Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет Інформаційно-аналітичних технологій та менеджменту
(повна назва)
Кафедра Інформатики
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА
Пояснювальна записка

рівень вищої освіти другий (магістерський)

ДОСЛІДЖЕННЯ ТА ПОРІВНЯННЯ МЕТОДІВ НАВЧАННЯ ДЛЯ ПРОНОЗУВАННЯ ПОГОДНИХ УМОВ НА ОСНОВІ МЕТЕОРОЛОГІЧНИХ ДАНИХ
(тема)

Виконав:
студент 2 курсу, групи ІНФм-23-1
Меженна І.Д.
(прізвище, ініціали)

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

Тип програми освітньо-професійна

Освітня програма Інформатика
(повна назва освітньої програми)

Керівник доц. Творошенко І.С.
(посада, прізвище, ініціали)

Допускається до захисту

Зав. кафедри _____ Кобилін О.А.
(підпис) (прізвище, ініціали)

2025 р.

Харківський національний університет радіоелектроніки

Факультет Інформаційно-аналітичних технологій та менеджменту
(повна назва)

Кафедра Інформатики
(повна назва)

Рівень вищої освіти другий (магістерський)

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

Тип програми освітньо-професійна

Освітня програма Інформатика
(повна назва освітньої програми)

ЗАТВЕРДЖУЮ:
Зав. кафедри _____
(підпис)
«____»_____ 2025 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

студентові Меженній Ірині Дмитрівні
(прізвище, ім'я, по батькові)

1. Тема роботи Дослідження та порівняння методів навчання для прогнозування погодних умов на основі метеорологічних даних

затверджена наказом по університету від 25 листопада 2024 року № 1246Ст

2. Термін подання студентом роботи до екзаменаційної комісії 23 грудня 2024 р.

3. Вихідні дані до роботи математичні моделі перетворювання зображень, перелік використовуваних програмних засобів: теоретичні відомості про методи сегментації текстурних зображень.

4. Перелік питань, що потрібно опрацювати в роботі

1. Огляд існуючих методів навчання для задачі прогнозування.
2. Особливості вибраних методів навчання для задачі прогнозування.
3. Формування методики для прогнозування погодних умов на основі метеорологічних даних.
4. Програмна реалізація вибраних методів.
5. Дослідження та критеріальний порівняльний аналіз обраних методів навчання.

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (п.5 включається до завдання за рішенням випускової кафедри) актуальність проблеми, об'єкт, мета та завдання дослідження, вихідні дані для дослідження, етапи виконання дослідження, аналіз отриманих результатів, перспективи подальших досліджень, висновки, апробація проведених досліджень.

6. Консультанти розділів роботи (п.6 включається до завдання 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

№ з/п	Назва етапів роботи	Терміни виконання етапів роботи	Примітка
1	Отримання завдання на кваліфікаційну роботу	25.11.2024	
2	Аналіз завдання, підбір літератури	25.11.24-28.11.24	
3	Аналіз літератури з досліджуваної проблеми	28.11.24-29.11.24	
4	Аналіз методів машинного навчання	29.11.24-30.11.24	
5	Виконання етапів дослідження	02.12.24-05.12.24	
6	Порівняльний аналіз отриманих результатів	06.12.24-10.12.24	
7	Оформлення пояснювальної записки	10.12.24-13.12.24	
8	Перевірка на плагіат	16.12.2024	
9	Рецензування	20.12.2024	
10	Підготовка презентації та доповіді	23.12.2024	
11	Занесення роботи в електронний архів	01.01.2025	
12	Попередній захист кваліфікаційної роботи	02.01.2025	

Дата видачі завдання 25 листопада 2024 р.

Студент _____
(підпис)

Керівник роботи _____ доц. Творошенко І.С.
(підпис) (посада, прізвище, ініціали)

РЕФЕРАТ/ABSTRACT

Пояснювальна записка до кваліфікаційної роботи: 80 с., 7 табл., 25 рис., 45 джерел.

МАШИННЕ НАВЧАННЯ, ПРОГНОЗУВАННЯ ПОГОДНИХ УМОВ, МЕТЕОРОЛОГІЧНІ ДАНІ, ЛІНІЙНА РЕГРЕСІЯ, ДЕРЕВА РІШЕНЬ, РЕКУРЕНТНІ НЕЙРОННІ МЕРЕЖІ, ЧАСОВА ЗАЛЕЖНІСТЬ, ТОЧНІСТЬ ПРОНОЗУВАННЯ, ПОРІВНЯЛЬНИЙ АНАЛІЗ, ОЦІНКА МОДЕЛЕЙ.

Об’єктом дослідження є якість прогнозу погодних умов на основі метеорологічних даних.

Метою дослідження є проведення порівняльного аналізу методів машинного навчання, що використовуються для задач прогнозування.

Проведено аналіз ефективності трьох різних підходів до прогнозування. Розглянуто їх здатність до моделювання складних патернів у даних та адаптації до змінюваних умов. Розроблено алгоритм прогнозування погоди на основі метеорологічних параметрів, таких як температура, вологість та атмосферний тиск. Використано введення штучного шуму в дані під час тренування для перевірки стійкості моделей до змін вхідних даних.

У результаті дослідження здійснена програмна реалізація обраних методів, виділено їх особливості для виконання задачі прогнозування погоди.

MACHINE LEARNING, WEATHER FORECASTING, METEOROLOGICAL DATA, LINEAR REGRESSION, DECISION TREES, RECURRENT NEURAL NETWORKS, TIME DEPENDENCE, PREDICTION ACCURACY, COMPARATIVE ANALYSIS, MODEL EVALUATION.

The object of the research is the quality of weather forecasting based on meteorological data.

The purpose of the research is to conduct a comparative analysis of machine learning methods used for forecasting tasks.

The efficiency of three different forecasting approaches is analyzed. Their ability to model complex patterns in data and adapt to changing conditions is considered. An algorithm for weather forecasting based on meteorological parameters such as temperature, humidity, and atmospheric pressure is developed. The introduction of artificial noise into the data during training was used to test the stability of the models to changes in input data.

As a result of the research, the software implementation of the selected methods was carried out, and their features for the weather forecasting task were highlighted.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

RNN – Recurrent Neural Network (рекурентна нейронна мережа)

MSE – Mean Squared Error (середня квадратична помилка)

MAE – Mean Absolute Error (середня абсолютна помилка)

ML – Machine Learning (машинне навчання)

ШНМ – штучна нейронна мережа

BPTT – Backpropagation Through Time (зворотне поширення в часі)

LSTM – Long Short-Term Memory (довга короткочасна пам’ять)

ReLU – Rectified Linear Unit (виправлений блок лінійної активації)

GRU – Gated Recurrent Unit (одинична рекурентна мережа з воротами)

с – секунда

ВСТУП

Машинне навчання – це галузь штучного інтелекту, яка фокусується на розробці алгоритмів і статистичних моделей, що дозволяють комп’ютерам навчатися, спираючись на дані, робити прогнози або приймати рішення [1].

Прогнозування, що визначається як випереджальне відбиття майбутнього, спрямоване на визначення тенденцій динаміки конкретного об’єкта або події на основі ретроспективних даних, тобто аналізу стану колись і тепер. Особливе значення мають задачі передбачення та прогнозування часових рядів.

Основне завдання машинного навчання в задачах прогнозування полягає в розробці моделей, які можуть виявляти закономірності в історичних даних і на основі цього передбачати майбутній розвиток подій. Для прогнозування, зокрема часових рядів, ключовими є алгоритми, що здатні обробляти послідовні дані, враховуючи тренди та сезонні коливання. Такі алгоритми широко застосовуються в аналізі фінансових ринків, погодних умов, поведінки клієнтів тощо.

Актуальність дослідження полягає у необхідності зрозуміти особливості різних методів, що використовуються в цій сфері. Аналіз різних підходів до прогнозування дозволить не лише оцінити їх ефективність, але й вибрати найбільш підходящі інструменти для розв’язання конкретних задач. Оскільки прогнози погоди впливають на різні сфери діяльності, від сільського господарства до енергетики, результати дослідження можуть стати основою для покращення точності прогнозів і оптимізації рішень, що приймаються на їх основі. Це, в свою чергу, забезпечить більш ефективне реагування на зміни погоди та підвищить готовність до можливих викликів, які постають у сучасному світі, зокрема в умовах змін клімату.

1 АНАЛІЗ ІСНУЮЧИХ МЕТОДІВ НАВЧАННЯ ДЛЯ ЗАДАЧІ ПРОГНОЗУВАННЯ

1.1 Аналіз сучасних методів навчання для задачі прогнозування

Машинне навчання – це галузь комп'ютерних наук, яка фокусується на використанні даних і алгоритмів, що дозволяють штучному інтелекту імітувати спосіб навчання людини, поступово підвищуючи точність відповіді. Загалом, машинне навчання використовується для задач прогнозування або класифікації. Тобто, на основі деяких вхідних даних, які можуть бути маркованими або немаркованими, розроблений алгоритм має зробити оцінку щодо закономірностей, які притаманні цим даним (рис. 1.1).

Рисунок 1.1 – Типи машинного навчання

Конкретно для задач прогнозування, частіше використовуються такі сучасні методи навчання:

– лінійна регресія;

– дерева рішень;

– ансамблеві методи;

– метод опорних векторів;

– рекурентні та згорткові нейронні мережі;

– гібридні моделі;

– автокодувальники.

Ці алгоритми широко застосовуються в різних галузях: від метеорології до медицини. Розглянемо детальніше кожен з них.

Лінійна регресія – це тип керованого алгоритму машинного навчання, який обчислює лінійний зв’язок між залежною змінною та однією або кількома незалежними ознаками шляхом підбору лінійного рівняння до даних спостережень. Коли є лише одна незалежна ознака, це називається простою лінійною регресією, а коли їх більше однієї, то це називається множинною лінійною регресією. Основним завданням лінійної регресії є знаходження лінії, яка найкраще описує залежність між змінними, що передбачає мінімізацію різниці між прогнозованими та фактичними значеннями (рис. 1.2 [2]). Лінія найкращої відповідності характеризується найменшою можливою помилкою.

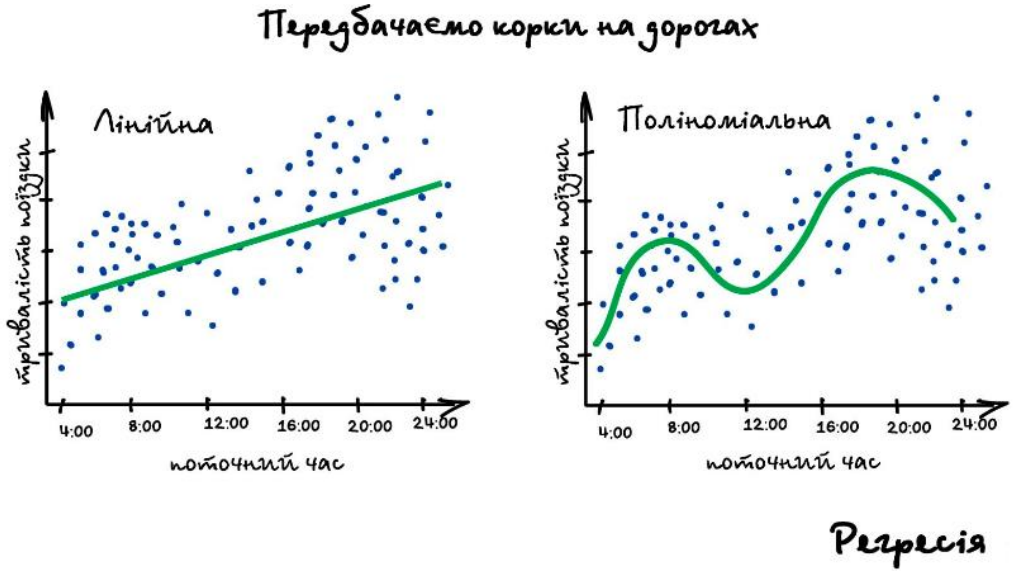


Рисунок 1.2 – Два основні типи регресії

Помітною перевагою даного методу є можливість інтерпретації. Рівняння моделі містить чіткі коефіцієнти, які пояснюють вплив кожної

незалежної змінної на залежну, що сприяє глибшому розумінню основної динаміки. Ще одна з головних переваг – простота. Оскільки лінійна регресія прозора та легка в реалізації, вона є не лише інструментом, а основою для різних складних моделей. Такі методи, як регуляризація та машина опорних векторів, базуються на лінійній регресії, розширюючи її корисність.

Дерево рішень у машинному навчанні – це універсальний алгоритм, який можна інтерпретувати та використовувати для прогнозного моделювання (рис. 1.3 [2]). Він структурує рішення на основі вхідних даних, що робить його придатним як для завдань класифікації, так і для регресії.

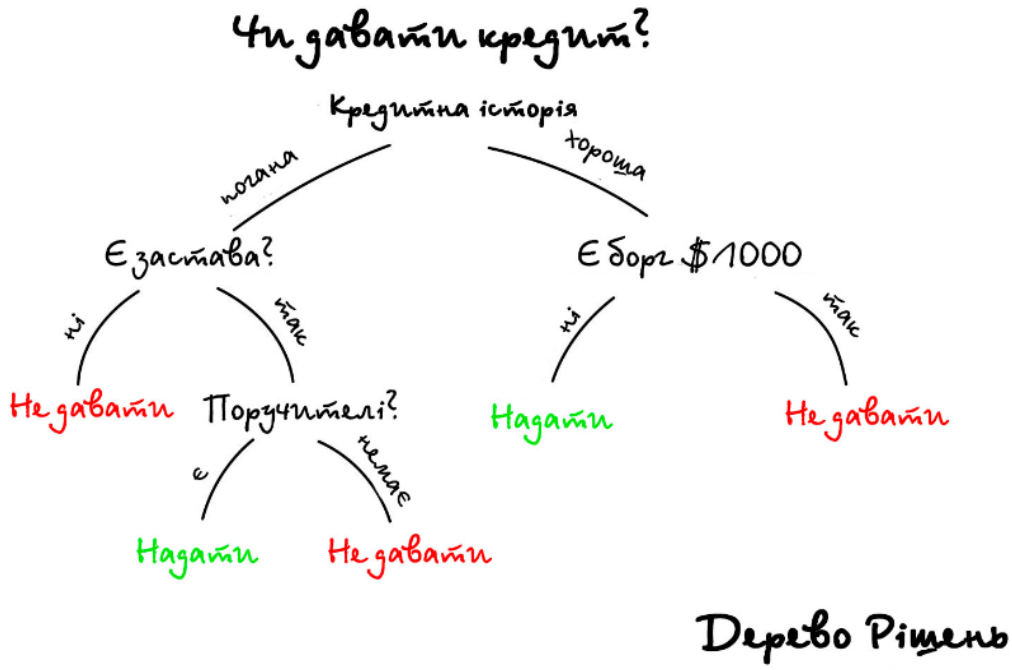


Рисунок 1.3 – Приклад дерева рішень

Як і попередній, алгоритм дерева рішень відноситься до категорії керованого навчання. Він зазвичай використовується для моделювання та прогнозування результатів на основі вхідних даних. Це деревоподібна структура, де кожен внутрішній вузол тестує атрибут, кожна гілка відповідає значенню атрибуту, а кожен листовий вузол представляє остаточне рішення або прогноз. Процес формування дерева рішень передбачає рекурсивне розбиття даних на основі значень різних атрибутів.

Алгоритм обирає найкращий атрибут для розділення даних у кожному внутрішньому вузлі на основі певних критеріїв, таких як інформаційний виграш або коефіцієнт Джині. Процес розбиття триває доти, доки не буде досягнуто критерію зупинки, наприклад, досягнення максимальної глибини або мінімальної кількості екземплярів у листовому вузлі.

Дерева рішень універсальні в моделюванні складних процесів прийняття рішень завдяки своїй інтерпретованості. Завдяки їхній ієрархічній структурі стає можливим врахування різноманітних причин та результатів. Дерева рішень надають зрозуміле уявлення про логіку прийняття рішень, добре працюють як з числовими, так і з категоріальними даними, і можуть легко адаптуватися до різноманітних наборів завдяки можливості автономного вибору ознак.

Наступний метод є одним із найпотужніших серед розглянутих – це ансамблеве навчання. Ансамбль вивчення – це використання багатьох моделей задля підвищення надійності та точності прогнозів (рис. 1.4). Тобто, коли потрібно вирішити складну обчислювальну задачу і жоден алгоритм не підходить ідеально, можна використати ансамбль – поєднання відразу декількох алгоритмів, які навчаються одночасно і виправляють помилки один одного.

Рисунок 1.4 – Підходи для збору ансамблів

На сьогоднішній день саме вони дають найточніші результати, тому саме їх найчастіше використовують усі великі компанії, для яких важлива швидка обробка великої кількості даних.

Однією з найпоширеніших методик ансамблю є пакетна вибірка, яка використовує бутстрап-вибірку для створення декількох наборів даних з вихідних даних і навчання моделі на кожному наборі даних. Інший метод – бустінг, який навчає моделі послідовно, кожна з яких фокусується на помилках попередніх моделей. Випадкові ліси – це популярний ансамблевий метод, який використовує дерева рішень як базові навчальні моделі і комбінує їхні прогнози для отримання остаточного.

Ансамблеві методи є ефективними, оскільки вони зменшують надмірне пристосування та покращують узагальнення, що призводить до більш надійних результатів. Вибір правильного методу ансамблю залежить від характеру даних, конкретної проблеми, яку намагаються вирішити, і доступних обчислювальних ресурсів. Часто для досягнення найкращих результатів потрібно експериментувати і вносити зміни.

Машина опорних векторів – це теж керований алгоритм машинного навчання, який класифікує дані, знаходячи оптимальну лінію або гіперплощину, що максимізує відстань між кожним класом у N-вимірному просторі. Цей метод широко використовуються в задачах класифікації. Він розрізняє два класи, знаходячи оптимальну гіперплощину, яка максимізує відстань між найближчими точками даних протилежних класів (рис. 1.5 [3]). Кількість ознак у вхідних даних визначає, чи є гіперплощина лінією у двовимірному просторі або площиною в n-вимірному просторі. Оскільки для розрізнення класів можна знайти кілька гіперплощин, максимізація відстані між точками дозволяє алгоритму знайти найкращу межу між класами. Це, в свою чергу, дозволяє йому добре узагальнювати нові дані і робити точні класифікаційні прогнози.

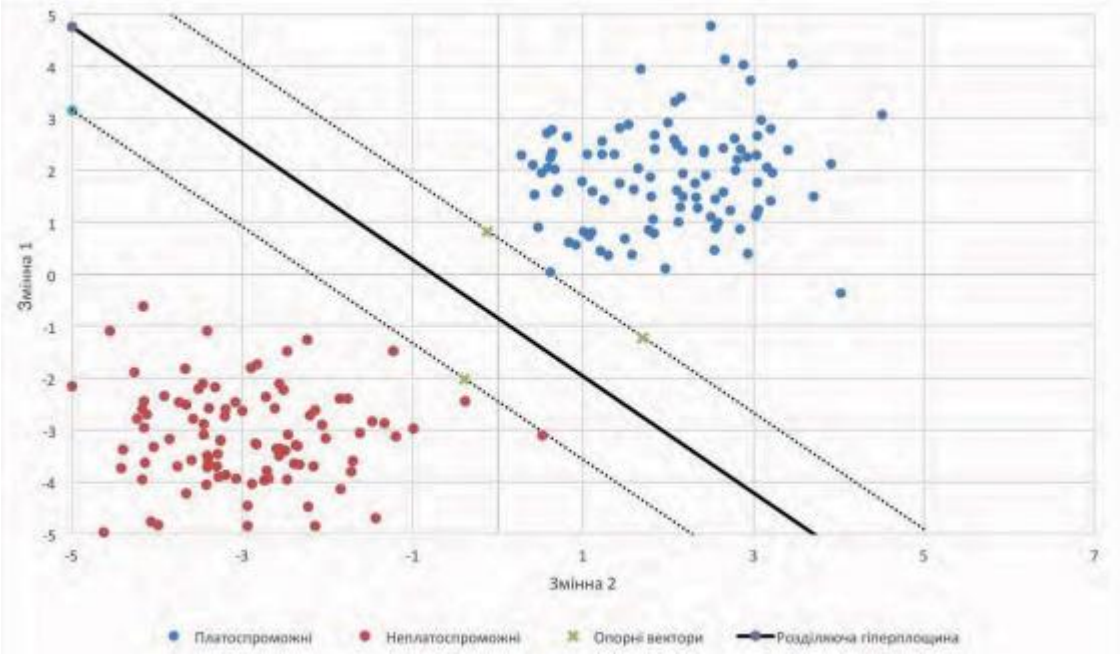


Рисунок 1.5 – Принцип методу опорних векторів

Лінії, які прилягають до оптимальної гіперплощини, називаються опорними векторами, оскільки ці вектори проходять через точки даних, які визначають максимальну відстань між ними.

Алгоритм широко використовується, оскільки може обробляти як лінійні, так і нелінійні завдання класифікації. Однак, коли дані не піддаються лінійному розділенню, використовуються функції ядра для перетворення простору даних у вимірному просторі, щоб уможливити лінійне розділення. Таке застосування ядерних функцій може бути відоме як «трюк ядра», і вибір ядерної функції, наприклад, лінійні ядра, поліноміальні ядра, ядра з радіально-базисною функцією або сигмоїдні ядра, залежить від характеристик даних і конкретного випадку використання.

Наступний інструмент, який широко використовується для різних задач, включаючи прогнозування, – це нейронна мережа. Згорткові нейронні мережі призначені для роботи з просторовими даними, такими як зображення, але також можуть бути адаптовані для аналізу часових рядів. Вони використовують згорткові шари, які здатні автоматично виявляти значущі ознаки у великих масивах даних, і зменшувати розмірність, що дозволяє зосередитися на найважливішій інформації (рис. 1.6 [2]).

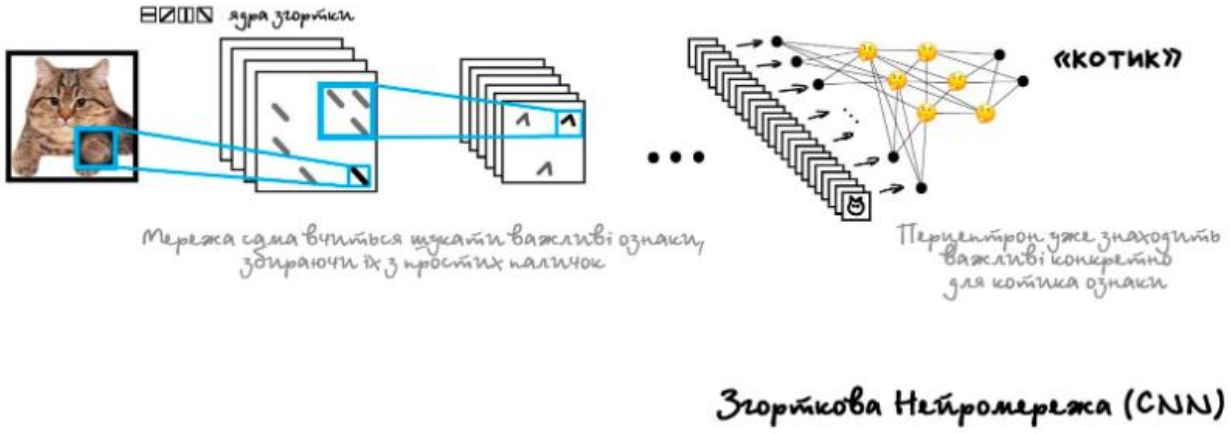


Рисунок 1.6 – Принцип роботи згорткової нейромережі

Це робить згорткові мережі ефективними для задач, де важливо розпізнавати локальні закономірності, наприклад, виявлення патернів у послідовностях даних або аналіз просторово-часових зображень, таких як метеорологічні карти.

Рекурентні нейронні мережі (рис. 1.7 [2]) створені для обробки послідовних даних. Завдяки здатності зберігати інформацію про попередні елементи у послідовності, в своїх прогнозах вони враховують залежності між даними на різних часових проміжках. Це робить їх корисними для прогнозування часових рядів, таких як фінансові показники, дані про температуру чи попит на ресурси.

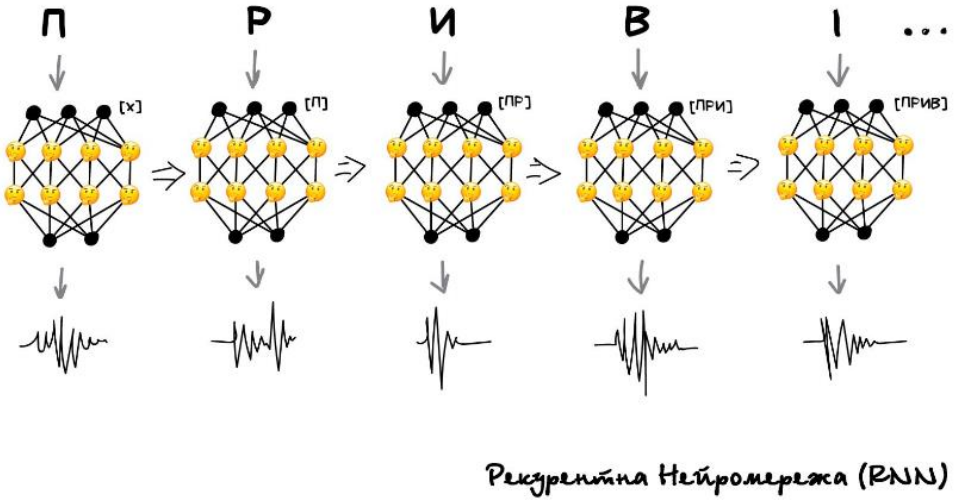


Рисунок 1.7 – Спрощений приклад роботи рекурентної нейромережі

Згорткові мережі можуть використовуватися для витягування корисних ознак із даних, таких як час, географічні показники або зображення, тоді як рекурентні мережі можна застосувати для побудови моделей, які врахують минулі значення для більш точного прогнозу. Тобто, згорткові нейромережі будуть використані як перший крок для вилучення важливих ознак, які потім проаналізуються рекурентними для довгострокових прогнозів. Таким чином, можна отримати гібридну модель – ще один з методів навчання.

Гібридні моделі об’єднують різні види нейронних мереж, щоб скористатися їхніми сильними сторонами та забезпечити більш точне та комплексне прогнозування.

Переваги гібридних моделей полягають у їхній здатності одночасно враховувати просторові та часові характеристики даних, що робить їх ідеальними для роботи з комплексними та багатовимірними даними. Вони ефективні там, де класичні моделі не можуть розпізнати складні взаємозв’язки, і часто перевершують традиційні підходи завдяки своїй здатності адаптуватися до різних типів вхідної інформації.

Таким чином, сучасні методи машинного навчання охоплюють різноманітні підходи, кожен з яких має свої сильні та слабкі сторони залежно від специфіки завдань. Комплексне розуміння їхніх особливостей дає змогу більш обґрунтовано підбирати інструменти для різних задач прогнозування. З метою отримання більш точних прогнозів, необхідно враховувати не лише загальні особливості алгоритмів, але й складність кліматичних моделей та змінні, які впливають на результати.

Погода – це атмосферні умови в певному місці в певний час.

Існує шість основних показників, таких як:

– температура;

– атмосферний тиск;

– вітер;

– вологість;

– опади;

– хмарність.

Разом ці компоненти описують погоду в будь-який момент часу. Вивчення погоди називається метеорологією і включає в себе спостереження, вимірювання та аналіз даних для прогнозування майбутніх умов. Прогнози погоди можуть укладатись як для загального користування, без певної специфіки, так і спеціалізовані, наприклад, авіаційні тощо.

Температура вимірюється термометром і показує, наскільки гарячою або холодною є атмосфера. Метеорологи повідомляють про температуру двома способами: за Цельсієм (C) і за Фаренгейтом (F). У Сполучених Штатах Америки використовується система Фаренгейта, в інших частинах світу – система Цельсія. Майже всі вчені вимірюють температуру за шкалою Цельсія.

Атмосферний тиск – тиск, з яким атмосфера Землі діє на земну поверхню і всі тіла, що на ній розташовані. Зміни атмосферного тиску сигналізують про зміну погоди. Високий тиск зазвичай приносить прохолодну температуру і ясне небо, низький – теплішу погоду, шторми та дощі. Метеорологи виражають атмосферний тиск в одиниці виміру, яка називається атмосферою. Атмосфери вимірюються в мілібарах або дюймах ртутного стовпчика. Середній атмосферний тиск на рівні моря становить близько однієї атмосфери (близько 1013 мілібар або 29,9 дюйма). Середній тиск у системі низького тиску, або циклоні, становить близько 995 мілібар (29,4 дюйма). Типова система високого тиску, або антициклон, зазвичай досягає 1030 мілібар (30,4 дюйма). Слово «циклон» відноситься до повітря, яке обертається по колу, як колесо.

Вітер – це рух повітря. Вітер утворюється через різницю в температурі та атмосферному тиску між сусідніми регіонами. Вітри, як правило, дмуть з областей високого тиску, де холодніше, в області низького тиску, де тепліше.

У верхніх шарах атмосфери сильні, швидкі вітри, які називаються струминними потоками, виникають на висоті від 8 до 15 кілометрів (від 5 до 9 миль) над Землею. Зазвичай вони дмуть зі швидкістю від 129 до 225 кілометрів на годину (80-140 миль на годину), але можуть досягати

понад 443 кілометри на годину (275 миль на годину). Ці вітри верхніх шарів атмосфери допомагають підштовхувати погодні системи по всьому світу.

Вологість – це кількість водяної пари в повітрі. Водяна пара – це газ в атмосфері, який допомагає утворювати хмари, дощ або сніг. Вологість зазвичай виражається як відносна вологість, або відсоток від максимальної кількості води, яку може утримувати повітря при певній температурі. Холодне повітря містить менше води, ніж тепле. При відносній вологості 100 відсотків повітря вважається насиченим, тобто воно не може утримувати більше водяної пари. Надлишок водяної пари випадає у вигляді опадів. Хмари і опади виникають, коли повітря охолоджується нижче точки насичення. Зазвичай це відбувається, коли тепле вологе повітря охолоджується, коли воно піднімається вгору.

Хмари бувають різних форм. Не всі вони дають опади. Наприклад, купчасті перисті хмари, як правило, сигналізують про м’яку погоду. Інші види хмар можуть принести дощ або сніг. Німбостратові хмари, схожі на ковдру, приносять стійкі, тривалі опади. Величезні купчасто-дощові хмари, або грозові хмари, викликають сильні зливи. Купчасто-дощові хмари також можуть викликати грози і торнадо. Хмари можуть впливати на кількість сонячного світла, що досягає поверхні Землі. У хмарні дні прохолодніше, ніж у ясні, тому що хмари затримують більшу частину сонячного випромінювання, що досягає поверхні Землі. Вночі все навпаки – хмари діють як ковдра, зберігаючи тепло на Землі.

Традиційно прогнозування погоди базується на фізичних моделях, які використовують рівняння динаміки атмосфери, такі як рівняння Нав’є-Стокса. Однак ці моделі є складними у виконанні та потребують великих обчислювальних потужностей. Через це все частіше для прогнозування починають використовувати методи машинного навчання, які дозволяють швидше обробляти великі обсяги даних і знаходити приховані залежності між параметрами. Машинне навчання допомагає не тільки підвищити

швидкість, але й покращити точність прогнозів, особливо завдяки здатності моделей виявляти нелінійні залежності в даних.

Отже, під час роботи буде досліджено можливості та ефективність різних алгоритмів машинного навчання для підвищення точності прогнозів. Особливу увагу буде приділено здатності моделей виявляти нелінійні залежності в даних, що є важливим для обробки складних погодних патернів. Також будуть розглянуті методи оптимізації моделей, включаючи налаштування параметрів та обробку великого обсягу даних для досягнення більш точних результатів. Це дослідження допоможе обґрунтувати вибір найефективніших підходів для прогнозування погодних умов та оптимізувати процес аналізу кліматичних даних.

Серед розглянутих методів для порівняння було обрано наступні:

– лінійна регресія;

– дерева рішень;

– рекурентна нейронна мережа.

Для оцінки роботи буде використано наступні ключові критерії:

– точність прогнозів;

– швидкість навчання та прогнозування;

– стабільність і робастність (стійкість моделей до змін у вхідних даних, шуму та пропущених значень);

– здатність моделей надавати інтерпретовані результати.

З метою підвищення точності буде досліджено використання механізмів уваги та різних підходів до налаштування гіперпараметрів. Передбачається введення штучного шуму у дані під час тренування для оцінки стійкості моделей.

Дослідження буде проводитися на вибірці з 100 записів метеорологічних даних, кожен з яких містить не менше п’яти параметрів, що дозволить оцінити здатність моделей до прогнозування в умовах невеликої кількості даних.

1.2 Аналіз джерел щодо апробації результатів застосування методів навчання для прогнозування погодних умов

З розвитком технологій обробки даних та збільшенням обсягів метеорологічних спостережень зростає і потреба в ефективних алгоритмах, що можуть точно й оперативно передбачати погоду. Оскільки традиційні статистичні методи мають обмежену здатність враховувати складні та нелінійні зв’язки у кліматичних даних, використання алгоритмів машинного навчання відкриває нові перспективи для точніших прогнозів. У світовій практиці наукових досліджень апробація різних моделей, таких як нейронні мережі, дерева рішень і методи ансамблю, показує, що вони здатні успішно обробляти складні часові ряди і виявляти патерни, які не завжди очевидні у традиційних підходах.

У джерелі [4] досліджено важливість значення середньострокового прогнозування погоди для прийняття рішень у багатьох соціальних та економічних сферах. Основною метою дослідження є знаходження шляхів покращення традиційних чисельних методів прогнозування погоди, які використовують збільшені обчислювальні ресурси для підвищення точності прогнозу, але не використовують безпосередньо історичні погодні дані для покращення базової моделі. Під час аналізу цієї роботи було досліджено GraphCast – метод на основі машинного навчання, який навчається безпосередньо на даних повторного аналізу. Він прогнозує сотні погодних змінних на наступні 10 днів по всьому світу менш, ніж за 1 хвилину. Було визначено, що GraphCast значно перевершує найточніші оперативні детерміновані системи на 90 відсотків з 1380 верифікаційних цілей, а його прогнози підтримують краще прогнозування важких подій, в тому числі відстеження тропічних циклонів, атмосферних річок і екстремальних температур.

У джерелі [5] досліджено потенціал моделювання на основі машинного навчання для прогнозування погоди. У цій статті порівнюються прогнози, отримані за допомогою машинного навчання, зі стандартними прогнозами на основі традиційного числового прогнозування у контексті, подібному до операційного, з однаковими початковими умовами. Зосереджуючись на детермінованих прогнозах, застосовуються поширені інструменти верифікації прогнозів, щоб оцінити, наскільки прогноз на основі даних, отриманий за допомогою однієї з нещодавно розроблених моделей PanguWeather, відповідає якості та атрибутам прогнозу однієї з провідних глобальних систем ECMWF IFS. Під час аналізу цієї роботи було визначено основні поточні недоліки прогнозів на основі машинного навчання: надмірно згладжені прогнози, збільшення похибки з часом прогнозування та низька ефективність у прогнозуванні інтенсивності тропічних циклонів.

У джерелі [6] досліджено оптимізацію обчислювальних витрат шляхом імітації прогнозів за допомогою моделей глибокої генеративної дифузії, отриманих із історичних даних. Основною метою дослідження є отримання моделей, що мають високу масштабованість щодо високопродуктивних обчислювальних прискорювачів і можуть створювати тисячі реалістичних прогнозів погоди за низькі витрати.

У джерелі [7] досліджено еволюцію передових моделей прогнозування штучного інтелекту та на основі виявлених спільних рис пропонується «Три великі правила» для вимірювання їхнього розвитку. Основною метою дослідження є обговорення потенціалу штучного інтелекту в революції в чисельному прогнозуванні погоди та коротке окреслення основних його причин. Під час аналізу цієї роботи було визнано високу точність, обчислювальну ефективність і легкість розгортання великих моделей прогнозування на основі штучного інтелекту.

У джерелі [8] досліджується тема прогнозування погоди за допомогою машинного навчання. Основною метою дослідження є ретельне вивчення кількох моделей машинного навчання, таких як *K*-Nearest Neighbors, машина

опорних векторів, підсилення градієнта, XGBOOST, логістична регресія та клас випадкового лісу для прогнозів погоди. Під час аналізу цієї роботи було обрано логістичну регресію як один із методів машинного навчання, які будуть використовуватися у практичних розділах запропонованого наукового дослідження.

У джерелі [9] досліджується модель загальної циркуляції, яка поєднує диференційований розв'язувач динаміки атмосфери з компонентами машинного навчання та показує можливість створювати прогнози детермінованої погоди нарівні з найкращими методами. Основною метою дослідження є демонстрація можливості покращення масштабного фізичного моделювання, яке є важливим для прогнозування погоди, з глибоким машинним навчанням.

У джерелі [10] досліджується покращення оцінки невизначеності, пов'язаної з прогнозами опадів у реальному часі, за допомогою методу машинного навчання. Основною метою дослідження є аналіз результатів прогнозів моделі, яка була навчена наземними даними Confederación Hidrográfica del Ebro. Під час аналізу цієї роботи було визначено, що ансамблеві методи на основі дерева рішень мають найменшу помилку узагальнення. Вивчені моделі прогнозування мають точність понад 90%, а середня квадратична помилка має подібні результати порівняно з результатами, отриманими за допомогою наземних даних. Було обрано дерева рішень як один із методів машинного навчання, які будуть використовуватися у практичних розділах запропонованого наукового дослідження.

У джерелі [11] досліджується використання передових методів інтелектуального аналізу даних для аналізу та прогнозування погодних умов із використанням обширних історичних даних про погоду. Основна мета дослідження – визначити найефективніші моделі машинного навчання для цього завдання шляхом порівняння різних алгоритмів. Дані про погоду, зібрані протягом десяти років із десяти різних наборів даних, були

проаналізовані за допомогою різноманітного набору моделей машинного навчання, зокрема на основі правил (OneR, Decision Table, JRIP, Ridor), на основі дерев (J48, LMT, Random Forest, CART), і функціональних підходів (MLR, MLP, SVM, LogitBoost, SMO, ANN). Кожну модель оцінювали на основі кількох показників продуктивності: точності, запам'ятовування, точності, F-міри, частоти істинних позитивних результатів (TPR), частоти помилкових позитивних результатів (FPR), частоти справжніх негативних результатів (TNR) і частоти помилкових негативних результатів (FNR). Під час аналізу цієї роботи було обрано метрики для порівняння моделей, які будуть використовуватися у практичних розділах запропонованого наукового дослідження.

У джерелі [12] досліджується ClimateLearn, бібліотека PyTorch з відкритим вихідним кодом, яка значно спрощує навчання та оцінку моделей машинного навчання для кліматичної науки на основі метеорологічних даних. Основною метою дослідження є доповнення функцій обширною документацією, посібниками для внесення вкладів і короткими посібниками, задля розширення доступу для спільноти.

У джерелі [13] досліджується доцільність навчання різних підходів машинного навчання на великому ансамблі кліматичних моделей. Основною метою дослідження є демонстрація того, що моделі прогнозування погоди на основі машинного навчання здатні конкурувати з існуючими динамічними моделями North American Multi Model Ensemble або перевершувати їх. Під час аналізу цієї роботи було обрано нейронну мережу як один з методів машинного навчання, які будуть використовуватися у практичних розділах запропонованого наукового дослідження.

Зважаючи на значення кожного підходу для досягнення точності прогнозування, було досліджено апробацію результатів у класичних літературних джерелах. Проте в сучасному світі, де обмін інформацією та науковими здобутками набуває безпрецедентної швидкості, необхідно звернути увагу й на електронні ресурси.

На сайті «Синоптик» [14] прогноз погоди представлений у зручному і зрозумілому форматі, що дозволяє швидко оцінити очікувані погодні умови на декілька днів або навіть тиждень вперед (рис. 1.8 [14]). Головний екран показує короткий прогноз із зазначенням температури та стану неба (ясно, хмарно, дощ тощо) для кожного дня, а для зручності навігації використано піктограми, що відображають погодні умови візуально.

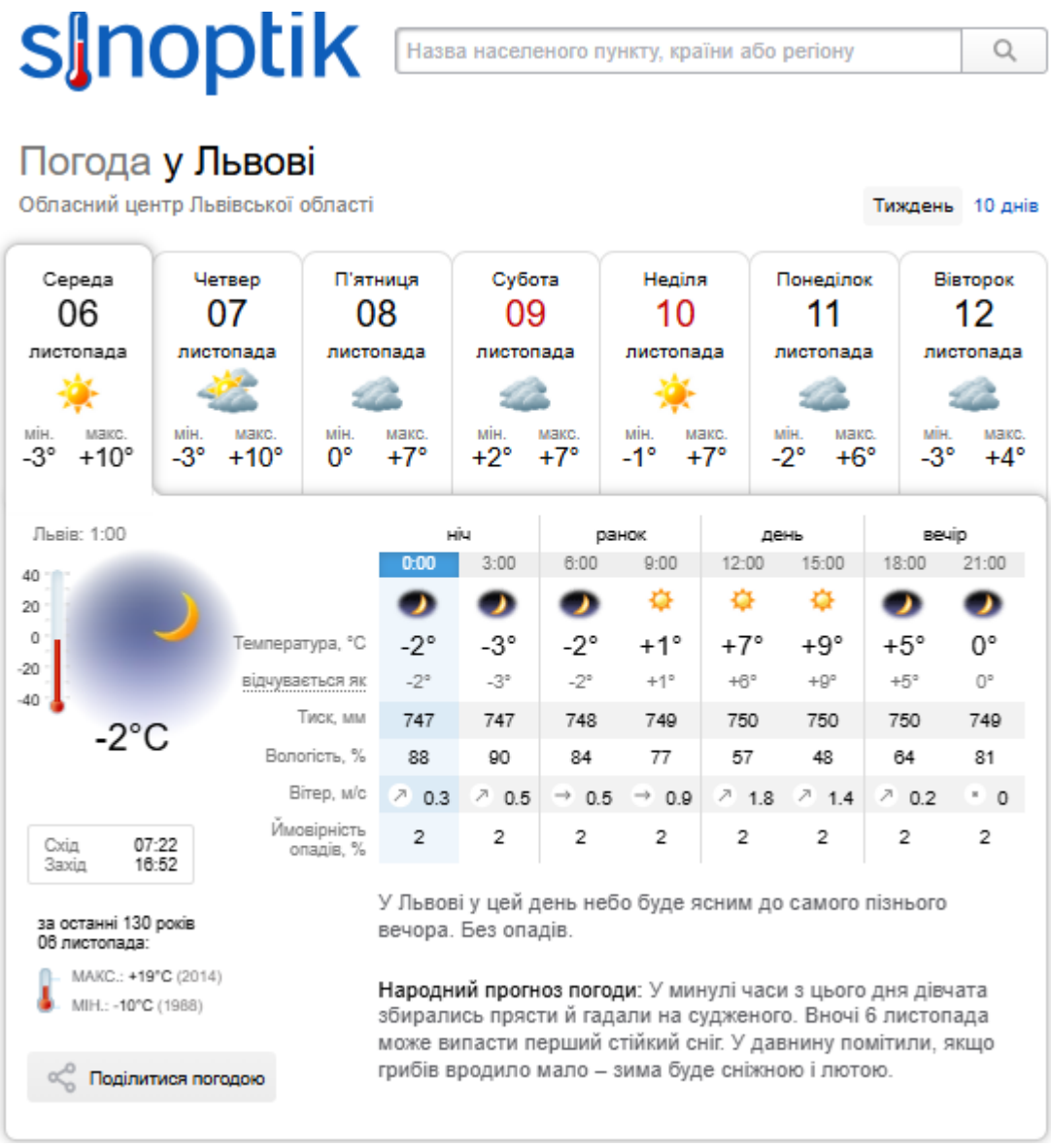


Рисунок 1.8 – Інтерфейс сайту з прогнозом погоди «Синоптик»

Прогноз деталізується за періодами дня – ранок, день, вечір та ніч – з відповідною температурою та можливими опадами. Крім того, для кожного дня надається інформація про швидкість і напрямок вітру, рівень вологості та

атмосферний тиск, що допомагає отримати більш повну картину про погодні умови. У розширеній версії прогнозу користувач може побачити імовірність опадів, яка часто подається у вигляді відсотків, що дозволяє більш обґрунтовано оцінити ймовірність дощу або снігу.

На сайті наведено прогноз із прийнятним рівнем деталізації на найближчі кілька днів, при цьому короткострокові прогнози (1–3 дні) зазвичай є найбільш точними, тоді як прогнози на більш тривалий період можуть мати нижчу точність. Сайт також використовує дані з різних метеорологічних джерел, що допомагає поліпшити точність прогнозу, але як і більшість погодних ресурсів, він має свої обмеження при прогнозуванні складних або швидкозмінних погодних ситуацій.

На сайті «Метеофор» прогноз погоди представлено у більш інтенсивному, інтерактивному інтерфейсі, який дозволяє налаштувати відображення даних під потреби користувача [15]. На сайті можна обрати, на який термін переглянути прогноз: доступні як короткострокові прогнози на кілька днів, так і більш тривалі – до двох тижнів, що зручно для планування наперед.

Сайт також дозволяє налаштовувати перегляд певних метеорологічних параметрів, серед яких є:

– температура повітря;

– середня швидкість повітря;

– напрямок вітру;

– опади;

– тиск;

– відносна вологість;

– УФ-індекс.

Хоча інтерфейс «Метеофор» виглядає насичено, він водночас інтуїтивно зрозумілий, а інформації тут достатньо для отримання повної картини про погодні умови (рис. 1.9 [15]).

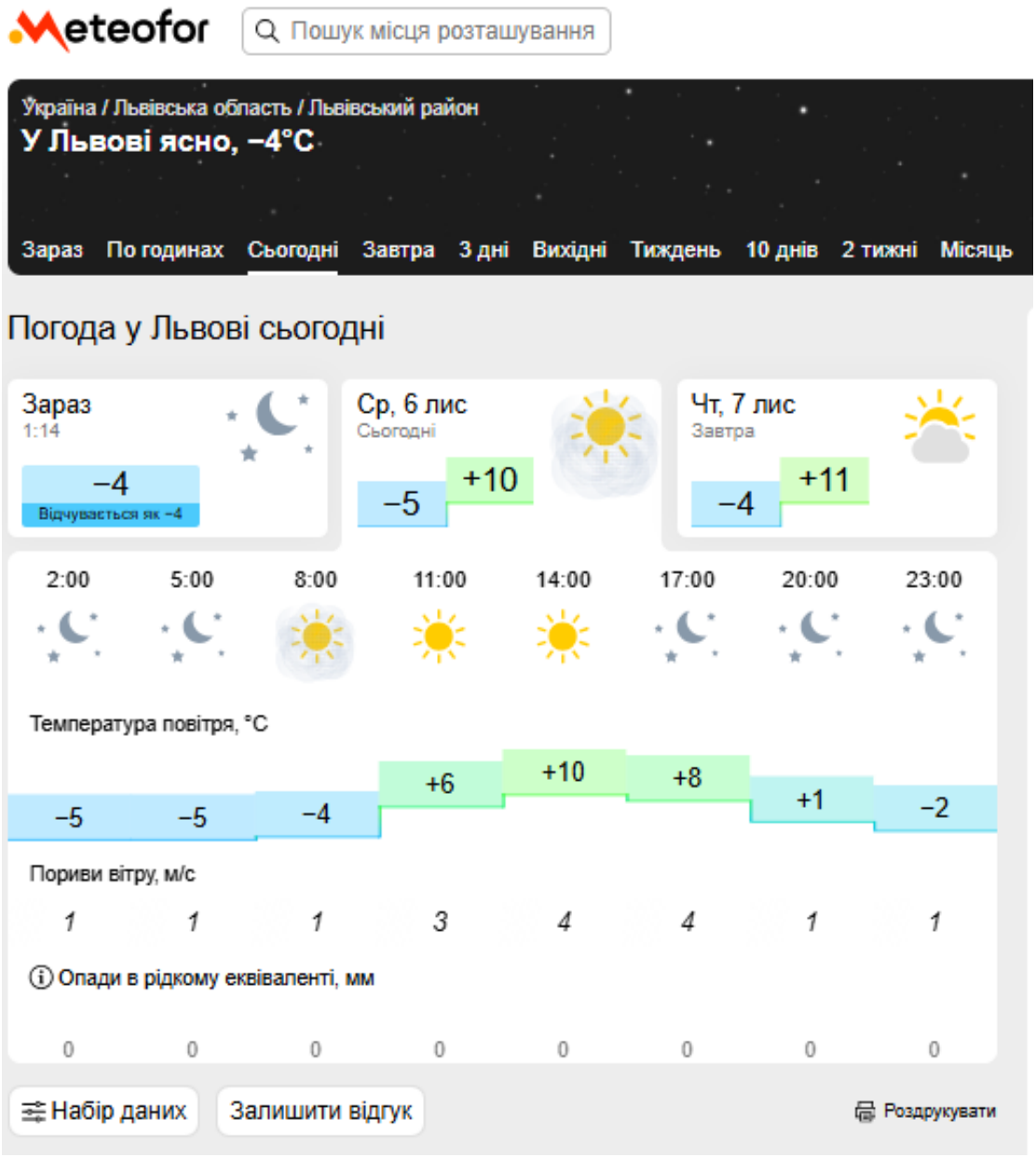


Рисунок 1.9 – Інтерфейс сайту з прогнозом погоди «Метеофор»

Всі основні показники надано в доступному форматі, і це дозволяє швидко орієнтуватися в прогнозах. У той же час відсутність зайвих деталей сприяє більш чіткому сприйняттю даних.

Отже, прогнози погоди на електронних ресурсах, як «Синоптик» і «Метеофор», зазвичай є відносно точними в короткостроковій перспективі. Таких сайтів достатньо для кількаденного планування. Користувачі отримують основні дані про температуру, опади, вологість, а також, залежно від сайту, додаткові показники швидкості вітру і атмосферного тиску.

Однак, попри зручність і доступність прогнозу, методи його розрахунку залишаються прихованими. Невідомо, які саме алгоритми чи джерела даних використовуються для прогнозування, а також чи застосовуються сучасні моделі машинного навчання або спеціалізовані метеорологічні моделі. Це обмежує розуміння точності та надійності прогнозу, оскільки користувач не має можливості оцінити, яким чином і за допомогою яких методів були отримані кінцеві результати.

Відносна точність короткострокових прогнозів, які пропонують сучасні погодні сервіси, зручність їх використання та доступність для широкого кола користувачів підтверджують попит на розвиток цієї галузі.

1.3 Постановка задачі дослідження

У результаті проведеного аналізу літературних та електронних джерел було виявлено, що тема прогнозування погодних умов є актуальною та перспективною для наукових досліджень. Тому ставиться завдання вивчення сучасних методів, таких як лінійна регресія, дерева рішень та рекурентні нейронні мережі для прогнозування погодних умов.

Об’єктом дослідження є якість прогнозу погодних умов на основі метеорологічних даних.

Метою дослідження є проведення порівняльного аналізу методів машинного навчання, що використовуються для задач прогнозування.

Для досягнення мети необхідно вирішити такі завдання:

– провести аналіз обраних методів машинного навчання для задачі прогнозування;

– розробити алгоритми, що використовують лінійну регресію, дерева рішень, рекурентні нейронні мережі для прогнозування погодних умов на основі доступних метеорологічних даних;

– дослідити вплив додавання штучного шуму до метеорологічних даних на точність прогнозування;

– реалізувати систему тестування та оцінки точності розроблених алгоритмів на вибірці даних;

– провести аналіз отриманих результатів;

– сформувати перелік особливостей та рекомендацій щодо вдосконалення існуючих методів прогнозування, а також запропонувати підходи для їх адаптації до змінюваних погодних умов.

2 ОСОБЛИВОСТІ ВИБРАНИХ МЕТОДІВ НАВЧАННЯ ДЛЯ ПРОГНОЗУВАННЯ ПОГОДНИХ УМОВ НА ОСНОВІ МЕТЕОРОЛОГІЧНИХ ДАНИХ

2.1 Аналіз методу на основі лінійної регресії

Регресія (у контексті машинного навчання) – це завдання оцінки числового значення якоїсь величини залежно від одного або кількох атрибутів (наприклад, оцінка ціни квартири на основі її площі). Лінійна регресія – це розв’язання задачі регресії за допомогою лінійної моделі, тобто моделі, яка є лінійною функцією (тобто зваженою сумою) оптимізованих параметрів [16]. Змінна, яку необхідно передбачити, називається залежною змінною. Змінна, яка використовується для прогнозування значення іншої змінної, називається незалежною змінною.

У простій лінійній регресії є одна незалежна змінна і одна залежна змінна. Модель оцінює нахил і перетин лінії найкращої відповідності, яка відображає зв’язок між змінними (рис. 2.1).

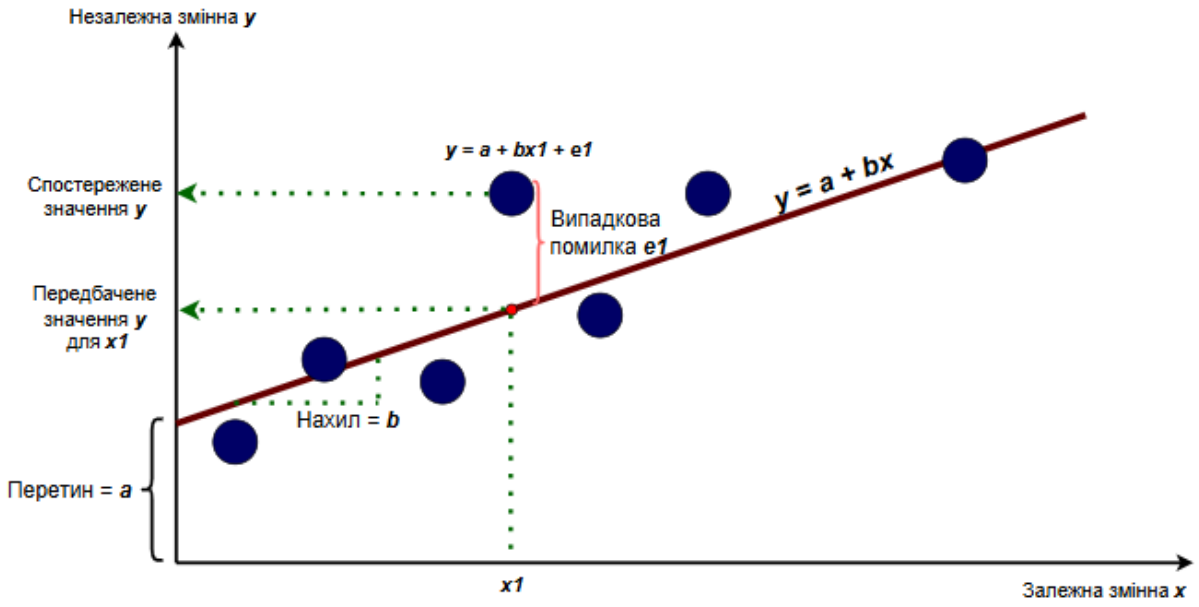


Рисунок 2.1 – Основні елементи лінійної регресії

Найпростішим прикладом лінійної регресії є пошук лінії $y = ax + b$, яка найкраще описує зв'язок між x та y координатами вказаних точок $D = \{(x_1, y_1),\ (x_2, y_2), \dots, (x_n, y_n)\}$. Така модель має два параметри (a, b), які необхідно визначити в процесі навчання та оптимізації. Нахил a показує зміну залежної змінної x на кожну одиницю зміни незалежної змінної y, в той час як перетин b представляє прогнозоване значення залежної змінної, коли незалежна дорівнює нулю. Різницю між спостережуваним значенням залежної змінної і прогнозованим значенням називають розбіжністю або помилкою.

Лінія найкращої відповідності – це лінія, яка має найменшу похибку, що означає, що помилка між прогнозованими $\hat{y}$ та фактичними значеннями y має бути мінімальною. Для того щоб вибрати «найкращу» лінію, нам потрібно встановити функцію помилок, яка описує, наскільки дана лінія відповідає заданим точкам. Найчастіше використовується квадратична похибка, яка визначається як сума квадратів відхилень індивідуальних оцінок від реальності. Цей підхід називають методом найменших квадратів, оскільки основна мета це мінімізація суми квадратів (рис. 2.2).

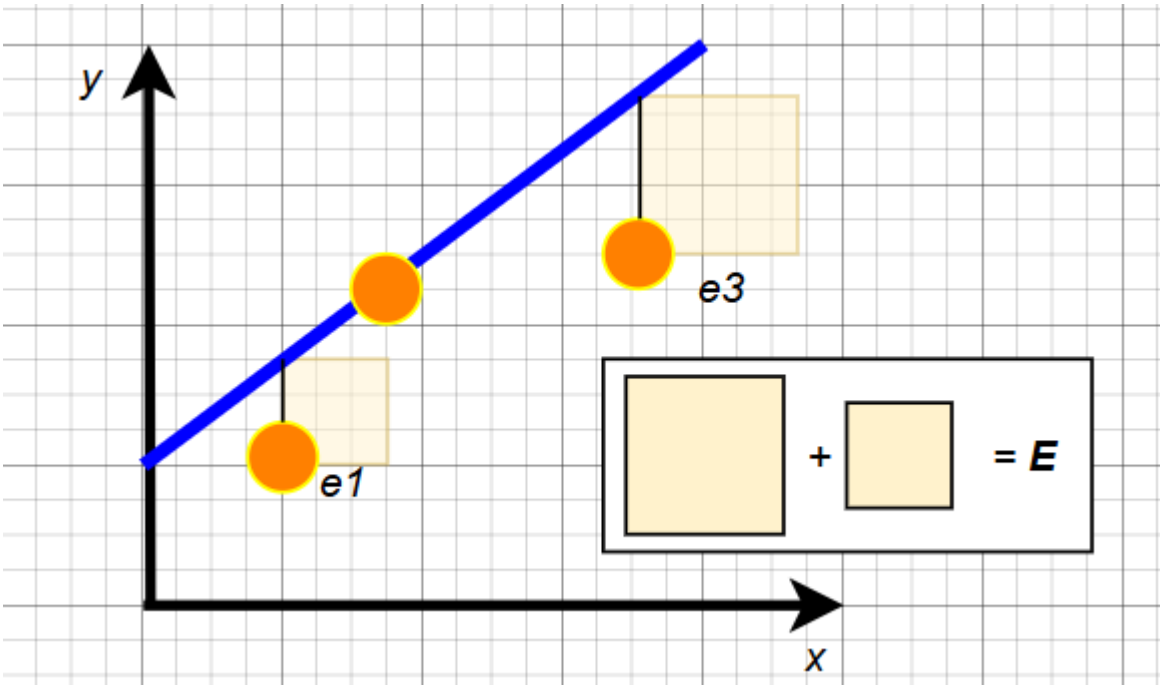


Рисунок 2.2 – Графічна інтерпретація квадратичної похибки

Квадратична помилка E залежить від параметрів a, b та набору точок D. Розраховується

$$E(a,b,D) = \sum_{(x,y)\in D}(\hat{y}-y)^2 = \sum_{(x,y)\in D}(ax+b-y)^2. \quad (0.1)$$

Квадратична помилка чутлива до викидів – точок, які значно відрізняються від інших – через піднесення відхилень до квадрату. Іноді викиди просто видаляються. У будь-якому випадку, про них важливо знати.

Як вже було згадано раніше, основне завдання полягає в пошуку таких значень a та b, які мінімізують квадратичну похибку. Існує декілька загальних підходів до цієї задачі. Найбільш відомі з них це систематичне тестування різних комбінацій значень параметрів (груба сила) та поступове вдосконалення однієї комбінації параметрів з незначними коригуваннями (градієнтний спуск). Проте у випадку лінійної регресії є інакший спосіб визначення оптимальних параметрів – аналітичний

$$a = \frac{\sum(x-\bar{x})(y-\bar{y})}{\sum(x-\bar{x})^2}, \quad (2.2)$$

$$b = \bar{y} - a\bar{x}, \quad (2.3)$$

де $\bar{x}$ – середнє значення x;

$\bar{y}$ – середнє значення y.

З формул можна помітити, що вищий перетин a свідчить про сильніший зв’язок між x та y. Рівняння для кута нахилу було отримано з інтуїтивно зрозумілого співвідношення середніх значень, яке можна вивести через похідну квадратичної похибки.

Якщо оцінка виконується на основі більше ніж одного атрибуту, тоді лінійна модель вже не є простою і має параметр для кожного з атрибутів. У загальному випадку лінійна модель має вигляд об’єкта з певним набором ознак та загальним числом.

Формула буде мати дещо інакший вигляд. Кількість доданків залежить від кількості ознак

$$y = \omega_0 + \omega_1 x_1 + \cdots + \omega_n x_n + \varepsilon, \quad (2.4)$$

де ω_i – параметри (ваги) моделі (як a та b);

x_i – ознаки (атрибути), на основі яких виконується оцінка;

ω_0 – вільний коефіцієнт (зсув);

ε – розбіжність між фактичними та передбаченими значеннями.

Ваги визначають, як кожен з атрибутів впливає на залежну змінну. Отже, в загальному випадку аналітична формула для лінійної моделі матиме вигляд функції

$$a(x) = \omega_0 + \sum_{i=1}^{n} \omega_i x_i, \quad (2.5)$$

де ω_0 – вільний коефіцієнт (зсув);

x_i – ознаки (атрибути), на основі яких виконується оцінка;

ε – розбіжність між фактичними та передбаченими значеннями.

Для лінійної моделі використовувати випадкову вибірку недоцільно, оскільки це може призвести до посередніх результатів. Щоб модель працювала ефективніше і точніше, дані слід підготувати належним чином.

На етапі попередньої обробки даних необхідно прибрати шум або непотрібну інформацію, залишаючи тільки найбільш важливі та інформативні ознаки. Тоді модель буде менше «плутатися», що позитивно вплине на кількість правильних прогнозів.

Задля підвищення ефективності навчання також є доцільним проведення шкалювання ознак. Шкалювання або масштабування – це процес конвертації виміряних результатів за деякою шкалою, якщо наявна певна норма, відповідне число. Тобто, якщо ознаки мають різні масштаби, то необхідно використати стандартизацію або нормалізацію. Це зробить модель

більш стійкою до шумів і аномалій, а також поліпшить її передбачувальну точність.

Для перетворення категоріальних ознак на числові, такі, з якими модель зможе працювати, можна використовувати кодування міток (Label Encoding). Кодування міток присвоює унікальний номер (починаючи з 0) кожному класу даних. Це може призвести до створення пріоритетних питань під час навчання моделі наборів даних. Мітка з високим значенням може вважатися такою, що має високий пріоритет, ніж мітка з нижчим значенням [17]. В такому випадку варто змінити підхід на one-hot кодування. Цей метод перетворює категоріальні змінні у двійковий формат. Кожна категорія у вихідному стовпці представлена як окремий стовпець, де значення 1 вказує на наявність цієї категорії, а 0 вказує на її відсутність. Тобто, якщо є три класи, мітки будуть кодуватися як [1, 0, 0], [0, 1, 0], і [0, 0, 1] для кожного класу відповідно.

Метод лінійної регресії має кілька параметрів, які можна налаштувати для визначення поведінки моделі:

– fit_intercept – параметр, що визначає наявність вільного члена у рівнянні (при значенні False модель буде проходити через початок координат);

– normalize – параметр, що визначає необхідність проведення нормалізації даних перед початком навчання;

– n_jobs – параметр, що визначає кількість задіяних у навчанні ядер CPU;

– positive – параметр, що визначає обмеження на позитивні коефіцієнти.

Для оцінювання результатів роботи моделі та визначення того, наскільки точно розроблений метод передбачає цільову змінну на нових даних, застосовують низку метрик:

– середньоквадратична помилка (MSE);

– модифікація середньоквадратичної помилки (RMSE).

Корінь з середньої квадратичної помилки вимірює середнє абсолютне відхилення між фактичними значеннями та передбаченими значеннями моделі – що менше значення, то краще модель відповідає даним. Ця метрика особливо актуальна, коли потрібно боротися із викидами [18].

Викиди можуть бути як аномаліями (помилками або аномальними подіями), так і реальними, але незвичайними значеннями. Подібні значення можна або замінити, щоб вони не спотворювали результати, або просто прибрати з набору даних, але це може призвести до втрати інформації.

З метою дослідження достовірності припущення про існування лінійного зв'язку між змінними, регресійну модель необхідно перевірити на залишки. Побудова залишків на осі *y* проти змінної на осі *x* виявляє будь-який можливий нелінійний зв'язок між змінними або може спонукати дослідити приховані змінні [19]. Прихована змінна існує тоді, коли на зв'язок між двома змінними суттєво впливає присутність третьої змінної, яка не була включена до процесу моделювання.

Отже, лінійна регресія є одним із базових та найпопулярніших методів прогнозування в машинному навчанні та статистиці, який базується на припущенні лінійної залежності між змінними. Цей алгоритм вирізняється швидкістю навчання і стійкістю до шуму в даних, що робить його ефективним для простих задач із невеликою кількістю факторів. Однак, у випадках із нелінійними залежностями лінійні моделі демонструють обмежену точність та ефективність, оскільки не можуть повністю відобразити складні зв'язки між змінними. До того ж, лінійна регресія є чутливою до викидів: навіть кілька аномальних точок можуть істотно викривити результати та знизити якість прогнозів. Таким чином, для успішного застосування лінійної регресії необхідна ретельна підготовка даних та врахування специфіки задачі, щоб забезпечити адекватну ефективність та точність.

2.2 Аналіз методу навчання на основі дерева рішень

Дерева рішень – це популярний і потужний метод, який використовується в різних сферах, таких як машинне навчання, інтелектуальний аналіз даних і статистика. Цей інструмент забезпечує чіткий та інтуїтивно зрозумілий спосіб прийняття рішень на основі даних шляхом моделювання взаємозв'язків між різними змінними. За своєю суттю, це контрольований алгоритм машинного навчання, який використовується як для завдань класифікації, так і для регресії. Основна ідея полягає в тому, щоб розділити дані на підмножини на основі значень ознак і прийти до рішення або прогнозу, приймаючи на кожному вузлі ті варіанти розв'язку, що максимізують певний критерій. Структура дерева схожа на блок-схему (рис. 2.3).

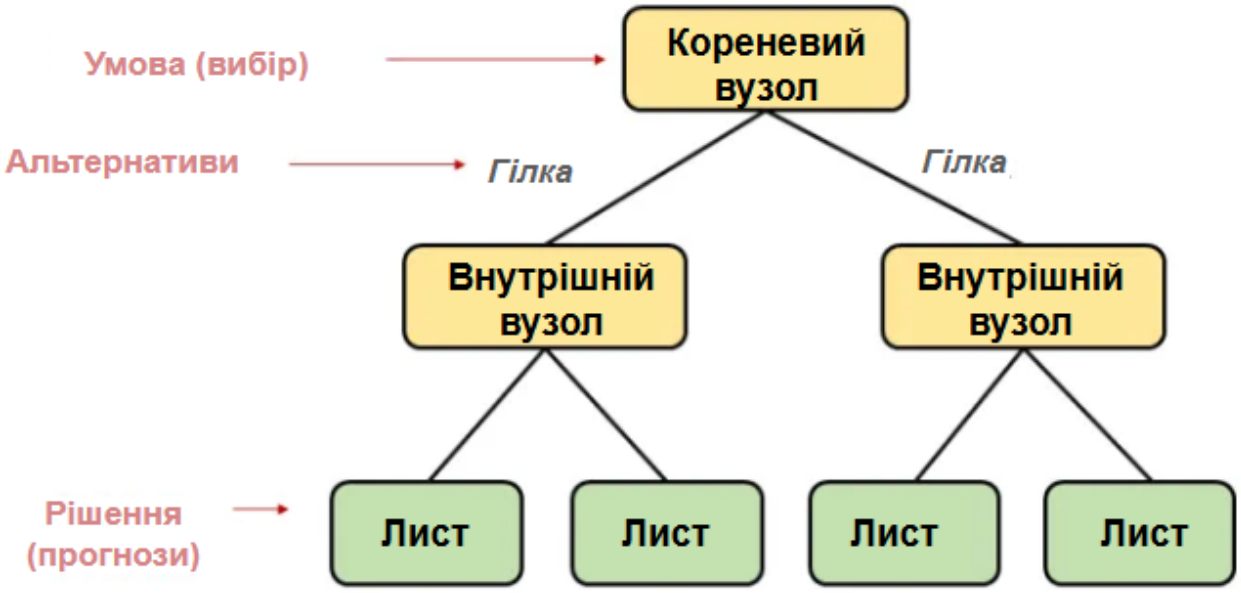


Рисунок 2.3 – Основні елементи дерева рішень

Основними компонентами дерева рішень є вузли та листи. Вузли представляють собою точки розгалуження, де відбувається вибір на основі певної умови або критерію. Листи, у свою чергу, розташовані в кінці кожної гілки й містять кінцеві рішення або прогнози моделі.

Починається дерево з кореневого вузла, який не має жодних вхідних гілок. Кореневий вузол представляє весь набір даних і початкове рішення, яке необхідно прийняти. Вихідні гілки від кореневого вузла надходять у внутрішні вузли, також відомі як вузли прийняття рішень. Внутрішні вузли представляють рішення або тести на атрибутах. Кожен внутрішній вузол має одну або декілька гілок. Гілки представляють результат рішення або тесту, що веде до іншої вершини. На основі доступних функцій проводиться оцінка для формування однорідних підмножин, які позначаються листовими вузлами або кінцевими вузлами. Листові вузли представляють остаточне рішення або прогноз. На цих вузлах більше не відбувається розгалужень.

Навчання дерева рішень використовує стратегію «розділяй і володарюй», проводячи жадібний пошук, щоб визначити оптимальні точки поділу. Цей процес поділу повторюється рекурсивним способом зверху вниз, доки всі або більшість записів не буде класифіковано за певними мітками класу. Чи всі точки даних класифікуються як однорідні набори, значною мірою залежить від складності дерева рішень. Меншим деревам легше отримати чисті вузли листя, тобто точки даних в одному класі. Однак із збільшенням розміру дерева стає все важче підтримувати цю чистоту, і це зазвичай призводить до фрагментації даних [20].

Щоб зменшити складність і запобігти надмірному заглибленню, зазвичай застосовують обрізання дерева – це процес, який видаляє гілки, що розгалужуються на ознаки з низьким рівнем важливості. Після обрізання відповідність моделі можна оцінити за допомогою перехресної перевірки. Інший спосіб зберегти точність дерев рішень – сформувати ансамбль за допомогою алгоритму випадкового лісу. Такий класифікатор прогнозує точніші результати, особливо коли окремі дерева не корелюють між собою.

Процес створення дерева рішень передбачає:

Крок 1. Вибір найкращого атрибута за допомогою певного критерію.

Крок 2. Поділ набору даних на підмножини на основі вибраного атрибута.

Крок 3. Повторення процесу рекурсивно для кожної підмножини, доки не буде виконано критерій зупинки.

Якщо набір даних складається з *N* атрибутів, то рішення про те, який атрибут розмістити в корені або на різних рівнях дерева (внутрішніх вузлах), є складним кроком. Просто випадковий вибір будь-якого вузла як кореневого може призвести до поганих результатів з низькою точністю. Для вирішення проблеми модна використати наступні критерії:

– ентропія;

– приріст інформації;

– коефіцієнт Джині;

– коефіцієнт приросту;

– зменшення дисперсії;

– *xi*-квадрат.

За цими критеріями будуть обчислені значення для кожного атрибута. Значення сортуються, а атрибути розміщуються в дереві відповідно до порядку, тобто атрибут із високим значенням розміщується в корені [21]. Використання приросту інформації передбачає роботу з категоричними атрибутами, використання коефіцієнту Джині – безперервними.

Ентропія – це концепція, запозичена з теорії інформації, і зазвичай використовується як міра невизначеності або безладу в наборі даних [22]. Низька ентропія вказує на більш впорядкований або однорідний набір, тоді як висока ентропія означає більший безлад або різноманітність (рис. 2.4).

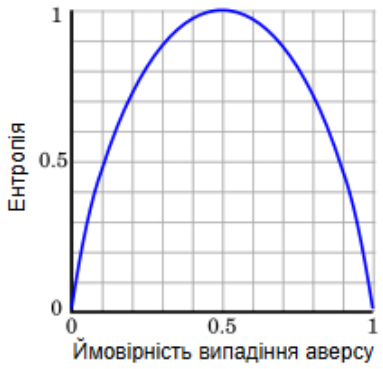


Рисунок 2.4 – Залежність ентропії від ймовірності випадіння аверсу

Мінімальна ентропія (0) виникає, коли всі екземпляри належать до одного класу, що робить набір ідеально впорядкованим. Максимальна ентропія (1) виникає, коли екземпляри рівномірно розподіляються між усіма класами, створюючи стан максимального безладу.

Математично ентропія для 1 атрибуту представляється так

$$E(S) = \sum_{i=1}^{c} -p_i \log_2 p_i, \tag{2.6}$$

де S – поточний стан;

p_i – ознаки (атрибути), ймовірність події i стану S або відсоток класу i у вузлі стану S.

Математично ентропія для кількох атрибутів представляється так

$$E(T, X) = \sum_{c \in x} P(c) E(c), \tag{2.7}$$

де T – поточний стан;

X – вибраний атрибут.

У кожному вузлі дерева рішень алгоритм оцінює ентропію для кожної функції та точки розділення. Для поділу вибирається функція та точка поділу, які призводять до найбільшого зменшення ентропії. Зменшення ентропії часто називають інформаційним приростом і обчислюють як різницю між ентропією до і після поділу.

Математично приріст інформації (IG) представляється так

$$IG = E_{parent} - \sum_{i=1}^{n} (\frac{|D_i|}{|D|} * E(D_i), \tag{2.8}$$

де D_i – підмножина D після розбиття за атрибутом.

Коефіцієнт Джині – це ймовірність неправильної класифікації випадкової точки в наборі даних, якщо вона була позначена на основі

розподілу класів набору даних. Подібно до ентропії, якщо множина S є чистою (тобто належить до одного класу), то її коефіцієнт дорівнює нулю.

$$Gini = 1 - \sum_{i=1}^{n}(p_i)^2, \quad (2.9)$$

де p_i – ймовірність класифікації екземпляра в певний клас.

Алгоритми дерева рішень спрямовані на мінімізацію коефіцієнта. Чим менший коефіцієнт Джині, тим «чистіші» групи даних у вузлах, тобто кожна підмножина більш однорідна й складається здебільшого з елементів одного класу. Мета алгоритму – вибрати такі розподіли, які максимізують «чистоту» підмножин, що допомагає побудувати дерево з кращою класифікаційною здатністю.

Отже, після проведеного аналізу можемо визначити, що дерева рішень є потужним інструментом для застосування в машинному навчанні. Цей метод ефективно використовуються для задач класифікації, таких як виявлення спаму, аналіз настроїв і діагностика захворювань, а також для регресійних задач, що включають прогнозування цін та погодних умов. Проте під час моделювання слід враховувати схильність дерев до надмірної пристосованості, особливо на зашумлених даних. Доцільно звернути увагу на вплив вибору критеріїв розділення, таких як коефіцієнт Джині чи ентропія, на продуктивність і точність моделі. Крім того, важливо дослідити вплив застосування методів обрізання, які зменшують розмір дерева і, відповідно, ризик надмірного заглиблення.

2.3 Аналіз методу навчання на основі рекурентної нейронної мережі

Штучні нейронні мережі – математичні моделі, а також їх програмні або апаратні реалізації, побудовані за принципом організації й функціонування біологічних нейронних мереж – мереж нервових кліток

живого організму. Це поняття виникло при вивченні процесів, що протікають у мозку, і при спробі змоделювати ці процеси. Першою такою спробою були нейронні мережі Маккалока й Піттса. Згодом, після розробки алгоритмів навчання, одержувані моделі стали використовувати в практичних цілях: у завданнях прогнозування, для розпізнавання образів, у завданнях керування та інших [23].

Штучні нейронні мережі, які не мають зациклених вузлів, називаються нейронними мережами прямого зв'язку. Інформація таких мережах переміщується від вхідного рівня до вихідного рівня – односпрямовано. Ці мережі підходять для завдань класифікації зображень, наприклад, де вхід і вихід незалежні. Тим не менш, нездатність автоматично зберігати попередні вхідні дані робить їх менш корисними для послідовного аналізу даних.

Повторювана нейронна мережа (RNN) – це тип нейронної мережі, де вихідні дані попереднього кроку подаються як вхідні дані для поточного. У традиційних нейронних мережах усі входи та виходи незалежні один від одного. Однак у випадках, коли, наприклад, потрібно передбачити наступне слово речення, необхідно мати уявлення про попередні слова, а отже, виникає потреба запам'ятати ці слова. Так виникла RNN, яка вирішила проблему запам'ятовування за допомогою прихованого шару (рис. 2.5).

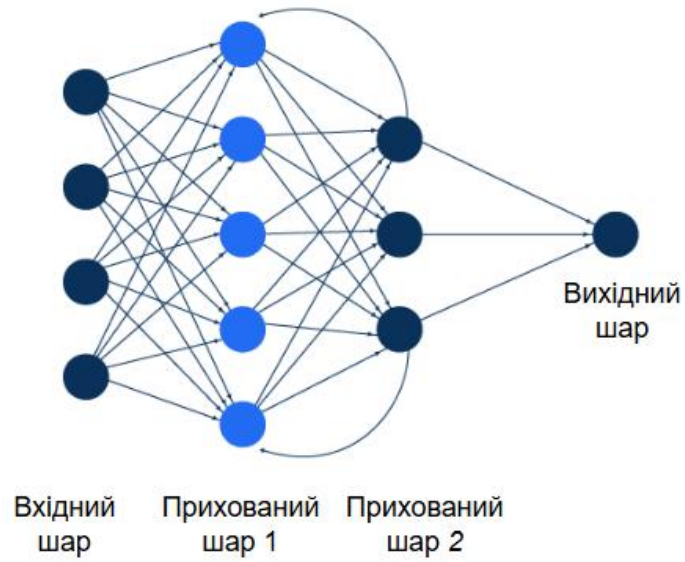


Рисунок 2.5 – Загальна структура рекурентної нейронної мережі

Головною і найважливішою особливістю RNN є прихований стан, який запам’ятовує деяку інформацію про послідовність. Цей стан також називають станом пам’яті, оскільки він запам’ятовує попередній вхід до мережі. Він використовує однакові параметри для кожного входу, оскільки виконує однакове завдання на всіх входах або прихованих шарах для створення виходу.

Комірки пам’яті в RNN знаходяться в центрі обчислень. Ці комірки керують потоком інформації між вхідним і вихідним рівнями та приймають вхідні дані від попередньої комірки, відстежуючи інформацію [24].

Існує чотири типи RNN на основі кількості входів і виходів у мережі:

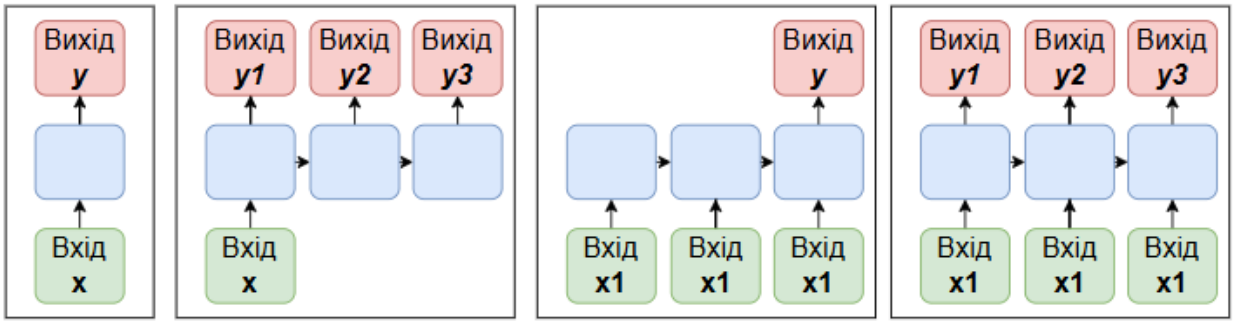
– один-до-одного;

– один-до-багатьох;

– багато-до-одного;

– багато-до-багатьох.

Тип «Один-до-одного» поводиться так само, як будь-яка проста нейронна мережа. У такої нейронної мережі є тільки один вхід і один вихід, як показано на рисунку 2.6 а).



а) б) в) г)

Рисунок 2.6 – Типи RNN:

а) один-до-одного; б) один-до-багатьох; в) багато-до-одного;

г) багато-до-багатьох

У типі RNN один-до-багатьох, як зображено на рисунку 2.6 б) є один вхід і багато виходів, пов’язаних з ним.

Одним із найбільш використовуваних прикладів цієї мережі є субтитри до зображень, де за зображенням ми передбачаємо речення, що містить кілька слів. У типі багато-до-одного, як на рисунку 2.6 в) декілька входів подаються в мережу в кількох станах, генерується лише один вихід.

Останній тип мереж – багато-до-багатьох, що зображений схематично на рисунку 2.6 г) – приймає кілька входів і генерує кілька виходів, що відповідають проблемі. Одним із прикладів таких задач є переклад, у якому надається кілька слів з однієї мови як вхідні дані та прогнозується кілька слів з іншої мови як вихідні дані.

RNN мають ту саму архітектуру введення та виведення, що й будь-яка інша глибока нейронна архітектура. Однак відмінності виникають у тому, як інформація переходить від входу до виходу. На відміну від глибоких нейронних мереж, де ми маємо різні матриці ваг, в RNN вага залишається незмінною на кожному рівні мережі. Тим не менш, ці ваги все ще коригуються за допомогою процесів зворотного поширення та градієнтного спуску [25].

Функція активації – це математична функція, яка застосовується до виходу кожного шару нейронів у мережі, щоб ввести нелінійність і дозволити мережі вивчати більш складні закономірності в даних. Без функцій активації ШНМ просто обчислювала би лінійні перетворення вхідних даних, що робило б модель нездатною вирішувати нелінійні задачі. Нелінійність має вирішальне значення для навчання і моделювання складних закономірностей, особливо в таких завданнях, як аналіз часових рядів і послідовне прогнозування даних.

Функція активації контролює величину виходу нейрона, утримуючи значення в заданому діапазоні, що допомагає запобігти занадто великому або занадто малому зростанню значень під час прямих і зворотних проходів. У ШНМ функції активації застосовуються на кожному часовому кроці до прихованих станів, контролюючи, як мережа оновлює свою внутрішню

пам’ять (прихований стан) на основі поточних вхідних даних і минулих прихованих станів (рис. 2.7).

Рисунок 2.7 – Основні функції активації

Сигмоїдна функція використовується для завдань, де вихід значень бажано обмежити в діапазоні між 0 і 1. Вона є плавною і підходить для класифікації, однак схильна до проблеми «зникнення градієнта», що робить її менш ідеальною для глибших мереж.

$$g(z) = \frac{1}{1+\partial^{-z}}. \tag{2.10}$$

Гіперболічний тангенс схожий на сигмоїд, але дає значення в діапазоні від -1 до 1, що часто допомагає нейронній мережі навчатися швидше. Відносно часто використовується в рекурентних, оскільки виводить значення з центром навколо нуля, що допомагає з вивченням довгострокових залежностей.

$$g(z) = \frac{\partial^{3} - \partial^{-z}}{\partial^{3}+\partial^{-z}}. \tag{2.11}$$

ReLU (Rectified Linear Unit) працює як лінійна функція для позитивних значень і дає нуль для всіх від’ємних значень. Є стандартною функцією активації для сучасних глибоких нейронних мереж завдяки своїй ефективності та здатності уникати «зникнення градієнта».

$$g(z) = max(0, z). \quad (2.12)$$

RNN мають унікальні алгоритми навчання через їх характер включення історичної інформації в послідовності. Алгоритми градієнтного спуску були успішно застосовані для оптимізації ваг RNN (для мінімізації помилки шляхом коригування ваги пропорційно до похідної помилки цієї ваги). Одним із популярних методів є зворотне поширення в часі (Backpropagation Through Time, BPTT), яке застосовує оновлення ваги шляхом підсумовування оновлень ваги накопичених помилок для кожного елемента в послідовності, а потім оновлення ваг наприкінці. Для великих вхідних послідовностей така поведінка може призвести до того, що вагові коефіцієнти або зникають, або вибухають.

Явища зникаючого та вибухаючого градієнтів виникають у рекурентних нейронних мережах під час навчання, особливо коли потрібно опрацьовувати довготривалі залежності. У процесі зворотного поширення похибки, градієнти параметрів, відповідальних за обробку ранніх елементів послідовності, множаться на ваги. Це може призвести до експоненційного зменшення градієнтів (зникаючий градієнт) або, навпаки, до їх різкого збільшення (вибухаючий градієнт).

Для боротьби з цією проблемою зазвичай використовуються гібридні підходи, у яких BPTT поєднується з іншими алгоритмами, такими як рекурентне навчання в реальному часі.

Отже, після проведеного аналізу рекурентних нейронних мереж ми виявили, що для моделювання часових рядів, зокрема для задачі прогнозування погоди, цей метод є перспективним завдяки здатності враховувати інформацію з попередніх часових кроків. RNN здатна зберігати довготривалі залежності в даних, що є критичним для прогнозування на основі послідовних значень, таких як температура, вологість чи тиск.

Для побудови ефективної моделі необхідно приділити особливу увагу контролю зникаючого та вибухаючого градієнтів, які можуть значно

вплинути на точність прогнозу. Використання методів, таких як відсікання градієнта, а також експериментування з архітектурами RNN, які мають покращену пам’ять, наприклад LSTM чи GRU, може допомогти уникнути цих проблем.

Також важливо перевірити на практиці ефективність обраної RNN моделі для довгих часових послідовностей і тестувати її з різними функціями активації, щоб знайти оптимальну конфігурацію. У результаті така модель зможе точніше передбачати майбутні метеорологічні умови, зважаючи на складні взаємозв’язки між атрибутами, які характеризують погоду.

2.4 Формування методики для прогнозування погодних умов на основі метеорологічних даних

Прогнози бувають малої (1–3 дні), середньої (4–10 днів) і великої (місяць, сезон) завчасності. Основними методами складання прогнозів малої і середньої завчасності є синоптичний і розрахунковий [26].

Суть синоптичного методу прогнозування погоди базується на складанні й аналізі висотних і приземних карт погоди, що дозволяє визначити характер розвитку атмосферних процесів та подальші найбільш вірогідні зміни ходу метеоумов. Ці синоптичні карти складаються на спеціальній бланковій основі, куди умовними знаками й цифрами наносяться відомості про погоду на станціях.

Розрахунковий метод базується на розрахунках руху атмосферних вихорів. Під час розв’язання системи прогнозних рівнянь, знаходяться значення майбутніх полів тиску, температури, вітру тощо. Рівняння досить складні і обчислюються на сучасних комп’ютерах.

Метеорологічні дані отримують з різних джерел, зокрема з метеорологічних станцій, супутників, радарів, а також від метеорологічних

буїв та інших спеціалізованих приладів. Дані включають широкий спектр параметрів:

– температуру повітря;

– вологість;

– швидкість і напрямок вітру;

– атмосферний тиск;

– рівень опадів;

– хмарність;

– тощо.

Отримані дані надходять у вигляді великих обсягів інформації, яка потребує попередньої обробки: очищення, нормалізації, заповнення відсутніх значень і зменшення шумів. Оброблені дані зберігаються у базах, з яких потім використовуються для побудови прогнозів погоди.

Прогнозування погоди – складний процес, що базується на математичних і фізичних моделях, які враховують вплив різних факторів і взаємодій між ними. Традиційні прогнози будуються на основі чисельного моделювання – це метод, у якому для розрахунку погодних умов використовуються рівняння гідродинаміки, термодинаміки, перенесення енергії та інших фізичних законів. Проте сучасні технології все частіше включають методи машинного навчання, які дозволяють використовувати великі обсяги даних для побудови моделей, здатних прогнозувати погодні умови на основі закономірностей у даних.

Для прогнозування погоди на основі метеорологічних даних за допомогою методів машинного навчання буде використано вибірку, що містить 100 записів про погодні умови, з п’ятьма атрибутами.

Основна задача – навчити прогнозувати температуру на основі інших параметрів. Методика прогнозування включає кілька ключових кроків:

Крок 1. Обробка даних, яка включає перевірку набору даних на наявність помилок та аномалій.

Крок 2. Навчання моделі на 80 відсотках даних.

Крок 3. Прогнозування та обробка результатів.

Крок 4. Перевірка на тестовій вибірці та вдосконалення моделі шляхом зміни налаштувань або параметрів.

Крок 5. Аналіз та формування висновків.

Кроки 2–5 можна повторювати декілька разів, щоб зафіксувати, як різні модифікації параметрів впливають на продуктивність моделі. Це дозволить провести більш детальний аналіз, дослідити вплив кожної зміни на точність, робастність та інші обрані критерії. Кожна ітерація допоможе виявити специфічні особливості різних методів і надати глибші висновки про їх ефективність у контексті прогнозування погодних умов.

2.4.1 Методика прогнозування за допомогою лінійної регресії

На етапі підготовки даних для лінійної регресії спочатку мають бути оброблені всі категоріальні змінні. Щоб зробити їх придатними для моделювання, необхідно перетворити ці змінні на числові значення за допомогою one-hot encoding методу.

Після нормалізації та аналізу на наявність аномалій, дані необхідно розділити на 2 вибірки: навчальну (70%) та тестову (30%).

Прогнозування буде виконуватися на основі 5 атрибутів:

– x_1: вологість;

– x_2: хмарність;

– x_3: тиск;

– x_4: опади;

– y: температура – значення, яке буде прогнозуватися.

Цільова змінна y буде залежною, а x_1, x_2, x_3, x_4 – незалежними. Лінійна модель буде намагатися знайти найкращу лінійну залежність між цими змінними.

$$y = \omega_0 + \omega_1 x_1 + \omega_2 x_2 + \omega_3 x_3 + \omega_4 x_4, \quad (2.13)$$

де ω_0 – вільний член (зміщення);

$\omega_1, \omega_2, \omega_3, \omega_4$ – ваги або коефіцієнти регресії, які визначають внесок кожного з атрибутів у результат.

Для визначення найкращих значень коефіцієнтів регресії потрібно мінімізувати різницю між фактичними значеннями температури y_i та прогнозованими значеннями $\hat{y}_i$ для *i*-го запису. Для цього використовується формула середньоквадратичної помилки *MSE*.

$$MSE = \frac{1}{N} \sum_{i=1} (y_i - \hat{y}_i)^2, \quad (2.14)$$

де *N* – кількість записів у вибірці.

Для мінімізації помилки буде використано метод найменших квадратів. Основне завдання полягатиме у знаходженні значень коефіцієнтів, які мінімізують функцію *MSE*:

Крок 1. Обчислення градієнта функції помилки за кожним атрибутом.

$$\frac{\partial MSE}{\partial w_j} = -\frac{2}{N} \sum_{i=1} (y_i - \hat{y}_i) x_{ij}, \quad (2.15)$$

де j = 0, 1, 2, 3, 4.

Крок 2. Застосування градієнтного спуску для корекції значень коефіцієнтів.

$$\omega_j = w_j - \alpha \frac{\partial MSE}{\partial \omega_j}, \quad (2.16)$$

де α – швидкість навчання, яка визначає розмір кроку.

Цей процес буде повторюватися ітеративно до тих пір, поки не буде досягнуто мінімуму функції помилки. Після процесу навчання можна буде перейти до формування прогнозів на тестовій вибірці.

Для оцінки якості прогнозу буде обчислюватися середньоквадратична помилка *MSE*, середня абсолютна похибка *MAE* та коефіцієнт детермінації R^2.

$$MAE = \frac{1}{N}\sum_{i=1}|y_i - \hat{y}_i|. \quad (2.17)$$

Коефіцієнт детермінації допоможе визначити, наскільки добре модель пояснює варіацію у залежній змінній, порівняно з середнім значенням

$$R^2 = 1 - \frac{\sum_{i=1}(y_i - \hat{y}_i)^2}{\sum_{i=1}(y_i - \bar{y})^2}, \quad (2.18)$$

де $\bar{y}$ – середнє значення реальних температур.

Чим ближче отримане значення буде до 1 тим краще модель описує залежність. Якщо коефіцієнт детермінації буде дорівнювати 0, то це буде означати що модель не вносить нічого нового в прогнозування, окрім того, що можна було б передбачити просто використовуючи середнє значення реальних спостережень.

2.4.2 Методика прогнозування за допомогою дерева рішень

Етап підготовки даних буде аналогічний описаному в пункті 2.4.1. Дерево рішень на кожному кроці буде обирати атрибут, який найкраще розділяє дані. Основні критерії, які можуть бути використані для вирішення цієї задачі, – приріст інформації та коефіцієнт Джині. Варто дослідити, як вибір критерії впливає на отриманні результати.

Кореневий вузол буде містити всі дані. Далі буде застосовано ітеративний процес, який складається з наступних кроків:

Крок 1. Вибір атрибуту для розбиття – для кожного вузла обчислюється критерій розбиття для кожного з атрибутів. Обирається атрибут, який мінімізує помилку.

Крок 2. Розбиття вузла – вузол розбивається на дві або більше частини на основі значення вибраного атрибута. Кожна частина формує новий вузол.

Крок 3. Рекурсивне повторення – Кроки 1 і 2 повторюються для кожного нового вузла, поки не буде досягнуто умов зупинки.

Умовами зупинки будуть:

– глибина дерева досягне певного максимуму;

– мінімальна кількість записів у вузлі стане меншою за певний поріг;

– середня помилка в вузлі стане меншою за обрану.

Після побудови та навчання, дерево рішень буде готове до тестування на основі тестових даних. Аналогічно попередньому методу, для оцінки якості прогнозу буде обчислюватися середньоквадратична помилка *MSE*, середня абсолютна похибка *MAE* та коефіцієнт детермінації R^2.

2.4.3 Методика прогнозування за допомогою рекурентної нейронної мережі

У рекурентній нейронній мережі кожен часовий крок t передбачає обчислення прихованого стану h_t, який зберігає інформацію про попередні стани та вхідні дані. Для кожного моменту часу наявні наступні елементи:

– вектор вхідних даних x_t;

– вектор прихованого стану h_t;

– передбачення температури y_t.

Прихований стан h_t на кожному кроці t оновлюється залежно від попереднього стану h_{t-1} і вхідного вектору x_t. Оновлене значення розраховується за формулою.

$$h_t = \sigma(W_{xh}x_t + W_{hh}h_{t-1} + b_h), \tag{2.19}$$

де W_{xh} – матриця ваг, що зв'язує вхід з поточним прихованим станом;

W_{hh} – матриця ваг для зв'язку попереднього прихованого стану з поточним;

b_h – вектор зсуву;

σ – функція активації.

Після обчислення прихованого стану на кожному кроці t обчислюється прогнозований вихід y_t. Оновлене значення розраховується так

$$y_t = W_{hy}h_t + b_y, \tag{2.20}$$

де W_{hy} – матриця ваг, що зв'язує прихований стан із виходом;

b_y – вектор зсуву для виходу.

Після побудови та навчання, мережа буде готова до випробовування на основі тестових даних. Аналогічно попереднім методам, для оцінки якості прогнозу буде обчислюватися середньоквадратична помилка *MSE*, середня абсолютна похибка *MAE* та коефіцієнт детермінації R^2.

2.5 Моделювання структури програмного застосунку для прогнозування погодних умов на основі метеорологічних даних

Застосунок для прогнозування погоди буде складатися з декількох послідовних блоків.

Кожен блок буде реалізовано у вигляді окремої функціональної частини, що дозволить зберігати чистоту коду та полегшить масштабування і підтримку програми (рис. 2.8).

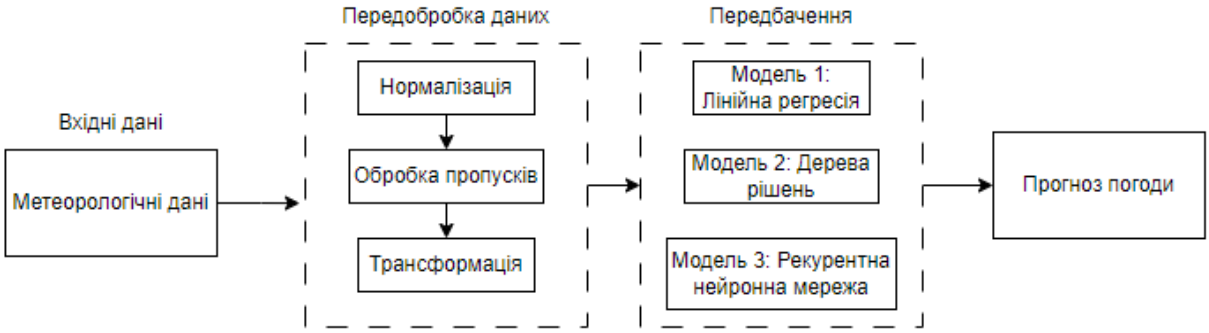


Рисунок 2.8 – Схема застосунку для прогнозування погодних умов

Перший блок програми буде складатися з функцій для передобробки даних. На цьому етапі здійснюється завантаження вибірки метеорологічних даних, проводиться очищення та нормалізація. Оброблені дані розділяються на навчальну і тестову вибірки.

Після цього буде розміщено блок першої моделі – лінійної регресії. Спочатку відбувається ініціалізація моделі, яка використовує вхідні параметри для передбачення цільової змінної. Модель навчається на навчальній вибірці, визначаючи коефіцієнти для кожного з атрибутів з метою мінімізації помилки. Потім модель тестується на тестовій вибірці. Проводиться оцінка її точності і стабільності. Результати тестування виводяться на екран, а також відображається графік для візуалізації прогнозованих і реальних значень.

Наступний блок буде містити модель дерева рішень. Модель навчається на навчальній вибірці, використовуючи процес побудови дерева для знаходження оптимальних розбиттів. Це забезпечує гнучкий і точний прогноз на основі навчальних даних. Після навчання модель тестується на тестовій вибірці. Результати виводяться на екран.

Останній блок буде присвячений рекурентній нейронній мережі. Створюється модель RNN з відповідною кількістю шарів та функцією активації. Модель навчається на послідовності даних, враховуючи попередні стани для передбачення наступного. Після навчання модель проходить тестування на нових даних, а результати – точність прогнозу та середня похибка – виводяться на екран. Для порівняння прогнозованих і реальних значень також буде побудовано графік.

У результаті розробки структури програмної реалізації для прогнозування погодних умов на основі метеорологічних даних Було сформовано основні блоки, кожен з яких виконує специфічні завдання. Кожен з цих блоків організований у послідовну структуру, яка дозволяє поступово переходити від простіших моделей до складніших. Обрана структура забезпечує гнучкість та інтеграцію різних підходів для прогнозування.

3 ДОСЛІДЖЕННЯ ТА ПОРІВНЯННЯ МЕТОДІВ НАВЧАННЯ ДЛЯ ПРОГНОЗУВАННЯ ПОГОДНИХ УМОВ НА ОСНОВІ МЕТЕОРОЛОГІЧНИХ ДАНИХ

3.1 Вибір інструментальних засобів для реалізації вибраних методів

У рамках кваліфікаційної роботи було розроблено методи для прогнозування погодних умов на основі доступних метеорологічних даних. Пошук найкращої мови програмування для реалізації застосунку має вирішальне значення у світі технологій, що постійно змінюються. Знання про відмінні характеристики мов і про те, як їх можна використовувати для покращення методів ML, може допомогти зробити обґрунтований вибір, який буде відповідати потребам проєкту [27]. Хоча жодна з існуючих мов не є ідеальною для будь-якої ситуації, ключовими факторами, які слід враховувати, є наступні:

– синтаксис і семантика коду;

– еластичність коду;

– інструменти та підтримка;

– продуктивність коду;

– сумісність та взаємодія;

– популярність мови.

Загалом можна виділити десять найбільш відомих мов для завдань машинного навчання (рис. 3.1). У кожної є свої плюси, мінуси та специфічні можливості. Враховуючи це та перелічені вище фактори, було вирішено обрати Python.

Python вважається найпопулярнішою мовою у світі машинного навчання та науки про дані завдяки простоті використання, чіткості та надійній підтримці бібліотек і фреймворків. Це кращий варіант як для експертів, так і для початківців.

Рисунок 3.1 – Найпопулярніші мови для ML

Близько 57% дослідників даних і розробників машинного навчання так чи інакше покладаються на Python, а 33% віддають йому пріоритет у розробці [28].

Синтаксис Python простий та інтуїтивно зрозумілий, що робить його доступним та ефективним. Надійна підтримка спільноти надає велику кількість навчальних посібників, документації та форумів, які полегшують навчання та вирішення проблем. Гнучкість інструменту дозволяє використовувати його і для розробки, автоматизації, і для аналізу даних тощо. Це мова загального призначення високого рівня, що робить її виконання повільнішою, в порівнянні з такою мовою, як C++. Однак це компенсується простотою та універсальністю.

Вбудовані бібліотеки та пакети Python, такі як TensorFlow, PyTorch, Scikit-learn, Pandas, Numpy та інші, значно полегшують процес розробки моделей машинного навчання. Вони пропонують готові рішення для численних задач, таких як передобробка, побудова моделей, навчання та оцінка ефективності.

Завдяки цьому можна зосередитися на самих алгоритмах та їх оптимізації, а не витрачати час на розробку базових інструментів для обробки та аналізу даних. Розглянемо детально бібліотеки, які будуть використані від час програмної реалізації.

NumPy – це бібліотека для обчислень. Це популярний інструмент для роботи з великими багатовимірними масивами, матрицями тощо. NumPy також надає набір математичних функцій для операцій над різними структурами даних. Ця бібліотека є основною для розрахунків в Python, особливо в наукових і інженерних задачах. Часто використовується в поєднанні з іншими бібліотеками.

Matplotlib – це бібліотека для візуалізації у Python, яка дозволяє створювати статичні, анімовані та інтерактивні діаграми. Вона є однією з найбільш популярних бібліотек для побудови графіків. Основна мета Matplotlib – допомогти користувачам візуалізувати дані, для їх подальшого легкого аналізу та інтерпретації.

Pandas – це бібліотека для обробки та аналізу даних. Вона надає зручні структури, такі як DataFrame та Series, що дозволяють легко маніпулювати, очищати та аналізувати великі обсяги даних. DataFrame – це двовимірна таблиця, яка дозволяє працювати з різними типами змінних, зручно здійснювати індексацію та фільтрацію, а також обчислювати статистичні показники. Pandas підтримує безліч операцій з даними, таких як об’єднання, групування, агрегування та перетворення.

Scikit-learn – це одна з найбільш популярних бібліотек для машинного навчання. Вона надає інструменти для побудови, тренування та оцінки моделей. Scikit-learn підтримує багато різних алгоритмів, які використовуються для вирішення завдань класифікації, регресії, кластеризації, зниження розмірності тощо.

У якості середовища розробки було обрано Google Colab. Це потужне хмарне середовище для роботи з Jupyter Notebook, яке дозволяє програмістам і дослідникам зручно виконувати обчислення, аналіз даних, машинне

навчання та інші завдання прямо в хмарі, не витрачаючи час на налаштування власної інфраструктури.

Однією з ключових переваг Google Colab є безоплатна доступність графічних процесорів, що дає змогу прискорити виконання складних обчислень і моделей машинного навчання. Крім того, Colab підтримує популярні бібліотеки Python, має зручний інтерфейс і надає можливість спільної роботи над проєктами.

3.2 Програмна реалізація вибраних методів

Розглянемо основні етапи програмної реалізації застосунку для дослідження методів прогнозування погодний умов на основі метеорологічних даних.

Етап 1. Передобробка вхідних даних.

На першому етапі було використано StandardScaler для того, щоб привести всі атрибути до одного масштабу. Тобто, було проведено нормалізацію кожної ознаки так, щоб середнє значення стало нулем, а стандартне відхилення – одиницею. Це запобігає спотворенню результатів моделювання через домінування одного атрибуту над іншим.

Після цього, для оцінки якості моделі, дані було розбито на тренувальний та тестовий набори. Візуальне зображення температури, передбачення якої буде досліджуватися, виводиться у вигляді графіку.

Лістинг 3.1 Передобробка даних:

```
scaler_X = StandardScaler()
scaler_y = StandardScaler()
X = scaler_X.fit_transform(X)
y = scaler_y.fit_transform(y)
```

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
plt.figure(figsize=(10, 6))
plt.plot(weather_data['time_of_day'], weather_data['temperature'], label="Temperature (°C)", color='blue')
plt.xlabel('Час')
plt.ylabel('Температура')
plt.title('Зміна температури у часі')
plt.grid(True)
plt.show()
```

Етап 2. Реалізація методу лінійної регресії.

Було реалізовано два варіанти лінійної регресії: з використанням градієнтного спуску та з L2-регуляризацією.

Лістинг 3.2 Клас для лінійної регресії з градієнтним спуском:

```
class LinearRegressionGD:
    def __init__(self, learning_rate=0.01, n_iterations=1000):
        self.learning_rate = learning_rate
        self.n_iterations = n_iterations
        self.theta = None

    def fit(self, X, y):
        X_b = np.c_[np.ones((X.shape[0], 1)), X]
        m = X_b.shape[0]
        self.theta = np.random.randn(X_b.shape[1], 1)

        for iteration in range(self.n_iterations):
            gradients = 2/m * X_b.T.dot(X_b.dot(self.theta) - y)
            self.theta -= self.learning_rate * gradients
```

```
    def predict(self, X):
        X_b = np.c_[np.ones((X.shape[0], 1)), X]
        return X_b.dot(self.theta)
```

Лістинг 3.3 Реалізація лінійної регресії з L2-регуляризацією:

```
class LinearRegressionGD:
    def __init__(self, learning_rate=0.01, n_iterations=1000, lambda_reg=0.1):
        self.learning_rate = learning_rate
        self.n_iterations = n_iterations
        self.lambda_reg = lambda_reg  # Лямбда для L2-регуляризації
        self.theta = None

    def fit(self, X, y):
        m = len(y)
        self.theta = np.zeros(X.shape[1])
        for _ in range(self.n_iterations):
            gradients = -2/m * X.T.dot(y - X.dot(self.theta)) + 2 * self.lambda_reg * self.theta
            self.theta -= self.learning_rate * gradients

    def predict(self, X):
        return X.dot(self.theta)
```

Етап 3. Реалізація дерева рішень.

Було розроблено клас, який реалізує деревом рішень для регресії. В роботі використовується MSE для оцінки якості розбиттів і побудови дерева. Метод *fit* запускає процес побудови дерева. Він викликає допоміжну функцію, яка будує дерево, починаючи з кореня.

Дані рекурсивно розбиваються на вузли до досягнення умов зупинки.

Лістинг 3.4 Метод дерева рішень:

```
class DecisionTreeClassifier:
    def __init__(self, max_depth=None):
        self.max_depth = max_depth
        self.tree = None

    def fit(self, X, y):
        self.tree = self._build_tree(X, y)

    def _build_tree(self, X, y, depth=0):
        if len(set(y)) == 1:
            return np.mean(y)

        if self.max_depth is not None and depth >= self.max_depth:
            return np.mean(y)

        best_split = self._best_split(X, y)
        left_tree = self._build_tree(X[best_split['left_indices']], y[best_split['left_indices']], depth + 1)
        right_tree = self._build_tree(X[best_split['right_indices']], y[best_split['right_indices']], depth + 1)

        return {
            'split': best_split,
            'left': left_tree,
            'right': right_tree
        }
```

```
def _best_split(self, X, y):
    best_split = None
    min_mse = float('inf')
    for feature_index in range(X.shape[1]):
        thresholds = np.unique(X[:, feature_index])
        for threshold in thresholds:
            left_indices = X[:, feature_index] <= threshold
            right_indices = ~left_indices
            left_y = y[left_indices]
            right_y = y[right_indices]
            mse = self._calculate_mse(left_y, right_y)

            if mse < min_mse:
                min_mse = mse
                best_split = {
                    'feature_index': feature_index,
                    'threshold': threshold,
                    'left_indices': left_indices,
                    'right_indices': right_indices
                }

    return best_split

def _calculate_mse(self, left_y, right_y):
    left_mse = np.mean((left_y - np.mean(left_y)) ** 2)
    right_mse = np.mean((right_y - np.mean(right_y)) ** 2)
    return left_mse + right_mse

def predict(self, X):
    predictions = np.array([self._predict_single(x, self.tree) for x in X])
```

```
        return predictions

    def _predict_single(self, x, tree):
        if isinstance(tree, dict):
            if x[tree['split']['feature_index']] <= tree['split']['threshold']:
                return self._predict_single(x, tree['left'])
            else:
                return self._predict_single(x, tree['right'])
        else:
            return tree
```

Етап 4. Реалізація рекурентної нейронної мережі.

Було використано Long Short-Term Memory модель RNN, у яку було послідовно додано три шари. Кількість нейронів, кількість епох та розмір пакета можна змінювати від час тренування задля досягнення найбільш точних результатів.

Лістинг 3.5 Налаштування рекурентної нейронної мережі:

```
model = Sequential()
model.add(LSTM(units=50, return_sequences=False, input_shape=(X_train.shape[1], X_train.shape[2])))
model.add(Dropout(0.2))
model.add(Dense(units=1))
model.compile(optimizer='adam', loss='mean_squared_error')
history = model.fit(X_train, y_train, epochs=50, batch_size=32, validation_data=(X_test, y_test))
```

Етап 5. Навчання моделей.

Етап 6. Тестування отриманих моделей.

Етап 7. Візуалізація та вивід результатів.

Етапи 5 – 7 є критичними для забезпечення результатів дослідження. З метою отримання найбільш точних прогнозів ці етапи будуть повторюватися після внесення змін у параметри моделей.

3.3 Дослідження та критеріальний порівняльний аналіз обраних методів прогнозування погодних умов на основі метеорологічних даних

Кожен з етапів роботи був записаний у окремий блок в блокноті Google Colab. Візуалізацію температури, прогнозування якою досліджувалося, наведено на рисунку 3.2.

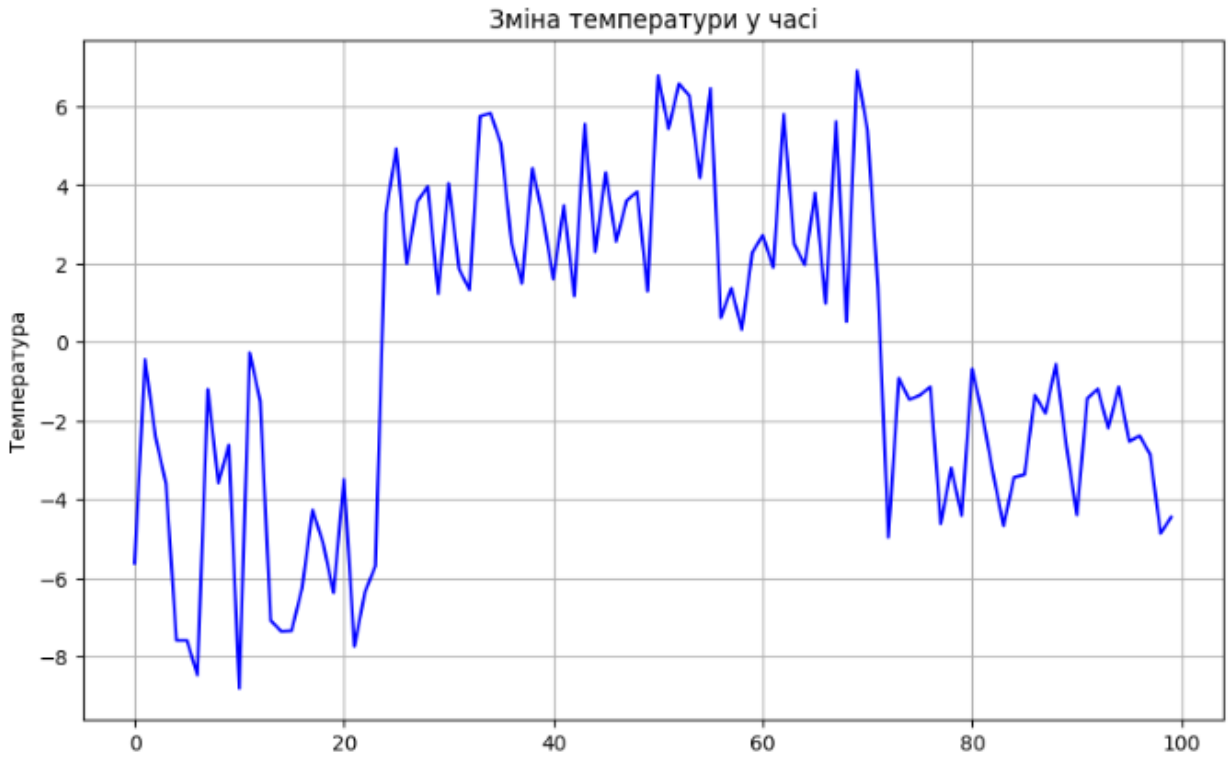


Рисунок 3.2 – Візуалізація даних (температури)

Спочатку запускається блок ініціалізації, в якому прописано усі бібліотеки, що будуть використовуватися, а також завантажуються та передоброблюються дані.

Для дослідження роботи моделей використовувалась вибірка даних зі 100 записів. 80% використовувалися для навчання і 20% – для тестування точності прогнозів.

На рисунках 3.3 та 3.4 представлено результати роботи лінійної регресії.

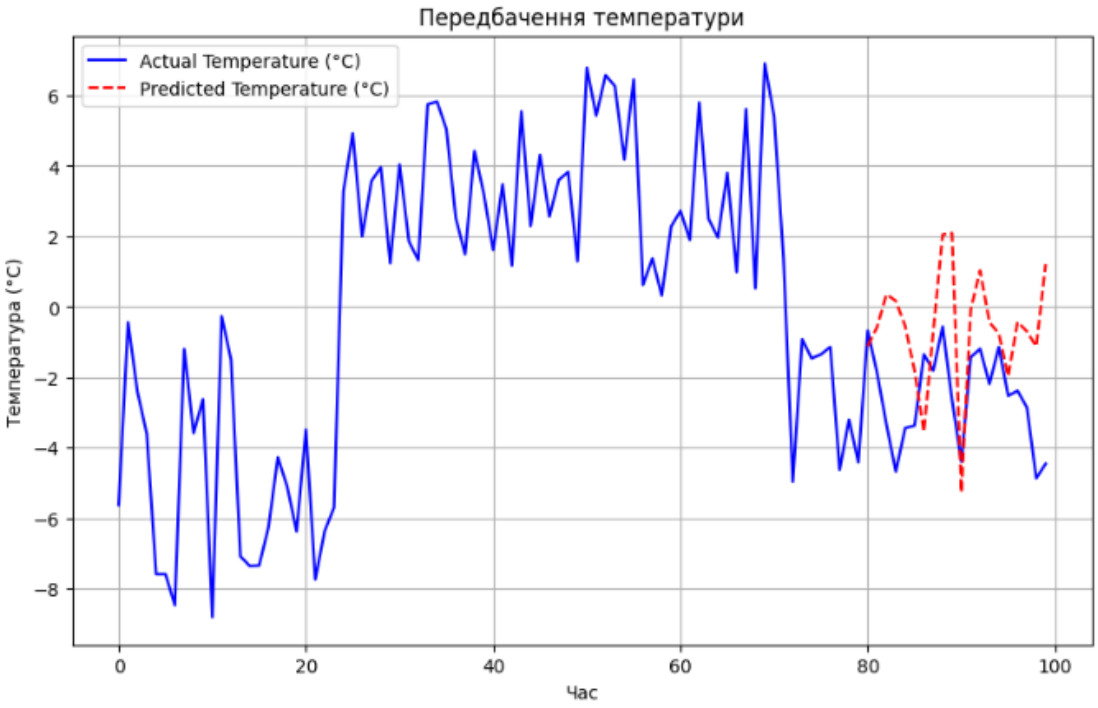


Рисунок 3.3 – Робота лінійної регресії з градієнтним спуском

Зміна параметрів *learning_rate* та *n_iterations* призводила до незначних коливань значення середньоквадратичної помилки. Найбільш точного прогнозування було досягнуто при *learning_rate* = 0,01, *n_iterations* = 1500.

Якщо значення *learning_rate* велике, модель починає «переплигувати» через оптимальні значення під час навчання. Передбачення стають менш точними. Мале значення призводить до повільного навчання, що затягує процес оптимізації та призводить до менш точного передбачення на початкових етапах навчання. Збільшення кількості ітерацій покращує точність, оскільки модель краще адаптується до вхідних даних, велика кількість веде до перенавчання і погіршує передбачення на тестових даних.

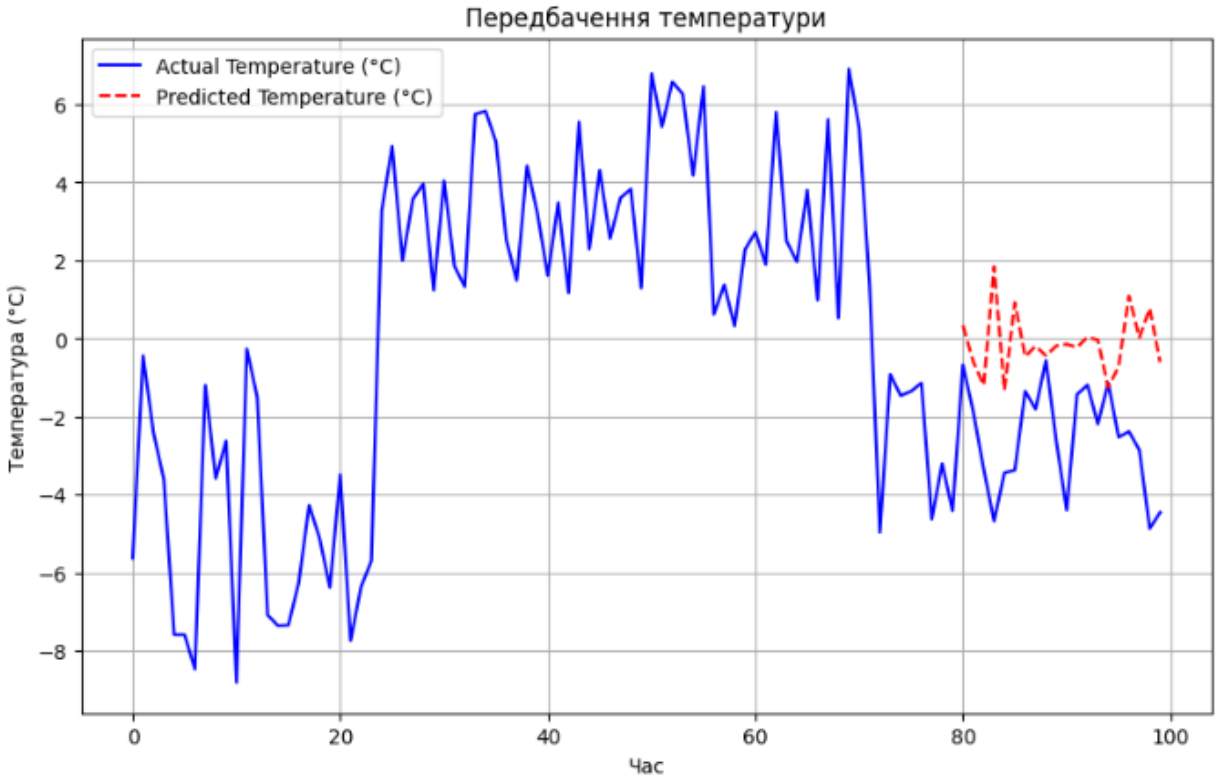


Рисунок 3.4 – Робота лінійної регресії з L2-регуляризацією

На графіках можна помітити, що використання L2-регуляризації забезпечує більш плавний та менш схильний до шумів прогноз. Модель з регуляризацією має кращу робастність і менше перенавчається на тренувальних даних, тому прогнози виглядають більш близькими до реальних.

Найкращі отримані значення точності для лінійної регресії наведені в таблиці 3.1.

Таблиця 3.1 – Результати прогнозів лінійної регресії

Тип оптимізації	**MSE**	**MAE**	**Час виконання**
Градієнтний спуск	22,07	3,81	0 с (дуже швидко)
L2-регуляризація	14,32	3,06	1 с (швидко)

На рисунку 3.5 представлено результати роботи дерева рішень. Графіки важливості критеріїв наведені на рисунках 3.6 та 3.7.

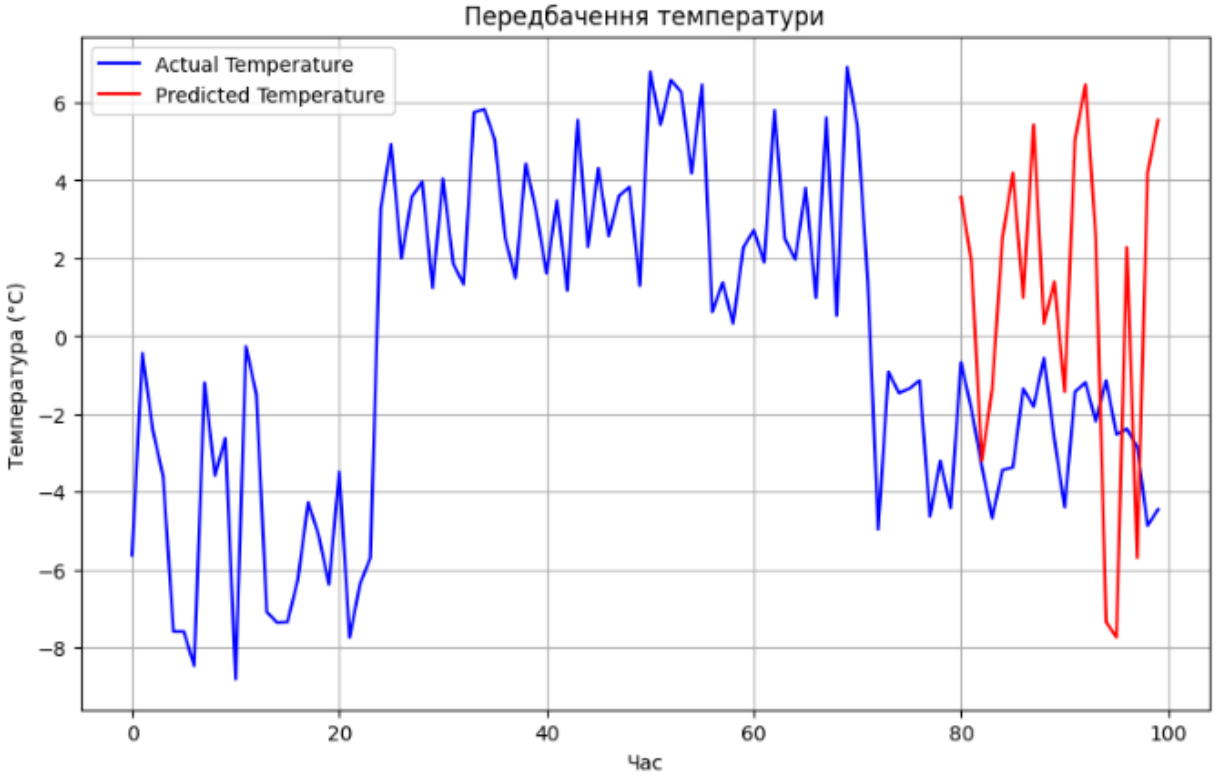


Рисунок 3.5 – Результат прогнозів дерева рішень

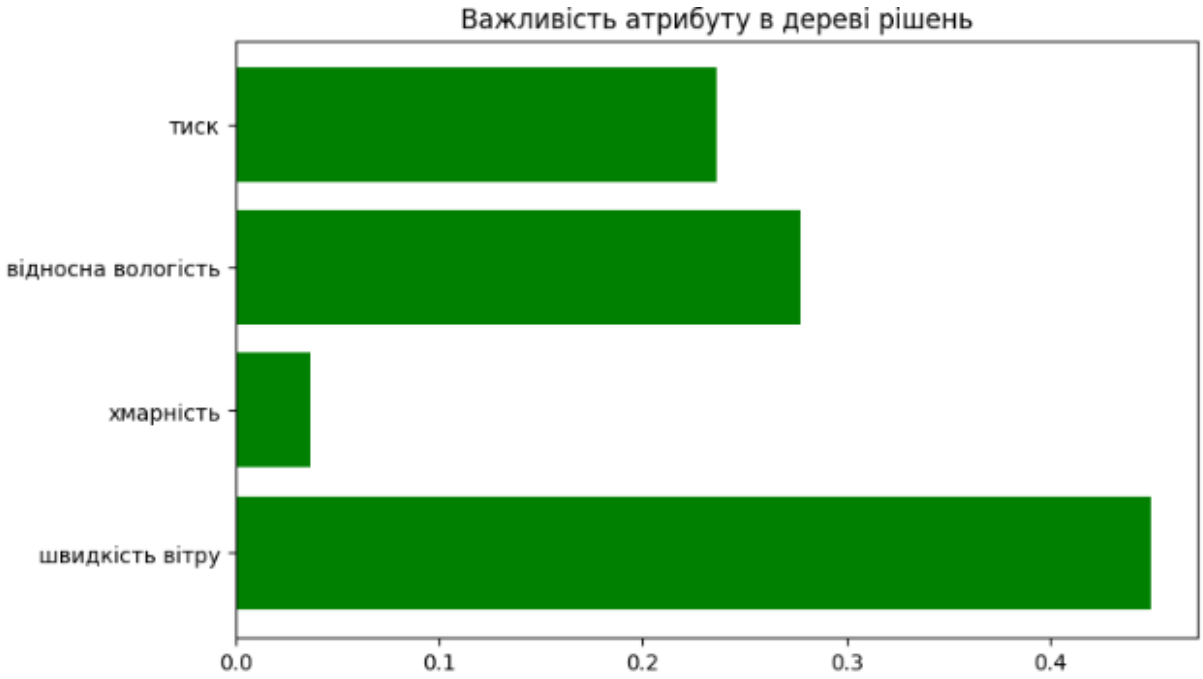


Рисунок 3.6 – Важливість атрибуту (коефіцієнт Джині)

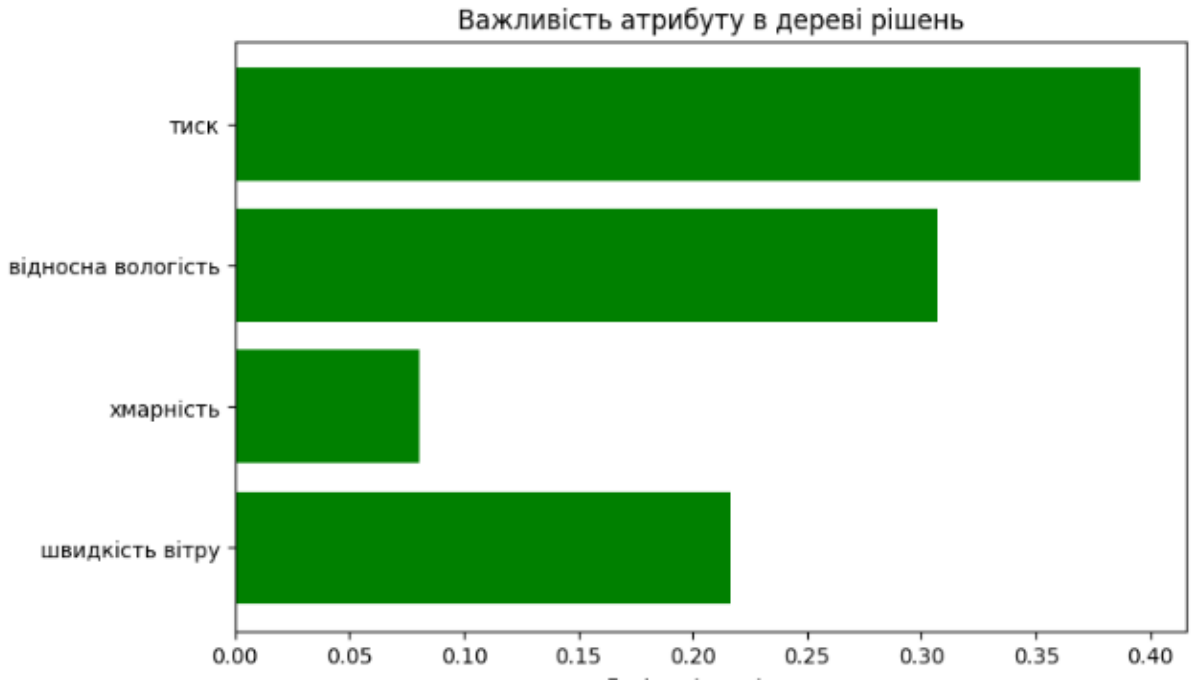


Рисунок 3.7 – Важливість атрибуту (ентропія)

Найкращі отримані значення точності для дерева рішень було досягнуто для моделі, яка приймала рішення на основі критерію ентропії. При використанні коефіцієнту Джині, модель показувала нижчу точність на близько 5%.

Досягнуто показників:

– MSE = 1,6258541312687644 (1,63);

– MAE = 1,0788740301927775 (1,08).

Час роботи був однаковий – близько 2 секунд.

На рисунку 3.8 представлено результати роботи RNN. Графік побудовано для масштабованих даних. Результати дослідження впливу основних параметрів мережі наведено в таблицях 3.2 – 3.4.

Таблиця 3.2 – Вплив на результати кількості нейронів в шарі

Кількість нейронів	**MSE**	**MAE**	**Час виконання**
20	0,03	0,17	7 с
50	0,03	0,17	8 с
100	0,02	0,15	9 с
200	0,03	0,16	12 с

Рисунок 3.8 – Результат роботи RNN

Таблиця 3.3 – Вплив на результати кількості епох в навчанні

Кількість нейронів	**MSE**	**MAE**	**Час виконання**
10	0,04	0,21	4 с
50	0,02	0,15	9 с
100	0,02	0,16	13 с
200	0,03	0,16	23 с

Таблиця 3.4 – Вплив на результати кількості зразків, що оброблюються за один крок

Кількість зразків	**MSE**	**MAE**	**Час виконання**
2	0,02	0,16	17 с
32	0,02	0,15	10 с
80	0,04	0,20	8 с

Після дослідження впливу кількості нейронів на результат прогнозування, для результуючої мережі було обрано показник в 100 нейронів.

Після дослідження впливу кількості епох на результат прогнозування, для результуючої мережі було обрано показник в 50 епох, а кількість зразків, що оброблюється за один крок, – 32.

Досягнуто показників:

– MSE = 0,023067022184496216 (0,02);

– MAE = 0,1518783137399682 (0,15).

Час роботи – близько 9 секунд. Порівняння розроблених методів наведено в таблиці 3.5.

Таблиця 3.5 – Критеріальний порівняльний аналіз обраних методів прогнозування погодних умов

Критерій	**Лінійна регресія**	**Дерево рішень**	**RNN**
Складність реалізації	Низька. Легко реалізувати, потребує мінімум налаштувань	Середня. Потрібна попередня обробка та налаштування параметрів дерева	Висока. Вимагає значної обробки даних, складної структури та параметризації
MAE	3,06	1,08	0,15
MSE	14,32	1,63	0,02
Швидкість навчання	Дуже висока	Помірна (залежить від набору даних)	Повільна (залежить від набору даних)
Швидкість прогнозу	Дуже висока	Висока	Помірна (залежить від будови мережі)
Робастність	Слабка	Помірна	Висока
Якість роботи з часовими рядами	Низька	Помірна	Висока
Інтерпретація результатів	Висока, добре інтерпретується	Помірна, складна інтерпретація на великих глибинах	Низька, модель важко інтерпретувати без додаткових інструментів

Для проведення остаточного аналізу результатів кожного методу прогнозування доцільно ввести тестову шкалу за кожним із критеріїв, що дозволить об’єктивно порівняти їхню ефективність.

Бали для оцінки результатів за кожним із критеріїв наведені в таблиці 3.6.

Таблиця 3.6 – Оцінка критеріїв

Критерій	**Низький показник**	**Середній показник**	**Високий показник**
Складність реалізації	1	0,5	0
MSE	(більше 5) 0	(від 1 до 5 включно) 1,5	(менше одиниці) 3
MAE	(більше 2) 0	(від 0,5 до 1) 1,5	(менше 0,5) 3
Швидкість навчання	0	0,5	1
Швидкість прогнозу	0	0,5	1
Робастність	0	1,5	3
Якість роботи з часовими рядами	0	1,5	3
Інтерпретація результатів	0,5	1	2

Результати оцінювання методів наведені в таблиці 3.7.

Таблиця 3.7 – Результати оцінки

Критерій	**Лінійна регресія**	**Дерево рішень**	**RNN**
Складність реалізації	1	0,5	0
MAE	0	1,5	3
MSE	0	1,5	3
Швидкість навчання	1	0,5	0
Швидкість прогнозу	1	0,5	0
Робастність	0	1,5	3
Якість роботи з часовими рядами	0	1,5	3
Інтерпретація результатів	2	1	0,5
Результат	**Оцінка «5»**	**Оцінка «9»**	**Оцінка «13»**

Отже, проведений аналіз показав, що кожен з методів має свої унікальні переваги та недоліки. Лінійна регресія відрізняється простотою реалізації та високою швидкістю навчання і прогнозування, що робить її зручною для швидких і базових прогнозів. Однак, цей метод демонструє найнижчу точність у порівнянні з іншими підходами, що обмежує його застосування у задачах з високими вимогами до якості прогнозів.

Дерево рішень займає проміжну позицію за багатьма критеріями, демонструючи кращу точність та робастність, ніж лінійна регресія, проте поступається RNN у роботі з часовими рядами. Дерево рішень забезпечує середній баланс між швидкістю, точністю та стабільністю, що робить його гарним вибором для випадків, де необхідна певна гнучкість та зрозумілість результатів.

Рекурентна нейронна мережа продемонструвала найвищу точність і робастність, що особливо важливо для роботи з часовими рядами. Проте варто враховувати, що реалізація методу складніша, і потребується більше часу на навчання та прогнозування.

Таким чином, вибір методу залежить від конкретних вимог до точності, швидкості та складності задачі. Висока здатність до роботи з послідовними даними робить RNN найкращим варіантом для прогнозування погодних умов, хоча цей метод вимагає значних обчислювальних ресурсів і є менш інтерпретованим.

3.4 Перспективи подальшої роботи

Розроблені методи та визначені особливості можна використовувати для подальших досліджень. З огляду на результати, є доцільним продовження роботи над оптимізацією обраних моделей, або над подальшим покращенням задля отримання максимальної точності прогнозу за допомогою методу рекурентних нейронних мереж.

Робота може бути розширена дослідженням інших алгоритмів, таких як градієнтний бустінг, глибокі нейронні мережі та інші. Це дозволить отримати ширший спектр підходів і краще розуміння сильних та слабких сторін кожного методу. Розширити також можливо і набор вхідних даних новими метеорологічними параметрами, більшою кількістю записів. Аналіз кореляції між параметрами допоможе краще зрозуміти взаємозв'язки, що впливають на точність прогнозів, і дозволить будувати більш точні моделі.

Проведене дослідження може бути корисним для подальшого розвитку моделей на основі ансамблів, таких як випадкові. Застосування ансамблевих методів дозволить комбінувати сильні сторони різних моделей, що може призвести до підвищення загальної точності та робастності прогнозування. Також ансамблі здатні зменшити вплив окремих похибок і, таким чином, підвищити стійкість прогнозів до шуму та зміни вхідних даних.

Ще одним перспективним напрямком, для якого результати проведеного дослідження можуть стати основою, є розробка прототипу інтегрованої системи, яка автоматично здійснюватиме прогнозування погодних умов, використовуючи різні моделі в залежності від характеру даних та вимог до точності і швидкості. Така система дозволить автоматично обирати найбільш оптимальний метод для кожного конкретного випадку.

ВИСНОВКИ

У рамках кваліфікаційної роботи було розроблено та досліджено три методи навчання для прогнозування погодних умов на основі метеорологічних даних. Проведено аналіз ефективності трьох різних підходів до прогнозування. Розглянуто їх здатність до моделювання складних патернів у даних та адаптації до змінюваних умов.

Під час теоретичного дослідження було визначено найбільш відомі та широко використовувані методи машинного навчання, які застосовуються для різних задач, таких як класифікація [29–32], кластеризація [33, 34], прогнозування та аналіз даних [35, 36]. Кожен з розглянутих методів демонструє свою ефективність у певних ситуаціях. Наприклад, дерева рішень мають значний потенціал для побудови моделей класифікації, де важлива складна структура даних [37, 38], а згорткові нейронні мережі [39–41] можуть бути використані для аналізу та класифікації [42, 43] візуальних даних, зокрема для обробки зображень, відео [36]. Для дослідження якості прогнозування погодних умов було обрано три методи: лінійна регресія, дерева рішень, рекурентні нейронні мережі.

На основі сформованої постановки задачі, було обрано інструменти для програмної реалізації та реалізовано застосунок на платформі Google Colab. Було проведення дослідження особливостей обраних методів в контексті задачі прогнозування температури на основі інших погодних умов.

Проведений аналіз показав, що усі моделі здатні виконати прогнозування погоди з певним рівнем точності, проте рекурентна нейронна мережа забезпечила найбільш точні прогнози, хоча й потребувала більше часу на навчання порівняно з іншими.

Наукова новизна полягає у дослідженні особливостей алгоритмів для прогнозування погоди, що використовують різні підходи до моделювання та аналізу метеорологічних даних. Виявлено, що рекурентні нейронні мережі, незважаючи на більший час навчання, демонструють найбільшу точність

прогнозування, що вказує на їх потенціал для вирішення задач прогнозування погодних умов у реальних умовах. Запропоновано критерії для порівняння моделей, що дозволяють точно оцінювати ефективність методів в контексті задачі прогнозування температури на основі інших погодних параметрів.

Практична значущість дослідження включає:

– створення моделей прогнозування, які можуть бути застосовані для аналізу наборів метеорологічних даних;

– розробку програмних засобів для інтеграції запропонованих методів у системи прогнозування погоди та системи підтримки прийняття рішень у метеорології.

Дослідження має важливе значення для подальшого розвитку алгоритмів машинного та глибокого навчання в контексті метеорології та їх практичного застосування в системах підтримки прийняття рішень. Результати можуть слугувати основою для розробки інтегрованих систем прогнозування, що автоматично вибирають оптимальні моделі в залежності від характеристик даних та вимог до точності і швидкості прогнозу.

Отримані результати можуть бути використані для покращення якості прогнозів погоди та оптимізації обчислювальних витрат у реальних умовах, що робить дослідження актуальним та прикладним для метеорологічних служб. Результати роботи можуть стати основою для подальших досліджень у галузі машинного та глибокого навчання [44], ансамблів моделей та інших алгоритмів. Перспективним напрямком є розробка прототипу інтегрованої системи, яка автоматично здійснюватиме прогнозування погодних умов, використовуючи різні моделі в залежності від характеру даних та вимог до точності і швидкості.

Результати дослідження апробовано у вигляді тез доповіді під час Міжнародного молодіжного форуму «РАДІОЕЛЕКТРОНІКА І МОЛОДЬ У XXI СТОЛІТТІ» [45].

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Що таке машинне навчання: як працює та де використовується. GigaCloud: Хмарні Технології та Хмарний Сервіс для Бізнесу. URL: https://gigacloud.ua/blog/navchannja/scho-take-mashinne-navchannja-jak-pracjue-ta-de-vikoristovuetsja (дата звернення 12.11.2024).

2. Машинне навчання простими словами. Частина 1. Механіко-математичний факультет Львівського національного університету імені Івана Франка. URL: https://mmf.com.ua/ar/1739 (дата звернення 12.11.2024).

3. Економетрична модель Національного банку України для оцінки кредитного ризику банку та альтернативний метод опорних векторів. Visnyk of the National Bank of Ukraine. URL: https://journal.bank.gov.ua/ua/article/2015/234/03 (дата звернення 12.11.2024).

4. Conti, S. (2024). Artificial intelligence for weather forecasting. Nature Reviews Electrical Engineering, 1(1), 8-8.

5. Ben Bouallègue, Z., Clare, M. C., Magnusson, L., Gascon, E., Maier-Gerber, M., Janoušek, M., ... & Pappenberger, F. (2024). The Rise of Data-Driven Weather Forecasting: A First Statistical Assessment of Machine Learning–Based Weather Forecasts in an Operational-Like Context. Bulletin of the American Meteorological Society, 105(6), E864-E883.

6. Li, L., Carver, R., Lopez-Gomez, I., Sha, F., & Anderson, J. (2024). Generative emulation of weather forecast ensembles with diffusion models. Science Advances, 10(13), eadk4489.

7. Ling, F., Ouyang, L., Larbi, B. R., Luo, J. J., Han, T., Zhong, X., & Bai, L. (2024). Is Artificial Intelligence Providing the Second Revolution for Weather Forecasting?. arXiv preprint arXiv:2401.16669.

8. babu Nuthalapati, S., & Nuthalapati, A. (2024). Accurate weather forecasting with dominant gradient boosting using machine learning. International Journal of Science and Research Archive, 12(2), 408-422.

9. Kochkov, D., Yuval, J., Langmore, I., Norgaard, P., Smith, J., Mooers, G., ... & Hoyer, S. (2024). Neural general circulation models for weather and climate. Nature, 1-7.

10. Calvo-Olivera, C., Guerrero-Higueras, Á. M., Lorenzana, J., & García-Ortega, E. (2024). Real-Time Evaluation of the Uncertainty in Weather Forecasts Through Machine Learning-Based Models. Water Resources Management, 38(7), 2455-2470.

11. Jones, C. B. (2024). ACCURATE WEATHER FORECASTING OVER WIDE DATASETS USING MACHINE LEARNING MODELS. ICTACT Journal on Soft Computing, 15(1).

12. Nguyen, T., Jewik, J., Bansal, H., Sharma, P., & Grover, A. (2024). Climatelearn: Benchmarking machine learning for weather and climate modeling. Advances in Neural Information Processing Systems, 36.

13. Gibson, P. B., Chapman, W. E., Altinok, A., Delle Monache, L., DeFlorio, M. J., & Waliser, D. E. (2021). Training machine learning models on climate model output yields skillful interpretable seasonal precipitation forecasts. Communications Earth & Environment, 2(1), 159.

14. SINOPTIK: Погода в Україні, детальний прогноз погоди на тиждень. Погода сьогодні, завтра в Україні та Світі. Sinoptik. URL: https://ua.sinoptik.ua/ (дата звернення 12.11.2024).

15. Погода в Україні та світі. METEOFOR. URL: https://meteofor.com.ua/ (дата звернення 12.11.2024).

16. Лінійна регресія – Практикування онлайн – Znaiemo Informatyku. Ми знаємо інформатику. URL: https://www.znaiemoinformatyku.org/liniina-rehresiia (дата звернення 12.11.2024).

17. Label Encoding in Python – GeeksforGeeks. GeeksforGeeks. URL: https://www.geeksforgeeks.org/ml-label-encoding-of-datasets-in-python/ (дата звернення 12.11.2024).

18. Бондаренко Сергій. Що таке лінійна регресія: повний гайд від robot_dreams. robot_dreams – онлайн-курси для фахівців у сфері big data,

machine learning, data science | Робот Дрімс. URL: https://robotdreams.cc/uk/blog/437-shcho-take-liniyna-regresiya (дата звернення 12.11.2024).

19. Linear Regression. Welcome | Department of Statistics and Data Science. URL: http://www.stat.yale.edu/Courses/1997-98/101/linreg.htm (дата звернення 12.11.2024).

20. IBM. What is a Decision Tree? | IBM. IBM – United States. URL: https://www.ibm.com/topics/decision-trees (дата звернення 12.11.2024).

21. Decision Tree Algorithm, Explained. URL: https://www.kdnuggets.com/2020/01/decision-tree-algorithm-explained.html (дата звернення 12.11.2024).

22. Jain A. All about decision trees. Medium. URL: https://medium.com/@abhishekjainindore24/all-about-decision-trees-80ea55e37fef (дата звернення 12.11.2024).

23. Прогнозування за допомогою нейронних мереж – Wiki ТНТУ. Wiki ТНТУ. URL: https://wiki.tntu.edu.ua/Прогнозування_за_допомогою_нейронних_мереж (дата звернення 12.11.2024).

24. Recurrent Neural Network: Types and Advantages | BotPenguin. Free Chatbot maker | Chatbot for Website, WhatsApp | BotPenguin. URL: https://botpenguin.com/glossary/recurrent-neural-network (дата звернення 12.11.2024).

25. IBM. What is a Recurrent Neural Network (RNN)? | IBM. IBM - United States. URL: https://www.ibm.com/topics/recurrent-neural-networks (дата звернення 12.11.2024).

26. Методи прогнозування погоди. StudFiles. URL: https://studfile.net/preview/6048011/page:7/ (дата звернення 12.11.2024).

27. GeeksforGeeks. 10 Best Language for Machine Learning – GeeksforGeeks. GeeksforGeeks. URL: https://www.geeksforgeeks.org/best-language-for-machine-learning/ (дата звернення 14.11.2024).

28. Яка найкраща мова для машинного навчання? (листопад 2024). Unite.AI. URL: https://www.unite.ai/uk/ (дата звернення 14.11.2024).

29. Daradkeh Y.I., Gorokhovatskyi V., Tvoroshenko I., and Zeghid M. (2024) Improving the effectiveness of image classification structural methods by compressing the description according to the information content criterion, *Computers, Materials & Continua*, vol. 80, no. 2, pp. 3085-3106.

30. Gorokhovatskyi V., Tvoroshenko I., and Yakovleva O. (2024) Transforming image descriptions as a set of descriptors to construct classification features, *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 33, no. 1, pp. 113-125.

31. Gorokhovatskyi V., Tvoroshenko I., and Yakovleva O. (2024) Transforming image descriptions as a set of descriptors to construct classification features, *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 33, no. 1, pp. 113-125.

32. Гороховатський В., Передрій О., Творошенко І., Марков Т. (2023) Матриця відстаней для множини компонентів структурного опису як інструмент для створення класифікатора зображень, *Сучасні інформаційні системи*, 7(1), С. 5-13.

33. Gorokhovatskyi, V., Tvoroshenko, I., Kobylin, O., & Vlasenko, N. (2023). Search for visual objects by request in the form of a cluster representation for the structural image description, *Advances in Electrical and Electronic Engineering*, 21(1), pp. 19-27.

34. Gorokhovatskyi V., Tvoroshenko I. (2023) Identification of visual objects by the search request. *International scientific symposium «INTELLIGENT SOLUTIONS-S». Computational intelligence (results, problems and perspectives). Decision making theory: proceedings of the international symposium, September 28, 2023, Kyiv-Uzhorod, Ukraine*, pp. 25-27.

35. Tvoroshenko I., and Gorokhovatskyi V. (2022) The Application of Hybrid Intelligence Systems for Dynamic Data Analysis, *International Journal of Engineering and Information Systems*, 6(2), pp. 40-48.

36. Yakovleva O., Matúšová S., Tvoroshenko I., and Isaiev Y. (2024) Visitor counting based on video stream analysis from surveillance cameras to solve various business problems, *Verejná správa a regionálny rozvoj ekonómia, manažment a marketing*, XX(1), pp. 67-87.

37. Gorokhovatskyi V., Tvoroshenko I., Yakovleva O., Hudáková M., and Gorokhovatskyi O. (2024) Application a committee of Kohonen neural networks to training of image classifier based on description of descriptors set, *IEEE Access*, vol. 12, pp. 73376-73385.

38. Daradkeh Y.I., Gorokhovatskyi V., Tvoroshenko I., Gadetska S., and Al-Dhaifallah M. (2023) Statistical data analysis models for determining the relevance of structural image descriptions, IEEE Access, vol. 11, pp. 126938-126949.

39. Pomazan V., Tvoroshenko I., and Gorokhovatskyi V. (2023) Handwritten character recognition models based on convolutional neural networks, International Journal of Academic Engineering Research, 7(9), pp. 64-72.

40. Tvoroshenko I., Pomazan V., Gorokhovatskyi V., and Kobylin O. (2023) Application of video data classification models using convolutional neural networks, International Journal of Academic and Applied Research, 7(11), pp. 134-145.

41. Pomazan, V., Tvoroshenko, I., & Gorokhovatskyi, V. (2023). Development of an application for recognizing emotions using convolutional neural networks, *International Journal of Academic Information Systems Research*, 7(7), pp. 25-36.

42. Daradkeh Y.I., Gorokhovatskyi V., Tvoroshenko I., and Zeghid M. (2022) Cluster representation of the structural description of images for effective classification, *Computers, Materials & Continua*, 73(3), pp. 6069-6084.

43. Гороховатський В.О., Творошенко І.С., Чмутов Ю.В. (2022) Застосування систем ортогональних функцій для формування простору ознак у методах класифікації зображень, *Сучасні інформаційні системи*, 6(3), С. 5-12.

44. Tvoroshenko I., Gorokhovatskyi V., Kobylin O., and Tvoroshenko A. (2023) Application of deep learning methods for recognizing and classifying culinary dishes in images, International Journal of Academic and Applied Research, 7(9), pp. 57-70.

45. Меженна. І. (2024) Особливості сучасних методів машинного навчання щодо вирішення задачі прогнозування даних, Тези доповідей 28-го Міжнародного молодіжного форуму «Радіоелектроніка і молодь у XXI столітті» (Травень, 2024). Харків, Україна, С. 87-89.