МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ЛЬВІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ПРИРОДОКОРИСТУВАННЯ
ФАКУЛЬТЕТ МЕХАНІКИ, ЕНЕРГЕТИКИ ТА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
КАФЕДРА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

КВАЛІФІКАЦІЙНА РОБОТА

другого (магістерського) рівня вищої освіти

на тему: **«Розробка інтелектуальної інформаційної системи прогнозування кількості опадів на основі балансування даних та використання моделі машинного навчання»**

Виконав: студент групи Іт-62

Спеціальності 126 «Інформаційні системи та технології»
(шифр і назва)
Куць Ярослав Ігорович
(Прізвище та ініціали)

Керівник: к.т.н., доцент Татомир А.В.
(Прізвище та ініціали)

Рецензент: к.т.н., доцент Шолудько Я.В.
(Прізвище та ініціали)

ДУБЛЯНИ-2024

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ЛЬВІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ПРИРОДОКОРИСТУВАННЯ
ФАКУЛЬТЕТ МЕХАНІКИ, ЕНЕРГЕТИКИ ТА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
КАФЕДРА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

Другий (магістерський) рівень вищої освіти
Спеціальність 126 «Інформаційні системи та технології»

«ЗАТВЕРДЖУЮ»

Завідувач кафедри_____

д.т.н., проф. А.М. Тригуба
«____» _____2024 р.

ЗАВДАННЯ

на кваліфікаційну роботу студенту

Куцю Ярославу Ігоровичу

1. Тема роботи: «Розробка інтелектуальної інформаційної системи прогнозування кількості опадів на основі балансування даних та використання моделі машинного навчання»

Керівник роботи Татомир Андрій Володимирович, доцент
затверджені наказом по університету від 12.09.2024 року № 616/к-с.

2. Строк подання студентом роботи 10.12.2024 р.

3. Вихідні дані до роботи: *дані для прогнозування кількості опадів на основі балансування даних та використання моделі машинного навчання; алгоритми створення моделей машинного навчання.*

4. Зміст розрахунково-пояснювальної записки (перелік питань, які необхідно розробити)
Вступ.
Аналіз стану об'єкту дослідження в теорії та практиці.
Прогнозування кількості опадів на основі балансування даних та обгрунтування моделі машинного навчання.
Результати розробки інтелектуальної інформаційної системи прогнозування кількості опадів.
Охорона праці та безпека у надзвичайних ситуаціях.
Економічна ефективність від використання інтелектуальної інформаційної системи прогнозування кількості опадів.
Висновки та пропозиції.
Список використаної літератури.

5. Перелік ілюстраційного матеріалу (з точним зазначенням обов»язкових слайдів): *аналіз стану об'єкту дослідження в теорії та практиці; прогнозування кількості опадів на основі балансування даних та обгрунтування моделі машинного навчання; результати розробки інтелектуальної інформаційної системи прогнозування кількості опадів; економічна ефективність від використання інтелектуальної інформаційної системи прогнозування кількості опадів.*

6. Консультанти з розділів:

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
1, 2, 3, 5	*Татомир А.В., доцент кафедри інформаційних технологій*		
4	*Городецький І.М., доцент кафедри фізики, інженерної графіки та безпеки виробництва*		

7. Дата видачі завдання 12 вересня 2024 р.

Календарний план

№ з/п	Назва етапів кваліфікаційної роботи	Терміни виконання етапів роботи	Примітка
1	*Написання першого розділу*	*12.09-20.09.24*	
2	*Виконання другого розділу та аркушів ілюстраційного матеріалу до нього*	*21.09-14.10.24*	
3.	*Виконання третього розділу та аркушів ілюстраційного матеріалу до нього*	*15.010-10.11.24*	
4.	*Написання розділу «Охорона праці та безпека у надзвичайних ситуаціях»*	*11.11-20.11.24*	
5.	*Оцінення ефективності запропонованої системи*	*21.11-30.30.24*	
6.	*Завершення оформлення розрахунково-пояснювальної записки та аркушів ілюстраційного матеріалу*	*01-04.12.24*	
7.	*Завершення роботи вцілому*	*05-10.12.24*	

Студент _____ Куць Я.І.
(підпис)

Керівник роботи _____ Татомир А.В.
(підпис)

УДК 004.8: 551.57

Розробка інтелектуальної інформаційної системи прогнозування кількості опадів на основі балансування даних та використання моделі машинного навчання.

Куць Я.І. Кафедра інформаційних технологій – Дубляни, ЛНУП, 2024.

Кваліфікаційна робота: 75 с. текст. част., 18 рис., 7 табл., 14 арк. ілюстраційного матеріалу, 53 джерела.

У роботі подано загальну характеристику сучасних підходів до прогнозування кількості опадів, зокрема в контексті кліматичних умов Львівської області. Проведено аналіз методів прогнозування, включаючи статистичні, фізичні та машинного навчання моделі. Описано особливості балансування даних для забезпечення точності прогнозів та вирішення проблем, пов’язаних із обробкою кліматичних даних.

Розглянуто процес підготовки наборів даних для прогнозування, обґрунтовано вибір методів машинного навчання та досліджено вплив балансування даних на точність прогнозів. На основі отриманих результатів розроблено моделі прогнозування кількості опадів, що демонструють ефективність у різних кліматичних умовах.

У роботі представлено архітектуру інтелектуальної інформаційної системи прогнозування кількості опадів, обрано технологічний стек для її реалізації та розроблено алгоритми обробки вхідних даних. Запропоновано інтерфейс користувача, який дозволяє візуалізувати результати прогнозів. Реалізовано серверну частину системи з використанням Flask та фронтенд на основі React. Оцінено економічну ефективність впровадження системи та розроблено рекомендації для забезпечення безпеки праці виконавців.

Ключові слова: прогнозування опадів, машинне навчання, балансування даних, інтелектуальна інформаційна система, кліматичні дані, економічна ефективність, Flask, React, візуалізація прогнозів.

ЗМІСТ

ВСТУП

На даний час прогнозування кількості опадів є завданням, яке має значний вплив на сільське господарство, управління водними ресурсами, транспортну інфраструктуру та планування екологічних заходів. Проте висока нестабільність кліматичних умов та обмеженість історичних даних для певних регіонів впливають на точність прогнозів [7].

Традиційні методи прогнозування не завжди враховують складні закономірності у великих масивах даних, що призводить до зниження їх ефективності [8]. Використання моделей машинного навчання в поєднанні з технологіями формування даних відкриває нові можливості для підвищення точності прогнозів та розробки адаптивних систем, здатних працювати в умовах неповноти чи нерівномірності даних.

Мета дослідження – розробити інтелектуальну інформаційну систему прогнозування кількості опадів, що використовує методи балансування даних та моделі машинного навчання для підвищення точності та адаптивності прогнозів.

Об’єкт дослідження – процес прогнозування мінімальних опадів у межах задач кліматичного моніторингу.

Предмет дослідження – методи балансування даних та алгоритми машинного навчання, що застосовуються для побудови точних моделей прогнозування.

У роботі для проведення для дослідження використано методи збору та аналізу даних щодо кліматичних джерел, методи балансування даних, моделі машинного навчання та методи оцінки точності моделей за допомогою метрики RMSE, MAE, R^2.

Результати роботи будуть корисними для створення систем екологічного моніторингу, управління сільськогосподарськими процесами, попередження стихійних лих, таких як повені. Розроблена інтелектуальна інформаційна система забезпечує підвищення точності прогнозування опадів, що сприяє

оптимізації використання ресурсів, мінімізації збитків від несприятливих погодних умов та підвищенню ефективності виконання рішень.

Таким чином, виконані дослідження у кваліфікаційній роботі та створена система сприяють розв’язанню актуальних прикладних задач кліматичного прогнозування та відкривають нові можливості для аналізу та використання великих кліматичних даних.

РОЗДІЛ 1.
АНАЛІЗ СТАНУ ОБ'ЄКТУ ДОСЛІДЖЕННЯ В ТЕОРІЇ ТА ПРАКТИЦІ

1.1. Загальна характеристика кліматичних умов Львівської області

Львівська область розташована на заході України і відзначена помірно-континентальним кліматом. Вона охоплює території, що включають низини, пагорби, передгір'я та гірські масиви, що створює різноманіття кліматичних умов. Основними кліматичними характеристиками області є лише м'яка зима, порівняно тепле літо та достатньо велика кількість опадів протягом року.

Температурні показники Львівської області варіюються залежно від висоти над рівнем моря та пори року. Взимку середні температури становлять в межах від -10°C до 0°C. В Карпатах часто спостерігаються сильні морози. Температури в літній період становлять від 18°C до 25°C, у гірських районах прохолодніше. Весна та осінь характеризується значною мінливістю температури, яка становить від 5°C до 18°C.

Рисунок 1.1 – Тенденції зміни середньорічної температури у м. Львів за багаторічний період

Львівська область є одним із найбільших зволожених регіонів України. Річна кількість опадів становить від 600 до 1100 мм, причому максимальні значення фіксуються в Карпатах.

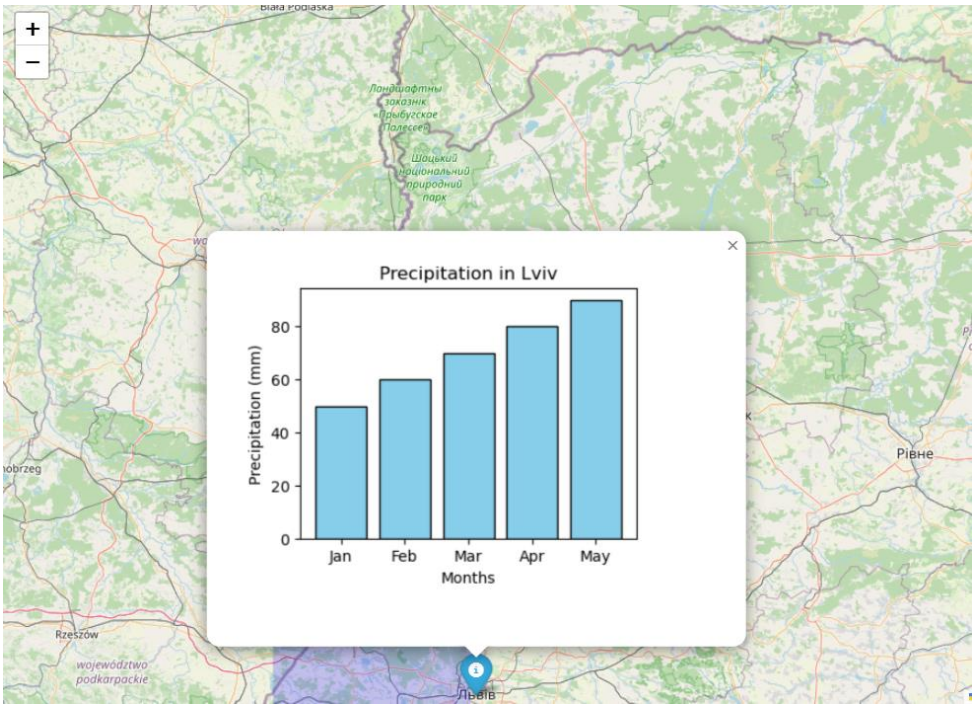

Рисунок 1.2 – Мапа та гістограма кількості опадів у Львові

Весна та осінь характеризується помірною кількістю опадів, їх середньомісячна кількість становить від 50 до 150 мм. Літні місяці (червень-серпень) характеризуються найбільшою кількістю опадів, часто у вигляді злив. Середньомісячні показники можуть досягати 300-500 мм у гірських районах. Взимку опади переважно випадають у вигляді снігу, їх кількість зростає від 10 до 80 мм на місяць.

Вологість повітря в регіонах коливається в межах від 60% до 90% залежно від сезону. Найбільші значення спостерігаються в осінньо-зимовий період, а найменші – влітку під час посуху.

У Львівській області переважають західні та північно-західні вітри. У гірських районах можливості місцеві бризи та сильні пориви вітру. Середня швидкість вітру становить 3-5 м/с, але в Карпатах вона може досягати 15-20 м/с.

Львівська область поділяється на декілька кліматичних зон:

- ✓ рівнина частина – помірно теплий клімат із середньою кількістю опадів;
- ✓ передгір’я – зволожений клімат із прохолодним літом;
- ✓ гірські райони (Карпати) – холодний та вологий клімат із значною кількістю опадів.

Кліматичні умови Львівської області характеризуються високою вологістю, значною кількістю опадів та помірними температурами. Це створює сприятливі умови для сільського господарства, але водночас спричиняє ризики паводків і зсувів у гірських районах. Розуміння кліматичних особливостей регіону є місцем для розробки системи прогнозування кількості опадів та адаптації до зміни клімату.

1.2. Аналіз сучасних методів прогнозування кількості опадів

Середньострокове та довгострокове прогнозування опадів є важливою частиною гідрологічної науки і завжди відіграє ключову роль у боротьбі з повенями, зменшенні кількості стихійних лих та комплексному використанні водних ресурсів. Проте зі збільшенням періоду прогнозу фактори впливу на середньо- та довгостроковий прогноз опадів все більше призводять до збільшення невизначеності в прогнозі та спричиняють зниження точності прогнозу. Це завжди було складним моментом у сфері прогнозування опадів. Таким чином, поглиблене вивчення теорії та методів середньо- та довгострокового прогнозування має не лише важливе наукове значення для збагачення та розвитку теорії прогнозування опадів, але також має важливе практичне значення для зменшення та запобігання стихійним лихам та соціально-економічного сталого розвитку [34; 36].

Однак середньо- та довгострокове прогнозування опадів, з точки зору надання загальної кількості опадів за певний період часу в майбутньому, вважається одним із найскладніших завдань у моделях глобального клімату,

оскільки точність прогнозу залежить від багатьма факторами, такими як місце випадання опадів, тривалість, частота та інтенсивність, орографія та землекористування [46; 17].

Традиційні середньо- та довгострокові прогнози переважно використовують статистичні методи, динамічні методи та комбінацію статистики та динаміки для створення прогнозів. В останні роки, зі швидким розвитком глобального супутникового дистанційного зондування, хмарних обчислень і хмарних технологій зберігання, можливість і стабільність роботи моделей загальної циркуляції (GCM) були ще більше вдосконалені, і GCM поступово замінили класичну статистичну модель і стати основним інструментом для випуску щомісячної прогнозної інформації в сезонному масштабі в реальному часі для основних метеорологічних і гідрологічних центрів прогнозування в усьому світі [30; 15].

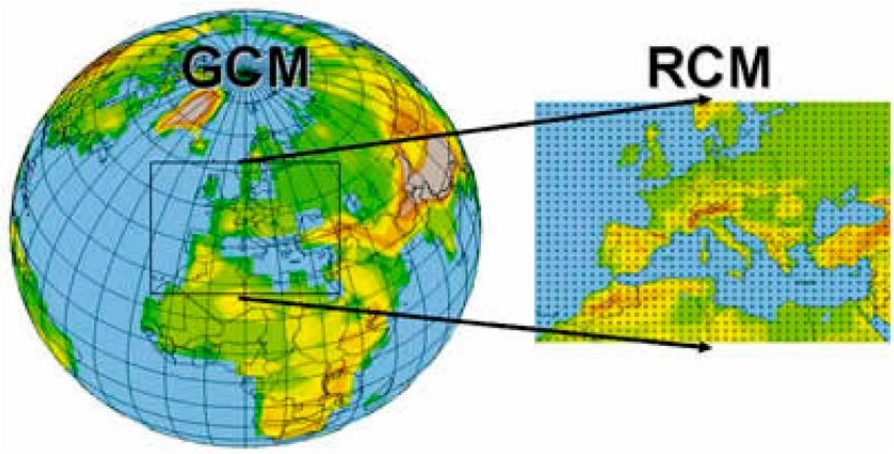

Рисунок 1.3 – Модель загальної циркуляції (GCM)

Водночас із швидким розвитком комп’ютерних технологій метод машинного навчання (ML), заснований на технології видобутку великих даних, поступово застосовувався для середньо- та довгострокових прогнозів опадів через його високу здатність до узагальнення та надійність. Методи середньо- та довгострокового прогнозування опадів, засновані на ML, в основному будують кореляції між опадами та предикторами. На опади впливає багато факторів. На додаток до добре відомих метеорологічних і кліматичних факторів,

прогностичні фактори, такі як зондування [10], локальні ефекти, такі як циркуляційний ефект гірської долини [18], атмосферне середовище [27] та інформація супутникових зображень, що спостерігаються в преконвективного середовища, сильно відрізняються [53].

В останні роки з безперервним прогресом науки і техніки та стрімким розвитком інформаційних технологій обсяг даних у виробництві та житті людини геометрично зріс і поступово перейшов від байтів до гігабайтів, терабайтів, петабайтів і навіть йоттабайтів. Технологія великих даних виникла і поступово стала центром наукових досліджень. Однак різноманітність і маса великих даних є одночасно благом і викликом для середньо- та довгострокового гідрологічного прогнозування. Завдяки активному просуванню інформатизації охорони водних ресурсів як дані спостережень поверхневих метеорологічних станцій, так і дані спостережень на основі супутникового дистанційного зондування досягли значного прогресу за останні роки в «якості» та «кількості» даних із сильними часовими та просторовими атрибутами і поступово започаткували «еру великих даних» у гідрології. Ці нові джерела інформації збагачують наше розуміння та покращують наші можливості моделювання.

Однак, перед обличчям інформації з різних джерел і структур, як використовувати технологію інтелектуального аналізу даних для дослідження її внутрішньої цінності та зв’язку з масивною метеорологічною та гідрологічною інформацією є передовою галуззю досліджень розвитку гідрологічного прогнозування [47]. Що ще важливіше, без передових технологій ми можемо навіть не усвідомлювати, яку приховану та абстрактну інформацію можна отримати, або обмеження точності цього вилучення, що призводить до недостатнього використання доступних даних [32].

Для вирішення вищезазначених проблем були запропоновані деякі методи, керовані даними, такі як методи ML, які широко використовуються перед лицем складних і великих змінних зв’язків, щоб допомогти нам витягти корисну інформацію із зростаючих даних [33; 16; 51; 24; 50] . Таким чином,

поєднання передових методів ML із традиційними гідрологічними методами для реалізації середньо- та довгострокового прогнозування опадів є не лише розширенням і вдосконаленням традиційного прогнозування опадів, але й великим прогресом у міждисциплінарному розвитку гідрологічної роботи. Методи ML можна розділити на дрібне ML і глибоке навчання відповідно до глибини мережі. Методи неглибокого ML широко використовуються в гідрології, такі як випадковий ліс (RF) [11; 31; 29], машина опорних векторів (SVM) [14; 28; 19], екстремальне підвищення градієнта (XGB) [13; 1; 23], Light Gradient Boosting Machine (LGB) [21; 35; 12] тощо. Методи глибокого навчання, такі як рекурентна нейронна мережа (RNN) [20; 25; 52] і довготривала короткочасна пам'ять (LSTM)) [49] не мають широкого застосування в гідрології. Крім того, у сфері штучного інтелекту більш важливим вирішальним фактором часто є обсяг даних, які використовуються для побудови моделі, а не те, чи може модель ML продемонструвати кращу продуктивність, здатність налагоджувати модель або саму модель.

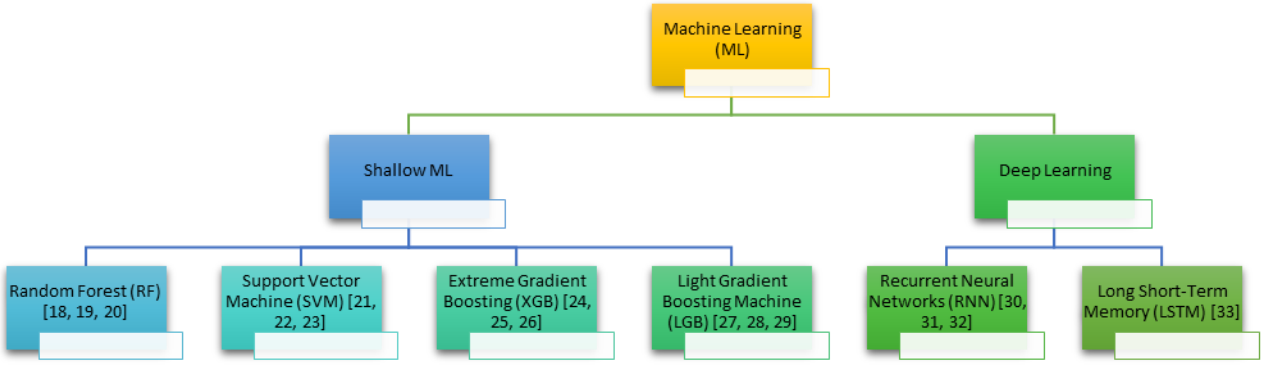

Рисунок 1.4 – Використовувані методи ML для прогнозування кількості опадів

У галузі гідрології довжина гідрологічних рядів обмежена, і іноді важко задовольнити кількість зразків, необхідних ML для побудови кращої моделі, що значно впливає на точність прогнозу. Таким чином, критично важливо розширити дані гідрологічного ряду в розумному діапазоні, щоб відповідати основним вимогам моделювання моделі ML, щоб модель ML відігравала кращу роль у середньо- та довгостроковому прогнозуванні опадів.

1.3. Особливості балансування даних для прогнозування опадів

Прогнозування кількості опадів є важливою задачею в метеорології, сільському господарстві, міському плануванні та багатьох інших галузях. Однією з основних проблем під час побудови моделей прогнозування є дисбаланс даних – ситуація, коли кількість днів з низькими або нульовими опадами значно перевищує кількість днів із середніми чи високими опадами. Це може суттєво вплинути на якість роботи моделей машинного навчання, які схильні адаптуватися до більшості класу і ігнорувати менш представлені приклади. Балансування даних є необхідним кроком для забезпечення рівномірного врахування всіх класів у задачах прогнозування.

У реальних даних для прогнозування кількості опадів більшість днів характеризується невеликими або нульовими значеннями опадів. Це пояснюється природними умовами: дощі, як правило, є нерегулярними, і велика частина часу спостерігається суха погода. Однак, для задач прогнозування важливо точно враховувати дні з інтенсивними опадами, оскільки вони часто є критичними для аналізу.

Моделі машинного навчання, треновані на даних із дисбалансом, можуть показувати високу загальну точність, однак не справляються із прогнозуванням днів з інтенсивними опадами, що належать до меншості класу. Наприклад, модель може прогнозувати «нульові» опади для більшості днів, забезпечуючи вражаючу загальну точність, але при цьому неефективно працювати з екстремальними подіями. Для вирішення цієї проблеми використовують різні підходи до балансування даних.

Балансування даних у задачах прогнозування опадів може бути реалізовано за допомогою двох основних підходів: oversampling (збільшення прикладів меншості класу) та undersampling (зменшення прикладів більшості класу) (табл. 1.1). У багатьох випадках використовують також їх комбінації.

Oversampling. Цей підхід передбачає збільшення кількості прикладів меншості класу шляхом дублювання існуючих даних або створення нових.

Таблиця 1.1 – Основні підходи до балансування даних у контексті прогнозування кількості опадів:

Метод	Опис	Переваги	Недоліки
Oversampling	Збільшення кількості прикладів меншості класу шляхом дублювання чи створення нових точок	- Покращує моделі для екстремальних опадів. - Зберігає всі приклади днів без опадів.	- Ризик перенавчання через дублювання. - Синтетично створені дані можуть бути некоректними.
- SMOTE (Synthetic Minority Oversampling Technique)	Генерує нові приклади меншості шляхом інтерполяції між існуючими точками	- Вирішує проблему дублювання. - Покращує врахування днів з рідкісними інтенсивними опадами.	- Може створювати точки в нерелевантних областях. - Чутливий до шуму в даних.
- ADASYN (Adaptive Synthetic Sampling)	Генерує нові точки більш адаптивно до складних зон розподілу даних	- Створює більше точок у складних для класифікації зонах. - Добре підходить для нерівномірного розподілу опадів.	- Ризик генерування шуму. - Висока обчислювальна складність.
Undersampling	Зменшення кількості прикладів більшості класу	- Зменшує час і ресурси на обробку даних. - Забезпечує баланс у даних.	- Можливість втрати важливої інформації про дні без опадів. - Менше даних для тренування.

- Random Undersampling	Випадкове видалення частини прикладів із більшості класу	- Простий у реалізації. - Швидке досягнення балансу.	- Може видаляти релевантні приклади днів без опадів.
- Tomek Links	Видаляє точки, які межують між класами	- Покращує чіткість меж між класами. - Ефективний для великих розподілів опадів.	- Малоефективний для невеликих наборів даних або великої кількості класів.
Комбіновані підходи	Поєднання oversampling і undersampling	- Оптимальний баланс між збереженням днів без опадів і врахуванням екстремальних значень.	- Складність налаштування параметрів. - Великі обчислювальні витрати.
- Balanced Bagging	Балансування даних у кожному дереві ансамблю	- Покращує стійкість прогнозів для складних розподілів. - Добре працює для ансамблевих моделей.	- Висока складність реалізації. - Потребує більше часу для навчання.
- Hybrid Sampling	Поєднує збільшення меншості та зменшення більшості	- Забезпечує кращу репрезентативність даних. - Добре підходить для різноманітних розподілів опадів.	- Висока обчислювальна складність. - Необхідність додаткового аналізу для оптимізації.

Найбільш поширеним методом є SMOTE (Synthetic Minority Oversampling Technique), який генерує нові точки меншості класу шляхом інтерполяції між існуючими прикладами. Для прогнозування опадів це може бути корисним, оскільки дозволяє збільшити кількість днів із високими значеннями опадів, які є важливими для точності моделі. Іншим підходом є ADASYN (Adaptive Synthetic Sampling), який адаптується до складності розподілу даних і генерує більше точок у зонах, де меншості класу найменше.

Undersampling. Цей метод передбачає зменшення кількості прикладів більшості класу. Наприклад, для задачі прогнозування опадів це може означати зменшення кількості днів із низькими або нульовими опадами. Один із підходів – random undersampling, який випадково видаляє приклади більшості класу для вирівнювання кількості даних у класах. Також використовують метод Tomek Links, який видаляє точки, що межують між класами, покращуючи розділення даних. Основним недоліком undersampling є ризик втрати важливої інформації, яка може бути критичною для побудови моделі.

Комбіновані підходи. У задачах прогнозування опадів часто застосовують поєднання oversampling та undersampling. Це дозволяє зберегти релевантність даних більшості класу та одночасно збільшити представленість меншості. Наприклад, balanced bagging поєднує балансування даних у кожному дереві ансамблевої моделі, а hybrid sampling забезпечує створення синтетичних прикладів меншості та видалення надлишкових прикладів більшості.

Балансування даних має низку переваг. По-перше, воно дозволяє моделі враховувати всі класи рівномірно, що покращує прогнозування екстремальних випадків. По-друге, збалансовані дані знижують упередженість моделі до більшості класу. Однак, методи балансування мають і обмеження. Наприклад, oversampling може призводити до перенавчання моделі, особливо якщо використовувати просте дублювання даних. З іншого боку, undersampling може втратити важливі приклади більшості класу, що негативно позначається на точності моделі.

У задачах прогнозування кількості опадів використання методів балансування є критично важливим. Дні з екстремальними опадами часто мають вирішальне значення для аналізу, наприклад, у контексті попередження про повені або оцінки водних ресурсів. Балансування даних дозволяє побудувати більш стійкі моделі, які враховують не лише загальні тренди, але й рідкісні події.

Найбільш ефективними підходами для прогнозування опадів є SMOTE та ADASYN для збільшення кількості днів із високими опадами, а також комбіновані підходи, які зберігають баланс між різними класами. Такі методи забезпечують точніше прогнозування, покращуючи метрики точності, recall і F1-score для класів із низькою представленістю.

Балансування даних є важливим етапом у задачах прогнозування кількості опадів. Воно забезпечує рівномірний розподіл класів, покращуючи точність моделей і враховуючи екстремальні погодні явища. Використання сучасних методів, таких як SMOTE, ADASYN та комбінованих підходів, дозволяє отримати більш точні та стабільні результати, що особливо важливо в контексті задач кліматичного аналізу та управління ресурсами.

1.4. Проблеми обробки та балансування кліматичних даних

Під час обробки кліматичних даних для прогнозування кількості опадів стикається з низкою викликів, які зумовлені природою даних та їх розподілом. Основні проблеми включають складові, які подано на рис. 1.5.

Дані, отримані з метеостанцій чи супутників, часто мають прогалини через технічні несправності, людські помилки чи обмеження в доступі до інформації. Неповні записи можуть суттєво вплинути на точність моделей прогнозування.

Кліматичні показники, такі як температура, вологість чи кількість опадів, можуть містити шуми, викликані несправністю обладнання, похибками

вимірювань або локальними екстремальними подіями. Шум у даних може збивати з пантелику моделі машинного навчання, що ускладнює адекватний аналіз.

Рисунок 1.5 – Складові проблеми обробки та балансування кліматичних даних

Кількість днів із нульовими чи низькими опадами значно перевищує кількість днів із сильними дощами або іншими екстремальними погодними умовами. Це призводить до дисбалансу класів, ускладнюючи навчання моделей і викликаючи їх упередженість до більшості класу.

Кліматичні фактори, такі як температура, вологість чи атмосферний тиск, тісно пов’язані між собою, що ускладнює побудову моделей. Неврахування цих залежностей може призвести до помилок у прогнозуванні.

Кліматичні дані часто охоплюють багаторічні періоди і великий просторовий масштаб, що вимагає значних обчислювальних ресурсів для їх обробки та аналізу.

Балансування кліматичних даних є ще однією складною задачею, яка безпосередньо впливає на якість моделювання. Дні з високими опадами, які є критично важливими для аналізу, значно менш представлені, ніж дні без опадів. Це призводить до упередженості моделей, які зосереджуються на прогнозуванні більшості класу.

Використання методів undersampling для зменшення кількості днів без опадів може призвести до видалення ключових прикладів, які необхідні для моделювання.

Методи oversampling, такі як SMOTE чи ADASYN, створюють нові точки, які можуть не повністю відображати реальну природу кліматичних явищ. Це може викликати похибки в прогнозах. Балансування великих наборів даних потребує значних обчислювальних ресурсів, особливо якщо застосовуються комбіновані методи. Штучно створені точки можуть бути чутливими до шуму, що знижує загальну якість даних.

Проблеми, пов’язані з обробкою та балансуванням кліматичних даних, зумовлюють необхідність розробки інтелектуальної інформаційної системи прогнозування кількості опадів. Метою кваліфікаційної роботи є подолання зазначених викликів за допомогою сучасних методів машинного навчання та балансування даних.

Прогнозування опадів є ключовим фактором для попередження паводків, планування сільськогосподарських робіт та управління водними ресурсами. Надійні прогнози сприяють мінімізації економічних та екологічних ризиків. Використання сучасних алгоритмів балансування даних (SMOTE, ADASYN) та методів машинного навчання (SVM, Random Forest, LSTM) дозволяє досягти високої точності прогнозів навіть за умов дисбалансу даних.

Інтелектуальна інформаційна система може бути інтегрована з існуючими метеорологічними платформами для автоматичного збору, аналізу та прогнозування кліматичних даних. Система надаватиме точні прогнози кількості опадів, що сприятиме прийняттю обґрунтованих рішень у різних галузях, включаючи сільське господарство, енергетику та транспорт. Розробка ефективних методів балансування дозволить підвищити якість даних, що використовуються для навчання моделей, забезпечуючи точніше прогнозування навіть у складних умовах.

Проблеми обробки та балансування кліматичних даних є серйозним викликом для сучасних систем прогнозування. Розробка інтелектуальної

інформаційної системи прогнозування кількості опадів із використанням методів машинного навчання та балансування даних дозволить вирішити ці проблеми, забезпечуючи точність і стабільність прогнозів. Це має велике значення для підвищення ефективності управління природними ресурсами та попередження наслідків екстремальних погодних явищ.

РОЗДІЛ 2.
ПРОГНОЗУВАННЯ КІЛЬКОСТІ ОПАДІВ НА ОСНОВІ БАЛАНСУВАННЯ ДАНИХ ТА ОБГРУНТУВАННЯ МОДЕЛІ МАШИННОГО НАВЧАННЯ

2.1. Особливості набору даних для прогнозування кількості опадів

Дані про погоду мають велике значення для розробки моделей прогнозування кількості опадів. Проте доступність і якість цих даних залишається серйозною проблемою для більшості дослідників у всьому світі. В Україні отримати дані спостережень про погоду дуже складно через рідкісне розміщення метеостанцій і непослідовні записи даних. Це створило критичні прогалини в доступності даних для запуску та розробки ефективних моделей прогнозування кількості опадів.

Щоб подолати цю прогалину, ми отримали часові ряди даних щомісячних спостережень за даними для умов Львівської області. Період даних з 1992 по 2024 рік із Львівського регіонального центру гідрометеорології. Доступ до даних було отримано зі сховища даних (рис. 2.1).

```
[3]: df.head()
```

[3]:

	Year	Month	Day	Specific Humidity	Relative Humidity	Temperature	Precipitation
0	1992.0	1.0	1.0	11.1	79.3	-9.8	40.1
1	1992.0	2.0	1.0	10.9	71.7	-7.7	32.6
2	1992.0	3.0	1.0	12.0	64.1	17.8	40.9
3	1992.0	4.0	1.0	11.9	81.5	10.6	56.0
4	1992.0	5.0	1.0	12.2	63.1	6.7	82.7

Рисунок 2.1 – Фрагмент масиву даних для розробки моделі прогнозування кількості опадів

Дані, що представлені на рис. 2.1, мають наступні характеристики. Кількість рядків становить 394 (по одному запису для кожного місяця за 33

роки). Кількість колонок становить 7 од: Year (Рік), Month (Місяць), Day (День), Specific Humidity (Специфічна вологість), Relative Humidity (Відносна вологість), Temperature (Температура) та Precipitation (місячні опади) (рис. 2.2).

```
[4]: df.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 394 entries, 0 to 393
Data columns (total 7 columns):
 #   Column             Non-Null Count  Dtype
---  ------             --------------  -----
 0   Year               394 non-null    float64
 1   Month              394 non-null    float64
 2   Day                394 non-null    float64
 3   Specific Humidity  394 non-null    float64
 4   Relative Humidity  394 non-null    float64
 5   Temperature        394 non-null    float64
 6   Precipitation      394 non-null    float64
dtypes: float64(7)
memory usage: 21.7 KB
```

Рисунок 2.2 – Інформація про дані для прогнозування кількості опадів

Діапазон значень Specific Humidity становить від 10.9 до 12.2 (грамів водяної пари на кілограм повітря), що описує описує кількість водяної пари в повітрі. Relative Humidity (Відносна вологість) має діапазон значень від 63.1% до 81.5%, що відображає відносний рівень вологості повітря. Temperature (Температура) має діапазон значень від -9.8°C до 17.8°C та містить середню температуру повітря для кожного місяця. Precipitation (Опади) має діапазон значень від 32.6 до 82.7 мм і вказує на кількість опадів у міліметрах за місяць.

Набір даних повний (без пропущених значень), що видно зі 100% заповнення кожної колонки (394 рядки у кожній колонці). Дані представлені у форматі середніх місячних значень, що полегшує аналіз тенденцій.

Цей набір даних добре підходить для аналізу погодних тенденцій, виявлення сезонності, і розробки моделей прогнозування кількості опадів, наприклад, за допомогою машинного навчання чи статистичних методів.

Для детальнішого дослідження сезонних тенденцій нами побудовано графіки залежностей опадів від інших факторів (температура, вологість) або аналізувати зміни по роках.

У подальшому нами виконано перетворення df з таблиці з окремими колонками для року, місяця і дня в таблицю з індексом типу datetime, що дозволяє більш ефективно працювати з часовими рядами (рис. 2.3).

[9]:
```
## setting date as index
df['DATE'] = pd.to_datetime(df[['Year', 'Month','Day']])
df = df.drop(columns=['Year', 'Month','Day'])
df.index = df["DATE"]
df.drop(columns=["DATE"],inplace = True)
```

[10]:
```
df
```

[10]:

DATE	Specific Humidity	Relative Humidity	Temperature	Precipitation
1992-01-01	11.1	79.3	-9.8	40.1
1992-02-01	10.9	71.7	-7.7	32.6
1992-03-01	12.0	64.1	17.8	40.9
1992-04-01	11.9	81.5	10.6	56.0
1992-05-01	12.2	63.1	6.7	82.7
...	...	...	...	...
2024-06-01	7.0	76.3	24.5	128.2
2024-07-01	10.2	82.1	23.2	114.9
2024-08-01	10.8	87.6	20.6	92.9
2024-09-01	14.0	71.1	8.5	82.3
2024-10-01	11.3	73.6	15.2	62.1

394 rows × 4 columns

Рисунок 2.3 – Фрагмент коду та результати перетворення df з таблиці з окремими колонками для року, місяця і дня в таблицю з індексом типу datetime

Для перетворення df використано функцію pd.to_datetime(), яка створює об’єкт типу datetime64[ns], об’єднавши три числові колонки (Year, Month, Day) в одну колонку DATE. Колонку DATE використано як індекс df. Індексація за часом дозволяє ефективно виконувати операції над часовими рядами (наприклад, фільтрацію, агрегацію за періодами, вибір піддіапазонів тощо). Багато моделей прогнозування часу очікують, що індекс даних буде у форматі дати.

```
[11]: # Створення DataFrame з колонкою Precipitation
rain = pd.DataFrame(df["Precipitation"])
rain.index.freq = 'MS'

# Налаштування стилю графіка
sns.set_theme(style="whitegrid")  # Тема оформлення

# Побудова графіка
plt.figure(figsize=(12, 6))  # Розмір графіка
sns.lineplot(data=rain, x=rain.index, y="Precipitation", color="royalblue", linewidth=2.5, label="Precipitation")

# Додаткові стилі та підписи
plt.title('Monthly Precipitation (1992-2024)', fontsize=18, fontweight='bold')
plt.xlabel('Date', fontsize=14)
plt.ylabel('Precipitation (mm)', fontsize=14)
plt.xticks(fontsize=12, rotation=45)  # Обернуті підписи на осі X
plt.yticks(fontsize=12)
plt.legend(fontsize=12)
plt.tight_layout()  # Автоматичне коригування
plt.show()
```

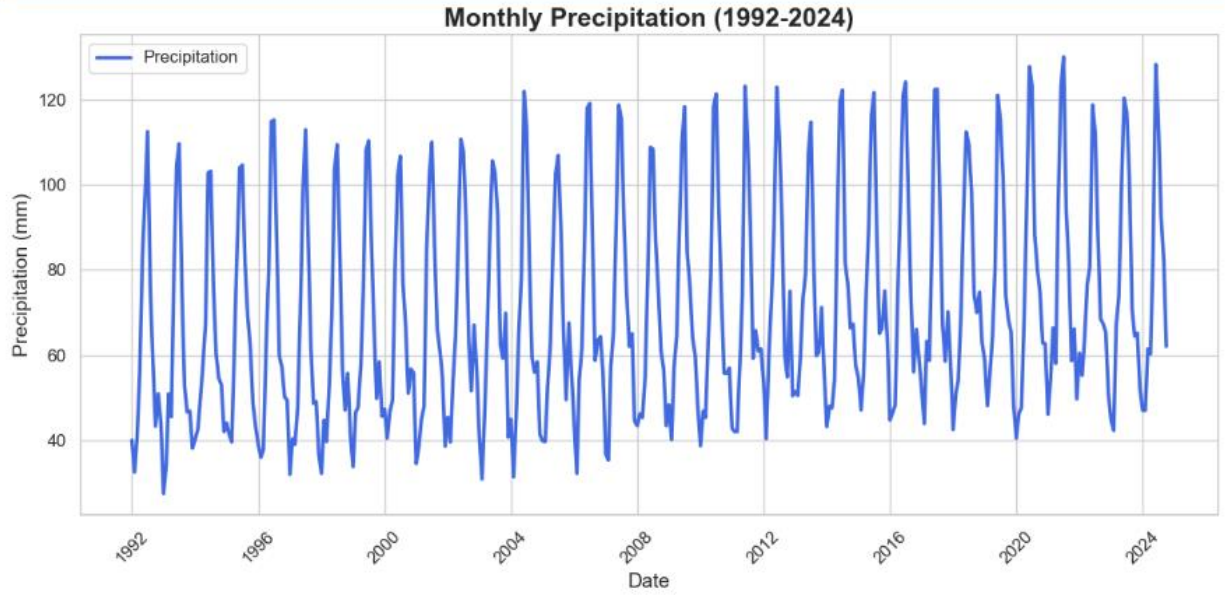

Рисунок 2.4 – Фрагмент коду та результати створення графіка кількості опадів (Precipitation)

Нами написано код, який створює графік кількості опадів (Precipitation) з використанням даних у часовому форматі, де індекс представлений як початок місяця. Вибрано лише колонку Precipitation з оригінального DataFrame (df) і створено новий DataFrame rain. Колонка Precipitation містить місячні значення опадів у міліметрах. Вказано, що індекс rain має частоту MS (Month Start – початок місяця). Це дозволяє Python коректно обробляти дані як часовий ряд із регулярною частотою.

Отримано графік, який показує, як змінювалася кількість опадів протягом кожного місяця з 1992 по 2024 рік. Цей графік допомагає візуально оцінити тенденції в кількості опадів, виявити сезонність або пікові періоди опадів, а також підготувати дані до подальшого аналізу чи моделювання.

У подальшому нами проаналізовано сезонну декомпозицію. Дані часового ряду розглядаються з позиції трендів та їх сезонності (рис. 2.5).

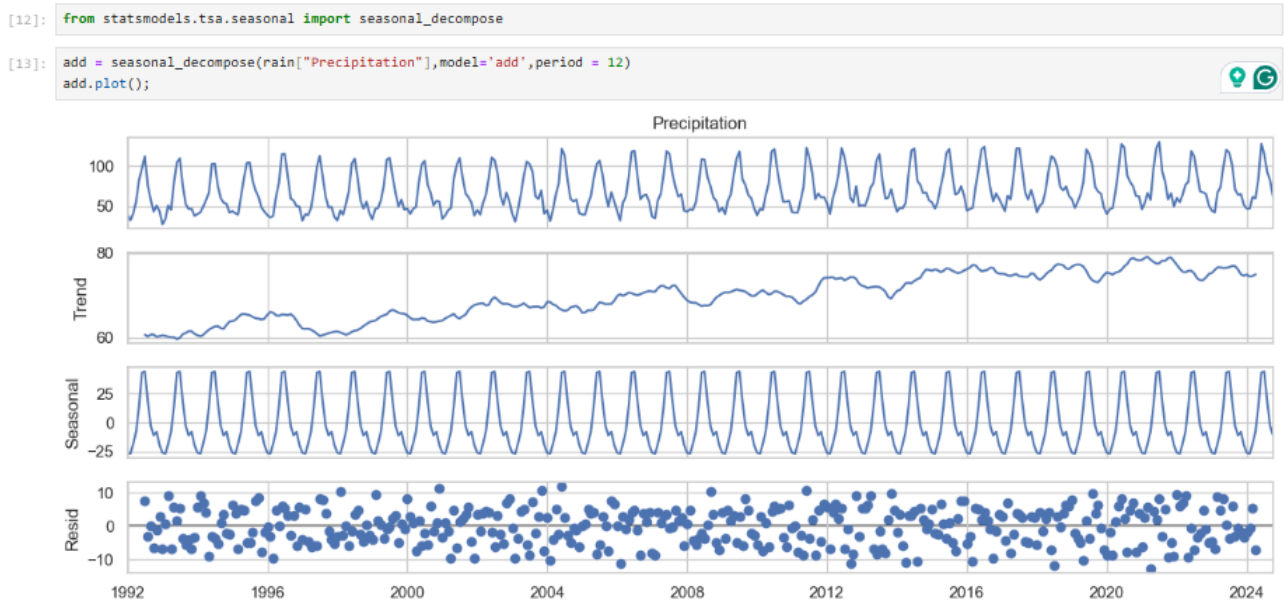

Рисунок 2.4 – Фрагмент коду та результати створення графіків сезонної декомпозиції

Тренд – це загальний напрямок даних. Сезонність – це періодичний компонент, який повторюється протягом певного періоду часу. Залишки –це те, що залишається після видалення тренду та сезонності. Вони являють собою випадкові коливання.

Нами виконано імпортування функції seasonal_decompose з бібліотеки statsmodels, яка дозволяє розкласти часовий ряд на основні компоненти – тренд, сезонність і залишки (residuals). Метод plot() автоматично створює чотири графіки, які відображають основні компоненти декомпозиції:

- ✓ Observed (Спостереження) – оригінальний часовий ряд;
- ✓ Trend (Тренд) – загальний напрямок змін у часі (довгострокові тенденції);
- ✓ Seasonal (Сезонність) – повторювані зміни, які залежать від річного циклу;
- ✓ Residual (Залишки) – непередбачувані компоненти, які залишаються після видалення тренду та сезонності.

Оригінальний графік кількості опадів за період із 1992 по 2024 рік демонструє загальну тенденцію та сезонні коливання в даних. Встановлено, що існує сезонність кількості опадів впродовж року. Найбільше припадає на весняно-літньо-осінній період, а найменше у зимовий період. Також спостерігається за досліджуваний період тренд, який показує загальний напрямок змін опадів у часі, згладжуючи короткострокові варіації. Можна побачити, що впродовж досліджуваного періоду кількість опадів збільшується, у довгостроковій перспективі.

Графік Residual відображає залишкові значення, які не можна пояснити трендом чи сезонністю. При цьому спостерігаються випадкові відхилення або аномальні явища в даних.

Декомпозиція дозволяє краще зрозуміти структуру часового ряду, виділивши його основні компоненти. Це особливо корисно для аналізу сезонних і довгострокових тенденцій у кількості опадів, а також для підготовки даних до моделювання або прогнозування.

2.2. Вибір методів та балансування даних

Під час підготовки даних для прогнозування кількості опадів виникає задача їх балансування. Балансування даних є важливим етапом попередньої обробки, який дозволяє забезпечити більш точні та стабільні результати моделювання. Нами здійснено вибір методів балансування даних, що відповідають специфіці задачі та характеристикам початкового набору даних.

Дисбаланс даних виникає, коли одна чи декілька категорій у наборі даних значно переважають інші. Це може вплинути на ефективність моделі, оскільки більшість стандартних алгоритмів схильні орієнтуватися на домінуючий клас. Як наслідок модель демонструє високу загальну точність, але низьку ефективність для менш представлених класів. Існує підвищений ризик надмірного навчання (overfitting) на домінуючих класах. У прогнозуванні

часових рядів або регресії нерівномірний розподіл значень може призводити до помилкових прогнозів. Для вирішення задачі балансування даних існують різні підходи, які можна класифікувати на три основні групи – методи підвибірки (undersampling), методи надвибірки (oversampling) та методи створення синтетичних даних.

Балансування даних є критично важливим етапом підготовки до прогнозування кількості опадів. Оскільки Precipitation (кількість опадів) є числовою змінною, необхідно використовувати спеціалізовані методи, які враховують особливості регресійних задач. Дисбаланс у даних може виникати, якщо значення опадів сильно варіюються (наприклад, значно більше малих значень у порівнянні з великими). Це може впливати на якість прогнозів та стабільність моделі.

Для балансування даних у нашій задачі обрано чотири методи (табл. 2.1).

Таблиця 2.1 – Переваги та недоліки методів балансування даних

Група	Назва	Характеристика	Переваги	Недоліки
Метод перетворення даних	Logarithmic Transformation	Логарифмічне перетворення даних для зменшення впливу великих значень	Простота реалізації; Підвищує стабільність моделей	Потребує зворотного перетворення для аналізу результатів
Метод зважування	Weighted Sampling	Введення ваг для кожного запису залежно від частоти його значення	Уникає дублювання та зайвої генерації; Зберігає початкову структуру даних	Важко підібрати оптимальні ваги
Метод кластеризації	KMeans Sampling	Кластеризація значень із рівномірною вибіркою з кожного кластера	Добре працює для складних багатовимірних залежностей	Потребує оптимального підбору кількості кластерів

Logarithmic Transformation рекомендовано використовувати для вирівнювання значень, якщо кількість опадів має великий діапазон. Його використовують коли модель чутлива до великих значень у вихідних даних.

Weighted Sampling забезпечує найкращий вибір, якщо потрібно зберегти структуру даних без додавання шумових значень або видалення важливих прикладів. Його використовують для уникнення дублювання даних і забезпечення стабільного навчання моделі.

KMeans Sampling є ефективним методом для багатовимірних даних, що дозволяє зберегти різноманітність у кожному кластері. Його використовують якщо дані мають складну структуру залежностей між змінними.

Для задачі прогнозування кількості опадів рекомендується використовувати зазначені методи балансування даних, що дозволяють суттєво покращити якість моделювання, особливо в задачах з дисбалансом даних. Обраний підхід спрямований на збереження варіативності даних та запобігання втратам інформації. Це створює міцну основу для ефективного моделювання та точних прогнозів.

Нами створено код для балансування даних за різними методами, який представлено у додатку А. Фрагменти із кодом різних методів балансування даних подано на рис. 2.5.

```
# 1. Логарифмічне перетворення (Logarithmic Transformation)
transformer = FunctionTransformer(np.log1p, validate=True)  # log1p для log(1+x), щоб уникнути log(0)
y_log = transformer.transform(y.values.reshape(-1, 1)).flatten()
df_log = pd.concat([X.reset_index(drop=True), pd.DataFrame(y_log, columns=["Precipitation"])], axis=1)

# 2. Зважування (Weighted Sampling)
weights = compute_sample_weight("balanced", y)
df_weighted = pd.concat([X.reset_index(drop=True), pd.DataFrame(y, columns=["Precipitation"]), pd.DataFrame

# 3. Кластеризація і вибірка (KMeans Sampling)
kmeans = KMeans(n_clusters=5, random_state=42)
df["Cluster"] = kmeans.fit_predict(y.values.reshape(-1, 1))  # Кластеризація по значеннях Precipitation
dfs_kmeans = []
```

Рисунок 2.5 – Фрагмент із кодом різних методів балансування даних

Метод логарифмічного перетворення (Logarithmic Transformation) передбачає застосовує логарифмічне перетворення до цільової змінної y для вирівнювання розподілу великих значень. Використовується log1p (логарифм log(1 + x)), щоб уникнути помилок для значень 0. Результати об'єднуються у новий DataFrame df_log (рис. 2.6).

```
[15]: # Перегляд результатів
print("\nSample of Logarithmic Transformation DataFrame:")
df_log.head()
```

```
Sample of Logarithmic Transformation DataFrame:
```

[15]:

	Specific Humidity	Relative Humidity	Temperature	Precipitation
0	11.1	79.3	-9.8	3.716008
1	10.9	71.7	-7.7	3.514526
2	12.0	64.1	17.8	3.735286
3	11.9	81.5	10.6	4.043051
4	12.2	63.1	6.7	4.427239

Рисунок 2.6 – Фрагмент коду та отримані результати балансування даних із використанням методу Logarithmic Transformation

Метод зважування (Weighted Sampling) передбачає для кожного значення виконувати розрахунок ваги зразка за допомогою compute_sample_weight з параметром «balanced». Результати додаються до нового DataFrame df_weighted, який містить ознаки x, цільову змінну y та колонку з вагами «Weights» (рис. 2.7).

```
[16]: # Перегляд результатів
print("\nSample of Weighted Sampling DataFrame:")
df_weighted.head()
```

```
Sample of Weighted Sampling DataFrame:
```

[16]:

	Specific Humidity	Relative Humidity	Temperature	Precipitation	Weights
0	11.1	79.3	-9.8	NaN	1.270968
1	10.9	71.7	-7.7	NaN	1.270968
2	12.0	64.1	17.8	NaN	1.270968
3	11.9	81.5	10.6	NaN	0.423656
4	12.2	63.1	6.7	NaN	1.270968

Рисунок 2.7 – Фрагмент коду та отримані результати балансування даних із використанням методу Weighted Sampling

Метод кластеризації і вибірки (KMeans Sampling) передбачає цільову змінну кластеризувати на 5 груп за допомогою алгоритму KMeans. Для кожного кластеру випадковим чином вибирається до 50 записів. Зразки з усіх кластерів об’єднуються у новий DataFrame df_kmeans. Колонка «Cluster», яка позначала кластери, видаляється (рис. 2.8).

```
[17]: # Перегляд результатів
print("\nSample of KMeans Sampling DataFrame:")
df_kmeans.head()
```

```
Sample of KMeans Sampling DataFrame:
```

[17]:

DATE	Specific Humidity	Relative Humidity	Temperature	Precipitation
2003-01-01	13.6	69.5	-2.1	40.8
2020-02-01	13.1	87.1	-6.3	46.5
1994-12-01	14.8	80.9	-9.4	42.1
2009-02-01	11.5	60.8	-8.5	40.2
2005-02-01	7.8	87.5	-2.6	39.8

Рисунок 2.7 – Фрагмент коду та отримані результати балансування даних із використанням методу KMeans Sampling

Для кожного методу (логарифмічне перетворення, зважування, кластеризація) виводиться кількість рядків та колонок у відповідному DataFrame. Отже, у подальшому пропонується балансувати цільову змінну Precipitation трьома різними методами, щоб покращити якість моделей машинного навчання.

Нами написано код, що створює функцію для візуалізації розподілу цільової змінної (Precipitation) у різних наборах даних після застосування методів балансування. Потім функція викликається для відображення результатів, які представлено на рис. 2.8. Це дало можливість створити функцію для візуалізації розподілу цільової змінної Precipitation після застосування різних методів балансування даних. Графіки побудовані у вигляді гістограм, які відображають частоту значень Precipitation у кожному наборі даних. Для покращення інтерпретації на гістограмах також додається лінія щільності (KDE), яка візуалізує розподіл значень. Окрім цього, на кожному

графіку позначено середнє значення змінної у вигляді вертикальної червоної лінії, що забезпечує додатковий контекст для аналізу.

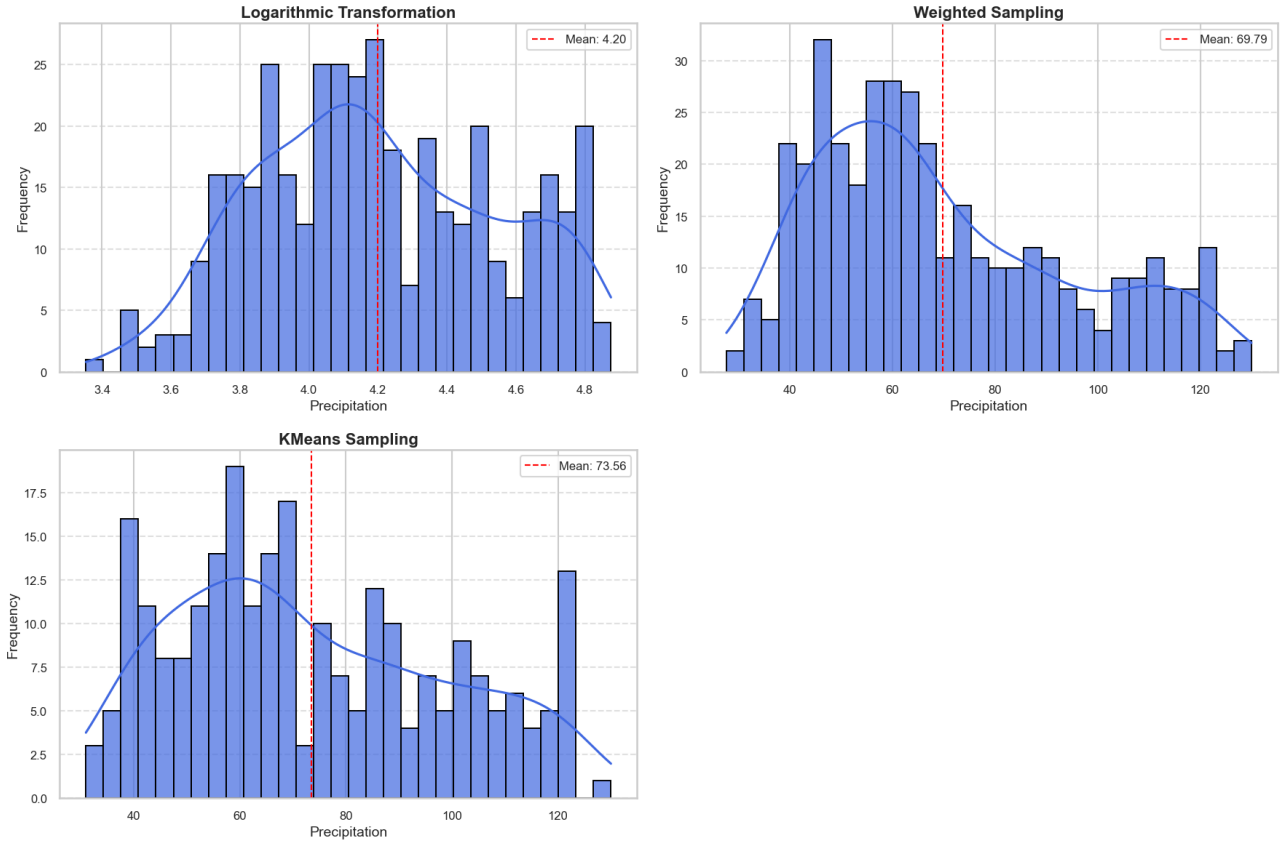

Рисунок 2.8 – Графіки розподілів Precipitation після використання різних методів балансування

У результаті отримали три окремі графіки, які ілюструють розподіл Precipitation після використання різних методів балансування:

✓ Logarithmic Transformation – графік демонструє логарифмічно трансформовані значення цільової змінної, що допомагає вирівняти розподіл і зменшити вплив великих значень.

✓ Weighted Sampling – розподіл Precipitation залишається незмінним, проте враховуються ваги кожного запису, що дозволяє компенсувати нерівномірність даних під час моделювання.

✓ KMeans Sampling – графік показує розподіл, збалансований за допомогою кластеризації, що дозволяє забезпечити рівномірне представлення різних кластерів у даних.

Ці візуалізації допомагають порівняти ефекти різних методів балансування та зрозуміти, як кожен із них вплинув на структуру розподілу даних.

2.3. Розроблення моделей прогнозування кількості опадів на основі методів машинного навчання

Нами написано код із навчання моделей прогнозування кількості опадів на основі методів машинного навчання, який представлено у додатку Б. Першим кроком виконано обробку пропущених значень у збалансованих наборах даних. Для цього створено функцію preprocess_data, яка заповнює всі пропущені значення середніми значеннями відповідних колонок (рис. 2.9).

```
# Функція для обробки пропущених значень
def preprocess_data(df):
    return df.fillna(df.mean())  # Заповнення пропущених значень середнім

# Обробка всіх збалансованих наборів даних
datasets = {
    "Logarithmic Transformation": preprocess_data(df_log),
    "Weighted Sampling": preprocess_data(df_weighted.drop(columns=["Weights"])),
    "KMeans Sampling": preprocess_data(df_kmeans)
}

# === Моделі ===
models = {
    "Gradient Boosting": GradientBoostingRegressor(n_estimators=100, random_state=42),
    "CatBoost Regressor": CatBoostRegressor(iterations=100, learning_rate=0.1, depth=6, verbose=0),
    "XGBoost Regressor": XGBRegressor(n_estimators=100, learning_rate=0.1, max_depth=6, random_state=42)
}
```

Рисунок 2.9 – Фрагмент коду обробки пропущених значень у збалансованих наборах даних та створення моделей

Це забезпечує коректність роботи моделей, оскільки більшість моделей машинного навчання не можуть працювати з пропущеними даними (NaN).

На кожному з підготовлених наборів даних навчаються три моделі: 1) Gradient Boosting Regressor – ансамблевий метод, який поступово зменшує похибку шляхом додавання нових моделей; CatBoost Regressor – модель градієнтного бустингу, спеціально оптимізована для високої швидкості та

точності; XGBoost Regressor – покращена версія бустингу з підтримкою регуляризації, що забезпечує високу продуктивність.

Дані розділили на тренувальні і тестові набори у співвідношенні 80:20. Моделі тренуються на тренувальних даних, а їх точність оцінюється на тестових.

2.4. Дослідження впливу методів балансування даних на точність прогнозування кількості опадів

Нами проведено дослідження впливу різних методів балансування даних на точність прогнозування кількості опадів із використанням трьох моделей машинного навчання – Gradient Boosting, CatBoost Regressor та XGBoost Regressor. Для оцінки точності застосовано метрики MSE, MAE та R^2. У таблиці 2.2 наведено результати, отримані на трьох збалансованих наборах даних.

Таблиця 2.2 – Результати дослідження впливу методів балансування даних на точність прогнозування кількості опадів

Model	Dataset	MSE	MAE	R^2
Gradient Boosting	Logarithmic Transformation	0.047	0.163	0.64
CatBoost Regressor	Logarithmic Transformation	0.05	0.169	0.62
XGBoost Regressor	Logarithmic Transformation	0.058	0.187	0.56
Gradient Boosting	Weighted Sampling	332.569	10.964	-0.0016
CatBoost Regressor	Weighted Sampling	332.566	10.971	-0.0016
XGBoost Regressor	Weighted Sampling	332.569	10.964	-0.00163
Gradient Boosting	KMeans Sampling	153.094	10.19	0.79
CatBoost Regressor	KMeans Sampling	163.835	10.113	0.78
XGBoost Regressor	KMeans Sampling	147.744	8.679	0.805

Аналіз результатів використання методу Logarithmic Transformation свідчить про те, що уцьому методі найкращі результати демонструє Gradient

Boosting, який має найменше значення MSE (0.047), найменше MAE (0.164) і R^2, близький до 0.644. CatBoost Regressor також показує конкурентоспроможні результати, але трохи гірші за Gradient Boosting. XGBoost Regressor демонструє найгірші результати серед трьох моделей для цього методу, що свідчить про його меншу ефективність для логарифмічно трансформованих даних.

Щодо використання методу Weighted Sampling, то у цьому методі всі три моделі показують однаково низьку продуктивність – значення MSE перевищує 332, а MAE становить близько 10.97. Коефіцієнт R^2 практично дорівнює 0 або навіть від'ємний, що свідчить про те, що моделі не змогли пояснити варіацію цільової змінної. Метод Weighted Sampling не забезпечує адекватного балансування для прогнозування кількості опадів.

Стосовно використання методу KMeans Sampling, то цей метод демонструє найкращі результати серед усіх трьох методів балансування. При цьому XGBoost Regressor показує найменше значення MSE (147.74) і найвищий R^2 (0.806), що свідчить про його здатність добре пояснювати варіацію цільової змінної. Gradient Boosting також має конкурентоспроможні результати, із R^2, близьким до 0.799, але трохи більшим значенням MSE (153.09). CatBoost Regressor показує гідні результати, але поступається іншим двом моделям.

На основі графіків, побудованих для кожної метрики (MSE, MAE, R^2), було виявлено наступне (рис. 2.10).

Встановлено, що найменше значення MSE спостерігається для XGBoost Regressor з методом KMeans Sampling. Метод Weighted Sampling має значно більші значення MSE, що робить його непридатним.

Найменше значення MAE також спостерігається для XGBoost Regressor з методом KMeans Sampling. Логарифмічне перетворення також показує низькі значення MAE, особливо для Gradient Boosting.

Найвище значення R^2 (0.806) отримано для XGBoost Regressor з методом KMeans Sampling. Логарифмічне перетворення забезпечує середній рівень точності ($R^2 \approx 0.644$ для Gradient Boosting).

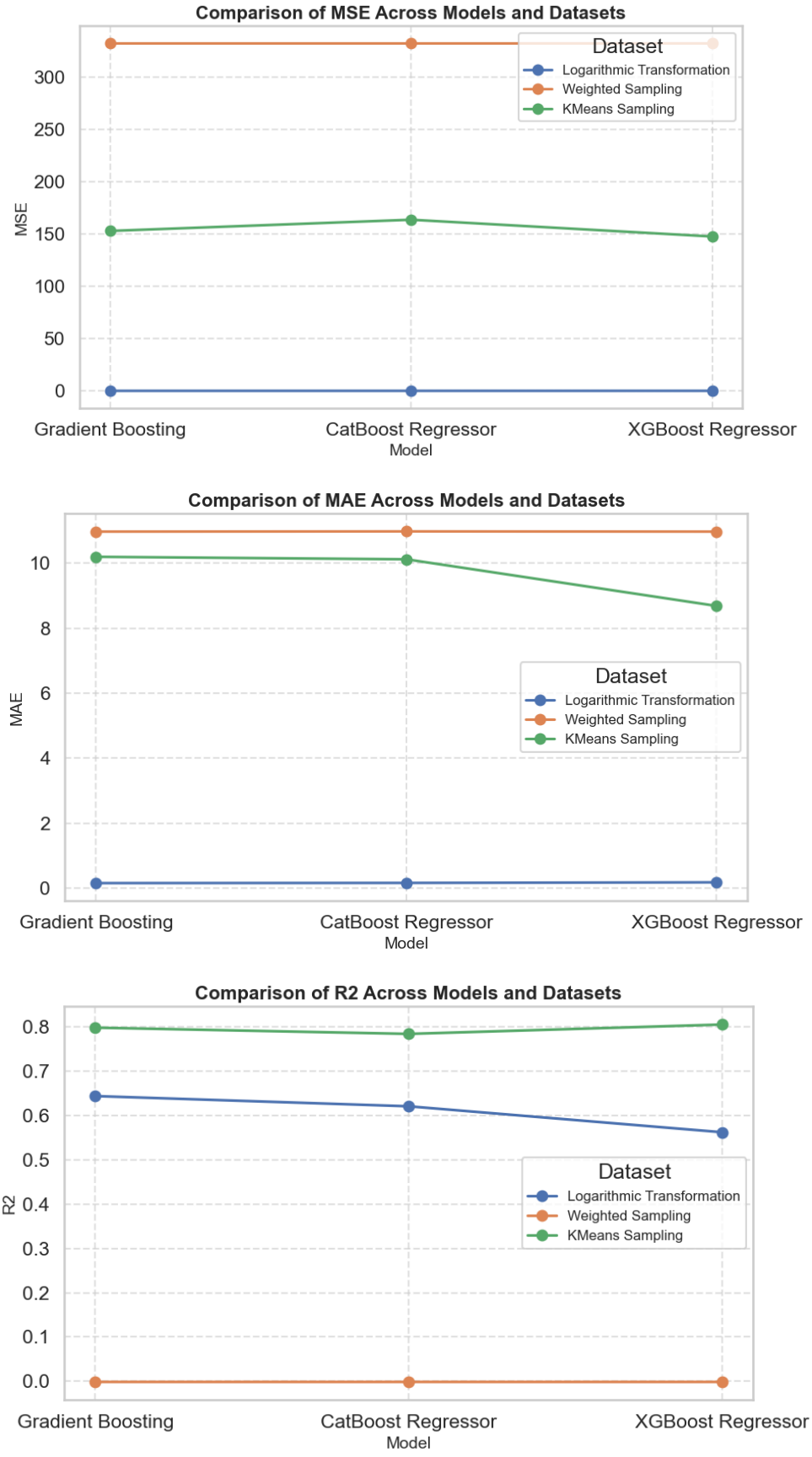

Рисунок 2.10 – Графіки зміни метрик MSE, MAE, R^2 за використання різних методів балансування даних та навчання моделей прогнозування кількості опадів

Найкращі результати для прогнозування кількості опадів демонструє метод KMeans Sampling. Метод Weighted Sampling виявився непридатним, оскільки не покращив продуктивність моделей.

Найвищу точність забезпечує XGBoost Regressor, особливо в комбінації з методом KMeans Sampling. Gradient Boosting показує стабільно високі результати і може бути альтернативою.

Для задачі прогнозування кількості опадів рекомендовано використовувати метод KMeans Sampling у поєднанні з моделлю XGBoost Regressor.

РОЗДІЛ 3.
РЕЗУЛЬТАТИ РОЗРОБКИ ІНТЕЛЕКТУАЛЬНОЇ ІНФОРМАЦІЙНОЇ СИСТЕМИ ПРОГНОЗУВАННЯ КІЛЬКОСТІ ОПАДІВ

3.1. Архітектура системи та вибір технологічного стеку

Розроблена інтелектуальна система для прогнозування кількості опадів базується на моделі XGBoost Regressor, яка показала найкращі результати у поєднанні з методом балансування даних KMeans Sampling. Система має багаторівневу архітектуру, яка включає компоненти для обробки даних, машинного навчання, серверної взаємодії та візуалізації.

Загальна архітектура системи побудована у вигляді модульної структури, що забезпечує гнучкість, масштабованість та інтеграцію з іншими інструментами. Архітектура складається з таких основних компонентів, які представлені у табл. 3.1.

Таблиця 3.1 – Основні компоненти інтелектуальної системи для прогнозування кількості опадів

Компонент	Опис
Дані	Сховище історичних даних про кліматичні умови: вологість, температура тощо.
Обробка даних	Балансування даних, обробка пропущених значень, підготовка до навчання.
Модуль ML	Реалізація моделі XGBoost Regressor для прогнозування кількості опадів.
API-сервер	Серверна частина для отримання запитів, обробки даних та повернення прогнозів.
Фронтенд	Веб-інтерфейс для введення вхідних даних і перегляду результатів.
Візуалізація	Побудова графіків і таблиць для аналізу результатів прогнозування.

Схема архітектури інтелектуальної системи для прогнозування кількості опадів представлена на рис. 3.1.

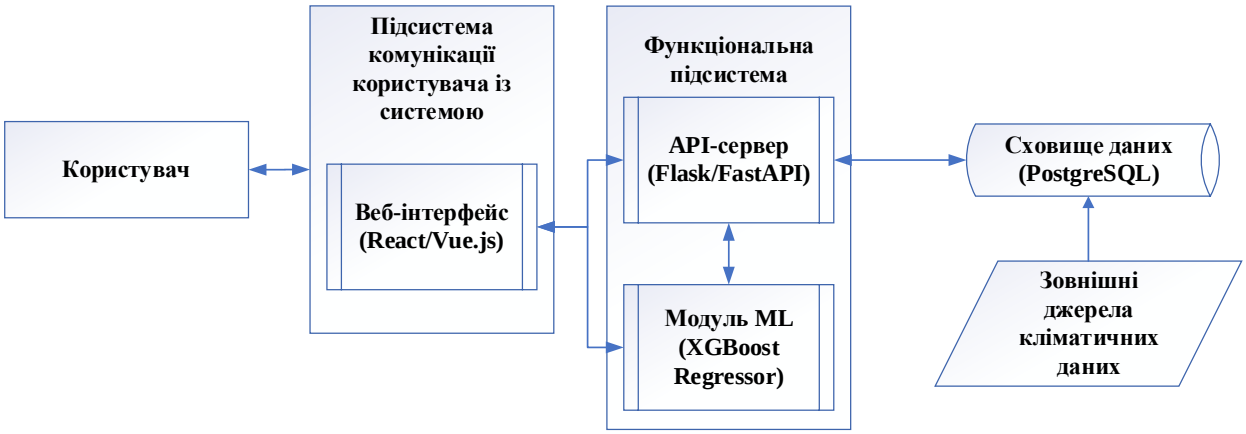

Рисунок 3.1 – Схема архітектури інтелектуальної системи для прогнозування кількості опадів

Пропонується використовувати історичні кліматичні дані, які зберігаються у базі даних PostgreSQL. Дані включають атрибути Specific Humidity, Relative Humidity, Temperature та Precipitation, опис яких виконано у розділі 2 цієї роботи.

Обробка даних передбачає виконання попередньо балансування даних за допомогою методу KMeans Sampling. Пропущені значення заповнюються середніми значеннями відповідних колонок.

Передбачається використовувати Модуль ML. Модель XGBoost Regressor навчається на збалансованих даних і прогнозує кількість опадів. Інтеграція з серверною частиною забезпечує автоматичне використання моделі для нових вхідних даних.

API-сервер побудований за допомогою фреймворку Flask або FastAPI. Він забезпечує обробку HTTP-запитів, передачу вхідних даних у модель і повернення прогнозів у JSON-форматі.

Фронтенд реалізований за допомогою React або Vue.js. Він забезпечує зручний інтерфейс для введення вхідних параметрів і відображення результатів.

Для візуалізації передбачено, що результати прогнозування виводяться у вигляді графіків і таблиць за допомогою бібліотеки Plotly або Chart.js. Графіки

дозволяють візуально оцінити розподіл прогнозованих значень і точність моделі.

Таблиця 3.2 – Результати вибору технологічного стеку

Компонент	Технологія	Призначення
Мова програмування	Python	Основна мова для обробки даних і реалізації моделі.
Бібліотеки ML	XGBoost, Scikit-learn	Реалізація моделі і підготовка даних.
Фреймворк API	Flask/FastAPI	Створення серверної частини.
Фронтенд	React/Vue.js	Інтерактивний веб-інтерфейс.
База даних	PostgreSQL	Зберігання історичних даних про кліматичні умови.
Візуалізація	Plotly/Chart.js	Побудова графіків і таблиць для аналізу результатів.
Деплоймент	Docker, AWS/Heroku	Контейнеризація та хостинг системи.

Запропонована архітектура системи забезпечує ефективне прогнозування кількості опадів за допомогою сучасних технологій. Використання моделі XGBoost Regressor разом із методом балансування KMeans Sampling дозволяє досягти високої точності прогнозів. Технологічний стек на основі Python, Flask, PostgreSQL та React/Vue.js забезпечує гнучкість, масштабованість і інтерактивність системи.

3.2. Алгоритми обробки вхідних даних та прогнозування

Алгоритм обробки вхідних даних і прогнозування складаються з кількох основних етапів – підготовка вхідних даних, балансування, навчання моделі,

прогнозування та оцінка точності. Нижче представлено блок-схему роботи системи (рис. 3.2) та опис кожного етапу.

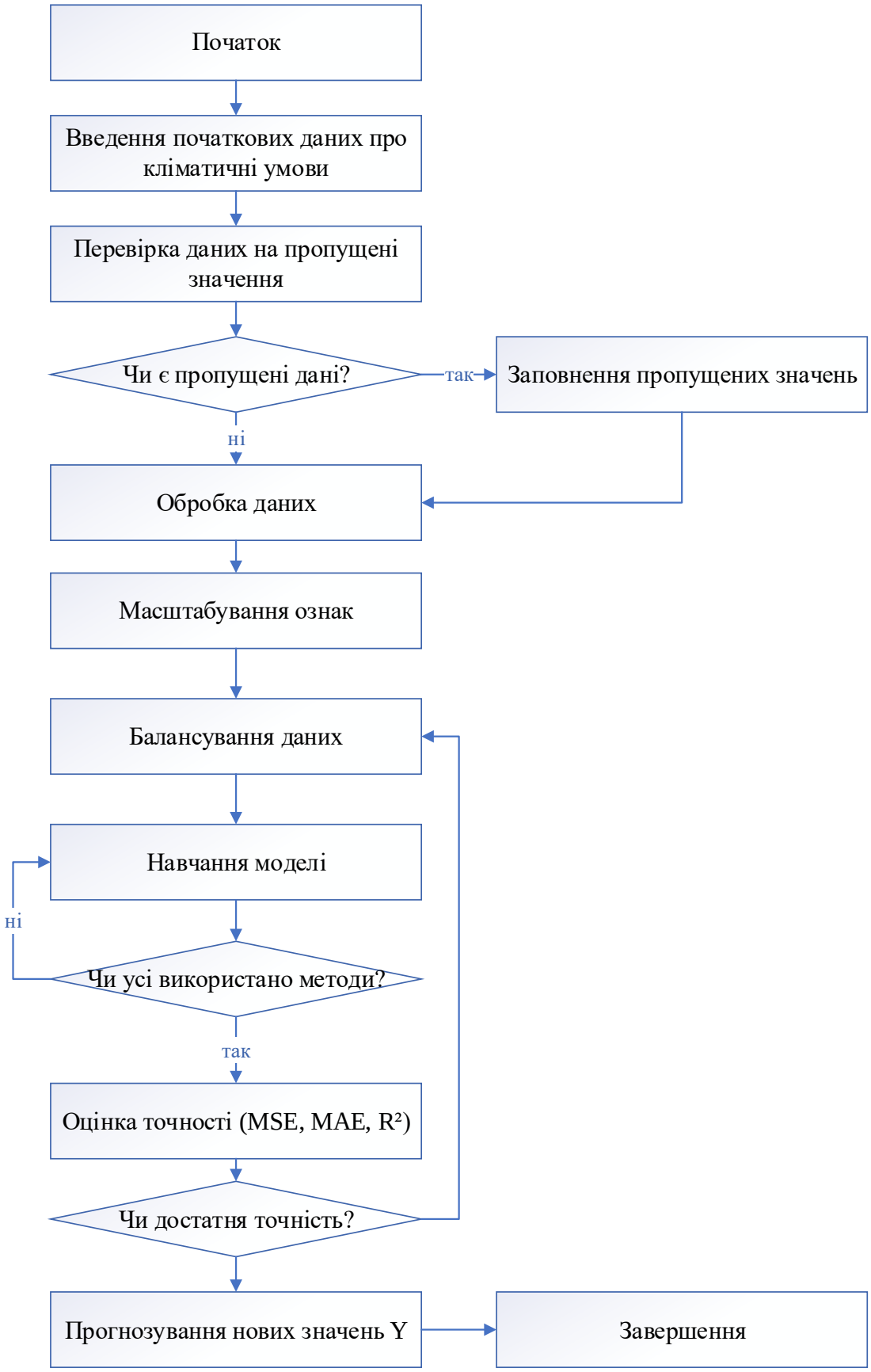

Рисунок 3.2 – Блок-схема алгоритму обробки вхідних даних та прогнозування

У представленій блок-схемі алгоритму обробки вхідних даних та прогнозування виконуються наступні кроки:

1. Початок. Алгоритм починається з ініціалізації системи, що включає підключення до сховища даних і завантаження вхідних параметрів. Вхідні дані: X_1 (Specific Humidity), X_2 (Relative Humidity), X_3 (Temperature), і Y (Precipitation).

2. Введення початкових даних про кліматичні умови. Дані отримують із зовнішніх джерел (API, бази даних) або надані користувачем. У формалізованому вигляді це виглядає як матриця $X \in \mathbb{R}^{n \times m}$, де n – кількість записів, а m – кількість ознак.

3. Перевірка на пропущені значення. Виконується перевірка, чи є у вхідних даних X пропущені значення NaN:

$$\text{isnull}(X) = \begin{cases} \text{True}, & \text{якщо } X_{ij} = \text{NaN} \\ \text{False}, & \text{інакше.} \end{cases}, \qquad (3.1)$$

Якщо пропущені значення присутні, запускається процедура їх заповнення.

4. Заповнення пропущених значень. Пропущені значення заповнюються середніми значеннями кожного стовпця:

$$X_{ij}^{\text{filled}} = \begin{cases} \dfrac{\sum_{i=1}^{n} X_{ij}}{n}, & \text{якщо } X_{ij} = \text{NaN}, \\ X_{ij}, & \text{інакше.} \end{cases} \qquad (3.2)$$

5. Масштабування ознак. Для нормалізації даних використовується мінімакс-скейлінг:

$$X_{ij}^{\text{scaled}} = \frac{X_{ij} - \min(X_j)}{\max(X_j) - \min(X_j)}, \qquad (3.3)$$

Це дозволяє привести всі ознаки до діапазону [0,1][0, 1][0,1].

6. Балансування заданим методом. Використовується метод для створення рівномірного розподілу значень цільової змінної Y. Кластери визначаються як:

$$C_k = \{Y_i \mid d(Y_i, \mu_k) < d(Y_i, \mu_j)\ \forall j \neq k\}, \qquad (3.4)$$

де μ_k – центр *k*-го кластера, d – евклідова відстань.

Із кожного кластера випадково вибирається рівна кількість записів для формування збалансованого набору.

7. Навчання моделі. Модель навчається на збалансованих даних із функцією втрат:

$$L(\theta) = \sum_{i=1}^{n} \left(Y_i - f(X_i, \theta)\right)^2 + \lambda \| \theta \|^2, \qquad (3.5)$$

де $f(X_i, \theta)$ – прогноз моделі; $\lambda \| \theta \|^2$ – регуляризаційний член для уникнення перенавчання.

Параметри моделі оновлюються за градієнтним методом:

$$\theta_{t+1} = \theta_t - \eta \nabla L(\theta_t), \qquad (3.6)$$

де η – швидкість навчання.

8. Прогнозування нових значень Y. Після навчання модель прогнозує кількість опадів для нових даних

$$Y = (X_{\text{new}}, \hat{\theta}), \qquad (3.7)$$

де $\hat{\theta}$ – оптимальні параметри моделі.

9. Оцінка точності. Точність прогнозування оцінюється за трьома метриками:

- ✓ MSE (Mean Squared Error)

$$\text{MSE} = \frac{1}{n} \sum_{i=1}^{n} \left(Y_i - \hat{Y}_i\right)^2, \qquad (3.8)$$

- ✓ MAE (Mean Absolute Error)

$$\text{MAE} = \frac{1}{n} \sum_{i=1}^{n} \left|Y_i - \hat{Y}_i\right|, \qquad (3.9)$$

- ✓ R^2 (R-Squared)

$$R^2 = 1 - \frac{\sum_{i=1}^{n} \left(Y_i - \hat{Y}_i\right)^2}{\sum_{i=1}^{n} \left(Y_i - \bar{Y}\right)^2}. \qquad (3.10)$$

10. Завершення. Алгоритм завершується, результати прогнозування та метрики точності передаються в інтерфейс для подальшого аналізу користувачем.

Запропонований алгоритм обробки вхідних даних та прогнозування кількості опадів забезпечує ефективну роботу системи. Поєднання методів балансування даних та навчання моделі дозволяє досягти високої точності за рахунок зменшення дисбалансу даних і використання сучасного методу машинного навчання.

3.3. Архітектура інтерфейсу користувача

Для інтерактивної взаємодії з користувачем у системі прогнозування кількості опадів реалізовано веб-інтерфейс на базі технологій React/Vue.js із серверною частиною, створеною за допомогою Flask. Система дозволяє вводити вхідні дані, отримувати прогнози моделі та візуалізувати результати у вигляді графіків.

Архітектура інтерфейсу користувача представлена у табл. 3.3.

Таблиця 3.3 – Архітектура інтерфейсу користувача

Компонент	Технологія	Призначення
Мова програмування	Python	Реалізація серверної логіки.
Бібліотеки ML	XGBoost, Scikit-learn	Реалізація моделі прогнозування.
Фреймворк API	Flask	Сервер для обробки запитів.
Фронтенд	React/Vue.js	Інтерактивний веб-інтерфейс для введення даних і перегляду результатів.
Візуалізація	Plotly/Chart.js	Відображення прогнозів у вигляді графіків.
Деплоймент	Docker, AWS/Heroku	Хостинг і забезпечення масштабованості.

Подана таблиця 3.3 описує ключові компоненти системи прогнозування кількості опадів із визначенням технологій, що використовуються, та їхнього функціонального призначення. Нами вибрана мова програмування Python, яка є основною мовою програмування для створення серверної частини. Вона забезпечує обробку даних, інтеграцію з бібліотеками машинного навчання (XGBoost, Scikit-learn) та побудову RESTful API за допомогою Flask.

Для навчання моделі XGBoost вибрано бібліотеку ML – Scikit-learn. XGBoost – високопродуктивна бібліотека для задач регресії та класифікації. Scikit-learn – базова бібліотека для попередньої обробки даних і обчислення метрик. Забезпечує реалізацію моделі прогнозування, підготовки даних і оцінки продуктивності. XGBoost демонструє високу точність і швидкість завдяки використанню бустингу. Scikit-learn забезпечує легкість у використанні для аналізу та валідації моделі.

Нами вибрано фреймворк API – Flask, який використовується для побудови серверної частини. Приймає HTTP-запити, передає дані до моделі машинного навчання, а також повертає результати у форматі JSON. Не підтримує асинхронність за замовчуванням (порівняно з FastAPI).

Фронтенд базується на React/Vue.js. Технологія React або Vue.js призначена для створення інтерактивного веб-інтерфейсу. Дозволяє користувачам вводити параметри, переглядати прогнози та аналізувати результати. Вона забезпечує динамічність і високий рівень інтерактивності, а також підтримку компонентного підходу для повторного використання коду.

Для візуалізації використано Plotly/Chart.js, що забезпечує використання інтерактивних графічних бібліотек для відображення прогнозів. Її призначення є для візуалізації історичних даних, прогнозів і їхніх порівнянь.

Деплоймент пропонується виконати із використанням Docker, AWS/Heroku. Docker призначено для контейнеризації додатків. AWS або Heroku для хостингу забезпечують масштабованість, безперервну роботу та легку розгортання системи. Дають можливість масштабування системи в

залежності від навантаження. Також наявна автоматизація розгортання та підтримка CI/CD.

Запропонована комбінація компонентів і технологій забезпечує високу точність прогнозування завдяки бібліотекам XGBoost і Scikit-learn. Зручний інтерфейс для взаємодії з користувачем, створений на React/Vue.js. Стабільність і масштабованість через використання Docker і хмарних сервісів (AWS/Heroku). Цей технологічний стек оптимальний для розробки інтерактивної системи прогнозування кількості опадів із підтримкою сучасних методів машинного навчання.

3.4. Створення серверної частини програми для прогнозування кількості опадів

Нами написано код для реалізації серверної частини програми для прогнозування кількості опадів за допомогою Flask. Серверна частина обробляє HTTP-запити від клієнтської частини, виконує прогнозування на основі завантаженої моделі машинного навчання, а також будує візуалізації результатів.

Насамперед виконано імпорт необхідних бібліотек (рис. 3.3): 1) Flask – використовується для створення веб-сервера та обробки запитів; 2) XGBoost – для роботи з попередньо навченою моделлю прогнозування; 3) Pandas і NumPy – для обробки вхідних даних; 4) Plotly – для побудови інтерактивних графіків; 5) JSON – для передачі даних у форматі, зручному для фронтенду.

У подальшому виконується ініціалізація Flask-додатка. Зокрема, app=Flask(__name__) створюється екземпляр Flask-додатка.

Модель XGBoost завантажується з файлу xgboost_model.json. Це дозволяє використовувати її для прогнозування без необхідності повторного навчання.

У подальшому обґрунтовуються маршрути для обробки запитів.

```python
from flask import Flask, request, jsonify
import xgboost as xgb
import pandas as pd
import numpy as np
import plotly.express as px
import json

app = Flask(__name__)

# Завантаження попередньо навченої моделі
model = xgb.XGBRegressor()
model.load_model("xgboost_model.json")  # Файл моделі

# Обробка POST-запиту для прогнозування
@app.route('/predict', methods=['POST'])
def predict():
    data = request.json
    try:
        # Перетворення вхідних даних у DataFrame
        features = pd.DataFrame([data])

        # Прогнозування
        prediction = model.predict(features)

        # Формування відповіді
        return jsonify({'prediction': float(prediction[0])})
    except Exception as e:
        return jsonify({'error': str(e)}), 400
```

Рисунок 3.3 – Фрагмент коду для реалізації серверної частини (Flask)

Фрагмент коду для реалізації серверної частини (Flask) показано на рис. 3.4.

```python
# Візуалізація прогнозів (графіки)
@app.route('/visualize', methods=['POST'])
def visualize():
    data = request.json
    try:
        # Створення набору даних для графіку
        historical = pd.DataFrame(data["historical"], columns=["Date", "Precipitation"])
        forecast = pd.DataFrame(data["forecast"], columns=["Date", "Prediction"])

        # Побудова графіка
        fig = px.line(historical, x="Date", y="Precipitation", title="Historical and Forecast Data")
        fig.add_scatter(x=forecast["Date"], y=forecast["Prediction"], mode='lines', name='Forecast')

        # Вивід графіка у форматі JSON
        return jsonify(fig.to_json())
    except Exception as e:
        return jsonify({'error': str(e)}), 400
```

Рисунок 3.4 – Фрагмент коду для реалізації серверної частини (Flask)

Візуалізація (/visualize) забезпечує обробку POST-запити. При цьому отримуються історичні дані та прогнозовані значення у форматі JSON. У результаті створюється графік з використанням Plotly. Також повертає графік у форматі JSON для відображення на фронтенді.

Сервер запускається у режимі відладки (debug=True), що дозволяє розробнику швидко тестувати і налагоджувати додаток.

Працює серверна частина наступним чином. Користувач через фронтенд відправляє запит. Зокрема, POST-запит на маршрут /predict для прогнозування

та POST-запит на маршрут /visualize для створення графіка. Flask обробляє запит, що виконує читає вхідні дані. Виконує потрібну операцію (прогнозування або побудову графіка). У подальшому сервер повертає відповідь у вигляді прогнозованого значення (у випадку /predict) та JSON-об’єктів із графіком (у випадку /visualize).

Особливостями реалізації є те, що сервер може обробляти як числові дані для прогнозування, так і часові ряди для візуалізації. Інтеграція з фронтендом дозволяє надавати користувачам зрозумілу графічну інформацію. Flask легко інтегрується з іншими сервісами (наприклад, базами даних або Docker для контейнеризації).

3.5. Реалізація фронтенду (React)

Нами написано код для реалізації інтерфейсу користувача для інформаційної системи прогнозування окремих опадів за допомогою React. Інтерфейс забезпечує інтерактивний зв’язок із серверною частиною, дозволяє вводити параметри, отримувати результати прогнозування та візуалізувати їх графік. Структура та основні компоненти фронтенду показані у таблиці 3.4.

Таблиця 3.4 – Структура та основні компоненти фронтенду

Компонент	Опис
Поля введення	Інтерфейс для введення початкових даних користувачем.
Кнопки	Елементи для запуску процесів прогнозування та візуалізації.
Відображення результатів	Вивід прогнозованої кількості опадів і графіків із даними.
HTTP-запити	Використовуйте Axios для зв'язку із серверною частиною.
Графічний модуль	Plotly.js для побудови інтерактивних графіків.

Код реалізації фронтенду (React) подано на рис. 3.5. Насамперед виконується імпорт бібліотек. Для роботи фронтенду створені такі основні

бібліотеки: 1) React для створення компонентів і управління станом; 2) Axios для відправки HTTP-запитів до серверної частини; 3) Plotly.js для візуалізації даних.

```
import React, { useState } from "react";
import axios from "axios";
import Plot from "react-plotly.js";

function App() {
  const [inputData, setInputData] = useState({ SpecificHumidity: "", RelativeHumidity: "", Temperature: "" });
  const [prediction, setPrediction] = useState(null);
  const [graphData, setGraphData] = useState(null);

  const handleChange = (e) => {
    setInputData({ ...inputData, [e.target.name]: e.target.value });
  };

  const handlePredict = async () => {
    try {
      const response = await axios.post("http://localhost:5000/predict", inputData);
      setPrediction(response.data.prediction);
    } catch (error) {
      console.error("Error predicting:", error);
    }
  };
```

Рисунок 3.5 – Фрагмент коду для реалізації фронтенду (React)

У коді передбачається обробка подій. При цьому передбачається зміна значення у полях введення. Функція handleChange оновлює inputData, зберігаючи введені значення.

```
  return (
    <div>
      <h1>Прогнозування кількості опадів</h1>
      <input
        type="number"
        name="SpecificHumidity"
        placeholder="Specific Humidity"
        value={inputData.SpecificHumidity}
        onChange={handleChange}
      />
      <input
        type="number"
        name="RelativeHumidity"
        placeholder="Relative Humidity"
        value={inputData.RelativeHumidity}
        onChange={handleChange}
      />
      <input
        type="number"
        name="Temperature"
        placeholder="Temperature"
        value={inputData.Temperature}
        onChange={handleChange}
      />
      <button onClick={handlePredict}>Отримати прогноз</button>
      <button onClick={handleVisualize}>Візуалізувати</button>

      {prediction && <h2>Прогнозована кількість опадів: {prediction.toFixed(2)}</h2>}
      {graphData && <Plot data={graphData.data} layout={graphData.layout} />}
    </div>
  );
}

export default App;
```

Рисунок 3.6 – Фрагмент коду для виконання прогнозування

Для прогнозування функція handlePredict відправляє дані на серверний маршрут /predict, а також отримує прогноз і зберігає його в prediction. Для візуалізації функція handleVisualize відправляє дані на серверний маршрут /visualize, отримує графік у форматі JSON і зберігає його в graphData.

Користувач вводить початкові дані. Він вводить параметри (Specific Humidity, Relative Humidity, Temperature) у поле введення. При натисканні на кнопку «Отримати прогноз» відбувається обробка введених даних і повертається прогнозована кількість опадів (рис. 3.6).

Візуалізація виконується на підставі побудови графіка, який містить прогнозоване значення.

Реалізований інтерфейс забезпечує зручність взаємодії користувача із системою. Здійснюється автоматична обробка введених даних. Візуалізація результатів прогнозування виконується у вигляді інтерактивних графіків.

Використання React дозволяє швидко адаптувати та розширювати функціональний інтерфейс, забезпечуючи високий рівень інтерактивності для користувачів.

РОЗДІЛ 4.
ОХОРОНА ПРАЦІ ТА БЕЗПЕКА У НАДЗВИЧАЙНИХ СИТУАЦІЯХ

4.1. Аналіз небезпек під час використання інтелектуальної інформаційної системи прогнозування кількості опадів

Використання інтелектуальної інформаційної системи прогнозування кількості опадів має як прямий, так і непрямий вплив на здоров'я працівників, які покладаються на результати прогнозів для планування діяльності. Нами проаналізовано небезпечні ситуації, які можуть виникати, а також запропоновані заходи для їх уникнення.

Таблиця 4.1 – Аналіз небезпек для здоров'я працівників

Категорія небезпеки	Опис	Можливі наслідки	Запобіжні заходи
1	2	3	4
Неточні прогнози	Система може надати помилкову інформацію про кількість опадів через модельні обмеження.	Ризик роботи в екстремальних умовах (зливи, грози), травми через негоду.	Регулярне оновлення моделі, врахування додаткових кліматичних факторів.
Затримка або збій у роботі системи	Система може не надати прогноз вчасно через технічні проблеми.	Працівники можуть бути не готові до змін погодних умов, що загрожує їхній безпеці.	Створення резервної копії системи та автоматичне аварійне сповіщення.

Продовження табл. 4.1

1	2	3	4
Неправильна інтерпретація прогнозу	Користувачі можуть помилково зрозуміти результати прогнозу через недостатню інформативність.	Недооцінка погодних ризиків, що може призвести до травм або захворювань.	Забезпечення зрозумілих інструкцій і пояснень до результатів прогнозу.
Довготривала робота за комп'ютером	Працівники можуть працювати з системою у статичній позі без перерв.	Погіршення стану зору, болі у спині та інших частинах тіла.	Рекомендації щодо ергономіки робочого місця та регулярних перерв.
Відсутність персоналу для моніторингу	Немає достатньо кваліфікованих працівників для роботи із системою або моніторингу даних.	Некоректна реакція на зміну погодних умов, збільшення ризиків травматизму.	Організація навчання та інструктажів із користування системою.

Система може надати прогноз, який не відповідає реальним умовам. Це може статися через обмеження моделі (недостатність даних для навчання), відсутність врахування додаткових факторів (вітер, атмосферний тиск) та використання застарілої версії моделі.

Працівники можуть потрапити під сильні опади, що спричиняє ризик травм або переохолодження. Для цього слід виконувати регулярну перевірку точності моделі. Виконувати інтеграцію додаткових джерел даних для підвищення якості прогнозу.

У разі технічних проблем система не встигає надати прогноз, що створює небезпеку для працівників, які планують діяльність залежно від погодних умов. При цьому робота в умовах негоди призводить до травм. Також можлива

відсутність інформації для прийняття обґрунтованих рішень. Для усунення цієї небезпеки слід налаштувати резервну систему сповіщення. Також можливе використання хмарних платформ для забезпечення безперервної роботи.

Аналіз небезпек показує, що під час використання інтелектуальної інформаційної системи прогнозування кількості опадів важливо враховувати не лише технічні аспекти, але й безпеку працівників. Для цього необхідно забезпечити точність прогнозів через оновлення моделі, резервні системи сповіщення для уникнення збоїв, інтуїтивно зрозумілий інтерфейс для запобігання помилкам інтерпретації, а також дотримання ергономічних стандартів для збереження здоров’я працівників.

4.2. Розробка заходів із покращення умов праці виконавців

Покращення умов праці є важливим елементом забезпечення продуктивності та безпеки виконавців, які використовують інтелектуальну інформаційну систему прогнозування кількості опадів. Нами запропоновано заходи, спрямовані на підвищення комфортності, безпеки та ефективності роботи.

Основні напрями покращення умов праці представлено на рис. 4.1.

Тривала робота за комп’ютером може викликати фізичний дискомфорт, який негативно впливає на продуктивність працівників. Для працівників рекомендується забезпечити правильне розташування монітора, клавіатури та миші відповідно до ергономічних стандартів. Використовувати офісні крісла з підтримкою спини та регульованими підлокітниками. Забезпечити якісне освітлення робочої зони для зменшення навантаження на зір. Встановити нагадування про регулярні перерви для розминки та зняття напруги. Це дасть можливість зменшити ризик болю в спині, шиї та очах, а також підвищити загальну зручність праці.

Рисунок 4.1 – Основні напрями покращення умов праці

Правильне використання інтелектуальної інформаційної системи залежить від рівня знань та навичок працівників. Недостатня підготовка може призвести до помилок у роботі. Рекомендується для працівників проводити регулярні тренінги з використання функціоналу системи, включаючи інтерпретацію результатів прогнозів. Розробити інструкції та документацію, що описують кроки взаємодії з системою. Впровадити симуляційні сценарії для відпрацювання дій у різних погодних умовах. Це дасть можливість підвищити впевненість працівників у використанні системи. Зменшити кількість помилок при роботі з прогнозами.

Багато завдань, пов’язаних із прогнозуванням і аналізом погодних умов, можуть бути автоматизовані для зменшення навантаження на працівників. Для цього слід налаштувати автоматичне сповіщення про критичні погодні зміни (зливи, бурі) через електронну пошту або месенджери. Використовувати автоматичну обробку великих обсягів даних для формування звітів. Інтегрувати систему з календарями для автоматичного планування робочих завдань

відповідно до прогнозу. Це дасть можливість зменшити час, витрачений на рутинну обробку даних. Своєчасно реагування на зміни погодних умов.

Впровадження зазначених заходів дозволить не лише покращити умови праці виконавців, але й підвищити ефективність використання інтелектуальної системи прогнозування кількості опадів. Завдяки ергономічним рішенням, навчальним програмам, автоматизації рутинних завдань, забезпеченню психологічного комфорту та підвищенню безпеки працівники зможуть максимально ефективно виконувати свої обов’язки, зберігаючи здоров’я та підвищуючи продуктивність.

4.3. Розробка заходів із забезпечення безпеки виконавців під час надзвичайних ситуацій

Робота виконавців у складних погодних умовах, зокрема в надзвичайних ситуаціях, пов'язаних із погодними явищами (зливи, бурі, повені тощо), потребує впровадження заходів безпеки. Ці заходи спрямовані на захист здоров’я та життя працівників, а також на забезпечення ефективності виконання завдань у небезпечних умовах.

Таблиця 4.2 – Аналіз потенційних ризиків

Ризик	Можливі наслідки	Приклади ситуацій
Травматизм через негоду	Пошкодження через падіння, удари або сильні опади.	Робота під час зливи або ожеледиці.
Гіпотермія або перегрівання	Переохолодження або тепловий удар.	Робота в умовах надмірного холоду чи спеки.
Порушення зв’язку	Відсутність комунікації з іншими працівниками.	Збої у зв’язку через погодні умови.
Невчасна евакуація	Неможливість залишити небезпечну зону вчасно.	Повені або різкі погодні зміни.
Втрата орієнтації	Неможливість визначити місце перебування через умови.	Робота під час густого туману або грози.

Нами запропоновано заходи із забезпечення безпеки. Насамперед слід забезпечити працівників засобами індивідуального захисту (ЗІЗ). Рекомендується використовувати водонепроникний одяг, захисні чоботи, шоломи та окуляри. Оснастити працівників світловідбивними елементами для роботи в умовах слабкого освітлення. Надати портативні обігрівачі або охолоджуючі елементи для регуляції температури тіла.

Заслуговує на увагу організація системи сповіщення. Слід впровадити автоматичну систему оповіщення про небезпечні погодні явища (SMS, мобільні додатки). Забезпечити можливість зв’язку між працівниками та диспетчерським центром через рації чи супутникові телефони. У результаті отримаємо своєчасне інформування працівників про ризики. Зменшення часу на реагування в екстрених ситуаціях.

Рекомендується розробити плани евакуації. Зокрема, розробити детальні маршрути евакуації з небезпечних зон. Оснастити працівників портативними GPS-навігаторами для орієнтації. Провести тренування з евакуації для підготовки до різних сценаріїв. У результаті зменшиться ризик паніки під час небезпечних ситуацій та підвищиться швидкість евакуації.

Запропоновані заходи дозволяють мінімізувати ризики для здоров’я та життя працівників під час надзвичайних ситуацій, пов’язаних із погодними умовами. Систематичне впровадження засобів індивідуального захисту, організація ефективної системи сповіщення, моніторинг стану працівників, а також інтеграція прогнозів із плануванням робіт забезпечать безпеку виконавців і ефективне виконання завдань.

РОЗДІЛ 5.
ЕКОНОМІЧНА ЕФЕКТИВНІСТЬ ВІД ВИКОРИСТАННЯ ІНТЕЛЕКТУАЛЬНОЇ ІНФОРМАЦІЙНОЇ СИСТЕМИ ПРОГНОЗУВАННЯ КІЛЬКОСТІ ОПАДІВ

Нами проведено оцінку економічної ефективності від впровадження інтелектуальної інформаційної системи прогнозування кількості опадів. Аналіз виконується на основі порівняння витрат на реалізацію системи з використаними вигодами, які отримуються внаслідок її використання.

Економічна ефективність розраховується за формулою:

$$E = B - C, \tag{5.1}$$

де E – економічна ефективність, грн; B – вигоди від впровадження системи, грн; C – витрати на впровадження та експлуатацію системи, грн.

Вигоди B розраховуються як сума прямих економічних вигод, зокрема від зменшення витрат на ліквідацію наслідків погодних ризиків (B_1) та оптимізації ресурсів, наприклад, зменшення витрат на полив (B_2).

$$B = B_1 + B_2, \tag{5.2}$$

Таблиця 5.1 – Початкові дані для розрахунку економічної ефективності

Параметр	Значення	Одиниця виміру
Зменшення витрат на ліквідацію наслідків (B_1)	500000	грн
Економія на оптимізації поливу (B_2)	200000	грн
Вартість розробки системи (C_1)	300000	грн
Витрати на навчання персоналу (C_2)	50000	грн
Експлуатаційні витрати (C_3)	100000	грн/рік

Витрати (C) включають вартість розробки системи (C_1), витрати на навчання персоналу (C_2) та експлуатаційні витрати (C_3):

$$C = C_1 + C_2 + C_3. \tag{5.3}$$

На підставі вище поданих формул виконаємо розрахунок економічної ефективності.

Проводимо визначення загальних вигод (B):

$$B = 500000 + 200000 = 700000\,грн.$$

Визначаємо загальні витрати (C):

$$C = 300000 + 50000 + 100000 = 450000\,грн.$$

Розраховуємо економічну ефективність (E):

$$E = 700000 - 450000 = 250000\,грн.$$

Результати виконаного розрахунку подано у таблиці 5.2.

Таблиця 5.2 – Результати розрахунку економічної ефективності

Показник	Розрахункове значення, грн
Загальні вигоди (B)	700000
Загальні витрати (C)	450000
Економічна ефективність (E)	250000

Розрахунок економічної ефективності показав, що впровадження інтелектуальної інформаційної системи прогнозування кількості опадів є економічно доцільним. Система дозволяє заощадити 250000 грн за рахунок зменшення витрат на ліквідацію наслідків погодних ризиків та оптимізацію використання ресурсів.

Окрім цього, впровадження системи має непрямі вигоди, такі як покращення організації роботи, зменшення ризиків для працівників та забезпечення точнішого планування. Для максимізації економічної ефективності рекомендується регулярне оновлення моделей прогнозування та розширення функціональності системи.

ВИСНОВКИ І ПРОПОЗИЦІЇ

На даний час прогнозування кількості опадів є завданням, яке має значний вплив на сільське господарство, управління водними ресурсами, транспортну інфраструктуру та планування екологічних заходів. Проте висока нестабільність кліматичних умов та обмеженість історичних даних для певних регіонів впливають на точність прогнозів. У нашій роботі пропонується розробити інтелектуальну інформаційну систему прогнозування кількості опадів, що використовує методи балансування даних та моделі машинного навчання для підвищення точності та адаптивності прогнозів.

Нами проаналізована загальна характеристика кліматичних умов Львівської області. Кліматичні умови характеризуються високою вологістю, значною кількістю опадів та помірними температурами. Це створює сприятливі умови для сільського господарства, але водночас спричиняє ризики паводків і зсувів у гірських районах. Розуміння кліматичних особливостей регіону є місцем для розробки системи прогнозування кількості опадів та адаптації до зміни клімату.

Аналіз сучасних методів прогнозування кількості опадів показав, що традиційні статистичні та фізичні підходи мають обмеження в умовах складних погодних змін, тоді як методи машинного навчання забезпечують високу точність, адаптивність і здатність працювати з великими обсягами даних. Це обґрунтовує доцільність розроблення інтелектуальної інформаційної системи прогнозування, яка зможе інтегрувати передові алгоритми машинного навчання для створення гнучкого та ефективного інструменту підтримки прийняття рішень у сфері кліматичного планування.

Встановлено, що балансування даних є важливим етапом у задачах прогнозування кількості опадів. Воно забезпечує рівномірний розподіл класів, покращуючи точність моделей і враховуючи екстремальні погодні явища. Використання сучасних методів, таких як SMOTE, ADASYN та комбінованих

підходів, дозволяє отримати більш точні та стабільні результати, що особливо важливо в контексті задач кліматичного аналізу та управління ресурсами.

Проблеми обробки та балансування кліматичних даних є серйозним викликом для сучасних систем прогнозування. Розробка інтелектуальної інформаційної системи прогнозування кількості опадів із використанням методів машинного навчання та балансування даних дозволить вирішити ці проблеми, забезпечуючи точність і стабільність прогнозів. Це має велике значення для підвищення ефективності управління природними ресурсами та попередження наслідків екстремальних погодних явищ.

В Україні отримати дані спостережень про погоду дуже складно через рідкісне розміщення метеостанцій і непослідовні записи даних. Це створило критичні прогалини в доступності даних для запуску та розробки ефективних моделей прогнозування кількості опадів. Щоб подолати цю прогалину, ми отримали часові ряди даних щомісячних спостережень за даними для умов Львівської області. Період даних з 1992 по 2024 рік із Львівського регіонального центру гідрометеорології. Доступ до даних було отримано зі сховища даних (рис. 2.1).

Нами написано код, який створює графік кількості опадів (Precipitation) з використанням даних у часовому форматі, де індекс представлений як початок місяця. Отримано графік, який показує, як змінювалася кількість опадів протягом кожного місяця з 1992 по 2024 рік. Цей графік допомагає візуально оцінити тенденції в кількості опадів, виявити сезонність або пікові періоди опадів, а також підготувати дані до подальшого аналізу чи моделювання. У подальшому нами проаналізовано сезонну декомпозицію. Дані часового ряду розглядаються з позиції трендів та їх сезонності (рис. 2.5).

Для задачі прогнозування кількості опадів рекомендується використовувати зазначені методи балансування даних, що дозволяють суттєво покращити якість моделювання, особливо в задачах з дисбалансом даних. Обраний підхід спрямований на збереження варіативності даних та запобігання

втратам інформації. Це створює міцну основу для ефективного моделювання та точних прогнозів.

Нами створено код для балансування даних за різними методами. Фрагменти із кодом різних методів балансування даних подано на рис. 2.5. Для кожного методу (логарифмічне перетворення, зважування, кластеризація) виводиться кількість рядків та колонок у відповідному DataFrame. Отже, у подальшому пропонується балансувати цільову змінну Precipitation трьома різними методами, щоб покращити якість моделей машинного навчання.

Нами написано код, що створює функцію для візуалізації розподілу цільової змінної (Precipitation) у різних наборах даних після застосування методів балансування. Потім функція викликається для відображення результатів, які представлено на рис. 2.8. Це дало можливість створити функцію для візуалізації розподілу цільової змінної Precipitation після застосування різних методів балансування даних. Це допомагає порівняти ефекти різних методів балансування та зрозуміти, як кожен із них вплинув на структуру розподілу даних.

На кожному з підготовлених наборів даних навчаються три моделі: 1) Gradient Boosting Regressor – ансамблевий метод, який поступово зменшує похибку шляхом додавання нових моделей; CatBoost Regressor – модель градієнтного бустингу, спеціально оптимізована для високої швидкості та точності; XGBoost Regressor – покращена версія бустингу з підтримкою регуляризації, що забезпечує високу продуктивність.

Нами проведено дослідження впливу різних методів балансування даних на точність прогнозування кількості опадів із використанням трьох моделей машинного навчання – Gradient Boosting, CatBoost Regressor та XGBoost Regressor. Для оцінки точності застосовано метрики MSE, MAE та R^2. У таблиці 2.2 наведено результати, отримані на трьох збалансованих наборах даних.

Встановлено, що найменше значення MSE спостерігається для XGBoost Regressor з методом KMeans Sampling. Метод Weighted Sampling має значно більші значення MSE, що робить його непридатним. Найменше значення MAE

також спостерігається для XGBoost Regressor з методом KMeans Sampling. Логарифмічне перетворення також показує низькі значення MAE, особливо для Gradient Boosting. Найвище значення R^2 (0.806) отримано для XGBoost Regressor з методом KMeans Sampling. Логарифмічне перетворення забезпечує середній рівень точності ($R^2 \approx 0.644$ для Gradient Boosting). Для задачі прогнозування кількості опадів рекомендовано використовувати метод KMeans Sampling у поєднанні з моделлю XGBoost Regressor.

Запропонована архітектура системи забезпечує ефективне прогнозування кількості опадів за допомогою сучасних технологій. Використання моделі XGBoost Regressor разом із методом балансування KMeans Sampling дозволяє досягти високої точності прогнозів. Технологічний стек на основі Python, Flask, PostgreSQL та React/Vue.js забезпечує гнучкість, масштабованість і інтерактивність системи.

Запропонований алгоритм обробки вхідних даних та прогнозування кількості опадів забезпечує ефективну роботу системи. Поєднання методів балансування даних та навчання моделі дозволяє досягти високої точності за рахунок зменшення дисбалансу даних і використання сучасного методу машинного навчання.

Запропонована комбінація компонентів і технологій забезпечує високу точність прогнозування завдяки бібліотекам XGBoost і Scikit-learn. Зручний інтерфейс для взаємодії з користувачем, створений на React/Vue.js. Стабільність і масштабованість через використання Docker і хмарних сервісів (AWS/Heroku). Цей технологічний стек оптимальний для розробки інтерактивної системи прогнозування кількості опадів із підтримкою сучасних методів машинного навчання.

Нами написано код для реалізації серверної частини програми для прогнозування кількості опадів за допомогою Flask. Серверна частина обробляє HTTP-запити від клієнтської частини, виконує прогнозування на основі завантаженої моделі машинного навчання, а також будує візуалізації результатів.

Нами написано код для реалізації інтерфейсу користувача для інформаційної системи прогнозування окремих опадів за допомогою React. Інтерфейс забезпечує інтерактивний зв’язок із серверною частиною, дозволяє вводити параметри, отримувати результати прогнозування та візуалізувати їх графік. Структура та основні компоненти фронтенду показані у таблиці 3.4. Використання React дозволяє швидко адаптувати та розширювати функціональний інтерфейс, забезпечуючи високий рівень інтерактивності для користувачів.

Розроблено заходи з охорони праці під час створення та використання інтелектуальної інформаційної системи прогнозування кількості опадів, дотримання яких забезпечить безпечні умови роботи для виконавців.

Розрахунок економічної ефективності показав, що впровадження інтелектуальної інформаційної системи прогнозування кількості опадів є економічно доцільним. Система дозволяє заощадити 250000 грн за рахунок зменшення витрат на ліквідацію наслідків погодних ризиків та оптимізацію використання ресурсів.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Жидецький В.Ц., Джигирей В.С., Мельников О.В. Основи охорони праці. Підручник. Вид. 5-е, доповнене. Львів: Афіша, 2012. 350с.

2. Класифікація в Python з Scikit-Learn та Pandas. URL: https://stackabuse.com/classification-in-pythonwith-scikit-learn-and-pandas/ (дата звернення: 18.10.2024).

3. Лехман С.Д., Рублев В.І., Рябцев Б.І. Запобігання аварійності і травматизму у сільському господарстві. К.: Урожай, 1993. 267 с.

4. Павлиш В. А. ,. Гліненко Л. К, Шаховська Н. Б. Основи інформаційних технологій і систем: підручник. Львів: Львівська політехніка, 2018. 620 с.

5. Ситнік Б.Т. Основи інформаційних систем і технологій: навч. посіб. Харків: УкрДУЗТ, 2018. 130 с.

6. Tryhuba, A., Boyarchuk, V., Tryhuba, I., Ftoma, O., Padyuka, R., Rudynets, M. Forecasting the Risk of the Resource Demand for Dairy Farms Basing on Machine Learning. Proceedings of the 2nd International Workshop on Modern Machine Learning Technologies and Data Science (MoMLeT+DS 2020). 2020; I, 327-340.

7. Харченко В. О. Основи машинного навчання : навч. посіб. / В. О. Харченко. Суми : Сумський державний університет, 2023. 264 с.

8. Ямпольський Л. С. Системи штучного інтелекту в плануванні, моделюванні та управлінні :підручник для студентів вищ. навч. закл. / Л. С. Ямпольський, Б П. Ткач, О. І. Лісовиченко. К.:ДП Видавничий дім «Персонал», 2011. 544 с.

9. Abba S. et al. Evolutionary computational intelligence algorithm coupled with self-tuning predictive model for water quality index determination. J. Hydrol., 2020, 587, P. 1-12.

10. Botes D., Mecikalski J. R. and Jedlovec G. J. Atmospheric infrared sounder (AIRS) sounding evaluation and analysis of the pre-convective environment. J. Geophysical Res. Atmos., 2012, 117, P. 1-21.

11. Breiman L. Random forest. Mach. Learn., 2001, 45, P. 5-32.

12. Cai J. et al. An assembly-level neutronic calculation method based on LightGBM algorithm. Ann. Nucl. Energy, 2020, 150, P. 1-12.

13. Chen T. and Guestrin C. XGBoost: A scalable tree boosting system. Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining, 2016, P. 785-794.

14. Cortes C. and Vapnik V. Support-vector networks. Mach. Learn., 1995, 20, 3, P. 273-297.

15. Cottrill A. et al. Seasonal forecasting in the Pacific using the coupled model POAMA-2. Weather Forecasting, 2013, 28, 3, P. 668-680.

16. Fang K. and Shen C. Full-flow-regime storage-streamflow correlation patterns provide insights into hydrologic functioning over the continental US. Water Resour. Res., 2017, 53, P. 8064-8083.

17. Guo J. et al. Investigation of near-global daytime boundary layer height using high-resolution radiosondes: First results and comparison with ERA-5 MERRA-2 JRA-55 and NCEP-2. Atmospheric Chem. Phys., 2021, 21, 22, P. 17079-17097.

18. Guo J. et al. Viurnal variation and the influential factors of precipitation from surface and satellite measurements in Tibet. Int. J. Climatol., 2014, 34, P. 2940-2956.

19. He Z. et al. A comparative study of artificial neural network adaptive neuro fuzzy inference system and support vector machine for forecasting river flow in the semiarid mountain region. J. Hydrol., 2014, 509, P. 379-386.

20. Hopfield J. J. Neural networks and physical systems with emergent collective computational abilities. Proc. Nat. Acad. Sci., 1982, 79, 8, P. 2554-2558.

21. Ke G. et al. LightGBM: A highly efficient gradient boosting decision tree. Adv. Neural Inf. Process. Syst., 2017, 30, P. 3146-3154.

22. Koval N., Tryhuba A., Kondysiuk I., Tryhuba I., Boiarchuk O., Rudynets M., Grabovets V., Onyshchuk V. Forecasting the Fund of Time for Performance of Works in Hybrid Projects Using Machine Training Technologies. Proceedings of the 3nd International Workshop on Modern Machine Learning Technologies and Data Science Workshop. Proc. 3rd International Workshop (MoMLeT&DS 2021). 2021; I, 96-206.

23. Li Y. et al. A multi-model integration method for monthly streamflow prediction: Modified stacking ensemble strategy. J. Hydroinform., 2020, 22, 2, P. 310-326.

24. Liang Z. et al. Long-term streamflow forecasting using SWAT through the integration of the random forests precipitation generator: Case study of Danjiangkou reservoir. Hydrol. Res., 2018, 49, 5, P. 1513-1527.

25. Lipton Z. C., Berkowitz J. and Elkan C. A critical review of recurrent neural networks for sequence learning. 2015, [Online]. Available: https://arxiv.org/abs/1506.00019.

26. Malanchuk, O., Tryhuba, A., Tryhuba, I., Sholudko, R., Pankiv, O. A Neural Network Model-based Decision Support System for Time Management in Pediatric Diabetes Care Projects. International Scientific and Technical Conference on Computer Sciences and Information Technologies, 2023.

27. Min M. et al. Estimating summertime precipitation from Himawari-8 and global forecast system based on machine learning. IEEE Trans. Geosci. Remote Sens., 2019, 57, 5, P. 2557-2570.

28. Noori R. et al. Assessment of input variables determination on the SVM model performance using PCA Gamma test and forward selection techniques for monthly stream flow prediction. J. Hydrol., 2011, 401, 3-4, P. 177-189.

29. Sadlera J. M., Goodalla J. L., Morsyab M. M. and Spencerc K. Modeling urban coastal flood severity from crowd-sourced flood reports using Poisson regression and random forest. J. Hydrol., 2018, 559, P. 43-55.

30. Saha S. et al. The NCEP climate forecast system version 2. J. Climate, 2014, 27, 6, P. 2185-2208.

31. Schoppa L., Disse M. and Bachmair S. Evaluating the performance of random forest for large-scale flood discharge simulation. J. Hydrol., 2020, 590, P. 1-25.

32. Shen C. A transdisciplinary review of deep learning research and its relevance for water resources scientists. Water Resour. Res., 2018, 54, 11, P. 8558-8593.

33. Shortridge J. E., Guikema S. D. and Zaitchik B. F. Machine learning methods for empirical streamflow simulation: A comparison of model accuracy interpretability and uncertainty in seasonal watersheds. Hydrol. Earth Syst. Sci., 2016, 7, 20, P. 2611-2628.

34. Simmons A. J., Willett K. M., Jones P. D., Thorne P. W. and Dee D. P. Low frequency variations in surface atmospheric humidity temperature and precipitation: Inferences from reanalyses and monthly gridded observational data sets. Geophys. Res. Atmos., 2010, 115, D1, P. 1-21.

35. Sutinen R. and Middleton M. Soil water drives distribution of northern Boreal conifers Picea abies and Pinus sylvestris. J. Hydrol., 2020, 588, P. 1-20.

36. Thompson G., Field P. R., Rasmussen R. M. and Hall W. D. Explicit forecasts of winter precipitation using an improved bulk microphysics scheme. Part II: Implementation of a new snow parameterization. Monthly Weather Rev., 2008, 136, 12, P. 5095-5115.

37. Tryhuba, A., Boyarchuk, V., Tryhuba, I., Ftoma, O., Padyuka, R. & Rudynets, M. Forecasting the risk of the resource demand for dairy farms basing on machine learning. CEUR Workshop Proceedings. https://www.scopus.com/authid/detail.uri?authorId=57205225539. 2021; 2631: 327–340.

38. Tryhuba, A., Kondysiuk, I., Tryhuba, I., Boiarchuk, O., Tatomyr, A. Intellectual information system for formation of portfolio projects of motor transport enterprises. CEUR Workshop Proceedings. 2022; 3109, 44–52.

39. Tryhuba, A., Tryhuba, I., Ftoma, O. & Boyarchuk, O. Method of quantitative evaluation of the risk of benefits for investors of fodder-producing

cooperatives. International Scientific and Technical Conference on Computer Sciences and Information Technologies, https://www.scopus.com/authid/detail.uri?authorId=57205225539. 2019; 3: 55–58.

40. Tryhuba, A., Tryhuba, I., Malanchuk, O., Marmulyak, A. A deep neural network model for predicting the competitive score of social projects for community development. CEUR Workshop Proceedings, 2024, 3711, pp. 55–74.

41. Tryhuba, A., Vovk, M., Batyuk, B., Holomsha, O., Sava, A. Improving the quality of management in the system of forecasting milk procurement in communities usage the technology of neutron networks. Journal of Hygienic Engineering and Design, 2022, 40, pp. 201–209

42. Tryhuba, A., Koval, N., Tryhuba, I. & Boiarchuk, O. Application of sarima models in information systems forecasting seasonal volumes of food raw materials of procurement on the territory of communities. CEUR Workshop Proceedings. 2022; 3295: 64–75. https://www.scopus.com/authid/detail.uri?authorId=57205225539.

43. Tryhuba, A., Malanchuk, O., Tryhuba, I. Prediction of the Duration of Inpatient Treatment of Diabetes in Children Based on Neural Networks. CEUR Workshop Proceedings. 2023; 3426, 122–135.

44. Tryhuba, I., Hutsol, T., Tryhuba, A., Tulej, W., Sojak, M. An Approach to Assessing the State of Organic Waste Generation in Community Households Based on Associative Learning. Sustainability (Switzerland), 2023, 15(22), 15922.

45. Tryhuba, I., Tryhuba, A., Hutsol, T., Tulej, W., Sojak, M. Prediction of Biogas Production Volumes from Household Organic Waste Based on Machine Learning. Energies, 2024, 17(7), 1786.

46. Volker W. et al. The convective and orographically-induced precipitation study (COPS): The scientific strategy the field phase and research highlights. Quart. J. Roy. Meteorological Soc., 2011, 137, P. 3-30.

47. Wakin M. Dimensionality Reduction. New York, NY, USA: Wiley, 2007.

48. XGBoost Core Team, Python API Reference, 2024. URL: https://xgboost.readthedocs.io/en/stable/python/python_api.html

49. Xiang Z., Yan J. and Demir I. A rainfall-runoff model with LSTM-based sequence-to-sequence learning. Water Resour. Res., 2020, 56, 1, P. 1-17.

50. Yaseen Z. M. et al. Stream-flow forecasting using extreme learning machines: A case study in a semi-arid region in Iraq. J. Hydrol., 2016, 542, P. 603-614.

51. Yu P. S. et al. Comparison of random forests and support vector machine for real-time radar-derived rainfall forecasting. J. Hydrol., 2017, 552, P. 92-104.

52. Zhang J. et al. Developing a long short-term memory (LSTM) based model for predicting water table depth in agricultural areas. J. Hydrol., 2018, 561, P. 918-929.

53. Zou X., Qin Z. and Weng F. Improved coastal precipitation forecasts with direct assimilation of GOES-11/12 imager radiances. Monthly Weather Rev., 2011, 139, 1, P. 3711-3728.

ДОДАТКИ

Додаток А

Код для балансування даних за різними методами

```
import pandas as pd
import numpy as np
from sklearn.preprocessing import FunctionTransformer
from sklearn.utils.class_weight import compute_sample_weight
from sklearn.cluster import KMeans

# Копія даних для роботи
df = df.copy()

# Цільова змінна і ознаки
X = df.drop(columns=[«Precipitation»])  # Ознаки
y = df[«Precipitation»]  # Цільова змінна

# 1. Логарифмічне перетворення (Logarithmic Transformation)
transformer = FunctionTransformer(np.log1p, validate=True)  # log1p для log(1+x), щоб
уникнути log(0)
y_log = transformer.transform(y.values.reshape(-1, 1)).flatten()
df_log = pd.concat([X.reset_index(drop=True), pd.DataFrame(y_log, columns=[«Precipitation»])],
axis=1)

# 2. Зважування (Weighted Sampling)
weights = compute_sample_weight(«balanced», y)
df_weighted = pd.concat([X.reset_index(drop=True), pd.DataFrame(y, columns=[«Precipitation»]),
pd.DataFrame(weights, columns=[«Weights»])], axis=1)

# 3. Кластеризація і вибірка (KMeans Sampling)
kmeans = KMeans(n_clusters=5, random_state=42)
df[«Cluster»] = kmeans.fit_predict(y.values.reshape(-1, 1))  # Кластеризація по значеннях
Precipitation
dfs_kmeans = []

for cluster in df[«Cluster»].unique():
    cluster_df = df[df[«Cluster»] == cluster]
    sampled_cluster = cluster_df.sample(n=min(len(cluster_df), 50), random_state=42,
replace=True)
    dfs_kmeans.append(sampled_cluster)

# Об’єднання результатів кластеризації та видалення колонки «Cluster»
df_kmeans = pd.concat(dfs_kmeans)
df_kmeans = df_kmeans.drop(columns=[«Cluster»])

# Вивід розмірів збалансованих DataFrame
print(«Logarithmic Transformation:», df_log.shape)
print(«Weighted Sampling:», df_weighted.shape)
print(«KMeans Sampling:», df_kmeans.shape)
```

Додаток Б

Код для навчання моделей прогнозування кількості опадів

```
from sklearn.ensemble import GradientBoostingRegressor
from xgboost import XGBRegressor
from catboost import CatBoostRegressor
import lightgbm as lgb

# Нові моделі
models = {
   «Gradient Boosting»: lgb.LGBMRegressor(n_estimators=100, random_state=42),
   «CatBoost Regressor»: CatBoostRegressor(iterations=100, learning_rate=0.1, depth=6,
verbose=0),
   «XGBoost Regressor»: XGBRegressor(n_estimators=100, learning_rate=0.1, max_depth=6,
random_state=42)
}

# Словник для результатів
results = []

# Цикл по наборах даних
for dataset_name, df_balanced in datasets.items():
   print(f»Evaluating on {dataset_name} data...»)

   # Розділення даних на ознаки і цільову змінну
   X = df_balanced.drop(columns=[«Precipitation»])
   y = df_balanced[«Precipitation»]

   # Розділення на тренувальний і тестовий набори
   X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

   # Тренування та оцінка нових моделей
   for model_name, model in models.items():
      print(f»Training {model_name}...»)
      model.fit(X_train, y_train)
      predictions = model.predict(X_test)

      # Обчислення метрик
      mse = mean_squared_error(y_test, predictions)
      mae = mean_absolute_error(y_test, predictions)
      r2 = r2_score(y_test, predictions)

      results.append({
         «Model»: model_name,
         «Dataset»: dataset_name,
         «MSE»: mse,
         «MAE»: mae,
         «R2»: r2
      })
```

```
# Результати у вигляді DataFrame
results_df = pd.DataFrame(results)

# Виведення таблиці результатів
print(«\nEvaluation Results:»)
print(results_df)

# Побудова графіків для метрик
for metric in [«MSE», «MAE», «R2»]:
    plt.figure(figsize=(10, 6))
    for dataset_name in datasets.keys():
        subset = results_df[results_df[«Dataset»] == dataset_name]
        plt.plot(subset[«Model»], subset[metric], marker=«o», label=dataset_name)

    plt.title(f»Comparison of {metric} Across Models and Datasets», fontsize=16,
fontweight=«bold»)
    plt.xlabel(«Model», fontsize=14)
    plt.ylabel(metric, fontsize=14)
    plt.legend(title=«Dataset», fontsize=12)
    plt.grid(alpha=0.6, linestyle=«--»)
    plt.tight_layout()
    plt.show()
```