати дослідження свідчать, що найвпливовішими факторами для короткострокового прогнозування рівня вологості повітря є сама поточна вологість та інтенсивність сонячного випромінювання. Водночас обігрів демонструє суттєвіший вплив на вологість, ніж на температуру, що підкреслює його роль у формуванні мікроклімату. Модель XGBoost виявилася здатною відтворювати складні взаємозв'язки між параметрами, оскільки дерево рішень для вологості має глибшу та розгалуженішу структуру порівняно з температурною моделлю. Це підтверджує ефективність XGBoost як інструмента точного прогнозування динаміки мікрокліматичних показників у великих приміщеннях.

Під час прогнозування концентрації CO₂ у приміщенні з горизонтом у 30 хвилин за допомогою моделі XGBoost було здійснено аналіз внеску кожного з вхідних параметрів, що дало змогу оцінити їх відносну значущість у побудові прогнозу (рисунок 4.7).

Аналіз показника F-скорю у контексті прогнозування концентрації CO₂ через 30 хвилин засвідчує, що основним чинником є поточне значення CO₂, яке демонструє найвищий рівень значущості. Друге місце за впливом посідає сонячна радіація (SRD), що свідчить про її тісний зв'язок із вуглекислим газом у повітрі. Значущими виявилися також внутрішня температура (TMPin), час доби (DATETIME) та опади (RAIN), кожен із яких має суттєвий вплив на динаміку концентрації CO₂. Натомість такі параметри, як швидкість та напрямок вітру, виявилися малозначущими, оскільки їхній внесок у зміну

концентрації вуглекислого газу є мінімальним.

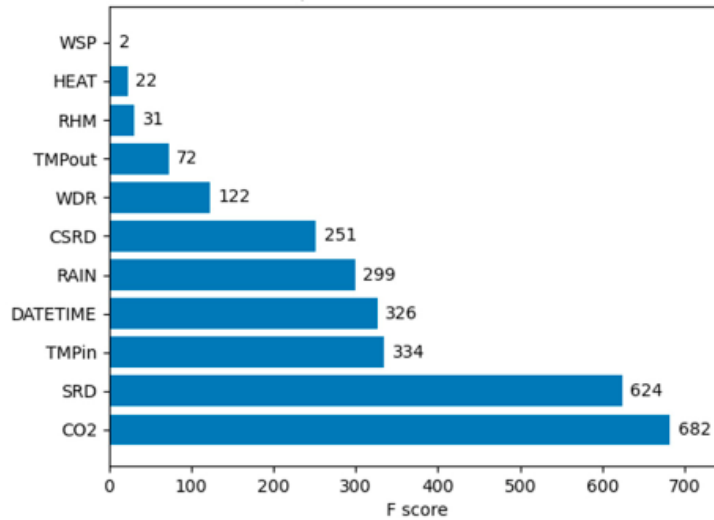


Рисунок 4.7 - Аналіз впливу структури рішення моделі XGBoost для прогнозування концентрації CO₂

Структура дерева рішень XGBoost, побудованого для прогнозування концентрації CO₂ (рисунок 4.8), демонструє, що на найвищому рівні моделі домінує поточне значення CO₂, що підтверджує його провідну роль у формуванні майбутніх значень. У подальших вузлах модель враховує сонячну радіацію (SRD) та внутрішню температуру (TMPin), що вказує на їхній вплив через механізми фотосинтезу, повітрообміну та температурної динаміки, які разом визначають коливання концентрації вуглекислого газу в просторі приміщення.

Модель XGBoost виявила два характерні сценарії динаміки концентрації CO₂ у приміщенні. У природних умовах, коли вміст вуглекислого газу залишається нижчим за медіанне значення, спостерігається його зниження із підвищенням сонячної радіації та температури, що свідчить про активізацію фотосинтезу і відповідне споживання CO₂ рослинами. У випадку карбонової фертизації, коли рівень CO₂ перевищує поріг 800 ppm, модель також фіксує його стрімке зменшення за умов інтенсивного сонячного випромінювання та високої температури, що підтверджує

здатність XGBoost враховувати ефекти штучного збагачення середовища вуглекислим газом. Така поведінка системи вказує на ефективне розпізнавання фотосинтетичних процесів і їх взаємозв'язку з керованими параметрами мікроклімату. Модель не лише прогнозує рівень CO₂, але й дозволяє відстежувати вплив управлінських рішень, таких як контроль вентиляції чи дозоване внесення CO₂, формуючи основу для адаптивного автоматизованого управління мікрокліматом у великих приміщеннях.

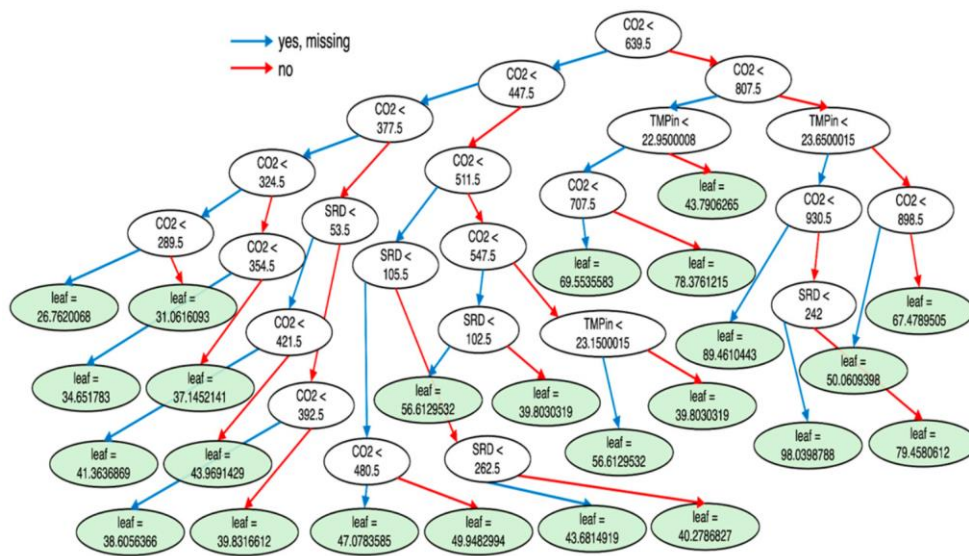


Рисунок 4.8 - Структура дерева рішень моделі XGBoost для прогнозування концентрації CO₂

Таким чином, у ході порівняння моделі за метриками точності було встановлено, що XGBoost перевершила альтернативні підходи, продемонструвавши найнижчі значення RMSE та найвищі показники R² для всіх трьох прогнозованих параметрів. Зокрема, точність прогнозу температури, вологості та концентрації CO₂ склала відповідно 0.9724, 0.9656 і 0.9929, що підтверджує високу ефективність цієї моделі. Значення RPD також засвідчили перевагу XGBoost, особливо у випадку CO₂, де RPD досяг значення 11.8464, суттєво випереджаючи інші моделі. Результати графічного аналізу розсіювання між фактичними та прогнозованими даними свідчать про близькість прогнозів XGBoost до ідеальної відповідності, що особливо

виразно проявляється для CO₂. Подальший аналіз важливості вхідних параметрів виявив ключову роль часу доби та рівня сонячного випромінювання у формуванні мікроклімату. Дослідження структури першого дерева рішень XGBoost показало здатність моделі розпізнавати складні залежності між параметрами внутрішнього середовища, фізіологічними особливостями культур і діями оператора. Це підтверджує доцільність використання XGBoost як інтелектуального інструмента для точного прогнозування та адаптивного керування температурою, вологістю та вмістом CO₂ у великих приміщеннях [12, 14]. У разі виявлення відхилень від оптимальних умов, така система здатна автоматично ініціювати відповідні виконавчі дії для забезпечення стабільного мікроклімату.

Окрім високої точності прогнозування, модель XGBoost забезпечує можливість інтерпретації логіки прийняття рішень, що дає оператору змогу краще розуміти реакцію кліматичних параметрів на дії системи управління та відповідно приймати більш обґрунтовані рішення.

ВИСНОВКИ

У кваліфікаційної роботи було реалізовано повноцінний цикл дослідження, проєктування та моделювання системи підтримки мікроклімату у великих приміщеннях на основі штучної нейронної мережі. Основною метою було створення інтелектуальної моделі, здатної забезпечувати ефективно та гнучке керування внутрішніми кліматичними параметрами — температурою повітря (TMP_{in}), вологістю (RHМ) та концентрацією вуглекислого газу (CO_2) — з урахуванням впливу зовнішніх чинників (TMP_{out}) і динаміки експлуатаційного середовища.

У рамках дослідження проведено ґрунтовний аналіз існуючих підходів до моделювання мікроклімату, зокрема було розглянуто моделі «білого», «чорного» та «сірого» ящика, виявлено їх переваги та недоліки у контексті реального застосування в умовах змінної поведінки системи. Особливу увагу приділено нейромережевим підходам, які забезпечують гнучкість, здатність до навчання та адаптації до нових вхідних даних. Було вивчено типи штучних нейронних мереж, їх архітектури, функції активації, а також проведено порівняння традиційних методів машинного навчання з ансамблевими алгоритмами, зокрема XGBoost.

На основі проведеного аналізу запропоновано нову нейромережеву модель прогнозування стану мікроклімату, яка навчається на базі сенсорних даних та дозволяє здійснювати адаптивне регулювання параметрів середовища в режимі реального часу. Модель інтегрується із гібридною системою збереження інформації, що дозволяє ефективно обробляти як оперативні, так і історичні дані, включаючи великі обсяги вхідних сигналів від сенсорів температури, вологості та CO_2 . Це дало змогу реалізувати предиктивний контроль із мінімальною затримкою й максимальною точністю.

На відміну від класичних моделей, які ґрунтуються на лінійних залежностях або статичних правилах, розроблена система має здатність до безперервного самонавчання та адаптації до змін зовнішніх умов, сезонних коливань, теплових навантажень та нестандартних ситуацій. Її особливістю є також поєднання аналітики в реальному часі з модульною архітектурою, яка дає змогу масштабувати систему на будь-які типи будівель і типові планування приміщень.

Результати моделювання показали високу точність прогнозування параметрів мікроклімату (низькі значення похибок RMSE, MAE), що свідчить про ефективність запропонованого підходу.

Таким чином, розроблена система має наукову новизну в частині використання адаптивної нейронної моделі у зв'язку з гібридним сховищем даних і є прикладом сучасного підходу до автоматизованого регулювання мікроклімату в умовах складної багатофакторної динаміки. Перспективи подальших досліджень полягають у розширенні типів даних, інтеграції з розумними енергосистемами, а також у впровадженні елементів розподіленого інтелекту для колективного керування мікрокліматом у великих інфраструктурних об'єктах.

За темою роботи опубліковано тези доповіді в рамках міжнародної науково-практичної конференції здобувачів вищої світи і молодих вчених "Інформаційні технології у сучасному світі" [15] та підготовлено наукову статтю до наукового журналу «Інформаційні технології та суспільство» [16] (додаток Б).

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Bennis N. et al. Greenhouse Climate Modelling and Robust Control // *Comput. Electron. Agric.* – 2008. – Vol. 61. – P. 96–107. DOI: <https://doi.org/10.1016/j.compag.2007.09.010>.
2. Shamshiri R.R. et al. Review of Optimum Temperature, Humidity, and Vapour Pressure Deficit for Microclimate Evaluation and Control in Greenhouse Cultivation of Tomato // *Int. Agrophys.* – 2018. – Vol. 32. – P. 287–302. DOI: <https://doi.org/10.1515/intag-2017-0038>.
3. Du J., Bansal P., Huang B. Simulation Model of a Greenhouse with a Heat-Pipe Heating System // *Appl. Energy.* – 2012. – Vol. 93. – P. 268–276. DOI: <https://doi.org/10.1016/j.apenergy.2011.12.004>.
4. Taki M. et al. Applied Machine Learning in Greenhouse Simulation; New Application and Analysis // *Inf. Process. Agric.* – 2018. – Vol. 5. – P. 253–268. DOI: <https://doi.org/10.1016/j.inpa.2018.03.003>.
5. Prek M. Exergy analysis of thermal comfort / M. Prek. // *International Journal of Exergy.* – 2004. – V. 1. – P. 303–315.
6. Collotta M., Messineo A., Nicolosi G. and Pau G. A Dynamic Fuzzy Controller to Meet Thermal Comfort by Using Neural Network Forecasted Parameters as the Input // *Energies.* – 2014. – № 7. – P. 4727–4756; doi:10.3390/en7084727.
7. Kramer R., J. van Schijndel, Schellen H. Simplified thermal and hygric building models: A literature review. *Frontiers of Architectural Research.* – 2012. – № 1(4). – P. 318–325.
8. Perera D.W.U., Pfeiffer C. F., Skeie N.-O. Control of temperature and energy consumption in buildings – A review // *International journal of Energy and Environment.* – 2014. – Vol. 5, Issue 4. – P. 471–484.
9. Rabadi N. J., Developing a Software to Predict Thermal Comfort of Humans at Work // *Jordan Journal of Mechanical and Industrial Engineering.* –

2011. – Vol. 5. – № 4. – P. 359–368.

10. Amjad M. et al. Prediction of Pile Bearing Capacity Using XGBoost Algorithm: Modeling and Performance Evaluation // *Appl. Sci.* – 2022. – Vol. 12. – Art. 2126. DOI: <https://doi.org/10.3390/app12042126>.

11. Chen T., Guestrin C. XGBoost: A Scalable Tree Boosting System // *Proc. 22nd ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining.* – San Francisco, CA, USA, 13–17 Aug. 2016. – P. 785–794. DOI: <https://doi.org/10.1145/2939672.2939785>.

12. Mariadass D.A.-L. et al. Extreme Gradient Boosting (XGBoost) Regressor and Shapley Additive Explanation for Crop Yield Prediction in Agriculture // *Proc. 12th Int. Conf. on Computer and Knowledge Engineering (ICCCKE).* – Mashhad, Iran, 2022. – P. 219–224.

13. Omid M. A Computer-Based Monitoring System to Maintain Optimum Air Temperature and Relative Humidity in Greenhouses // *Int. J. Agric. Biol.* – 2004. – Vol. 6. – P. 869–873.

14. Agarap A.F. Deep Learning Using Rectified Linear Units (ReLU) // *arXiv preprint.* – 2019. – arXiv:1803.08375.

15. Піскаръов О.М., Абрамович Д.О., Плу́гін В.В. Модель інтелектуальної підтримки мікроклімату у великих приміщеннях із використанням нейромережових підходів та гібридних сховищ інформації // *Збірка матеріалів міжнародної науково-практичної конференції здобувачів вищої світи і молодих вчених "Інформаційні технології у сучасному світі".* - Харків: ДБТУ. 2025. – С. 154-156.

16. Піскаръов О., Абрамович Д., Плу́гін В. Застосування нейромереж та гібридних сховищ при створенні комп'ютерних систем. Інформаційні технології та суспільство. Науковий журнал. Випуск 2 (17). - Київ: МАУП, - 2025. - С. (прийнята до друку).